Government of Karnataka
Department of Technical Education

Diploma Curriculum
Computer Science and Engineering

C 25 Scheme of Studies
(Effect from the AY 2025-26)

Government of Karnataka

DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

III Semester Scheme of Studies - Diploma in Computer Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	CS	25CS31I	Database Concepts and Technologies	4	0	4	8	6	50	20	50	20	-	-	100
2	CS	25CS32I	Computer Networks	4	0	4	8	6	50	20	50	20	-	-	100
3	CS	25CS33I	Data Structures with Python	3	0	4	7	5	50	20	-	-	50	20	100
4	CS	25CS34I	System and Network Administration	3	0	4	7	5	50	20	-	-	50	20	100
Audit Course															
5	KA	25KA31T	Kannada –I (ಸಾಹಿತ್ಯ ಸಿಂಚನ-I/ಬಳಕೆ ಕನ್ನಡ-I)	2	0	0	2	2	50	20	-	-	-	-	50
Total				**16**	**0**	**16**	**32**	**24**	**250**	**-**	**100**	**-**	**100**	**-**	**450**

Government of Karnataka

DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

IV Semester Scheme of Studies - Diploma in Computer Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	CS	25CS41I	Linux Administration	4	0	4	8	6	50	20	50	20	-	-	100
2	CS	25CS42I	Object Oriented Programming with Java	4	0	4	8	6	50	20	50	20	-	-	100
3	CS	25CS43I	Data Analysis and Algorithm Design	3	0	4	7	5	50	20	-	-	50	20	100
4	CS	25CS44I	Web Development	3	0	4	7	5	50	20	-	-	50	20	100
Audit Course															
5	KA	25KA41T	Kannada –II (ಸಾಹಿತ್ಯ ಸಿಂಚನ-II/ಬಳಕೆ ಕನ್ನಡ-II)	2	0	0	2	2	50	20	-	-	-	-	50
Total				**16**	**0**	**16**	**32**	**24**	**250**	**-**	**100**	**-**	**100**	**-**	**450**

SEMESTER 3

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	3
Course Name	**Database Concepts and Technologies**	**Type of Course**	Integrated
Course Code	**25CS31I**	**Contact Hours**	8 per week
Teaching Scheme	4:0:4	**Credits**	6
CIE Marks	50	**SEE Marks**	50 (Theory)

1. Rationale:

This course is designed to equip students with a comprehensive understanding of modern database systems by integrating essential theoretical concepts with practical skills. The course focuses on enabling students to conceptualize, design, and implement both relational and NoSQL databases, with an emphasis on performing basic CRUD (Create, Read, Update, Delete) operations.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Analyse real-world database requirements and design a structured ER diagram that accurately represents entities and relationships.
CO-02	Translate the ER diagram into a relational schema and verify its compliance with integrity constraints and normalize the design up to the third normal form (3NF)
CO-03	Create database objects and perform data manipulation operations using suitable tool.
CO-04	Query the database to retrieve the required information using suitable SQL statements.
CO- 05	Design and implement a NoSQL database, and perform fundamental CRUD operations.

3. Course Content

Week	**CO**	**PO**	**Lecture(4HRS) (Knowledge Criteria)**	**Practice(4HRS) (Performance Criteria)**
1	**1**	**1,7**	**Introduction to Database Systems** ▪ Overview of databases and their significance ▪ Types of database systems (RDBMS, NoSQL, Distributed, and Cloud Databases.) ▪ Advantages of database systems over file systems ▪ Overview of Database Management Systems	• Setting up database environments (MySQL/PostgreSQL and MongoDB Atlas) • Explore the interface (MySQL Workbench/ PostgreSQL PgAdmin and local or cloud MongoDB)

			(definition, key features and examples)	
2	**1**	**1,7**	**Database Models and Architecture** ▪ Database models: ER, Relational, and NoSQL ▪ Three-tier architecture and DBMS components ▪ Data Independence: Logical and Physical	a. Consider a database scenario and identify the levels of abstraction (external, conceptual, and internal). b. Analyze a case study where a database underwent schema evolution (e.g., adding/removing tables or columns). Identify how logical independence was maintained. **c.** Study a database migration example where physical storage was optimized (e.g., moving to SSDs or cloud-based storage) without affecting application functionality. d. Set up database connections (MySQL/PostgreSQL/MongoDB) in Python using appropriate modules.
3	**1**	**1,2,3,4,7**	ER Modeling ▪ Introduction to Database Design ▪ Importance of ER Modeling in Database Systems ▪ Key ER Modeling Concepts: – Entities, Attributes, and Relationships – Primary Keys and Foreign Keys – Cardinality and Participation	a. Identify the ER elements for the given requirements. b. Use online tools (Lucidchart, draw.io or similar tools) to create basic entity diagrams. c. Design an ER diagram for a library or e-commerce system. d. Develop an ER model for an online library system like Kindle or Audible, focusing on user subscriptions and book rentals e. Design a comprehensive ER diagram for a healthcare management system, including cardinalities and participation constraints for entities like "Doctor," "Patient," "Appointment," and "Medicine." f. Use database modeling tools like MySQL Workbench/ PostgreSQL PgAdmin to create ER diagrams and validate them against requirements.

4	**2**	**1,2,3,4,7**	**Relational Model and Integrity Constraints** ▪ Introduction to Relational Model ▪ Integrity Constraints ▪ Steps for Converting ER Diagrams into Relational Schema	a. Convert ER model to Relational model b. Translate any given ER diagram into Relational schema c. Use MySQL Workbench or PgAdmin to create database objects and apply integrity constraints through the GUI, without writing SQL statements
5	**3**	**1,2,3,4,7**	**Structured Query Language (SQL) – Basics** ▪ Introduction to SQL; ▪ Categories of SQL Commands with definition and syntax ▪ Data types in SQL ▪ Update anomalies	Create database, tables and enforce data integrity constraints using DDL commands. Perform basic CRUD operations through programming language such as PYTHON. ▪
6	**2**	**1,2,3,4,7**	**Functional Dependency(FD)** ▪ Definition and examples. ▪ Types of functional dependencies ▪ Role of FD in normalization. **Normalization** ▪ Purpose and benefits of normalization. ▪ Normal forms: 1NF, 2NF, 3NF	▪ Identify Functional Dependencies in a Table ▪ Consider relations and apply normalization rules to bring them to 1NF, 2NF, and 3NF. ▪ Consider an unstructured e-commerce database and apply normalization techniques to eliminate redundancy while maintaining data integrity.
7	**4**	**1,2,3,4,7**	**Querying Databases** ▪ SQL Clauses: Where, Distinct, Order By, Limit; ▪ SQL Operators: Logical Operators, Comparison Operators ▪ Aggregate functions	▪ Write SQL queries to filter, sort, and limit data using clauses. ▪ Use operators to refine queries and match complex conditions. ▪ Apply aggregate functions to summarize data and perform calculations.
8	**4**	**1,2,3,4,5,7**	**Advanced Database Querying** ▪ Joins: INNER, OUTER, CROSS ▪ GROUP BY and HAVING clauses ▪ Subqueries: Single-row subqueries; Multi-row subqueries; Scalar	▪ Perform different types of joins to combine data from multiple tables. ▪ Group data and filter the results using GROUP BY and HAVING clauses. ▪ Write and optimize subqueries for filtering and data retrieval.

			subqueries; Correlated subqueries; ▪ Views ▪ Query optimization	▪ Create and use views to simplify complex queries and manage data.
9	**3**	**1,2,3,4,7**	**Transactions and Concurrency** Introduction to Transactions Concurrency Control: ▪ Need for Concurrency Control; ▪ Concurrency Issues; ▪ Concurrency Control Techniques; TCL commands DCL commands	▪ Create, manage, and roll back transactions ▪ Understand and apply locking mechanisms to control concurrent access to data. ▪ Execute DCL commands to provide privileges to different users
10	**5**	**1,2,3,4,7**	**Introduction to NoSQL Databases** ▪ Overview; Key Features; Types; Advantages and Disadvantages; ▪ CAP theorem ▪ Comparison between RDBMS and NoSQL ▪ Overview of MongoDB	MongoDB Shell Basic commands : show dbs, use <dbname>, show collections
11	**5**	**1,2,3,4,7**	**MongoDB Data Model and CRUD Operations** MongoDB Data Model: Collections and Documents; Data types CRUD Operations ▪ Create: *insertOne(), insertMany()* ▪ Read: *find(), findOne()* ▪ Update: *updateOne(), updateMany()* ▪ Delete: *deleteOne(), deleteMany()*	• Create MongoDB for E-Commerce Website. • Perform CRUD operations
12	**5**	**1,2,3,4,7**	**Aggregation in MongoDB** ▪ Aggregation Pipeline ▪ Stages in the Aggregation Pipeline ▪ Expression Operators	Write the aggregation pipeline
13	**1,2,3,4,5**	**1,2,3,4,7**	**Mini Project**	Students will design, implement, and optimize a database system for a real-world application of their choice, integrating SQL and NoSQL databases where applicable.

4. References:

Sl No	Description
1	"Database System Concepts" – Abraham Silberschatz, Henry F. Korth, S. Sudarshan
2	"Fundamentals of Database Systems" – Ramez Elmasri and Shamkant B. Navathe
3	"SQL for Data Analysis" – Cathy Tanimura
4	"Learning SQL" – Alan Beaulieu
5	"NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" – Pramod J. Sadalage and Martin Fowler
6	"MongoDB: The Definitive Guide" – Kristina Chodorow
7	MySQL Official Docs: https://dev.mysql.com/doc/
8	W3Schools SQL Tutorial: https://www.w3schools.com/sql/
9	PostgreSQL Documentation: https://www.postgresql.org/docs/
10	MongoDB Documentation: https://www.mongodb.com/docs/
11	"Databases and SQL for Data Science" – Coursera
12	"Introduction to NoSQL Databases" – Udacity

5. Suggestive Online courses

Sl no	Topic Name	Reference Courses	Self Assessment Link	Source
1	Introduction to Database Systems	TOC - Database Management System Part - 1 | Infosys Springboard	AssessmentDB1 - Viewer Page | Infosys Springboard	Infosys Wingspan
2	Database Models and Architecture	TOC - Database Management System Part - 2 | Infosys Springboard	AssessmentDB2 - Viewer Page | Infosys Springboard	Infosys Wingspan
3	ER Modeling	TOC - Database Management System (DBMS) and SQL : Complete Pack | Infosys Springboard		Udemy
4	Relational Model and Integrity Constraints			Udemy
5	Structured Query Language (SQL) – Basics	TOC - Database Management System Part - 1 | Infosys Springboard	AssessmentDB1 - Viewer Page | Infosys Springboard	Infosys Wingspan
6	Functional Dependency(FD) and Normalization	TOC - Database Management System (DBMS) and SQL : Complete Pack | Infosys Springboard		Udemy
7	Querying Databases	TOC - Database Management System Part - 1 | Infosys Springboard	AssessmentDB1 - Viewer Page | Infosys Springboard	Infosys Wingspan
8	Advanced Database Querying	TOC - Database Management System Part - 1 | Infosys Springboard	AssessmentDB1 - Viewer Page | Infosys Springboard	Infosys Wingspan
		TOC - Working with Subqueries | Infosys Springboard		Skillsoft
		TOC - SQL Views | Infosys Springboard		Skillsoft
9	Transactions and Concurrency	TOC - Database Management System (DBMS) and SQL :		Udemy

		Complete Pack | Infosys Springboard		
10	Introduction to NoSQL Databases	TOC - Database Management System Part - 2 | Infosys Springboard	AssessmentDB2 - Viewer Page | Infosys Springboard	Infosys Wingspan
11	MongoDB Data Model and CRUD Operations			Infosys Wingspan
12	Aggregation in MongoDB	TOC - Aggregation Techniques | Infosys Springboard		IIHT
13	Database Design and Implementation	TOC - Manage Data for an Online Grocer Using MySQL Workbench | Infosys Springboard		Coursera

6. Suggested programs

Week	Description
1	• Setting up database environments (MySQL/PostgreSQL and MongoDB Atlas) • Explore the interface (MySQL Workbench/ PostgreSQL PgAdmin and local or cloud MongoDB)
2	Identifying Levels of Abstraction in a Given Database Scenario Scenario: A retail company uses a database to manage inventory, sales, and customer information. The system allows: • Customers to browse products and place orders through an online shopping platform. • Store managers to track inventory levels and restock products. • The finance team to generate revenue reports and analyze sales trends.
3	Design ER model for the below database requirement. 1. A food delivery platform connects customers with multiple restaurants, allowing them to browse menus, place orders, and have food delivered to their location. The database must manage a wide range of data, including restaurant information, menu items, customer profiles, orders, payment processing, and delivery tracking. The system must also support delivery logistics, managing driver assignments, and tracking order status from preparation to delivery. 1. Restaurant Management: ▪ Store restaurant details, including name, location, contact information, and operational hours. ▪ Manage menus with categories, items, pricing, and availability status. 2. Menu Management: ▪ Maintain records of menu items, including ingredients, preparation time, price, and availability. ▪ Support special offers, discounts, and promotions for menu items. 3. Customer Management: ▪ Store customer profiles with information such as name, contact details, addresses, and order history. ▪ Allow customers to save favorite restaurants, past orders, and payment preferences. 4. Order Management:

	▪ Track customer orders from placement through preparation, dispatch, and delivery. ▪ Manage order status updates in real-time, including stages such as "Preparing," "Out for Delivery," and "Delivered." 5. Delivery Management: ▪ Assign delivery drivers to orders based on location, availability, and estimated delivery time. ▪ Track driver locations in real-time for efficient route optimization and status updates. 6. Payment Processing: ▪ Handle various payment methods including credit/debit cards, digital wallets, and cash on delivery. Maintain secure transaction records and support refunds for cancellations or disputes. 2. Design a database for a **To-Do List Application** that helps users manage their tasks effectively. The database should store and organize information about users, tasks, categories, and their relationships. Requirements: The system should store user information such as unique username, email, and password. Users should be able to create tasks with attributes like title, description, priority, status((e.g., Pending, In Progress, Completed), and due date. Users should be able to organize tasks into categories. A task can belong to multiple categories, and a category can include multiple tasks. Each task should be linked to the user who created it.
4	Convert above ER Diagram to Relational Schema.
5	Create Retail_Store_Management Database and Product Table with following requirements: ▪ each product should have a unique identifier ▪ Each product should have a Product Name, which must be **unique** and **cannot be NULL** ▪ Each product should have a Price, and the price must be **greater than zero** ▪ Each product should have a Description, whose value can be null. **Modify Structure**: After creating the table, ▪ **Remove** the Description field from the table structure (as per requirement). ▪ Add Quantity , and the default value for it should be set to **0**. Execute DML Commands on the Created Tables ▪ Insert Data into the Product Table ▪ Update Data in the Product Table ▪ Delete Data from the Product Table ▪ Select Data from the Product Table ▪ Examine update anomalies and understand their impact on data integrity Design a database for an Inventory Management System for a retail store. The system should keep track of the store's products, suppliers, and customer orders. The database must include the following requirements:

<table>
<tr><td></td><td>

Each product has a unique ID, name, category, quantity in stock, price per unit, and supplier details.
Each supplier has a unique ID, name, contact information, and the products they supply.
Each customer has a unique ID, name, contact details, and purchase history.

Each order must have an order ID, customer details, date of the order, products purchased, quantities, and total cost.
</td></tr>
<tr><td>6</td><td>
Normalize given schema up to 3NF
<table>
<tr><th>OrderID</th><th>CustomerName</th><th>ProductNames</th><th>ProductQuantities</th><th>ProductPrices</th><th>TotalAmount</th><th>ShippingAddress</th></tr>
<tr><td>1001</td><td>Alice</td><td>Laptop, Mouse</td><td>1, 2</td><td>1000, 50</td><td>1100</td><td>123 Elm St, NY</td></tr>
<tr><td>1002</td><td>Bob</td><td>Keyboard, Monitor</td><td>1, 1</td><td>80, 300</td><td>380</td><td>456 Maple Ave, CA</td></tr>
<tr><td>1003</td><td>Charlie</td><td>Laptop, Mouse, Keyboard</td><td>1, 2, 1</td><td>1000, 50, 80</td><td>1130</td><td>789 Birch Rd, TX</td></tr>
</table>
<table>
<tr><th>MovieID</th><th>MovieTitle</th><th>Cast</th><th>DirectorName</th><th>Genre</th></tr>
<tr><td>1</td><td>Inception</td><td>Leonardo DiCaprio, Joseph Gordon-Levitt</td><td>Christopher Nolan</td><td>Sci-Fi</td></tr>
<tr><td>2</td><td>The Dark Knight</td><td>Christian Bale, Heath Ledger</td><td>Christopher Nolan</td><td>Action</td></tr>
<tr><td>3</td><td>The Prestige</td><td>Christian Bale, Hugh Jackman</td><td>Christopher Nolan</td><td>Drama</td></tr>
</table>
<table>
<tr><th>BookID</th><th>BookTitle</th><th>Authors</th><th>PublisherName</th><th>AuthorContact</th></tr>
<tr><td>1</td><td>Python Programming</td><td>John Doe, Jane Smith</td><td>TechBooks</td><td>john@tech.com, jane@tech.com</td></tr>
<tr><td>2</td><td>Data Science Basics</td><td>Alice Brown, Bob White</td><td>DataPress</td><td>alice@datasci.com, bob@datasci.com</td></tr>
<tr><td>3</td><td>Web Development</td><td>Alice Brown, John Doe</td><td>WebWorks</td><td>alice@webworks.com, john@webworks.com</td></tr>
</table>
<table>
<tr><th>OrderID</th><th>CustomerName</th><th>ProductNames</th><th>ProductPrices</th><th>ShippingAddress</th></tr>
<tr><td>101</td><td>Alice</td><td>Laptop, Mouse</td><td>1000, 50</td><td>123 Elm St, NY</td></tr>
<tr><td>102</td><td>Bob</td><td>Keyboard, Monitor</td><td>80, 300</td><td>456 Maple Ave, CA</td></tr>
<tr><td>103</td><td>Charlie</td><td>Laptop, Mouse, Keyboard</td><td>1000, 50, 80</td><td>789 Birch Rd, TX</td></tr>
</table>
</td></tr>
<tr><td>7</td><td>
Suggestive Basic SQL Queries on To do list database

Retrieve User Information for specific email id
Count Tasks for all Users
Retrieve the First 5 Tasks
Retrieve Tasks Due Today
Find the Maximum Due Date of Tasks for a Specific User
Count the Number of Pending Tasks for a Specific User
Find the Average Number of Tasks for specfic user

</td></tr>
<tr><td>8</td><td>
Suggestive Advanced SQL Queries on To do list database

Retrieve Tasks with No Categories Assigned.
Count Tasks in Each Category.
Retrieve Categories Without Any Tasks
Retrieve Users Who Have Tasks with High Priority

</td></tr>
</table>

	▪ Retrieve Tasks and Their Categories using JOIN ▪ Retrieve Categories with More Than 5 Tasks Assigned. ▪ Retrieve Users with Their Total Task Count using subquery
9	Create a transaction for the scenario: ▪ A customer wants to transfer money from their savings account to their checking account. ▪ If any of the steps fail (for example, if there is insufficient balance in the savings account), roll back the transaction to ensure that no changes are made to the database. DCL ▪ Grant SELECT privilege on a table to a user ▪ Grant multiple privileges to a user ▪ Grant all privileges on a table to a user ▪ Grant privileges on all tables in a schema ▪ Revoke SELECT privilege from a user ▪ Revoke multiple privileges from a user ▪ Revoke all privileges from a user ▪ Revoke privileges on all tables in a schema
10	MongoDB Shell Basic commands : show dbs, use <dbname>, show collections
11	Create MongoDB for E-Commerce Website with following collections Products: Stores product details like name, description, price, category, etc. Users: Stores user details like username, email, password, etc Orders: Stores order details like user ID, product IDs, order status, etc. ▪ Adding a Product ▪ Adding a User ▪ Adding an Order ▪ Find All Products in a Category ▪ Find a Specific Product by Name ▪ Retrieve Orders for a Specific User ▪ **Find User Details by Username** ▪ Update Product Stock ▪ Update User Address ▪ Delete a Product ▪ Delete an Order
12	Write the aggregation pipeline ▪ to calculate the total sales for each product within a specified date range. ▪ to calculate the average price of products in each category ▪ to filter products within a certain price range and then calculate the total number of products in each category.

7. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(20)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

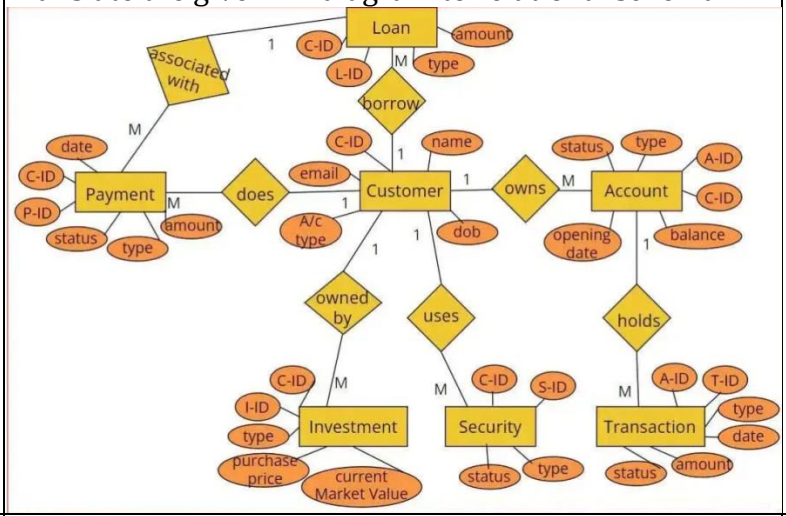
8. SEE - Theory Assessment Methodologies

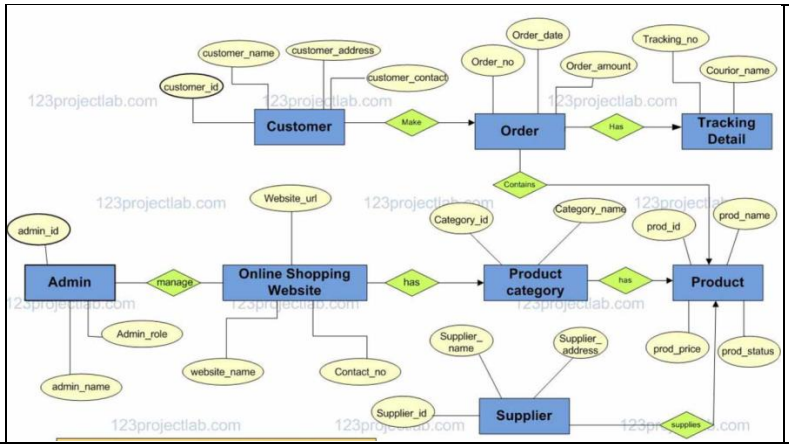
Sl. No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

9. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester -3	
Course Name	Database Concepts and Technologies			Test	I/III
Course Code	25CS31I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks

Section – 1				
1	a. Why is data integrity better maintained in databases than in file systems? Provide examples. – 5 b. Identify and explain the key features of a DBMS. – 10 c. Explain the difference between total and partial participation with examples. – 4 d. Identifying Levels of Abstraction in below given Database Scenario - 6 A university uses a database to manage information about students, courses, and enrollments. The database allows: ▪ Students to view their grades and register for courses via an online portal. ▪ Professors to input grades and access class rosters. ▪ The administration to generate reports on enrollment statistics and course performance.	L2,L3	1	25
2	a. Describe the three-tier architecture of a DBMS and explain the role of each tier. - 5 b. Create an ER diagram for a retail store that includes entities like "Product," "Customer," "Order," and "Supplier." Include attributes and relationships and constraints. – 20	L2,L3	1	
Section – 2				
3	Translate the given ER diagram to relational schema 	L3	2	25
4	Translate the given ER diagram to relational schema	L3	2	

				
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

10. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**3**
Course Name	**Database Concepts and Technologies**			**Test**	**II/IV**
Course Code	**25CS31I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
Develop a database system to support a Ticket Raising Application that facilitates efficient management and resolution of user-reported issues. The system should be capable of handling user information, ticket details, categorization, prioritization, assignment, and tracking of ticket resolution. **Requirements:** a. Store user information such as name, email, phone number, and role (e.g., Customer, Support Agent, Admin). b. Maintain secure login credentials for users. c. Allow customers to raise tickets with details such as: ▪ Issue description ▪ Category (e.g., Technical, Billing, General Inquiry) ▪ Priority level (e.g., Low, Medium, High, Critical) ▪ Assign a unique identifier and timestamp to each ticket. d. Track the status of each ticket (e.g., Open, In Progress, Resolved, Closed) and record timestamps for status changes. e. Store resolution details, including: ▪ Comments or actions taken by the support agent and resolution timestamp. f. Assign tickets to support agents based on expertise, workload, or availability.	**1,2,3,4**	**50**
Scheme of assessment **a. Database Design – 15** ▪ Entity-Relationship (ER) Diagram -5 ▪ Correct conversion of ER diagram into a normalized relational schema. - 5 ▪ Use of appropriate data types for attributes and Implementation of integrity constraints-5 **b. Database Implementation – 20** ▪ Correct creation of database objects using tools - 5		

▪ Application of constraints - 5 ▪ Data Population - 5 ▪ Execution of basic INSERT, UPDATE, DELETE, and SELECT statements - 5 **c. Querying (basic and advanced) - 10** **d. Documentation and Presentation – 5**	
Total Marks	**50**

Sign of the Course Coordinator **Signature of the HOD**

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	3
Course Name	**Computer Networks**	**Type of Course**	Integrated
Course Code	**25CS32I**	**Contact Hours**	8 per week
Teaching Scheme	4: 0:4	**Credits**	6
CIE Marks	50	**SEE Marks**	50 (Theory)

1. Rationale:

This course is designed to equip students with the foundational knowledge and practical skills needed to design, implement, and troubleshoot computer networks in various environments. It addresses the selection of appropriate communication technologies and the resolution of issues like signal loss, latency, and noise to ensure optimal network performance. Learners will gain hands-on experience in configuring IP addressing, routing, subnetting, and protocols like TCP/UDP and DNS using network simulators. The course also emphasizes designing secure and scalable networks, from home office setups to enterprise-level solutions, tailored to specific requirements. By mastering these skills, students will be well-prepared for roles in network administration, engineering, and IT consulting.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Evaluate and select appropriate communication technologies based on network requirements while mitigating transmission impairments
CO-02	Design a network for a given specification by using the right network components, devices, topologies, protocols and software.
CO-03	Identify, diagnose, and resolve common network connectivity and performance issues using troubleshooting tools and methodologies.
CO-04	Configure and manage IP addressing, routing, subnetting, and network services (TCP, UDP, DNS) in a WAN environment using simulation tools.
CO-05	Design and implement simple Home Office network for given specification.

3. Course Content

Week	CO	PO	Lecture(4HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	**1,2**	**1,7**	**Perspectives on Networking –** End user perspectives on Networking and Internet. **Communication Systems -** Basic elements of communication systems with block diagram, List commonly used terms in electronic communication systems.	**1.** Demonstrate different network cables and connectors**.** 2. Demonstrate connecting Computer to home modem/router through Ethernet cable or Wifi 3. Demonstrate automatic and manual IP configuration on both a PC and a mobile device, explaining IP address, subnet

			Data representation, Data flow. **Communication Channels:** **Guided Media (Over view)-** Twisted pair, Coaxial Cable, Optical Fibers. Analog and Digital Signals, Periodic and Non-Periodic Signals, Sine Wave, Phase, Wavelength, Digital Signals, Bitrate, Bit-length, Modulation, Demodulation. **Unguided Media:** Micro Wave, Radio wave, Infrared etc. **Electromagnetic waves** – what are electromagnetic waves and their properties **Electromagnetic spectrum** - classification and its applications.	mask, default gateway, and DNS settings.
2	**1,2,3**	**1,2,7**	**Transmission Impairment –** Attenuation, Distortion and Noise, **Performance -** Bandwidth, Throughput, Latency, Jitter (Basic concepts only). **Transmission Modes –** Parallel and Serial Transmission. Asynchronous and Synchronous Transmission**.** **Overview of Networking**. Categories of networks – Personal Network, LAN, MAN, WAN, Internetworking (Illustrate Network from LAN connected using a HUB to Internetwork). A brief History – ARPANET, MILNET, CSNET, NSFNET, ANSNET. The Internet today – Backbone ISP, Regional ISP, Local ISP. World Wide Web.	1. Explore all ISP in your area/locality and select best internet ISP/plan based on cost and performance. 2. Test the download/upload speed in your computer/mobile phone also check type, bandwidth and ISP. 3. Demonstrate Transmission mode (Video Demonstration) 4. Demonstrate Bandwidth, Throughput, Latency, Jitter (Video Demonstration) 5. Use pathping command to find actual path between source to destination with information about network latency/delay and network loss.
3	**1,2,3**	**1,2,3,7**	**Protocol and Standards-** The communication rules (Method, language, Confirmation) – Protocols. Standards. **Protocol layers** OSI Model : OSI Layers and Their Functions, OSI Layering Concepts	**Network Simulator**: Demonstrate Packet tracer or any other network simulator 1. Installation 2. User Interface3 3. Components and cables 4. GUI and CLI Configuration 5. Configure end devices.

			and Benefits, OSI Encapsulation Terminology**.**	6. Demonstrate Working of all 7 layers by opening a webpage in a simulated network using packet tracer
4	**1,2,3**	**1,2,3,7**	**TCP/IP Networking Model -** Overview of the TCP/IP Networking Model. TCP/IP Application Layer, HTTP Overview, HTTP Protocol Mechanisms. TCP/IP Transport Layer, TCP Error Recovery Basics, Same-Layer and Adjacent-Layer Interactions. TCP/IP Network Layer, Internet Protocol and the Postal Service, Internet Protocol Addressing Basics, IP Routing Basics. TCP/IP Link Layer (Data Link Plus Physical), TCP/IP Model and Terminology, Data Encapsulation Terminology. Names of TCP/IP Messages. Addressing : Physical, Logical, Port	1. Determine the IP Address Configuration of a Computer (Windows) and Test the Network Interface TCP/IP Stack (Ping). 2. Install and configure wireless access point over the LAN. 3. Determine the MAC Address of a Host (PC and Phone). 4. View Wireless and Wired NIC Information and make a table explaining each. 5. Manual and Automatic address assignment (Windows and Android Pone) a) IPv4 address b) Subnet mask c) DNS 6. Demonstrate Addressing in packet tracer
5	**1,2,3,5**	**1,2,3,4**	**Hardware and Software components of Network -** Common network devices - Computers, Access points, Hub, Switch, Router, repeaters, NIC, Modem. LAN Cables – Co-axial, twisted pair, optical fibre, LAN connectors- co-axial cable, and twisted pair cable, optical fibre, Connectors, Firm wares, ISPs. Wired Local Area Networks Protocol and technologies: Ethernet (IEEE 802.3) – Frame Format, Ethernet Evolution, Access Methods: CSMA/CD, CSMA/CA, Fast Ethernet, Gigabit Ethernet. Error Detection in Ethernet. Broadcast and Collision domain.	1. Demonstrate working of common network devices. 2. Install and configure NIC. 3. Twisted Pairs: UTP Cabling Pinouts for 10BASE-T and 100BASE-T, Straight Through Cable Pinout, Crossover Cable Pinout Demonstration. 4. Perform RJ45 crimping for both straight-through and crossover cables, and verify connectivity using a LAN tester. 5. Create and Demonstrate all possible network topologies using simulator. 6. Demonstrate Broadcast and Collision domain (Video demonstration)

			Overview of network topologies - Basic topologies- bus, ring, star, mesh and hybrid**.**	
6	**1,2,3,5**	**1,3,4**	**Wireless LAN (IEEE 802.11) –** Architecture, Frame Format,BSS, ESS, MAC Sublayer, CSMA/CA, Hidden and Exposed Station Problems, **Ethernet access layer devices –** Hub, Switch, The MAC address table, Ethernet Broadcast and Broadcast domain, ARP. **Hierarchical Network Design –** Physical and logical addresses, Benefits of a Hierarchical Design, Access, Distribution, and Core layers (Demonstrate Hierarchical Network Design in packet tracer)	1. Build a physical Ethernet LAN Network and demonstrate file sharing, printer sharing. 2. Install and configure wireless access point over the LAN. 3. Configure and install a ethernet switch/Hub (Use simulator if hardware devices are not available) 4. Create/model a simple Ethernet network using 3 hosts and a switch, Observe traffic behavior on the network and Observer data flow of ARP broadcasts and pings. 5. Demonstrate Collision and Broadcast domain in packet tracer.
7	**2,4,5**	**1,3,4**	Introduction to Network Layer Switching - Circuit Switching, Packet Switching. Packet Switching in Network Layer- Connectionless Service, Connection-Oriented Service. Network Layer Services – Logical Addressing, Services Provided at the Source Computer, Services Provided at Each Router, Services Provided at the Destination Computer. Other Network Layer Issues - Error Control, Flow Control, Congestion Control, Routing	1. Build a simple peer-to peer network and verify physical connectivity and Assign various IPv4 addresses to hosts and observe the effects on network communication. 2. Configure IP addresses of a network (real or simulated) and ping across to test and troubleshoot. 3. View Wireless and Wired NIC Information and make a table explaining each.
8	**2,4,5**	**1,3,4**	IP Addressing, Address Space, Notation. Classful addressing , Two Levelling addressing, Three-Level Addressing, Classless addressing – Subnetting, Subnet mask, Super netting, CIDR, Variable length subnet mask.	1. Subnetting : Your college is given an ip 192.168.100.0/24, you have 5 different departments, design and implement subnetting using packet tracer. Justify subnetting choices based on departmental needs and allocate appropriate IP ranges.

			Special addresses Network Address Translation (NAT). Delivery and forwarding of IP packets - Direct Delivery, Indirect Delivery. Forwarding. Structure of a Router. IP V4 – Datagram, fragmentation.	2. Connect to web server using simulator, Demonstrate how packets are sent across the Internet using IP addresses and working of HTTP services. 3. Implement Network Address Translation using packet tracer
9	**2,4,5**	**1,3,4**	Routing: The Need for Routing - Criteria for Dividing the Local Network. IPv4 Routing - IPv4 Host Routing, Router, Forwarding Decisions and the IP Routing Table, The default gateway, A Summary of Router Forwarding Logic, A Detailed Routing Example. Routing Protocols - IPv4 Routing Protocols – static and dynamic. Other Network Layer Features - Using Names and the Domain Name System, The Address Resolution Protocol, ICMP Echo and the ping Command. IGMP. IPv4 Issues - Need of IPv6, Ipv4 vs IPv6, IPv4 and IPv6 Coexistence. IPv6 features - IPv6 Address Representation. Concept of Virtual LAN's(VLAN's)	1. Implement simple static routing for the practice activity of week 8 (subnetting) 2. Troubleshooting of IP Addressing a) Change a routing table entry b) Wrong address c) incorrect subnet mask. 3. Demonstrate ICMP commands.
10	**2,4,5**	**1,3,4**	Transport Layer Services- Process-to-Process Communication, Addressing: Port Numbers, Multiplexing and Demultiplexing, Flow Control, Error Control – Sliding window. Congestion Control. Connectionless and Connection-Oriented Services. Transport Layer protocols- Simple protocol, Stop and Wait, Go back N.	1. Demonstration of Port addressing and well-known port numbers (HTTP, DNS, SMTP) using packet tracer 2. TCP vs. UDP Comparison (Compare the behavior of TCP and UDP protocols) a. Create a network with two PCs connected through a router. b. Configure one PC to run an HTTP (TCP) server and the other a DNS (UDP) server. c. Use the Web Browser and DNS Client tools on the other PC to access these services.

				d. In Simulation Mode, filter for TCP and UDP packets and compare reliability, acknowledgment, and retransmission behavior. 3. Demonstrate Sliding window, Congestion control, Stop and Wait, Go back N (Only demonstration)
11	**2,4,5**	**1,3,4**	The Client Server relationship – Client Server interaction. URI, URN, URL User datagram protocol – User datagram, UDP services, UDP Applications. Transmission Control protocol: TCP services , features. Popular TCP/IP Applications, Connection Establishment and Termination. Port Numbers – TCP and UDP. Socket pairs, The netstat command.	1. Create a client – server model in simulator and observe the client interaction between the server and PC using packet tracer. a. Create a network with one server, a switch, and two PCs. a) Configure IP addresses for all devices. b) Enable HTTP and FTP services on the server. c) Use the Web Browser and Command Prompt on the clients to access HTTP and FTP services. d) Open Simulation Mode to capture and analyze HTTP GET and FTP request/response packets. 2. Demonstrate Netstat command
12	**2,4,5**	**1,3,4**	**Introduction to application layer:** Client Server paradigm. HTTP DHCP DNP Telnet and SMTP,POP, IMAP, FTP.	1. Set up and configure a DNS server in Packet Tracer, demonstrating the resolution of domain names to IP addresses using nslookup. 2. Use simulator to demonstrate Telnet and SSH 3. Configure and test DHCP using packet tracer. 4. Configure a HTTP service and demonstrate working of HTTP

13	**1,2,3**	**1,2,3,4,7**	**Troubleshoot Common Network Problems -** The Troubleshooting Process, Network Troubleshooting Overview, Gather Information - Nature of problem, Equipment, Configuration and Topology, Previous Troubleshooting. Structured Troubleshooting Methods - Bottom-Up, Top-Down, Divide-and-Conquer, Follow-the-Path, Substitution, Comparison, Educated Guess. Guidelines for Selecting a Troubleshooting Method Troubleshoot Wireless Issues - Causes of Wireless Issues, Authentication and Association Errors**.**	1. Demonstrate troubleshooting Commands with a scenario- ipconfig, ping, netstat, tracert, nslookup. 2. Troubleshooting Common Issues Using Packet Tracer Simulation Mode a) set up a basic network with a router, switch, and PCs. b) Introduce issues like: c)Incorrect subnet mask. d) Disabled router interfaces. e) Use Simulation Mode to observe packet flow: f) Identify where packets drop or fail. g) Fix the configuration and test.

4. References

Sl No	Description
1	Computer Networking: A Top-Down Approach – Kurose and Ross
2	Data Communications and Networking – Behrouz A. Forouzan
3	Computer Networks – Andrew S. Tanenbaum, David J. Wetherall
4	CCNA 200-301 Official Cert Guide – Wendell Odom
5	Network Warrior – Gary A. Donahue
6	TCP/IP Illustrated, Volume 1: The Protocols – W. Richard Stevens
7	Mastering Python Networking – Eric Chou
8	https://www.netacad.com/
9	https://www.coursera.org/learn/computer-networking
10	https://www.udemy.com/course/complete-networking-fundamentals-course/
11	https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-networking.html
12	https://www.youtube.com/c/NetworkChuck
13	https://www.packettracernetwork.com/
14	https://www.wireshark.org/docs/wsug_html/
15	https://www.rfc-editor.org/

5. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE - Theory Assessment Methodologies

Sl. No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -3**	
Course Name	**Computer Networks**			**Test**	**I**
Course Code	**25CS321**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a) Illustrate Communication system with block diagram. 5 Marks b) Discuss optical fibers, Why are optical fibers preferred for long-distance communication? 10 Marks c) Why is serial transmission preferred for long-distance communication? 10 Marks	L2 L3 L3	1 1 1	25
2	a) How does data represented in computer? 5 Marks b) Why is line-of-sight important for microwave communication? 5 Marks c) What is bitrate and bit length? How they are connected to network speed? 5 Marks d) Illustrate different categories of networks. 10 Marks	L3 L3 L3 L2	1 1 1 1	
	Section – 2			
3	a) Illustrate the seven layers of the OSI model. 10 Marks b) What is encapsulation in the context of the OSI model? Why is encapsulation essential for data transmission in a network? 10 Marks c) Illustrate main functions of Network layer in OSI model. 5 marks	L2 L2 L2	2 2 2	25
4	a) Illustrate working of TCP/IP with diagram 10 Marks b) Differentiate between Physical, Logical, Port addressing 10 Marks c) Discuss the purpose of routing in the network layer 5 mark	L3 L3 L3	2 2 2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**3**
Course Name	**Computer Networks**			**Test**	**IV**
Course Code	**25CS32I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**

Your college is given an IP 192.168.100.0/24, you have 5 different departments and each department needs separate subnetwork, design and implement the network using packet tracer. Or Design a network with minimum 2 routers and 4 switches, configure network and demonstrate static routing using packet tracer.	**2,5**	**50**
Scheme of assessment a) Design IP Address scheme (Written)-15 b) Network design (Creating topology, configuring devices, IP assignment etc)-20 c) Verify connectivity / Routing-10 d)Demonstrate working / Troubleshoot if any error-05		
Total Marks		**50**

Signature of the Course Coordinator **Signature of the HOD**

9. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2
08	Crimping Tool Kit		2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	3
Course Name	**Data structures with Python**	**Type of Course**	Integrated
Course Code	**25CS33I**	**Contact Hours**	7 Hours per week
Teaching Scheme	3:0:4	**Credits**	5
CIE Marks	50	**SEE Marks**	50 (Practice)

1. Rationale:

The course is designed to provide learners with a comprehensive understanding of how data is organized, stored, and manipulated in computational systems. By covering both linear and nonlinear data structures, as well as Python's built-in data structures, the course equips students with the knowledge and practical skills necessary to design efficient algorithms and solve computational problems effectively.

2. Course Outcomes : At the end of the Course, the student will be able to:

CO-01	Analyze and compare different data structures to determine their efficiency and suitability for various computational problems.
CO-02	Design and implement solutions for computational problems by selecting appropriate built-in data structures.
CO-03	Develop a solution by integrating appropriate data structures and error-handling techniques.
CO-04	Explain and apply Abstract Data Types (ADT) such as Stacks, Queues, Lists, and Trees in problem-solving scenarios.
CO-05	Design and implement Abstract Data Types (ADT) using User-Defined Data Structures (UDT) for real-world problem-solving.

3. Course Content

Week	CO	PO	Theory	Practice
1	1	1,7	**Introduction to Data Structures** **Classification**: Built in data structures User defined data structures Linear and non linear **Operations on data structures**: Traversal, Insertion, Deletion, Searching, Sorting, and Merging.	Recap of python concepts [practice programs using functions] Explore and document real time applications of Built-in data structures and user defined data structures.
2	1,2	1,2, 3,4	**Sets:** ▪ Characteristics and uniqueness of elements.	Practice programs to understand the concept of ▪ Sets and its operations.

			▪ Set operations: Union, intersection, difference, symmetric difference. **Tuples:** ▪ Characteristics and immutability. ▪ Basic operations: Creating, Accessing elements and slicing ▪ Packing and unpacking tuples.	▪ Tuple and its operations
3	1,2	1,2, 3,4	**Lists:** ▪ Characteristics and use cases. ▪ Common operations: Creating, Indexing, slicing, appending, removing, sorting. ▪ List comprehensions**.** **Dictionaries:** ▪ Key-value pairs: Characteristics and use cases. ▪ Adding, updating, deleting entries. ▪ Iterating through dictionaries. ▪ Dictionary methods (e.g., keys(), values(), items())	Practice programs to understand the concept of ▪ Lists and its operations. ▪ Dictionaries and its operations
4	1,2	1,2, 3,4	**Strings:** ▪ Creating and assigning strings ▪ Indexing **String manipulation and built-in methods:** ▪ Concatenation ▪ Repetition ▪ Slicing **Files**: File attributes; operations	Practice programs to understand the concept of ▪ Strings and its methods. ▪ Files and its operations
5	1	1,2, 3,4	**Error and Exception Handling**: Types of errors; Exception handling	Practice programs to understand the concept of ▪ Exception handling
6	3	2,4	**Introduction to Recursion** ▪ How recursion works (stack memory and execution flow). ▪ Difference between recursion and iteration. **Structure of a Recursive Function** ▪ Base case. ▪ Recursive case. ▪ Identifying recursive patterns in problems. **Use cases of recursion:** Factorial calculation.	Practice programs to understand the concept of recursion
7	1,2, 3	1,2, 4	**Getting started with ADT and UDT:** Abstract data type(ADT):	Practice programs to understand the concept of ADT and UDT.

			Common ADTs: Stacks, Queues, Lists, Trees, Graphs, Hash Tables. User-Defined Data Type(UDT): class Using UDTs to Implement ADTs	
8	1,3	1,2, 3,4	**Stack data structure** LIFO (Last In, First Out) principle Real life examples Stack Operations	Practice programs to understand the concept of Stack data structure and its operations
9	1,3	1,2, 3,4	**Queue Data Structure** FIFO (First In, First Out) principle Real life examples Queue operations	Practice programs to understand the concept of Queue data structure and its operations
10	1,3, 4	1,2, 3,4	Linked Lists ▪ Definition and basic structure (Node, Head, Tail) ▪ Linked list vs. arrays ▪ Types : Singly Linked List, Doubly Linked List, Circular Linked List Singly Linked List ▪ Structure and representation ▪ Basic operations: a. Insertion (at beginning, end, and middle) b. Deletion (from beginning, end, and middle) c. Traversal d. Searching	Practice programs to understand the concept of singly linked list its operations
11	1,3, 4	1,2, 3,4	Doubly Linked List ▪ Structure and representation (with previous and next pointers) ▪ Basic operations: a. Insertion (at beginning, end, and middle) b. Deletion (from beginning, end, and middle) c. Traversal (both forward and backward)	Practice programs to understand the concept of doubly linked list and its operations
12	1,3, 5	1,2, 3,4	Introduction to Trees ▪ Definition and Terminology ▪ Properties of Trees ▪ Binary tree representation in memory **Basic Operations**: ▪ Insertion, Deletion, Searching ▪ Traversal: In-order, Pre-order, Post-order	Practice programs to understand the concept of tree data structure and its operations

13	1,3, 5	1,2, 3,4	Tree Traversal(recursive and iterative): ▪ In-order ▪ Pre-order ▪ Post-order	Practice programs to understand the concept of tree traversals

4. References:

Sl No	**Description**
1	"Python for Everybody" by Dr. Charles Severance
2	“Automate the Boring Stuff with Python" by Al Sweigart
3	"Python Data Structures and Algorithms" by Benjamin Baka
4	https://docs.python.org/3/tutorial/datastructures.html
5	https://www.geeksforgeeks.org
6	https://www.w3schools.com
7	https://www.tutorialspoint.com
8	https://www.khanacademy.org
9	https://www.datacamp.com
10	https://www.programiz.com

5. Suggestive Online courses

Sl no	**Topic Name**	***Reference Courses***	**Self Assessment Link**	**Source**
1	**Introduction to Data Structures Operations on data structures:**	TOC - Data Structures and Algorithms using Python - Part 1 | Infosys Springboard		Coursera
2	**Sets Tuples**	TOC - Data Structures and Algorithms using Python - Part 1 | Infosys Springboard		Infosys Wingspan
3	**Lists Dictionaries**	TOC - Data Structures and Algorithms using Python - Part 1 | Infosys Springboard		Infosys Wingspan
4	**Strings**	TOC - Data Structures and Algorithms using Python - Part 1 | Infosys Springboard		Infosys Wingspan
5	Error and Exception Handling		Basics of Python - Self Assessment - Viewer Page | Infosys Springboard	Infosys Wingspan
6	Introduction to Recursion	TOC - Programming Fundamentals using Python - Science Graduates - Foundation Program | Infosys Springboard	Assessment - Programming Fundamentals using Python - Science Graduates - Viewer Page | Infosys Springboard	Infosys Wingspan
7	**Operators and their Precedence Expressions**			
8	**Flow control Practice programs to understand the Conditional statements**			
9	**Iterative statement**			

10	**Nested loops**			
11	**Functions**			
12	**Modules and Packages**			

6. Suggestive programs

Week	Program description
2	A university needs a system to manage student course enrollments while ensuring data integrity and efficiency. Requirements: Use tuples to store immutable student details such as: – Student ID – Name – Date of Birth – Enrollment Year ▪ Use sets to manage course enrollments: ▪ Each student has a set of enrolled courses to prevent duplicate enrollments. ▪ Perform operations like adding, removing, and finding common courses among students. ▪ Identify students enrolled in multiple courses using set intersection. ▪ Find students who have dropped a course using set difference.
3	A company needs a system to efficiently manage employee payroll, including storing employee details, calculating salaries, and retrieving information as needed. Develop an Employee Payroll Management System that: Uses a list of dictionaries to store multiple employee records. Each dictionary contains: – Employee ID – Name – Designation – Basic Salary – Allowances and Deductions ▪ Calculates net salary based on salary structure and deductions. ▪ Provides functionalities to add, update, delete, and search employee records. ▪ Allows retrieval of employee salary details using Employee ID or Name. ▪ Ensures efficient data access using dictionary-based lookups
4	In large-scale applications, maintaining a log of system events such as user activities, errors, and warnings is crucial for debugging and auditing purposes. The challenge is to efficiently store and retrieve these logs in a structured format. Develop a log management system that: ▪ Captures system events (e.g., login attempts, errors, warnings) as structured log messages using string manipulation. ▪ Stores these log messages persistently in a file for future reference. ▪ Provides a mechanism to read and display logs when needed. ▪ Handles missing or corrupted log files gracefully. In an educational institution, generating and storing student report cards digitally is essential for record-keeping and easy retrieval. The challenge is to process student data, calculate grades, and store the information efficiently in a file.

	Develop a Student Report Card System that: ▪ Takes input for student details (name, roll number, subjects, and marks) and processes it using string manipulation. ▪ Calculates total marks, percentage, and grade based on predefined criteria. ▪ Stores the report card in a structured format within a file. ▪ Allows retrieval and display of student records when needed. ▪ Handles errors such as missing or corrupted report files gracefully.
5	An e-commerce platform allows customers to make payments online using credit cards, debit cards, or digital wallets. However, various issues may arise, such as: ▪ Insufficient funds ▪ Invalid card details ▪ Network failure Payment gateway timeout To ensure a smooth user experience, implement exception handling to catch and manage these errors gracefully.
6	Design and implement a recursive function to navigate through a file system, listing all files and directories within a given path. The function should: ▪ Traverse the directory recursively, printing all files and folders. ▪ Identify subdirectories and go deeper into them until no further subdirectories exist. ▪ Handle cases where directories may be empty. A robot needs to find its way out of a maze represented as a 2D grid. The robot can move up, down, left, or right, but it cannot pass through walls. The goal is to determine a path from the start position (S) to the exit (E) using recursion.
7	Design and implement a UDT-based system for a library to manage books, where: ▪ Each book is represented as an object with attributes like title, author, ISBN, available, and issued to. ▪ The system should allow adding books, issuing books, returning books, and checking book availability. Design and implement a UDT-based system for a hospital to manage patient records, where: ▪ Each patient is represented as an object with attributes like name, age, patient_id, disease, and admitted status. ▪ The system should allow admitting new patients, discharging patients, and updating patient information.
8	Design and implement a stack-based system to manage browser history navigation. The system should: ▪ Store visited web pages in a stack as the user browses. ▪ Allow the user to go "Back" to the previous page. ▪ Allow the user to go "Forward" if they had previously gone back. ▪ Display the current page and maintain navigation history. Design and implement a stack-based system to manage undo/redo functionality in a text editor. The system should: ▪ Store text changes in a stack as users type. ▪ Allow users to undo the last change and revert to the previous state. ▪ Allow users to redo an undone action. ▪ Maintain a record of changes to support multiple undo and redo operations.
9	Design and implement a queue-based system to manage print jobs in a printer. The system should:

	▪ Accept print requests from multiple users. ▪ Process print jobs in FIFO order (First job submitted should be printed first). ▪ Allow users to view the pending queue of print jobs. ▪ Remove a job from the queue once printing is completed. Design and implement a queue-based system to manage customer support tickets. The system should: ▪ Allow customers to submit support tickets with a brief description. ▪ Process tickets in FIFO order (First submitted ticket should be resolved first). ▪ Allow customers to view the current queue of pending tickets. ▪ Remove a ticket from the queue once it has been resolved.
10	Problem Statement: Design and implement a Singly Linked List (SLL) to manage a real-time customer support ticketing system. The system should support the following operations: ▪ Submit Ticket – Add a new ticket to the queue with customer details and issue description. ▪ Resolve Ticket – Remove the first ticket from the queue when resolved. ▪ Next Ticket – Display the next ticket without removing it. ▪ Display Queue – Show all pending tickets. Problem Statement: Design and implement a Singly Linked List (SLL) to manage a real-time ride-sharing queue. The system should support the following operations: ▪ Add Rider – Insert a new rider at the end of the queue when they request a ride. ▪ Assign Rider – Remove a rider from the front of the queue when a driver is available. ▪ Next Rider – Display the next rider without removing them. ▪ Display Queue – Print the list of all riders waiting for a ride.
11	Design and implement a Doubly Linked List (DLL) to manage a web browser's navigation history. The system should support the following operations: ▪ Visit Page – Add a new webpage to the history. ▪ Back – Move to the previous webpage in history. ▪ Forward – Move to the next webpage in history. ▪ Display History – Show all visited pages in order. Design and implement a Doubly Linked List (DLL) to manage a music playlist. The system should support the following operations: ▪ Add Song – Add a new song to the playlist. ▪ Play Current Song – Display the currently playing song. ▪ Next Song – Move to the next song in the playlist. ▪ Previous Song – Move to the previous song. ▪ Display Playlist – Show all songs in the playlist in order.
12	Design and implement a tree-based data structure to manage an organization hierarchy. The system should support the following operations: ▪ Add Employee – Add a new employee under a manager. ▪ Display Organization Hierarchy – Show the reporting structure in a tree format. ▪ Find Manager – Find the reporting manager of a given employee. Design and implement a tree-based data structure to manage a file system hierarchy. The system should support the following operations: ▪ Add Folder/File – Add a new file or folder inside a specific directory. ▪ Display File System – Show the complete directory structure in a tree format.

	▪ Search File – Find a file and return its path.
13	Design and implement a tree-based data structure to represent an organization hierarchy. Implement tree traversal methods to display employees in different orders: ▪ Pre-order Traversal (Top-Down View) – Start from the CEO and visit managers/employees before subordinates. ▪ Post-order Traversal (Bottom-Up View) – Start from employees and move up to managers and then the CEO. ▪ Level-order Traversal (Breadth-First Search) – Traverse the hierarchy level by level. Design and implement a tree-based data structure to represent a file system. Implement different tree traversal methods to explore files and directories: ▪ Pre-order Traversal (For File Path Display) – Displays the directory structure from root to leaf. ▪ Post-order Traversal (For Deleting a Directory) – Ensures that subdirectories and files are deleted before the parent directory. ▪ Level-order Traversal (For Folder Size Calculation) – Helps in calculating disk usage by processing folders level by level.

7. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 TheoryTest	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
Total					**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

8. SEE – Practice Assessment Methodologies

Sl NO	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

9. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -3rd**	
Course Name	**Data structure with python**			**Test**	**I/III**
Course Code	**25CS331**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section – 1			
1	a)What is the difference between a tuple and a list in Python? Give a scenario where you would prefer using a tuple over a list. (5marks) b)Write a Python function that takes a list of integers as input and returns a new list where:(10 marks) ▪ All even numbers are squared. ▪ All odd numbers are multiplied by 3. **Example Input**: [1, 2, 3, 4, 5, 6] **Example Output**: [3, 4, 9, 16, 15, 36] c)Write a Python function that takes a list of tuples, where each tuple contains the name and the age of a person. The function should return the name of the oldest person.(10marks) **Example Input**: [('John', 45), ('Alice', 50), ('Bob', 30)] **Example Output**: 'Alice'	L2,L3	1	25
2	a)Describe the basic operations that can be performed on a **set** in Python. Provide examples of how to compute the intersection, union, and difference between two sets.(5marks) b)Write a Python function that takes a list of tuples, where each tuple contains the name and the age of a person. The function should return the name of the oldest person.(10 marks) **Example Input**: [('John', 45), ('Alice', 50), ('Bob', 30)] **Example Output**: 'Alice'	L2,L3	1	

	c)Write a Python function that takes two sets and returns a tuple containing:(10 marks) ▪ The intersection of the two sets. ▪ The union of the two sets. ▪ The difference between the first set and the second set. **Example Input**: set1 = {1, 2, 3, 4, 5} set2 = {4, 5, 6, 7} **Example Output**: ({4, 5}, {1, 2, 3, 4, 5, 6, 7}, {1, 2, 3})			
	Section – 2			

3	**Analyze the code and write the output**
	def sum_nested_elements(nested_structure):
	total_sum = 0
	stack = [nested_structure] # Initialize the stack with the outer structure
	print(stack)
	while stack:
	current = stack.pop() # Take the last element from the stack
	print(current)
	for item in current:
	if isinstance (item, (list, tuple)): # If it's a list or tuple, add to the stack
	print(item)
	stack. append(item)
	print(stack)
	elif isinstance (item, (int, float)): # If it's a number, add to the total
	total_sum += item
	return total_sum
	
	nested_list = [1, 2, (3, 4), [5, (6, 7)], 8]
	total = sum_nested_elements(nested_list)
	print(f"Total sum of elements: {total}")
		L3	2	25

4	a)Create a Python program to Create a collection named trees – Oak, Pine, Mango, Neem, Maple, Teak, Mango, Neem, Maple, Pine, Mango, Oak, Pine, Mango, Peepal, Banyan, Sal, Compute and display the result with relevant messages for following: 1. Ask the user to input a tree name to count how many times it appears in the collection. 2. Ask the user to input a tree to find its position in the collection. 3. Get the last 5 tree names from the collection	L3	2	25

	4. Find the unique tree names from the collection Suggest the suitable python concept to solve above and justify your selection. b)Swami Vivekananda was a renowned Indian Hindu monk and one of the most influential spiritual leaders of the 19th century. For the above text, create a python program to 1. measure the length of the text. 2. count the number of words in the text. 3. Count the occurrence of these stop words – of, the, at, in, with 4. Capitalize these words and display the text – Narendranath, Vivekananda 5. Find numeric and alphanumeric words from the above text. Suggest the suitable python concept to solve above and justify your selection.			
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

10. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	
Course Name	**Data structure with python**			**Test**	
Course Code	**25CS331**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
Problem Statement: Text and Numerical Data Separation with Word Classification Write a program that processes a given paragraph and performs the following tasks: 1. Read Input Data: ○ Read a text file containing text data. Each line of the file may contain multiple words. 2. Process Words: ○ Extract all the words from the file, ignoring special characters and punctuation. 3. Separate Text and Numerical Data: ○ Parse the paragraph to separate numerical data (e.g., numbers, digits) and text (e.g., words or characters). 4. Classify Words: ○ From the separated text, classify words into the following categories:	**1,2,3**	**50**

▪ Uppercase Words: Words containing only uppercase letters (e.g., "HELLO"). ▪ Lowercase Words: Words containing only lowercase letters (e.g., "world"). ▪ Mixed Case Words: Words containing both uppercase and lowercase letters (e.g., "Python"). **5. Output:** ○ The original paragraph. ○ A list of numerical data extracted from the paragraph. ○ Separate lists for Uppercase, Lowercase, and Mixed Case words.		
Scheme of assessment (customize scheme as per requirement of your problem statement) a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10 b) Implementation and Execution - 30 c) Best Practices (Code Readability and Error Handling)- 10		
	Total Marks	**50**

Signature of the Course Coordinator **Signature of the HOD**

11. SEE- Model Practice Question Paper

Program	**Computer Science and Engineering**		**Semester**	**3**
Course Name	**Data Structures using python**	**Course Code 25CS33I**	**Duration**	**180 min**
Questions			**CO**	**Marks**
1. Design a student grade management system for your college to track and manage student grades. Let the system designed by you include the following requirements. • Add new student records with their name, subject, and grade. • Update a student's grade if there was a mistake. • Search for a student's grade by their name or subject. • Calculate and display the average grade for all students. The system should handle errors, such as trying to update or search for a non-existent student, and offer a simple menu for interaction.			**1, 2, 3, 4, 5**	**50**
Scheme of assessment (customize scheme as per requirement of your problem statement) a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10 b) Implementation and Execution - 30 c) Best Practices (Code Readability and Error Handling)- 10				
			Total Marks	**50**

1) Signature of the Examiner 1 **2) Signature of the Examiner 2**

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**3**
Course Name	System and Network Administration	**Type of Course**	Integrated
Course Code	**25CS34I**	**Contact Hours**	7 Hours per week Total 91
Teaching Scheme	3:0:4	**Credits**	5
CIE Marks	**50**	**SEE Marks**	**50 (Practice)**

1. Rationale:

The course System and Network Administration is designed to equip learners with the skills and knowledge required to manage, maintain, and troubleshoot computer systems and networks. This course integrates theoretical knowledge with hands-on practice, preparing learners for vital roles in IT infrastructure management. As technology forms the backbone of modern organizations, proficient administrators are critical to ensuring systems remain secure, reliable, and efficient.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Analyze hardware specifications and assemble a computer based on given technical requirements.
CO-02	Install an Operating system, configure and modify administration settings, carry out basic troubleshooting and performance monitoring
CO-03	Manage File systems through formatting, partitioning , data backup and recovery techniques.
CO-04	Configure Network, check and update network administration settings, provide basic network security and sharing activities.
CO-05	Design an optimized lab layout and implement e-waste disposal procedures in compliance with regulatory guidelines.

3. Course Content

Week	CO	PO	Theory	Practice
1	1	1	Computer hardware categories and their usage. External components Input devices Output devices Internal components Processing devices Storage devices Network devices	**Hardware Identification:** 1. Computer Case- Types, Features 2. Identify the front and rear panel controls and ports on a PC cabinet. 3. Exploring the internal components of a computer case 4. Identify different cables and

			Cables and connectors used with computers	CMOS battery, front and back panel connectors, power connectors. 5. Identify Storage devices(Hard disk, SSD), PSU, Cooling systems and their connection to the motherboard. 6. Identify the networking devices connected to the computer.
2	1	1,5	**Power supply unit - SMPS** What is SMPS? Need for SMPS SMPS parts and connectors. Working principles of SMPS Types of SMPS SMPS voltage color codes Measuring SMPS voltage levels. How to choose right power supply for PC Common issues in SMPS. Troubleshooting common problems in SMPS	1. Unmount the power supply from the PC cabinet. Identify the types of output connectors. 2. Identify output voltages using color coding. 3. Measure voltage levels using a multimeter. 4. Mount the power supply into the PC cabinet, connect different components and test the PC. 5. Troubleshoot Power supply through SMPS fan. 6. Diagnose power supply faults using PSU Tester.
3	1	1,5	**Exploring Motherboard** What is a motherboard? Types of motherboard(Form factor) Components of Motherboard How does a motherboard work? Main functions of motherboard Popular Motherboard Manufacturers. Desktop motherboard Vs Laptop motherboard Common motherboard issues and maintenance. Mother troubleshooting techniques Usage of CPUID CPU-Z tool.	1. Identify various components of the motherboard.(Draw motherboard layout) 2. Compare motherboards of various types 3. Precautions to be taken before removing the motherboard from the PC cabinet. 4. Using the CPUID CPU-Z tool, find different features of the CPU. 5. Check the Electric flow path and data flow path 6. Using Windows resource monitor to troubleshoot performance issues. 7. Using the CPUID CPU-Z tool, identify the CPU cache features of your working system.
4	1	1,4,5	**Chipset Architecture and components** Northbridge	1. Hands-on identification of Northbridge and Southbridge chips on motherboards

			Southbridge Booting process (FIRMWARE) POST: BIOS/UEFI Bootloader Execution OS Kernel Loading System Processes Initialization User Authentication	2. Identify the components connected to the north bridge and south bridge. 3. Implement basic power management features using BIOS/UEFI settings. 4. Checking System Information in BIOS/UEFI and Change the Boot Order 5. Analyze the chipset design of a specific device (e.g., smartphone, laptop) and suggest improvements
5	2	1,2,5	**Operating System Installation** Creating OS Image (Bootable Device) Operating system : Functions and types Understanding file systems and partitions	1. Install and boot Windows 10/11 or any Linux OS 2. Configure essential OS settings including user interface customization and system diagnostics. a. Inverting and rearranging screen b. Testing Microphone 3. Document OS installation procedures a. Windows / Linux
6	2	1,2,4,5	**Windows Command line interface** Basic Navigation commands File and directory management commands writing simple PowerShell scripts. File security. User Account and Permission Management User Profile Issues: Create a New Profile,Profile Repair	1. Using command prompt or powershell, a. Execute basic navigation commands b. Execute file and directory management commands. 2. Use command line to encrypt and decrypt files and folders 3. Adding, modifying, and deleting user accounts. 4. Create and manage user accounts on a Windows system: a) Using the Control Panel b) Using PowerShell c) Verify the user account creation by logging in with the new credentials and noting any administrative privileges. d) Document the steps taken for the above task

				5. PowerShell scripting: Reading from and writing to the console and files
7	2	1,2,4,5	**OS configuration and Administration** System Startup Issues and solutions: Safe Mode, Automatic Repair, System Restore, system reset, Image recovery Performance Issues: Task Manager, Resource Monitor, Disk Cleanup, Check for Malware, Update Drivers, Check Disk Health	1. Solve system startup issues using Safe Mode, Automatic Repair, System Restore, system reset, Image recovery 2. Check and/or Solve Performance Issues using Task Manager, Resource Monitor, Event Viewer, Disk Cleanup, Check for Malware, Update Drivers, Check Disk Health 3. Document the concept of Blue Screen of Death (BSOD), identify common causes, analyze a specific error code, explore basic diagnostic tools, and suggest preventive measures
8	3	1,2,4,5	**Basics of File Systems** ▪ File system types: FAT32, NTFS, exFAT ▪ Structure and functions ▪ File properties **System Configuration** ▪ Disk partitioning (MBR vs GPT), ▪ Drive formatting, ▪ Disk shrink/extend **Data Recovery Techniques** ▪ Deleted file recovery ▪ Registry restoration ▪ app recovery ▪ System Restore ▪ Windows Recovery **Windows Update Management** ▪ Manual vs Auto Updates ▪ Update testing, ▪ Rollback ▪ Update history	1. Create and Format a New Partition 2. MBR vs GPT Identification 3. Resize (Shrink/Extend) a Partition 4. Explore Command Line Disk Tools 5. Delete a file and recover it using Windows File Recovery Tool 6. Registry Backup and Restore 7. System Restore ▪ Create a manual restore point. ▪ Simulate a system issue (e.g., uninstall a program) and roll back using System Restore. 8. Uninstall a program, then use Windows Store or official setup to reinstall it. 9. Windows Update Management ▪ Check and Install Updates Manually ▪ Change Update Settings ▪ View Update History ▪ Rollback an Update
9	3	1,2,4,5	**Monitor and manage**: Performance monitoring; optimize windows services; tune scheduled tasks. Customizing windows desktop. File and System Corruption:	1. Monitor the processes, System performance using Task manager (Windows)/ Process related Commands in Linux.

			System File Checker (SFC) Deployment Imaging Service and Management Tool (DISM) Check Disk Utility Drivers Management: Device drivers, Network drivers, System drivers	2. Troubleshoot the system delaying for booting (disabling unnecessary Startup applications, Services) 3. Work with task scheduler: Windows/Linux 4. Install printer driver, Network driver disable/ enable, 5. System driver update
10	4	1,2,3 ,4,5	**Network Configuration and Management** : IP Addressing , Subnetting, DNS Configuration, Gateway Configuration, VLAN Configuration Network Troubleshooting: Identify the Problem, Establish a Baseline, Check Physical Layer, Verify the Data Link Layer, Test Network Connectivity, Use Traceroute, Diagnose and resolve common IP and DNS-related issues,Examine Firewall and Security Settings Network Security: Firewall, Network profile, Managing wireless network Network Sharing	1. Configure Network Connections, Configure IP Settings, Set DNS Server, Network Adapter Information, Disable or Enable Network Adapter. 2. Ping, Trace Route, Checking Open Ports, Network Statistics, Flush DNS Cache, Reset TCP/IP Stack, 3. Windows Firewall Configuration, Configure Network Profiles, Manage Wireless Networks 4. Share any Files and Folders with other System in the network. 5. Establish and manage remote desktop connections securely.
11	4	1,2,4 ,5	**Network Monitoring and Reporting**: Monitor Network Activity, View Network Usage Advanced Network Configuration: Create and Manage Virtual Networks (Hyper-V Manager), Manage Routing (Modem) and Remote Access Virtual Private Network: Definition, Types of VPN, Implementation	1. Monitor network activity using Task manager and resource monitor 2. Install a VM(Hypervisor) document VM Network Utilization Metrics 3. Run basic network commands in guest OS installed on VM 4. Download and install any third party VPN app and connect internet through VPN
12	5	7,8	**E-waste management:** E-Waste: Definition and types of electronic waste (e.g., computers, smartphones, home appliances). Global statistics on e-waste generation. Environmental and health impacts of improper e-waste management.	1. Visit https://greene.gov.in and https://kspcb.karnataka.gov.in/ to find the latest regulations and policies taken up by the Government of India. 2. Analyze e-waste management processes through on-site visits or virtual case studies.

			E-Waste Management Lifecycle: Lifecycle of electronic devices: production, usage, and disposal. Processes in e-waste recycling: collection, sorting, and recovery. Policies and Regulations on E-waste: Global e-waste policies National laws and policies on e-waste	3. Document how to reduce carbon footprint
13	1,2,3,4,5	1,2,3,4,5,6	Mini Project Activity- Layout Management Lab setup Management: Resource purchase, Installation, Electrical safety measurement	Develop a comprehensive lab setup plan, including hardware procurement, installation, electrical safety measures, and cybersecurity protocols.

4. References:

Week	Description
01	https://discover.hubpages.com/technology/The-Four-Main-Categories-Of-Computer-Hardware-Parts https://www.javatpoint.com/what-are-the-computer-cables
02	https://www.intel.com/content/www/us/en/gaming/resources/power-supply.html https://en.wikipedia.org/wiki/Switched-mode_power_supply https://www.amd.com/en/resources/support-articles/faqs/GPU-104.html
03	https://www.geeksforgeeks.org/what-is-a-motherboard/ https://www.studocu.com/en-us/document/purdue-university-global/information-technology-concepts/12-exploring-motherboards-processors-and-memory/37160479
04	https://www.baeldung.com/cs/chipset https://www.javatpoint.com/booting-in-operating-system https://learn.microsoft.com/en-us/windows-hardware/drivers/bringup/boot-and-uefi
05	https://www.javatpoint.com/how-to-install-windows-10 https://answers.microsoft.com/en-us/windows/forum/all/steps-to-install-windows-10/a499989c-f31f-49d0-8dbb-3b6425a5f47b https://www.geeksforgeeks.org/how-to-install-ubuntu-desktop/
06	https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands https://www.freecodecamp.org/news/command-line-commands-cli-tutorial/ https://www.geeksforgeeks.org/most-useful-cmd-commands-in-windows/ https://devblogs.microsoft.com/scripting/table-of-basic-powershell-commands/
07	https://learn.microsoft.com/en-us/troubleshoot/windows-client/performance/windows-boot-issues-troubleshooting https://learn.microsoft.com/en-us/troubleshoot/windows-client/performance/windows-startup-issues-troubleshooting https://support.microsoft.com/en-us/windows/system-configuration-tools-in-windows-f8a49657-b038-43b8-82d3-28bea0c5666b
08	https://learn.microsoft.com/en-us/windows-server/storage/disk-management/overview-of-disk-management https://support.microsoft.com/en-us/windows/windows-file-recovery-61f5b28a-f5b8-3cc2-0f8e-a63cb4e1d4c4
09	https://support.microsoft.com/en-us/windows/system-configuration-tools-in-windows-f8a49657-b038-43b8-82d3-28bea0c5666b

	https://support.microsoft.com/en-us/windows/update-drivers-manually-in-windows-ec62f46c-ff14-c91d-eead-d7126dc1f7b6
10	https://learn.microsoft.com/en-us/training/paths/configure-networking-windows-clients/ https://support.microsoft.com/en-us/windows/change-tcp-ip-settings-bd0a07af-15f5-cd6a-363f-ca2b6f391ace https://support.microsoft.com/en-us/windows/setting-up-a-wireless-network-in-windows-97914e31-3aa4-406d-cef6-f1629e2c3721 https://learn.microsoft.com/en-us/windows/security/operating-system-security/network-security/windows-firewall/ https://www.windowscentral.com/8-networking-command-tools-every-user-should-know-windows-10
11	https://www.vmware.com/topics/virtual-networking https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn
12	https://greene.gov.in/ https://kspcb.karnataka.gov.in/

5. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 TheoryTest	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (20)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

- Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7 CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester -III	
Course Name	System and Network Administration			Test	I
Course Code	25CS34I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a) Categorize the given devices/components into multiple categories i. Mouse ii. Phone Display iii. RAM iv. RJ45 v. Flash Drive vi. NIC vii. Printer viii. Projector ix. Cache -05 b) Ajay a computer science diploma II year student. One of his friends wanted to buy a new desktop, so he asked Ajay's suggestions, feeling he might know more about computers. What are your suggestions to him with respect to hardware? -10 c) Compare the hardware components of a computer with your mobile phone. -10	L3 L3 L2	1 1 1	25
2	a) How can you diagnose power supply faults using PSU Tester? 5 b) Rachana was sitting on a plastic chair before trying to check the system unit cabinet. Though the power supply is not connected she got an electric shock. How can you analyze this problem? What are the safety measures for this? (static electricity) -10 c)How does SMPS work? -10	L3 L3 L2	1 1 1	
	Section – 2			
3	a) Anil and Hruthvik are debating whether the north bridge is faster or the south bridge is faster? How can you conclude it? Justify your answer. -05 b) One of the computers in your lab was hanged when Vedant ran application 'X'. He doesn't want to restart the computer but has to kill application 'X' . How can you help him? -10	L3 L3	1 1	25

	c) Khushi wants to install an operating system from his pen drive. But it is not loading. Someone told her to change it to BIOS. She does not understand it clearly. Can you help her with this regard - 10	L3	1,2	
4	a) How can you differentiate single boot virtual machine installation? -5 b) Compare windows/Linux OS with android OS -10 c) Rehman installed an older version of the operating system on his computer. Some of his hardwares like Wi-Fi, is not working. How can you resolve this issue? (Chipset not installed) -10	L2 L2 L3	1,2 1,2 1,2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**1**
Course Name	**System and Network Administration**			**Test**	**II**
Course Code	**25CS34I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
1. You are instructed to install newer version of Operating system in a computer in your lab. Decide the OS, document hardware requirement for the same, before installing the OS check the BIOS/UEFI settings and do necessary updates in it. Install OS on right PC. Create a new account name CSE_LAB1_01 with a strong password	**1,2,3**	**50**
Scheme of assessment a) Choosing OS and document Hardware-10 b) BIOS/UEFI settings-10 b) Installation of OS and related operations-15 c) OS related operations-10 d) Portfolio Evaluation-05		
Total Marks		**50**

Sign of the Course Coordinator **Signature of the HOD**

9. SEE–Model Practical Question Paper

Program	Computer Science and Engineering		Semester	III
Course Nam	System and Network Administration		Marks	50
Course Code			Duration	180 Min
Questions		**Cognitive Levels**	**Course Outcomes**	**Marks**
Section -1	You are tasked with setting up and maintaining a computer system for a small office environment. The goal is to ensure that the hardware components are correctly identified, software is installed and configured, issues are resolved efficiently, and the network is operational for seamless communication. **Requirements:** **Hardware:** Verify hardware specifications and compatibility with the system requirements. **Software Installation:** Install an operating system (e.g., Windows/ Linux) on a computer. Install essential drivers and utility software required for system functionality Diagnose ,resolve and document any hardware and software issues faced **Network Configuration:** Set up a basic wired or wireless network for the office, ensuring connectivity between devices	L3	1,2,3,4,5	45
Section -2	Portfolio Evaluation/Viva	L3		5

1)Signature of the Examiner 1 **2) Signature of the Examiner 2**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers		10
02	PSU Tester		10
03	Multimeter		10
04	Individual components- SMPS/PSU	400 watts	10
05	Individual components- SMPS/PSU	800 watts	10
06	Motherboard	ATX	10
07	Motherboard	Micro ITX 10	10
08	RAM stick	DDR3	10
09	RAM stick	DDR4	10
10	CMOS battery		10
11	Windows 10/11 OS user license for multi users		-
12	POST diagnostic card		10

SEMESTER 4

Government of Karnataka

DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**4**
Course Name	**Linux Administration**	**Type of Course**	Integrated
Course Code	**25CS41I**	**Contact Hours**	8 per week
Teaching Scheme	4:0:4	**Credits**	6
CIE Marks	50	**SEE Marks**	50 (Theory)

1. Rationale:

In today's technology-driven world, Linux servers form the backbone of enterprise IT infrastructure, **Cloud Computing**, and cybersecurity systems. This course is designed to equip learners with essential Linux administration skills to deploy, manage, and secure Linux-based servers effectively. Through real-world scenarios and practical exercises, learners will not only gain technical proficiency but also develop problem-solving skills that are crucial for handling complex IT challenges. the course focuses on industry-relevant best practices to prepare learners for career opportunities in system administration, DevOps, cybersecurity, and **Cloud Computing**.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Install, configure, and manage Linux OS in both virtualized and physical environments.
CO-02	Administer system resources and automate tasks using shell scripting and job scheduling.
CO-03	Secure and harden Linux systems by implementing firewall rules, access controls, and monitoring tools.
CO-04	Set up and manage Linux server infrastructure, including web, database, and file servers.
CO-05	Troubleshoot and optimize Linux systems to ensure performance, reliability, and security.

3. Course Content

Week	**CO**	**PO**	**Theory**	**Practice**
1	**1,5**	**1,2,4, 6,7**	**Introduction to Operating Systems** ▪ Operating System – definition and its role in computing. ▪ Core functions, Types of OS ▪ Linux vs. Other OS (Windows, macOS) ▪ Understanding System Boot Process (UEFI/systemd-boot, Init/Systemd)	▪ Installation of Linux distribution on a Physical Machine by performing a manual installation process. ▪ Install Linux distribution on VirtualBox/VMware. ▪ Navigate the Linux CLI (ls, pwd, cd, mkdir).

			▪ Basic Linux Commands (ls, cd, pwd, mv, etc.) ▪ Virtualization : Definition, Purpose, Benefits, types, Hypervisors. Use Cases in Linux Server Administration	▪ Identify system info (uname -a, lsb_release -a). ▪ Configure systemd-boot boot parameters. **Scenario: A company is setting up a Linux-based infrastructure.** ▪ Install Linux OS (Ubuntu/CentOS) on a VM. ▪ Modify systemd-boot bootloader and recover from a boot failure. ▪ Identify system logs (journalctl -xe, /var/log/syslog). **scenario**: *A system fails to boot due to a misconfigured GRUB. Troubleshoot and restore the bootloader.*
2	**1,2 ,5**	**1,2,3, 4,6,7**	**Kernel and Shell** ▪ OS architecture: Kernel vs. User space. ▪ Kernel types: Monolithic, Microkernel, Hybrid Kernel. ▪ Kernel Module Management (lsmod, modprobe, rmmod) ▪ The role of the Shell: Command-line interpreters (Bash, Zsh, PowerShell). **Introduction to shell scripting** ▪ What is Shell Scripting? ▪ Writing and Executing Shell Scripts ▪ Variables, Input, Conditional Statements, Loops.	▪ Identify kernel version and running processes. ▪ Switch between different shells (`bash`, `zsh`). ▪ Load/unload kernel modules (`lsmod`, `modprobe`). ▪ Write simple shell scripts (Hello World, user greeting, loops). ▪ Check system uptime and notify if it's less than 1 hour. ▪ Create a backup script that runs every hour. ▪ Write a shell script to monitor disk usage and send an alert when usage exceeds 80%

				Scenario: *As an IT administrator, you need to write a startup script that logs system uptime and sends an email alert if uptime exceeds 24 hours.*

3	**1,3**	**1,2,3, 4,5,6, 7**	**Linux File System and Hierarchy**
			▪ Linux Directory Structure
			▪ File System Types (ext4, XFS, Btrfs)
			▪ File and Directory Permissions
			  ▪ Understanding Linux file permissions (rwx).
			  ▪ Permission types for users, groups, and others.
				▪ Explore the Linux file system structure (cd, ls, tree).
				▪ Create, move, copy, and delete files.
				▪ Modify file permissions (chmod, chown).
				▪ Mount and unmount drives (mount, umount).
				Scenario: Your team needs a secure file-sharing system.
				▪ Modify file permissions (chmod 750 file.txt).
				▪ Set up ACLs for shared directories (setfacl -m u:user1:rwx project_dir).
				▪ Format and mount a new disk (mkfs.ext4, mount).
				Scenario: A developer accidentally deleted critical logs. Set up a log directory with restricted access and implement a backup strategy using file permissions and ownership.

4	**1,3**	**1,2,3, 4,5,6, 7**	**User and Group Management** Understanding User Accounts Managing Users and Groups Sudo and Access Control Advanced Access Control using PAM (Pluggable Authentication Modules)	▪ List system users and groups. ▪ Create and manage users (`useradd`, `passwd`). ▪ Add users to groups (`usermod -aG`). ▪ Grant sudo access and configure permissions.

				▪ Implement ACLs (`setfacl`). ▪ Configure user access logs and track unauthorized access attempts using auditd **Scenario: A new employee joins your company. Configure his access.** ▪ Create a new user, set a password, assign to a group. ▪ Grant and revoke sudo privileges. ▪ Restrict SSH access for specific users. ▪ Implement password aging policies. **Scenario**: Your company needs a new user group for web developers with limited access to server files. Configure user groups and permissions accordingly.
5	**1,5**	**1,2,3, 4,6,7**	**Process Management** Introduction to Process Management ▪ Definition of a Process and its Role in OS ; Importance of Process Management ;Process vs. Program Understanding Processes ▪ Types of Processes (System, User) ▪ Process Lifecycle and States ▪ State Transitions and Their Significance; Process vs. Thread Anatomy of a Process ▪ Process Control Block (PCB): Structure and Role ; Process Creation (fork, exec);Process Termination and	▪ List running processes (`ps`, `top`, `htop`). ▪ Start a process in the background (`and`, `nohup`). ▪ Change process priority (`nice`, `renice`). ▪ Terminate processes (`kill`, `pkill`). **Scenario: High CPU usage troubleshooting.** ▪ Analyze CPU load using top and htop. ▪ Change process priority using nice.

			Resource Cleanup ;Parent and Child Processes Managing Processes	▪ Identify high memory usage (ps aux --sort=-%mem). **Scenario**: A critical process is consuming too much CPU. Identify and optimize it without shutting down the entire server.
6	**2,5**	**1,2,3, 4,6,7**	**Job Scheduling and Interprocess Communication** Introduction to Job Scheduling ▪ Importance of Job Scheduling ▪ One-time Scheduling (at, batch) vs. Recurring Scheduling (cron) Automating Tasks with cron ▪ Understanding cron and crontab Syntax ▪ Setting Up, Viewing, Editing, and Removing Cron Jobs Interprocess Communication (IPC) ▪ Overview of IPC and Synchronization ▪ Types of IPC: Pipes, Message Queues, Shared Memory, Semaphores ▪ Choosing the Right IPC Mechanism for a Task	▪ Schedule jobs using at and cron. ▪ Set up a cron job to clean temporary files. ▪ Implement IPC using named pipes. ▪ Monitor background jobs (jobs, fg, bg). ▪ A scheduled backup fails intermittently. Diagnose cron logs and fix the issue. **Scenario:** Automate log management. ▪ Schedule a cron job to delete old logs. ▪ Implement named pipes for inter-process communication. **Scenario:** *The system needs to generate automated reports at midnight daily. Configure a cron job to execute a script that emails the report to the manager.*
7		**1,2,3, 4,6,7**	**Resource Management** ▪ Introduction to Resource Management ▪ What is Resource Management?	▪ Identify system resource usage (top, htop). ▪ Monitor CPU and memory (free -m, vmstat).

	1,5		▪ Role of OS in Managing Resources (CPU, Memory, I/O, Storage) ▪ CPU Scheduling and Context Switching **Memory Management** ▪ Memory Allocation (Contiguous vs. Non-contiguous) ▪ Virtual Memory and Paging ▪ Performance Optimization	▪ Adjust process priority (nice, renice). ▪ Configure swap memory (swapon, swapoff). **Scenario**: Detect and fix memory leaks. ▪ Identify memory issues (free -m, top). ▪ Adjust swap settings (swapon -s). **Scenario**: *A web application is experiencing slow performance. Analyze resource usage and optimize system memory allocation.*
8	**1,3,4,5**	**1,2,3,4,5,6,7**	**Storage and Disk Management** ▪ Logical Volume Manager (LVM) Basics and RAID Configurations ▪ Disk Scheduling Algorithms **I/O Device Management** ▪ Device Drivers and Interrupt Handling ▪ Buffering, Spooling, and Caching ▪ Resource Monitoring and Optimization ▪ Resource Contention and Deadlocks ▪ Performance Tuning and Load Balancing	▪ Monitor disk usage (`df -h`, `du -sh`) ▪ Create and manage partitions (`fdisk`, `mkfs.ext4`). ▪ Configure LVM and RAID (`pvcreate`, `vgcreate`). ▪ Optimize disk I/O performance (`iotop`). ▪ A disk in a RAID 1 array fails. Replace it and rebuild the array. **Scenario**: Expanding storage for a database server. ▪ Expand a partition using LVM. ▪ Configure RAID 1 for redundancy. **Scenario:** *A database server is running out of storage. Set up a*

				new partition and implement RAID 1 for redundancy.
9	**1,3**	**1,2,3, 4,5,7**	**Package Management** Introduction to Package Management ▪ What is a Package Manager? Importance of Package Management ▪ Types of Packages ▪ Binary vs. Source Packages Dependencies and Package Repositories ▪ Working with Package Managers ▪ Handling Dependencies	▪ Identify installed packages (dpkg -l, rpm -qa). ▪ Install, update, and remove packages. ▪ Configure repositories (apt-add-repository). ▪ Fix broken dependencies. ▪ Set up a local package repository for offline package installations. **Scenario**: Upgrade all packages and resolve conflicts. ▪ Identify the package manager and install/remove software. **Scenario:** *Your team requires Python 3.10 for development. Install it and manage dependencies while ensuring system stability.*
10	**3,5**	**1,2,3, 4,5,6, 7**	**Networking and Security** ▪ Basics of Linux Networking (IP Addressing, DNS, Gateway) ▪ Configuring Network Interfaces (Static and Dynamic IP) ▪ Network Troubleshooting (ping, traceroute, netstat, ip route) ▪ Basics of Firewalls (`iptables`, `ufw`, `firewalld`) ▪ Secure Remote Access (SSH, SCP, SFTP) ▪ VPN Configuration (WireGuard/OpenVPN)	▪ Configure network interfaces (static/dynamic IP). ▪ Test network connectivity (ping, traceroute). ▪ Set up a firewall (iptables, ufw). ▪ Secure SSH access (disable root login, change port). ▪ Set up a VPN server and connect clients securely. ▪ Configure firewall rules to allow only HTTP, HTTPS, and

				SSH, and block all other traffic. **Scenario**: Hardening a web server. ▪ Configure static IP and test connectivity. ▪ Block all ports except SSH, HTTP, and HTTPS. **Scenario***: A company needs a secure remote connection for its employees. Configure SSH access and firewall rules to allow only specific IPs.*
11	**3,4**	**1,2,3, 4,5,6, 7**	**Linux Server Administration** Introduction to Server Administration ▪ What is a Server? Difference between Client and Server OS ▪ Common Linux Server Roles ▪ Server vs. Desktop Administration Setting Up a Linux Web Server ▪ Overview of Web Servers (Apache vs. Nginx) ▪ Hosting a Simple Web Page ▪ Managing Virtual Hosts and Configuring Firewall Rules (Apache, Nginx) ▪ Web Server Security (SSL/TLS Configuration) Managing Databases ▪ Installing and Configuring MySQL/PostgreSQL ▪ Database Backup and Restore (`mysqldump`, `pg_dump`)	▪ Install and configure Apache/Nginx. ▪ Set up a MySQL/PostgreSQL database. ▪ Configure SSH-based server management. ▪ Implement SSL/TLS for web security. ▪ Set up a secondary web server and configure load balancing with Nginx. **Scenario**: Host a secure website. ▪ Deploy a website using Apache/Nginx. ▪ Secure MySQL database. **Scenario**: *Deploy a website on an Nginx web server, configure a MySQL database, and secure it using an SSL certificate.*
12	**2,3**		**Remote Server Access and Backup Strategies** ▪ SSH Security Best Practices	▪ Set up SSH key-based authentication.

		1,2,3, 4,5,6, 7	**System Backup and Disaster Recovery** ▪ Backup Strategies for Linux Servers - Using `tar`, `rsync`, and `scp` for Data Backup	▪ Transfer files securely using SCP/SFTP. ▪ Configure user quotas (edquota). ▪ Implement system backup (tar, rsync). **Scenario**: Secure remote login and automate backups. ▪ Configure SSH with key-based authentication. ▪ Automate backups using cron. **Scenario**: *A remote developer needs access to project files. Set up secure file transfer via SCP and restrict unauthorized access.*
13	**3,5**	**1,2,3, 4,5,6, 7**	**Server Monitoring and Performance Optimization** ▪ Monitoring System Resources ▪ Log Management ▪ Optimizing Server Performance ▪ Identifying and Resolving Bottlenecks **Security Hardening for Linux Servers** ▪ Understanding Linux Firewalls (`firewalld`, `iptables`, `ufw`) ▪ Setting Up Fail2Ban for Brute-Force Protection ▪ Implementing SELinux and AppArmor for Extra Security ▪ Detecting Unauthorized Access (`auditd`, `who`, `last`)	▪ Monitor logs (journalctl, syslog). ▪ Detect unauthorized access (auditd, last). ▪ Implement Fail2Ban to prevent brute-force attacks. ▪ Optimize server performance (swap tuning, process priorities). ▪ Simulate a brute-force attack on SSH and analyze logs for unauthorized access patterns. **Scenario**: Prevent brute-force attacks. ▪ Configure Fail2Ban for SSH security. ▪ Set up and test firewall rules. **Scenario:** *A server is experiencing repeated login attempts.*

				Configure Fail2Ban and system auditing to block attackers and monitor logs for security threats.

4.References

Sl No	Description
1	Operating System Concepts by Silberschatz, Galvin, and Korth
2	Supplementary Text: Modern Operating Systems by Andrew S. Tanenbaum
3	The Linux Command Line: A Complete Introduction by William E. Shotts Jr.
4	Advanced Programming in the UNIX Environment by W. Richard Stevens, Stephen A. Rago
5	Supplementary Text: Bash Cookbook by Carl Albing, JP Vossen
6	https://methodist.edu.in/web/uploads/labmanual/OS%20Lab%20Manual.pdf
7	https://it.mitindia.edu/PDF/manuals/IT7411.pdf
8	https://people.iitism.ac.in/~download/lab%20manuals/cse/CSC211.pdf

5. CIE Assessment Methodologies

Sl.No	**CIE Assessment**	**Test Week**	**Duration (minutes)**	**Max marks**	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE - Theory Assessment Methodologies

Sl. No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester -4	
Course Name	Linux Administration			Test	
Course Code	25CS41I	Duration	90 min	Marks	50

Name of the Course Coordinator:

Note: Answer any one full question from each section. Each full question carries equal marks.

Q.No	Questions	Cognitive Level	CO	Marks
	Section - 1			
1	a. During system boot, if systemd fails to mount a file system, what could be the possible causes? How would you troubleshoot this? (5) b. Explain the core functions of an OS with real-world examples for each. c. Explain the significance of kernel modules and how they impact system performance and security.	L2,L3	1 5	25
2	a. You are troubleshooting a system that is missing an essential kernel module. Outline the steps to check, load, and verify the module. (5) b. Explain the importance of virtualization in modern IT infrastructure and discuss its advantages and challenges. c. How do system logs assist in system troubleshooting? Explain all logs with examples.	L2, L3	1 5	
	Section - 2			
3	a. What is the significance of the Linux directory structure? Explain the purpose of /etc, /var, /home, and /bin. (5) b. Describe and demonstrate the steps to change ownership and permissions of files in Linux. Explain permission notations (symbolic and numeric). Include examples showing permission change from read-only to read-write-execute for a specific user and group	L2,L3	3	25

	c. Critical system logs stored in /var/log/ are being accidentally deleted during manual cleanup. How would you prevent accidental deletion using Linux file permissions? Explain the steps clearly. (10)			
4	a. Compare the characteristics of different Linux file systems (ext4, XFS, and Btrfs). (5) b. A user reports that they cannot modify a file despite having write permissions. How would you diagnose and resolve the issue? c. Your organization needs a shared folder for project files where only team members can read and write files, but non-members should not have access. How would you configure: ▪ Group ownership and permissions (chown, chmod) ▪ Access Control Lists (setfacl) for finer access control	L2,L3	3	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**4**
Course Name	**Linux Administration**			**Test**	
Course Code	**25CS41I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
You've been hired by a company to prepare a Linux server environment for development teams. Your tasks include OS setup, automation, security, performance monitoring, and user access control. a. Install Linux distribution on a virtual machine and configure system hostname, time zone. b. Verify boot process by modifying boot parameters and identify kernel and OS version c. Create a group devteam and users alice, bob, and charlie. d. Set strong passwords and assign them to devteam e. Create /project directory, assign devteam ownership and permissions 770. f. Configure ACL so that only alice has full access (rwx) to /project/code. g. Write a shell script that: ▪ Monitors disk usage of /home ▪ If usage > 80%, logs warning to /var/log/disk_alert.log ▪ Schedule this script via cron to run every hour. h. Analyze resource usage and recommend one performance improvement step based on the output.	**1,2,3,5**	

Scheme of evaluation ▪ **Installation and Configuration - 10** ▪ User and File System Security Setup - 10 ▪ Automation and Shell Scripting – 10 ▪ Resource Management – 10 ▪ Troubleshooting and Viva – 10		
Total Marks		**50**

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

9. Equipment/softwarelistwithSpecificationforabatchof30students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**4**
Course Name	Object Oriented Programming with Java	**Type of Course**	Integrated
Course Code	25CS42I	**Contact Hours**	8 Hours / Week
Teaching Scheme	L: T:P	**Credits**	6
CIE Marks	**50**	**SEE Marks**	**50 (Theory)**

1. Rationale

This course is designed to equip students with a comprehensive understanding of object-oriented programming (OOP), a foundational paradigm in modern software development. The focus is on enabling students to write maintainable, modular, and scalable code while gaining practical experience in implementing these concepts using Java. As a versatile programming language, Java is widely used for enterprise applications, mobile apps, and web services, making it an ideal choice for mastering OOP concepts and SOLID principles.

2. Course Outcomes: At the end of the course, the student will be able to:

CO-01	Explain the core principles of object-oriented programming and their significance in software development.
CO-02	Create Java programs that leverage object-oriented concepts to solve real-world problems effectively.
CO-03	Apply refactoring techniques to improve code readability, maintainability, and efficiency while ensuring adherence to SOLID principles.
CO-04	Demonstrate the use of exception handling mechanisms in Java to identify and manage runtime errors, ensuring application reliability.
CO-05	Develop Java applications to perform CRUD operations using JDBC.

3. Course Content

Week	CO	PO	Theory	Practice
1	1	1	**Object-Oriented Paradigm** ▪ Overview ▪ Object oriented programming ▪ OOP core concepts ▪ OO design principles **OO Programming language : Java** Introduction to basic Java syntax: Variables, data types, operators, control statements	• Setting up the Java development environment Install JDK and set up an IDE such as VScode • Write your first java program "Hello, World!"

2	1,2,3	1,2,3	**Classes and Objects** Understanding classes and objects. Defining classes and creating objects in Java. instance variables, methods, constructors. Understanding *this* keyword. **S - Single Responsibility Principle** Definition, intent and benefits; Example of violation Example of correct SRP	▪ Create a class representing a real-world entity. ▪ Instantiate objects and manipulate object attributes. ▪ Demonstrate constructor usage and method calling. ▪ Create simple Java programs and implement SRP by dividing responsibilities across multiple classes. ▪ Given sample code, check compliance with SRP. ▪ Provided a simple class with a violation of SRP, refactor the code to adhere to SRP.
3	1,2	1,2,3	**Encapsulation and Access Modifiers** ▪ Introduction to encapsulation. ▪ Access modifiers: public, private, protected, default. ▪ Getter and setter methods. ▪ Why encapsulation is important in OOP.	▪ Implement encapsulation by creating classes with private variables and public getter/setter methods.
4	1,2,3	1,2,3	**Inheritance** ▪ Understanding inheritance. ▪ *extends* keyword and method overriding. ▪ *super* keyword and constructor chaining. ▪ Types of inheritance **OCP - Open/Closed Principle** ▪ Definition, intent and benefits; ▪ Example of violation ▪ Example of correct OCP	▪ Implement inheritance. ▪ Demonstrate method overriding and using the super keyword. ▪ Provided a simple class with a violation of OCP, refactor the code to adhere to OCP using inheritance. ▪ Given sample code, check compliance with OCP.
5	1,2,3	1,2,3	**Polymorphism** ▪ Understanding polymorphism ▪ Method overloading and method overriding. ▪ Compile-time vs runtime polymorphism. ▪ Interfaces and abstract classes. **Reinforce OCP with polymorphism** Using inheritance and interfaces to achieve OCP.	▪ Demonstrate method overloading and overriding with examples. ▪ Implement interfaces and abstract classes ▪ Use polymorphism to call methods on objects of different classes. ▪ Code programs demonstrating OCP with polymorphism.
6	1,2,3	1,2,3	**Abstraction** ▪ Introduction to abstraction in Java. ▪ Abstract classes vs interfaces.	▪ Create an abstract class with abstract methods and a concrete subclass.

			▪ When to use abstract classes and interfaces. ▪ Real-world examples of abstraction. **LSP - Liskov Substitution Principle** ▪ Definition, intent and benefits; ▪ Example of violation ▪ Example of correct LSP	▪ Provided a simple class with a violation of LSP, refactor the code to adhere to LSP.
7	2,4	1,2,3	**Exception Handling** ▪ Introduction to exception handling. ▪ try, catch, finally, throw, and throws keywords. ▪ Types of exceptions: Checked vs unchecked exceptions. ▪ Best practices for handling exceptions.	▪ Write Java programs to handle exceptions. ▪ Create custom exception classes and use them in a program.
8	2	1,2,3	**Collections Framework - I** ▪ Introduction to Java Collections Framework. ▪ List, Set, Map, and Queue interfaces.	▪ Create programs to manipulate collections
9	2,3	1,2,3	**Collections Framework - II** ▪ Working with ArrayList, and LinkedList. ▪ Iterators and enhanced for loop **ISP: Interface Segregation principle** ▪ Definition, intent and benefits; ▪ Example of violation ▪ Example of correct ISP	▪ Create programs to manipulate collections ▪ Design an interface that builds different types of toys and check Compliance with ISP.
10	2,4	1,2,3	**Multithreading** ▪ Introduction to multithreading. ▪ Creating threads using Thread class and Runnable interface. ▪ Thread synchronization and thread lifecycle.	▪ Create simple multi-threaded programs. ▪ Demonstrate thread synchronization using synchronized keyword.
11	2,4	1,2,3	**File Handling and Annotations** ▪ Introduction to file handling in Java. ▪ Reading from and writing to files ▪ Serialization and deserialization **Java Annotations** ▪ What are annotations in Java? ▪ Commonly used annotations (@Override, @Deprecated, @SuppressWarnings) ▪ Custom annotations and their use cases	Write a Java program that a. reads a text file and prints its contents to the console b. creates a new text file and writes some content to it. c. append data to an existing file handle exceptions properly. ▪ Write a Java program that uses annotations to store file paths or configurations.
12	3,5	1,2,3	**Database Connectivity in Java** ▪ Overview of JDBC (Java Database Connectivity): ▪ What is JDBC and why is it used?	▪ Performing CRUD Operations using JDBC ▪ Applying DIP to Database Connectivity

			▪ Basic components of JDBC: Driver, Connection, Statement, ResultSet. ▪ Steps for Database Connectivity in Java ▪ JDBC API and Database Operations **Dependency Inversion Principle (DIP)** ▪ Definition, intent and benefits; ▪ Example of violation ▪ Example of correct DIP	
13	1,2,3,4,5	1,2,3	**Mini Project**	▪ Create a simple java application such as To Do list

4. References

Sl No	Description
1	Java: The Complete Reference" by Herbert Schildt
2	"Head First Java" by Kathy Sierra and Bert Bates
3	"Effective Java" by Joshua Bloch
4	"Object-Oriented Software Engineering" by Timothy Lethbridge and Robert Laganière
5	"Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin (Uncle Bob)
6	https://docs.oracle.com/javase/
7	https://www.geeksforgeeks.org/java/
8	https://www.tutorialspoint.com/java/index.htm
9	https://hyperskill.org/tracks/1

5. Suggestive Online Courses

Sl no	Topic Name	Reference Courses	Self Assessment Link	Source
1	Introduction to Object-Oriented Programming	Programming using Java	https://infyspringboard.onwingspan.com/web/en/viewer/iap/lex_auth_0129116467125616642_shared?collectionId=lex_auth_01288046454761881634 7_sharedandcollectionType=Course	Infosys Wingspan
2	Classes and Objects	TOC - Design Patterns and SOLID Principles with Java | Infosys Springboard		Infosys Wingspan
3	Encapsulation and Access Modifiers	TOC - SOLID Principles | Infosys Springboard		Infosys Wingspan
4	Collections Framework - I	The Collections Framework		Infosys Wingspan
5	Collections Framework - II			
6	Multithreading	Java Language Features		Infosys Wingspan
7	File Handling and Annotations			
9	Database Connectivity in Java Dependency			

	Inversion Principle (DIP)			

6. Suggestive programs for SOLID principles

Sl no	Scenario
1	Invoice Management System You have a class InvoiceProcessor that: ▪ Calculates invoice totals ▪ Saves invoice data ▪ Sends invoice emails Refactor to check adherence with SRP
2	Online Payment Gateway A class PaymentProcessor handles CreditCard payments only. Now you need to add support for PayPal and UPI.
3	Transportation Booking App A class Vehicle has a method startEngine(). A subclass Bicycle inherits from Vehicle, but calling startEngine() causes issues.
4	Smart Home Devices An interface SmartDevice has methods like: playMusic(), turnOnLights(), setAlarm(), openGarage() You need to implement this for: ▪ SmartSpeaker ▪ SmartLight ▪ SmartAlarm
5	User Authentication System LoginService directly depends on MySQLUserDatabase. Now you want to switch to FirebaseAuth.

7. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (20)	1-13		50	

	Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.
Rubrics for the Mini Project (if included) should be defined by the course coordinator.

8. SEE – Theory Assessment Methodologies

Sl. No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

9. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester – IV**	
Course Name	**Object Oriented Programming with Java**			**Marks**	**50**
Course Code	**25CS42I**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a) Explain the concept of OOP and its key objectives.	L2	1	5
	b) Describe the different control statements used in Java.	L2	1	5
	c) Explain how Single Responsibility Principle (SRP) is violated in the below code and refactor the code to adhere to SRP.	L3	2	15

	``
	class OrderProcessor {
	    public void processOrder(Order order) {
	        // Process the order
	    }
	
	    public void generateInvoice(Order order) {
	        // Generate invoice for the order
	    }
	
	    public void sendEmailConfirmation(Order order) {
	        // Send email confirmation for the order
	    }
	}
	``			

2	a) Briefly explain importance of SOLID principles.
	b)
	``
	class Employee {
	    private String name;
	    public int age;
	    protected double salary;
	    String department;
	``
	Explain the use of different access modifiers in the Employee class.
	c) Explain how Open/Close Principle (SRP) is violated in the below code and refactor the code to adhere to OCP.
	``
	class AreaCalculator {
	    double calculateArea(String shapeType, double dimension) {
	        if (shapeType.equals("circle")) {
	            return Math.PI * Math.pow(dimension, 2);
	        } else if (shapeType.equals("square")) {
	            return Math.pow(dimension, 2);
	        } else {
	            throw new IllegalArgumentException("Shape not supported");
	        }
	    }
	}
	``
		L2
		
		L2
		
		L3
			1
			
			1
			
			2
				5
				
				5
				
				15

	Section - 2			

3	a) Explain with an example how encapsulation is implemented in Java using getter and setter methods.
	b) Write a Java program to demonstrate the use of constructors and the this keyword.
	c) Refactor the following code to adhere to encapsulation principles:
	``
	class Student {
	    public String name;
	    public int age;
	
	    public void displayDetails() {
	        System.out.println("Name: " + name + ", Age: " + age);
	    }
	}
	
	public class Main {
	    public static void main(String[] args) {
	        Student student = new Student();
	        student.name = "John"; // Direct access to the fields
	        student.age = 20; // Direct access to the fields
	        student.displayDetails();
	    }
	}
	``
		L2
		
		L2
		
		L3
			1
			
			1
			
			2
				5
				
				5
				
				15

4	a) Write a Java program to demonstrate the types of inheritance (single and multilevel). c) Refactor the following real-world example of a banking application to demonstrate the principle of abstraction. `class SavingsAccount { public void deposit(double amount) { System.out.println("Depositing $" + amount + " into Savings Account"); } } class CurrentAccount { public void deposit(double amount) { System.out.println("Depositing $" + amount + " into Current Account"); } } public class Main { public static void main(String[] args) { SavingsAccount savings = new SavingsAccount(); savings.deposit(500); CurrentAccount current = new CurrentAccount(); current.deposit(1000); } }`	L2 L3	1 2	10 15
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

10. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**IV**
Course Name	**Object Oriented Programming with Java**			**Marks**	**50**
Course Code	**25CS42I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
1. Suppose you are tasked with developing a simple Library Management System in Java that demonstrates key Object-Oriented Programming (OOP) principles and SOLID principles. The system should allow the following operations: a) Adding books to the library. b) Borrowing and returning books. c) Displaying available books. d) Handling exceptions such as borrowing unavailable books or returning books not in the library.				**1,2,3,4**	**50**
Scheme of assessment(Can be altered as per week wise concepts by the Faculty) **1. Writing the program by identifying and proper use of concepts - 30** a. *Defining classes – 5* b. *Encapsulation – 5* c. *Inheritance and Polymorphism- 10* d. *SRP/OCP adherence– 5* e. *Exception Handling – 5*					

2. Final Integration and execution - 20	
Total Marks	**50**

Signature of the Course Coordinator **Signature of the HOD**

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	4
Course Name	**Data Analysis and Algorithm Design**	**Type of Course**	Integrated
Course Code	**25CS43I**	**Contact Hours**	7 Hours per week
Teaching Scheme	3:0:4	**Credits**	5
CIE Marks	50	**SEE Marks**	50 (Practice)

1. Rationale:

This course aims to equip learners with the essential tools and skills necessary to handle data efficiently, apply advanced algorithm design strategies, and evaluate their performance in terms of time and space complexity. By integrating NumPy, Pandas, Matplotlib, and algorithmic concepts, the course prepares students to solve complex real-world problems and perform data-driven tasks efficiently.

2. **Course Outcomes**: At the end of the Course, the student will be able to:

CO-01	Perform numerical computations and data manipulation using Python libraries such as NumPy and Pandas.
CO-02	Analyze datasets and communicate insights through effective data visualization techniques.
CO-03	Apply appropriate algorithm design strategies (e.g., brute force, divide and conquer, dynamic programming) to solve computational problems.
CO-04	Evaluate algorithm efficiency using space and time complexity analysis.
CO- 05	Debug and optimize algorithm implementations to improve performance and correctness.

3. Course Content

Week	CO	PO	Theory	Practice
1	**1**	**1, 2, 3, 4, 5, 7**	**NumPy Arrays: Overview,** **Array Creation Methods**: array(), arange(), linspace(), zeros(), ones(), empty() **Random Number Generation:** Creating random arrays with: np.random.rand(), np.random.randn(), np.random.randint() **Array Indexing and Slicing** **Setting the random seed with** np.random.seed()	Implement a python code to ▪ Create and manipulate 1D and 2D arrays. ▪ Write code to display array attributes such as shape, size, and dimension. ▪ Create arrays of evenly spaced values using np.arange() and np.linspace(). ▪ Generate random arrays of integers and floats. ▪ Retrieve specific rows, columns, or subsets of a 2D array.

			Aggregation functions: np.sum(), np.mean(), np.median(), np.std(), np.var()	▪ Apply Boolean indexing to filter elements based on conditions. ▪ Modify a given array using slicing and indexing. ▪ Compute statistical measures (mean, median, standard deviation) for a dataset
2	**1**	**1, 2, 3, 4, 5, 7**	**NumPy Arrays contd.** **Array Reshaping:** **Changing the shape of arrays:** np.reshape(), np.resize() **Flattening arrays:** np.flatten(), np.ravel() **Transposing arrays:** np.transpose(), np.swapaxes() **Stacking arrays**: np.hstack(), np.vstack() **Splitting arrays**: np.split(), np.hsplit(), np.vsplit() **Broadcasting in NumPy** **Element-wise Operations Using ufuncs**: np.add(), np.subtract(), np.multiply(), np.divide()	Implement a python code to ▪ Reshape a 1D array into 2D and vice versa. ▪ Flatten a 2D array and explore the difference between np.flatten() and np.ravel(). ▪ Combine arrays horizontally and vertically, and split them into smaller arrays. ▪ Perform element-wise operations on two arrays ▪ Perform arithmetic operations between arrays of different shapes using broadcasting.
3	**2**	**1, 2, 3, 4, 5, 7**	**Getting Started with pandas** ▪ Differences between pandas and Excel/numpy. **Series and DataFrame Basics** ▪ Creating a Series and exploring its attributes (e.g., .values, .index). ▪ Creating DataFrames (from dictionaries, lists, numpy arrays). ▪ DataFrame attributes (.shape, .columns, .dtypes, .head(), .tail()). **Data Selection and Manipulation** **Indexing and Selection** ▪ Selecting rows and columns using .loc[] and .iloc[]. ▪ Boolean indexing to filter rows based on conditions. ▪ **Modifying Data** ▪ Adding new columns, Updating column values. ▪ Dropping columns and rows (.drop()). ▪ **Handling Missing Data** ▪ Detecting missing values (.isnull(), .notnull()). ▪ Filling missing values (.fillna()).	Implement a python code to ▪ Create and explore a Series for a small dataset (e.g., sales or student scores). ▪ Create DataFrames from dictionaries and NumPy arrays. ▪ Use .head() and .tail() to inspect the structure of the DataFrame. ▪ Retrieve specific rows and columns from a DataFrame using .loc[] and .iloc[]. ▪ Apply filters to a dataset using Boolean indexing (e.g., selecting rows based on conditions). ▪ Modify the index of a DataFrame. ▪ Add and delete columns in a DataFrame. ▪ Handle missing values in a dataset by filling or dropping them. ▪ Sort a DataFrame based on specific columns or index.

			▪ Dropping missing data (.dropna()).	
4	**2**	**1, 2, 3, 4, 5, 7**	**Pandas contd.** **Data Analysis and Grouping:** **Sorting and Ranking** ▪ Sorting values by rows or columns (.sort_values(), .sort_index()). ▪ Ranking data using .rank(). ▪ **Aggregation and Descriptive Statistics** ▪ Summary statistics (.mean(), .median(), .std(), .describe()). ▪ Aggregation methods (.sum(), .count(), .min(), .max()). ▪ **Grouping Data** ▪ Grouping data with .groupby(). ▪ Aggregating grouped data (.size(), .agg(), .apply()).	Implement a python code to ▪ Compute basic statistics (sum, mean, etc.) for a dataset. ▪ Group a dataset by a specific column and compute aggregated statistics for each group. ▪ Use multiple aggregation functions on grouped data.
5	**1** 2	**1, 2, 3, 4, 5, 7**	**Functional Programming** Built-in higher-order functions in Python: map(), filter(), reduce(). **Lambda functions:** ▪ Syntax and use cases for anonymous functions. ▪ Difference between lambda functions and regular functions.	▪ Given a list of numbers [1,2,3,4,5], use a lambda function with the map() function to square each element of the list. Return and display the new list of squared values. ▪ Given a list of integers [12,15,8,24,30], write a lambda function with filter() to select only the numbers that are divisible by 5. Display the filtered list. ▪ Using the list [1,2,3,4,5], write a lambda function that uses reduce() to compute the product of all elements in the list. Display the final result.
6	**2**	**1, 2, 3, 4, 5, 7**	**Data visualization with python** ▪ Importance of Data Visualization ▪ Types of visualizations: Line plots, bar charts, histograms, scatter plots, etc. ▪ Overview of Python visualization libraries: Matplotlib, Seaborn, and Plotly. **Matplotlib**: Structure of Matplotlib-Figure, Axes, and Plotting. Plot customization: ▪ Titles, labels, and legends. ▪ Adjusting axis limits and scales (logarithmic vs linear).	Install and set up Matplotlib and Seaborn. 1.Create basic plots using Matplotlib: ▪ Line plot ▪ Bar chart ▪ Scatter plot 2.Create a customized line plot, bar chart, and scatter plot. ▪ Modify titles, labels, and legends. ▪ Change plot colors and styles.

			▪ Styling plots (colors, markers, line styles).	
7	**2**	**1, 2, 3, 4, 5, 7**	**Seaborn:** Introduction to Seaborn and its advantages over Matplotlib. Seaborn's default aesthetic style. Types of plots in Seaborn: Boxplot, violin plot, pairplot, heatmap, etc. **Customizing Seaborn plots:** ▪ Changing plot colors, palettes, and styles. ▪ Adjusting plot sizes and axis labels. ▪ Plotting categorical data (e.g., barplot, countplot).	1.Create basic plots using Seaborn: ▪ Boxplot ▪ Histogram with KDE ▪ Pairplot Work with Seaborn's built-in datasets (e.g., tips, iris) 2.Create and customize Seaborn plots: ▪ Modify plot colors using Seaborn color palettes. ▪ Customize axis labels and titles. ▪ Create bar plots and count plots for categorical data.
8	**3** 4 5	**1, 2, 3, 4, 7**	**Introduction to algorithm design paradigms.** Importance of algorithm design and analysis. **Algorithm efficiency**: Time complexity and space complexity. **Asymptotic Analysis** ▪ Types of Asymptotic Notation ▪ Steps to Perform Asymptotic Analysis ▪ Best, worst, and average-case time complexities. ▪ Common Time Complexities in Asymptotic Analysis **Problem-solving strategies**: Brute force, Decrease and conquer, Divide and conquer, greedy, Dynamic programming.	1.Write a program to calculate the sum of the first n numbers using: · A loop (O(n)) · A formula (O(1)) Asymptotically analyse the space complexity and time complexity of both approaches by varying n value and plot the graph using asymptotic notations. 2.Write a program to calculate the Fibonacci sequence ▪ Recursively and ▪ Iteratively Asymptotically analyse the space complexity and time complexity of both approach by varying n value and plot the graph using asymptotic notations
9	**3** 4 5	**1, 2, 3, 4, 7**	**Brute force approach** ▪ Bubble sort, ▪ Selection Sort, ▪ Linear Search.	a. Implement Linear Search compute space and time complexities. Plot the graph using asymptotic notations. b. Implement Bubble, Selection sorting algorithms compute space and time complexities. Plot the graph using asymptotic notations
10	**3** 4	**1, 2,**	**Decrease and conquer**: Insertion Sort. **Greedy** – Concepts only.	a. Implement insertion sort algorithms compute space and

	5	**3, 4, 7**	**Dynamic programming**: Fibonacci sequence.	time complexities. Plot graph using asymptotic notations. b. Implement Fibonacci sequence with dynamic programming. Plot graph using asymptotic notations
11	**3** 4 5	**1, 2, 3, 4, 7**	**Divide and conquer** - Merge Sort, Quick Sort, Binary search. **Backtracking** – Concepts only	a. Implement Merge and quick sorting algorithms. compute space and tim**e** complexities. Plot graph using asymptotic notations and compare both solutions. b. Implement Binary Search using recursion. Compute space and time complexities. Plot graph using asymptotic notations.
12	**3** 4 5	**1, 2, 3, 4, 5, 7**	**Introduction to Graphs** Graph definitions, terminology. Types of graphs. Represent graphs in different ways: (adjacency matrix, adjacency list, and edge list). **Graph Traversal Algorithms** Breadth-First Search (BFS) Depth-First Search (DFS)	**Implementing BFS and DFS** **BFS Implementation**: ○ Use a queue to traverse the graph level by level. **DFS Implementation**: ○ Implement both recursive and iterative approaches for DFS.
13	**3** 4 5	**1, 2, 3, 4, 7**	Introduction to Hashing. Hashing - Perfect hashing functions. Hash table Hash Functions, Operations, Hash collision, Application.	Implement Password Verification System Using Hashing(use Python's hashlib module, which provides hashing functions like SHA-256 to demonstrate secure password handling.)

4. References:

Sl No	Description
1	"Data Analysis with Python" by David Taieb
2	"Python for Data Analysis" by Wes McKinney
3	"Algorithms: Design and Analysis" by Tim Roughgarden
4	https://docs.python.org/3/tutorial/datastructures.html
5	https://www.geeksforgeeks.org
6	https://www.w3schools.com
7	https://www.tutorialspoint.com
8	https://www.khanacademy.org
9	https://www.datacamp.com
10	https://www.programiz.com

5. CIE Assessment Methodologies

Sl	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1 TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 TheoryTest	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Practice Assessment Methodologies

Sl	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester 4**	
Course Name	**Data Analysis and Algorithm Design**			**Test**	
Course Code	**25CS43I**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	**Questions**		**CL**	**CO**	**Marks**
Section - 1					
1	**Problem Statement: Weather Data Analysis**		L2	1	25

	A meteorological department has collected temperature data for a city over a year. The data is stored as a 2D NumPy array, where each row represents a month's data, and each column represents the daily temperatures for that month. Your task is to analyze the temperature data using NumPy to answer the following questions: ○ Load the temperature data (a 12 x 31 NumPy array where rows represent months and columns represent days). ○ Verify the shape of the array to ensure it matches the expected dimensions. ○ Calculate the average temperature for each month. ○ Identify the month with the highest and lowest average temperatures. ○ Determine the day of the year with the highest temperature. ○ Find the day of the year with the lowest temperature.			
2	**Problem Statement: Sales Data Analysis** A retail store collects daily sales data for a week from three branches. The data is stored in a 2D NumPy array, where each row represents a day of the week, and each column represents the sales figures for a specific branch. Your task is to analyze the sales data using NumPy to answer the following questions: ○ Load the sales data into a 2D NumPy array. ○ Verify the shape of the array (it should be 7 x 3). ○ Calculate the total sales for each branch over the week. ○ Find the branch with the highest total sales. ○ Calculate the total sales for each day across all branches. ○ Identify the day with the highest total sales. ○ Compute the average daily sales for the entire store. ○ Identify the day with sales closest to the weekly average. ○ Sort the sales data for each branch in ascending order (row-wise). ○ Flatten the array to analyze all sales figures as a single list.	L2	1	
Section - 2				
3	**Problem Statement: Student Grades Analysis** A school maintains a record of students' grades for three subjects (Math, Science, and English) in a CSV file. Each row represents a student, and the columns represent the student's name, roll number, and their grades for the three subjects. ○ Load the student grades data into a Pandas DataFrame. ○ Display the first 5 and last 5 rows of the dataset. ○ Print the column names and the number of students (rows). ○ Calculate the average grade for each subject. ○ Find the subject in which students scored the highest average grade. ○ Find students who scored more than 90 in Math. ○ Sort the students by their total marks in descending order.	L2	2	25

	○ Assign a rank to each student based on their total marks.			
4	**Problem Statement: Employee Salary Data Analysis** A company maintains an employee salary record in a CSV file. The dataset contains the following columns: Employee ID, Name, Department, Salary, and Joining Date. Each row represents an employee. ○ Load the employee salary data into a Pandas DataFrame. ○ Display the first 5 rows of the dataset. ○ Show the total number of employees and the list of unique departments. ○ Calculate the average salary for the entire company. ○ Find the department with the highest average salary. ○ Identify the employee with the highest salary in the company. ○ Find employees whose salaries are greater than 80,000. ○ List all employees who joined the company before 2015. ○ Calculate the total salary expenditure for each department. ○ Count the number of employees in each department.	L2	2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**4**
Course Name	**Data Analysis and Algorithm Design**			**Test**	
Course Code	**25CS43I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
Problem Statement: Sales Data Analysis and Visualization **P**roblem Statement: A retail company wants to analyze its sales data for the past year, which is stored in a CSV file. The dataset includes daily sales figures for various product categories (e.g., Electronics, Clothing, and Groceries). The company wants to perform the following tasks to understand its sales performance and visualize key metrics: 1. Load the sales data from a CSV file into a Pandas DataFrame. 2. Clean the data by handling any missing values or outliers. 3. Use NumPy to perform calculations such as calculating the daily sales growth and determining the total sales for each month. 4. Group the data by product category and calculate the average sales per category, as well as the total sales for each product. 5. Plot key visualizations using Matplotlib, such as: ○ A line plot to show the sales trend over time. ○ A bar chart to compare the total sales per product category. ○ A pie chart showing the market share of each product category	**1,2,3**	**50**

Scheme of assessment (customize scheme as per requirement of your problem statement) a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10 b) Implementation and Execution - 30 c) Best Practices (Code Readability and Error Handling)- 10	
Total Marks	**50**

Signature of the Course Coordinator **Signature of the HOD**

9. SEE- Model Practice Question Paper

Program	Computer Science and Engineering		**Semester**	**4**
Course Name	Data Analysis and Algorithm Design	Course Code : 25CS43I	**Duration**	**180 min**
Questions			**CO**	**Marks**
Social Media Engagement Analysis **Problem:** You have data from a social media platform showing User_ID, Post_ID, Likes, Comments, Shares, and Date. ▪ Import the data into a Pandas DataFrame. ▪ Calculate the total number of interactions (likes, comments, shares) for each post. ▪ Group the posts by date and calculate the average engagement (total interactions per post). ▪ Implement a Divide and Conquer algorithm to segment users into different engagement levels based on their total interactions. ▪ Create a bar chart to visualize total interactions by post. ▪ Plot a time series graph of user engagement over time.				**50**
Scheme of assessment: a)Load and analyse data (10 Marks) b)Algorithm selection and design((10 marks) c)Implementation(20 marks) c)Visualization(5 marks) d)Presentation(5 marks)				
Total Marks				**50**

1) Signature of the Examiner **2) Signature of the Examiner**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1

03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	4
Course Name	**Web Development**	**Type of Course**	Integrated
Course Code	**25CS44I**	**Contact Hours**	7 per week
Teaching Scheme	3: 0:4	**Credits**	5
CIE Marks	50	**SEE Marks**	50 (Practice)

1. **Rationale**: The course aims to equip students with a comprehensive understanding of user interface (UI) and user experience (UX) design principles, alongside practical front-end development skills. By incorporating a design thinking approach, the course empowers students to create functional, visually appealing, and user-friendly web applications.

2. **Course Outcomes:** At the end of the Course, the student will be able to:

CO1	Design visually appealing user interfaces using industry-standard tools.
CO2	Transform UI designs into functional web interfaces using HTML and CSS.
CO3	Implement interactivity and dynamic behaviour in webpages using JavaScript.
CO4	Develop modular and reusable UI components using React.
CO-5	Develop a fully functional Single Page Application (SPA) by integrating state management, routing, and UI components

3. **Course Content**

Week	CO	PO	Theory	Practice
1	1	1,3,4	Introduction to UI/UX Design • Overview of UI/UX; Differences and their roles in web development; • Principles of UI/UX Design; UI elements (Buttons, Text Fields, Dropdowns, Checkboxes and Radio Buttons, Sliders, Date Pickers, Cards, Navigation Bars, Progress Bars, Notifications/Alerts, etc) • Examples of good and bad UI/UX. • Color theory and Typography	Learn wire framing and prototyping using Figma or similar design tools. Gain hands-on experience with the tool's interface, including its key features and functionalities, to create effective and user-friendly design mockups
2	1	1,2,3, 4	Design thinking - 5 stages Integrating Design Thinking with UI/UX design	• Create a simple wireframe for a login or e-commerce page. • Create wireframes to visualize layouts and structures. Case Study

				• Problem: Users find it challenging to navigate a banking app to transfer money. • Problem: Users frequently abandon their orders during the checkout process in a food delivery app. Use Design Thinking to redesign UI/UX to solve above problems
3.	2	1,2,3, 4	HTML Basics 1. Structure of an HTML document. Basic Tags, Elements, Attributes and comments Text formatting tags : <p>; <h1> -<h6>; <b>, <i>, <u>, , ,<b>,<hr> ▪ Lists: Ordered and Unordered ; ▪ Links: <a>; ▪ Images: <img>; ▪ block and inline elements: and	Prepare and get familiarize with Your Development Environment like VScode or similar IDE Practice using HTML tags to structure and design user interfaces, focusing on creating layouts, Refer table for suggestive exercises
4	2	1,2,3, 4	1. Tables - <table>, <tr>, <td>, <th> 2. Forms and Input Elements Form structure: <form> Input fields: <input>, <textarea>, <button> Dropdowns and checkboxes: <select>, <option>, <input type="checkbox"> Radio buttons: <input type="radio"> Media elements.	Practice using HTML tags to structure and design user interfaces, focusing on creating layouts, Refer table for suggestive exercises
5	2	1,2,3, 4	1. What is CSS? Why is it important? Different ways to apply CSS: Inline, Internal, External. 2. Understanding the CSS syntax: Selectors, properties, and values. 3. Color; Background; Fonts and Text Styling	Create a simple HTML file and apply inline, internal, and external CSS. Refer table for suggestive exercises
6	3	1,2,3, 4	What is JavaScript? Role of JavaScript in web development. ES6 features; Embedding JavaScript in HTML: <script> tag. ▪ Variables (var, let, const) and data types. Operators and expressions. Conditional statements **(**if-else**,** switch**).** ▪ Loops (for, while, do-while). Loop control statements: **break** and **continue**.	Refer table for suggestive exercises

7	3	1,2,3, 4	▪ Functions and scope ▪ DOM Manipulation: What is DOM? Accessing elements, modifying elements, creating and adding elements, removing elements ▪ Event handling : event and types [Mouse events, keyboard event, form events, windows events] Event Listeners Event propagation and delegation	Refer table for suggestive exercises Selecting and modifying HTML elements. Event handling (e.g., click, mouseover). Basic form validation
8	3	1,2,3, 4	**Arrays and Objects**: Arrays: Create, Indexing, array methods, ES6 enhancements for arrays Objects: create, accessing properties, object methods ES6 Enhancements for Objects	Refer table for suggestive exercises
9	3	1,2,3, 4,5,7	Introduction to JSON What is JSON? Structure; Syntax Rules; Common Use Cases Working with JSON in JavaScript Converting JavaScript Objects to JSON Converting JSON to JavaScript Objects Handling JSON in APIs Working with JSON in Web Applications	***Create and Parse JSON***
10	4	1,2,3, 4,5,7	Introduction to React: How traditional websites worked Evolution of websites What is React? How react app works? Basic Syntax of React .js React components and props	Setting up the environment- install required tools – vscode node.js Create your first React app using vite – create – run - stop Create your first component
11	4	1,2,3, 4,5,7	React Component Interaction State and lifecycle methods. React Hooks: What are React Hooks? Hook – purpose, syntax and basic usage [**useState, useEffect, useContext, useReducer]** React Forms and React Styles	Learn state management with React hooks Create and interact with react forms
12	5	1,2,3, 4,5,7	Introduction to React Router Basic Routing Dynamic Routing Nested Routing Redirects and Programmatic Navigation Protected Routes	Install React Router and Setup Dynamic Routing with Parameters Learn how to redirect users based on certain conditions using the <Redirect /> component.

				Learn how to create protected routes that only allow authenticated users to access certain pages. Create a login system with protected routes
13	5	1,2,3, 4,5,7	Introduction to SPA: What is a Single Page Application; Characteristics; Advantages; Basic Structure of a React SPA; Building a Basic SPA with React.	Build simple REACT SPA.

4. References:

Sl No	Description
1	"Don't Make Me Think" by Steve Krug
2	"The Design of Everyday Things" by Don Norman
3	Designing Interfaces by Jenifer Tidwell
4	"The Design Thinking Playbook" by Michael Lewrick, Patrick Link, and Larry Leifer
5	"Design Thinking: Understanding How Designers Think and Work" by Nigel Cross
6	“Learning Design Thinking”: Focuses on applying design thinking in real-world scenarios.
7	“Design Thinking Guide for Successful Professionals”: Practical and industry-oriented.
8	Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins
9	"HTML5: The Missing Manual" by Matthew MacDonald
10	https://www.geeksforgeeks.org/five-phase-model-for-design/
11	Learning React by Alex Banks and Eve Porcello
12	Fullstack React by Accomazzo, Murray, and Lerner
13	Eloquent JavaScript by Marijn Haverbeke
14	JavaScript: The Good Parts by Douglas Crockford
15	Figma Learn – https://help.figma.com/hc/en-us/categories/360002051373
16	Canva Design School – https://www.canva.com/learn/design-school/

5. Suggested Online Courses:

Sl no	**Topic Name**	***Reference Courses***	**Self Assessment Link**	**Source**
1	**Introduction to UI/UX Design**	TOC - UX and UI - Designing with Color Theory | Infosys Springboard	Assessment - Viewer Page | Infosys Springboard	IIHT
2	**Design thinking**	TOC - UX Design for Web Developers | Infosys Springboard	Assessment - Viewer Page | Infosys Springboard	IIHT
3	**HTML Basics**	TOC - HTML5 - The Language | Infosys Springboard	Assessment - HTML5 - Viewer Page | Infosys Springboard	Infosys Wingspan
4	**CSS**	TOC - Cascading Style Sheets | Infosys Springboard	Cascading Style Sheets - Assessment - Viewer Page | Infosys Springboard	Infosys Wingspan
5	**JavaScript**	TOC - JavaScript | Infosys Springboard	JavaScript - Assessment - Viewer Page | Infosys Springboard	Infosys Wingspan
6	**JSON**	TOC - JSON Essentials | Infosys Springboard		skillsoft

7	**React**	TOC - ReactJS | Infosys Springboard	React JS - Assessment - Viewer Page | Infosys Springboard	Infosys Wingspan
9	**React Router Principle (DIP)**	TOC - Hands-On React Router | Infosys Springboard	Assessment - Viewer Page | Infosys Springboard	IIHT
10	**SPA**	TOC - Building a Single Page Application with JavaScript | Infosys Springboard	Assessment - Viewer Page | Infosys Springboard	IIHT

6. Suggested Program

week	suggestive program/activity
1	Wireframe Login page participation registration form for an event, etc
2	Conduct warmup activities to ignite Design Thinking ▪ Empathize: Understanding user behaviors, needs, and pain points. ▪ Define: Clearly articulate the user problem ▪ Ideate: Brainstorm creative solutions to the problem ▪ Prototype: Create Visual Representations - Create wireframes to visualize layouts and structures. ▪ Test: Validate the prototype by gathering user feedback.
3	"Create Your First HTML Document" create a basic HTML document to demonstrate your understanding of fundamental HTML structure and tags. The document should meet the following requirements: a. Include a <head> section with a title for the webpage. b. Add a main heading (<h1>) to introduce the page with a welcoming message. c. Include a paragraph (<p>) describing the purpose of the page. d. Create an unordered list ( ) with at least three list items ( ) explaining basic facts about HTML. e. Add a hyperlink (<a>) to an external website, ensuring it opens in a new tab. Format the text to enhance the visual appeal and readability
4	Design a product detail page for an e-commerce website. The page should display the product name, image, description, price, and availability. There should also be options to select the quantity and add the product to the shopping cart. Design a login and registration page for a website. The login form should include fields for email and password, while the registration form should include additional fields such as username, email, password, and confirm password. Design a user profile page where users can view and edit their personal details such as name, email, phone number, profile picture, and address. The page should have a navigation bar at the top and a sidebar with links to other user settings (e.g., Account Settings, Privacy Settings). Design a product listing page for an e-commerce website. The page should display a list of products with the following details: product name, price, product image, description, and a "Add to Cart" button. The page should be responsive and adapt to different screen sizes (desktop, tablet, and mobile).

	Project - Create a basic webpage that includes: ▪ A title and headings. ▪ Paragraphs and images. ▪ A table and a form. ▪ Links to navigate between sections or pages. Adding a video and audio file to your webpage.
5	Style basic elements like <p>, <h1>, and Create an HTML page and apply different selectors to style elements. Experiment with background colors and text colors using various color formats. Style paragraphs and headings using different fonts and text properties. Enhance the visual appeal of the UI design in the above section by applying CSS
6 and 7	Write and execute your first JavaScript program. Practice using console.log() for debugging. Write code to compare two numbers and display the larger one Calculate the Total Bill (Shopping Cart) Create a program that calculates the total price of items in a shopping cart. The cart is represented as an array of prices. Use a loop to iterate through the array and calculate the total. Create a Personal Expenses Tracker Design and build a simple web-based personal expenses tracker that allows users to: Add an expenses (description, amount, and date). View the list of expenses. Calculate and display the total amount spent. HTML a. Create a form with the following fields: Description (text input) Amount (number input) Date (date input) b. A button to add the expense to the list. c. Add a table or list to display the expenses. CSS a. Style the form to make it visually appealing. b. Add hover effects on the "Add Expense" button. c. Use a color scheme to distinguish the form, expense list, and total amount display. JavaScript a. Validate the form fields (e.g., all fields must be filled). b. Add functionality to: c. Add the entered expense to the list when the button is clicked. d. Calculate and update the total amount after each addition. Create a To-Do List Application Design and build a simple web-based to-do list application that allows users to: ▪ Add tasks to a list. ▪ Mark tasks as completed. ▪ Delete tasks from the list. HTML a. Create a form with a single input field to enter a task description. b. Add a button to add the task to the list. c. Create a section to display the list of tasks, with each task having: ▪ A checkbox to mark it as completed.

- A delete button to remove it from the list.

CSS

a. Style the form and task list to make the application visually appealing.
b. Use different styles for completed tasks (e.g., strikethrough text or a light background color).
c. Add hover effects on buttons and smooth transitions for task updates.

JavaScript

Add functionality to:

a. Validate that the input field is not empty before adding a task.
b. Add the task to the list dynamically.
c. Mark tasks as completed when the checkbox is clicked.
d. Remove tasks from the list when the delete button is clicked.

Create a Temperature Converter

Design and build a simple web-based temperature converter that allows users to:

- Input a temperature in Celsius or Fahrenheit.
- Convert the temperature to the other unit.
- Display the result dynamically on the page.

HTML

Create a form with:

- An input field for the temperature value.
- A dropdown menu to select the unit (Celsius or Fahrenheit).
- A button to perform the conversion.
- Add a section to display the converted temperature.

CSS

- Style the form and result section to make the application visually appealing.
- Add a consistent color scheme and spacing for a clean layout.
- Use hover effects for the button

JavaScript

Add functionality to:

- Validate that the input field contains a valid number.
- Perform the conversion based on the selected unit:
- Celsius to Fahrenheit: $(C \times 9/5) + 32$
- Fahrenheit to Celsius: $(F - 32) \times 5/9$
- Display the result dynamically on the page.

Create a Currency Converter

Design and build a simple web-based currency converter that allows users to:

- Input an amount in one currency.
- Select the currencies for conversion (e.g., USD to EUR).
- Display the converted amount dynamically based on a fixed exchange rate.

HTML

Create a form with:

- An input field for the amount.
- Two dropdown menus to select the source currency and target currency.
- A button to perform the conversion.
- Add a section to display the converted amount.

CSS

- Style the form and result section for a professional look.
- Add a consistent color scheme and spacing.
- Use hover effects for the button and smooth transitions for dynamic updates.

JavaScript

Add functionality to:

- Validate the input to ensure it is a positive number.
- Use predefined exchange rates to calculate the conversion. For example:

	▪ USD to EUR: 1 USD = 0.85 EUR ▪ EUR to USD: 1 EUR = 1.18 USD ▪ USD to INR: 1 USD = 74.50 INR ▪ Add more currencies as needed. ▪ dynamically display the converted amount.
8	Create a simple contact list to store contact information and perform operations like searching and adding new contacts. Design a simple system to manage the items in the shopping cart of an online store using arrays and objects. The system should allow users to add, remove, and update items in the cart, calculate the total cost, and display the cart's contents. Create a system to manage movie rentals. Each movie has a title, genre, rental price, and availability status. The system should allow users to add and remove a movie, update movie info, display all movies and rent a movie. Develop a system to manage tasks for a project. Each task has a title, description, priority, and completion status. The system should allow following operations: ▪ Add Task: Add a new task to the system. ▪ Remove Task: Remove a task by taskID. ▪ Update Task Info: Update the title, description, priority, or status of a task. ▪ Mark Task as Completed: Mark a task as completed. ▪ Display All Tasks: Display the list of all tasks with their titles, priorities, and statuses. ▪ Filter Tasks by Priority: Display tasks with a specific priority level.
9	Create and Parse JSON a. Write a JSON string for a "Book" with attributes like title, author, price, and genres. Parse it into an object and access its properties. b. Create a JSON object for a "Student" with nested "Subjects" and "Scores". Parse it and calculate the average score. c. Convert JSON data (string) into a JavaScript object using JSON.parse(). d. Convert a JavaScript object into a JSON string using JSON.stringify().
10	▪ Create a Simple Greeting Component - Create a React component called Greeting that accepts a name as a prop and displays a message like "Hello, [name]!". ▪ Build a ColorBox component that displays a colored box. It should take a color prop (e.g., red, blue, green) and render the box in that color. ▪ Create a ProductCard component that displays product information like name, price, description, and an "Add to Cart" button. On clicking the button, log a message "Product added to cart".
11	Counter App: • Create a simple counter app with useState. • Add buttons to increment and decrement the counter. Effect on Mount: • Use useEffect to log a message to the console when the component is mounted. • Add a cleanup function in useEffect to demonstrate component unmounting. Form Validation App: • Create a form with multiple fields (e.g., name, email, password). • Use useState to manage the form data and useEffect to validate the fields. • Show validation messages when the fields are invalid. ▪ Create a LoginForm component with two input fields (username and password) and a submit button. Display a message saying "Welcome, [username]" when the user submits the form. ▪ Create a LoginForm component with two input fields (username and password) and a submit button. Display a message saying "Welcome, [username]" when the user submits the form.

	▪ Create a TodoList component where users can add new tasks to a list and remove tasks by clicking a delete button next to each task. ▪ Create a Rating component that displays 5 stars. Clicking on a star sets the rating, and all stars up to that point should be highlighted. ▪ Build a SearchFilter component that allows users to search through a list of items. Display only the items that match the search query. ▪ Create a ShoppingList component that renders an array of items (e.g., "Apples", "Bananas", "Oranges"). Display the items in a list. ▪ Create a ToggleButton component with a button that toggles between "ON" and "OFF" states when clicked. ▪ Build a Counter component with a button that increments the count each time it's clicked. ▪ Create a Stopwatch component with start, stop, and reset buttons. Display the time elapsed since the stopwatch was started. ▪ Build a ThemeSwitcher component with a button that toggles between light and dark themes. The background color and text color should change accordingly. ▪ Build a Slider component that allows users to select a value between a defined range (e.g., 0-100) using an input slider. Display the selected value.
12	Basic Routing ▪ Create a simple React app with three pages: Home, About, and Contact. ▪ Use react-router-dom to create routes for each page. ▪ Display a navigation menu with links to each page. Output: When you click on the links, the corresponding pages should render without reloading the entire page. Nested Routes ▪ Create a "Dashboard" page that contains two sub-pages: "Profile" and "Settings". ▪ Use nested routes to render these sub-pages inside the Dashboard component. ▪ Display links for "Profile" and "Settings" inside the Dashboard page. Output: Clicking on "Profile" or "Settings" should render the appropriate content inside the Dashboard page. Dynamic Routing ▪ Create a route to display a user profile page. ▪ The URL should contain a user ID (e.g., /profile/:id). ▪ Use useParams to extract the id from the URL and display it on the page. Output: Visiting /profile/1 should display a profile for user 1, and /profile/2 should display a profile for user 2. Redirecting ▪ Create a login page where a user can submit a form. ▪ After a successful login, redirect the user to the "Home" page using the Redirect component. Output: After the login form is submitted, the user should be redirected to the Home page. Programmatic Navigation ▪ Create a button that, when clicked, programmatically navigates to a new route (e.g., from "/home" to "/about"). Output: Clicking the button should redirect the user to the new route. Navigation Bar ▪ Create a navigation bar with links to different pages (Home, About, Contact). Output: The active link should be highlighted based on the current route.
13	E-Commerce Product Catalog

	Features: Display a list of products with filters (e.g., price, category). Add items to a shopping cart. Update the cart dynamically without refreshing the page. Key Concepts: Dynamic rendering of products. State management for the shopping cart. Online Quiz Application Features: Display multiple-choice questions one at a time. Show results and scores after completion. Include a timer for each question. Key Concepts: Dynamic rendering of questions. State management for answers and scores. Conditional rendering for results and feedback. To-Do List Application Features: Add, edit, and delete tasks. Mark tasks as completed. Filter tasks by status (e.g., All, Active, Completed). Key Concepts: Component-based design (e.g., using React or Vue). State management for tasks. Routing (e.g., different views for "All" and "Completed" tasks).

7. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam / NPTEL/AWS /any other (30)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.
Rubrics for the Mini Project (if included) should be defined by the course coordinator.

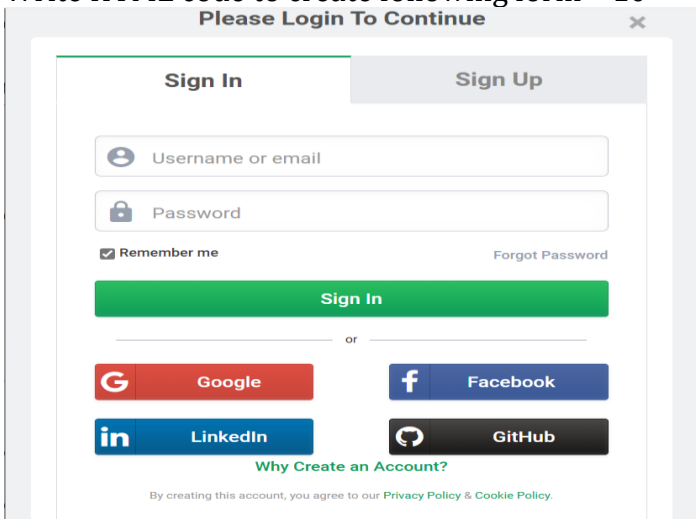
8. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

9. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester - 4	
Course Name	Web Development			Test	
Course Code	25CS44I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section – 1			
1	a. What are the five stages of design thinking? Briefly describe each stage with examples. - 10 b. Create a simple wireframe for a participation registration form for a seminar. -10 c. Describe core UI/UX design principles. - 5	L2,L3	1	25
2	a. How does empathy play a critical role in UI/UX design? Provide an example of how empathy can influence design decisions. - 5 b. Explain the difference between UI (User Interface) and UX (User Experience) with examples. Design a wireframe for a login page, including features like “Remember Me” and “Forgot Password” links. - 10 c. Explain the concept of progressive disclosure in UI/UX design. How can it be represented in a wireframe? - 10	L2,L3	1	
	Section – 2			
3	a. Explain the difference between block-level and inline elements with examples. b. How does the alt attribute in the <img> tag enhance accessibility?	L2	2	25

	c. How would you create a form with the following fields: Name, Email, Password, Gender, and Submit? Write the HTML code. – 10 **d.** Write HTML code to create following table - 10 **Two - Wheeler** **OEM** / **May-23** / **May-22** / **YOY** Hero MotoCorp / 5,08,309 / 4,66,466 / 9% HMSI / 3,11,144 / 3,20,844 / -3% TVS Motor / 2,52,690 / 1,91,482 / 32% Bajaj Auto / 1,94,811 / 96,102 / 103% Royal Enfield / 70,795 / 53,525 / 32% *Suzuki Motorcycle* / 64,000 / 60,518 / 6% **Total** / **14,01,749** / **11,88,937** / **18%**			
4	a. How can you set a default value for a text input field in HTML? b. Create a form to capture patient details during registration. -10 c. What is the difference between checkboxes and radio buttons in forms? Provide examples. d. Write HTML code to create following form – 10 e. 	L2	2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

10. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	4
Course Name	**Web Development**			**Test**	
Course Code	25CS44I	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
Create a Currency Converter Design and build a simple web-based currency converter that allows users to: ▪ Input an amount in one currency.				1,2,3	**50**

▪ Select the currencies for conversion (e.g., USD to EUR). ▪ Display the converted amount dynamically based on a fixed exchange rate. HTML Create a form with: ▪ An input field for the amount. ▪ Two dropdown menus to select the source currency and target currency. ▪ A button to perform the conversion. ▪ Add a section to display the converted amount. CSS ▪ Style the form and result section for a professional look. ▪ Add a consistent color scheme and spacing. ▪ Use hover effects for the button and smooth transitions for dynamic updates. JavaScript Add functionality to: ▪ Validate the input to ensure it is a positive number. ▪ Use predefined exchange rates to calculate the conversion. For example: ▪ USD to EUR: 1 USD = 0.85 EUR ▪ EUR to USD: 1 EUR = 1.18 USD ▪ USD to INR: 1 USD = 74.50 INR ▪ Add more currencies as needed. dynamically display the converted amount.		
Scheme of assessment **Wireframing - 15** ▪ Clarity and Structure - 05 ▪ Consistency and Visual Appeal - 05 ▪ Interactivity Representation - 05 **Implementation – 30** ▪ Use of correct HTML elements – 10 ▪ CSS styling and responsiveness – 10 ▪ JavaScript functionality – 10 **Presentation -05**		
	Total Marks	**50**

Signature of the Course Coordinator **Signature of the HOD**

11. SEE- Model Practice Question Paper

Program	**Computer Science and Engineering**			**Semester**	
Course Name	**Web Development**			**Test**	**II/IV**
Course Code	**25CS44I**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
Design and build a **Single Page Application (SPA)** using **React** to manage a **To-Do List**. The application should allow users to add, edit, delete, and mark tasks as completed. The tasks should be stored temporarily (using local state or a temporary storage solution) and should be dynamically updated on the page without requiring a page reload. The application should have a clean, user-friendly interface and should demonstrate key features of React such as state management, event handling, and component reusability.				1,2,3,4,5	**50**

Requirements ▪ Allow users to add tasks with validation. ▪ Enable task name editing. ▪ Allow users to remove tasks from the list. ▪ Toggle completion status. ▪ Filter tasks by completion status. ▪ Sort tasks by name or completion status. ▪ Add and display due dates for tasks.		
Scheme of assessment **Wireframing and Design - 10** ▪ Clarity and Structure - 05 ▪ Consistency and Visual Appeal - 05 **Implementation and Functionality– 30** ▪ Event Handling - 10 ▪ React Component Design and Creation- 10 ▪ State Management and Routing – 10 **Code Quality and Best Practices - 05** **Presentation and Demo -05**		
	Total Marks	**50**

1)Signature of the Examiner **2) Signature of the Examiner**

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

V and VI SEMESTER

Government of Karnataka

DEPARTMENT OF TECHNICAL EDUCATION

V Semester Scheme of Studies - Computer Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/ week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	CS	SP-1	Specialization Pathway 1	4	0	4	8	6	50	20	50	20	-	-	100
2	CS	SP-2	Specialization Pathway 2	3	0	4	7	5	50	20	-	-	50	20	100
3	CS	SP-3	Specialization Pathway 3	3	0	4	7	5	50	20	-	-	50	20	100
4	CS	25CS54I	Software Engineering and Entrepreneurship	4	0	4	8	6	50	20	50	20	-	-	100
Total				**14**	**0**	**16**	**30**	**22**	**200**		**100**	**-**	**100**	**-**	**400**

SP-1		SP-2		SP-3	
Course Code	**Course Name**	**Course Code**	**Course Name**	**Course Code**	**Course Name**
25CS51IA	Artificial Intelligence Concepts	25CS52IA	Machine Learning	25CS53IA	Deep Learning
25CS51IB	Cloud Computing Architecture	25CS52IB	Cloud Networking	25CS53IB	Cloud Security and Compliance
25CS51IC	REST API Development with Spring Boot	25CS52IC	Advanced Frontend Development with React	25CS53IC	DevOps: Software Development and Operations
25CS51ID	Cyber-Physical System	25CS52ID	Cyber Security	25CS53ID	Ethical Hacking

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

VI Semester Scheme of Studies - Computer Science and Engineering

Sl.No.	Department	Course Code	Course Name	Hours per week	No of Weeks	Credits	CIE Marks		Practice SEE Marks		Total Marks
							Max	Min	Max	Min	
1	CS	25CS61I	Internship/Capstone Project	40	13	13	50	20	50	20	100
Total				**40**	**13**	**13**	**50**	**20**	**50**	**20**	**100**

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Computer Science and Engineering	**Semester**	**V**
Course Name	Artificial Intelligence Concepts	**Type of Course**	Integrated
Course Code	**25CS51IA**	**Contact Hours**	8 hrs per week
Teaching Scheme	L: T:P 4:0:4	**Credits**	6
CIE Marks	**50**	**SEE Marks (Theory)**	**50**

1. Rationale:

This course provides a foundational understanding of Artificial Intelligence (AI), exploring its core concepts, methodologies, and applications across various domains. Without delving into complex algorithms, students will engage with AI tools and platforms to experience real-world AI applications in fields like healthcare, finance, and automation. The course emphasizes both theoretical knowledge and practical exploration through intuitive, no-code AI tools, ensuring accessibility for beginners. By the end of the course, students will have a strong conceptual grasp of AI's capabilities, ethical considerations, and its evolving role in society, preparing them for advanced AI studies.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain core AI principles and describe real-world applications
CO-02	Explain AI project cycle and analyse importance of data in AI applications.
CO-03	Explore and analyze various AI methodologies and technologies
CO-04	Apply AI concepts in real-world scenarios by utilizing practical AI tools
CO-05	Explore AI's future trends and analyse ethical considerations in major domains.

3. Course Content

Week	**CO**	**PO**	Theory (4 hours)	Practical (4 hours)
1	**1**	1, 2, 7	**Introduction to AI** ▪ What is AI? Definitions and Scope ▪ Difference between AI, ML, and DL ▪ AI in Everyday Life: Real-world applications (Google Search, Netflix, Alexa)	▪ Prepare timeline for evolution of AI ▪ Explore AI applications used in daily life ▪ Identify AI-powered features in mobile apps

			▪ History and Evolution of AI	
2	**2**	2, 5, 6	**AI Project Cycle and Problem Solving** Understanding the AI Project Cycle: Problem Identification, Data Acquisition, Model Training, Evaluation, Deployment Challenges in AI Projects: Bias, Data Limitations, Ethical Concerns Case Studies of AI Solutions	▪ Mini-Project Planning: Select an AI use case and define its AI project cycle ▪ Tool Exploration: Use Google AI Experiment to understand AI models
3	**2**	1, 3, 4	**Data and AI** ▪ Role of Data in AI: Importance of Data for Model Training ▪ Types of Data: Structured, Unstructured, Semi-structured ▪ Data Collection and Preprocessing: Sources, Cleaning, Annotation and labelling	▪ Hands-on Activity: Explore Teachable Machine - Train an image-based AI model ▪ Experiment with Datasets: Analyze structured and unstructured data
4	3 4 5	2, 4, 6	**Predictive Systems** ▪ What are Predictive Systems? ▪ Key components: · Data collection · Feature selection · Model training · Evaluation and deployment ▪ Importance and types: · -Supervised: Regression, classification. · -Unsupervised: Clustering, Anomaly Detection ▪ Applications and Challenges ▪ Ethical Considerations	▪ Google Teachable Machine (Link) ▪ Train simple classification models using images, audio, or poses ▪ BigML (Link) ▪ No-code machine learning tool for classification, regression, and clustering. ▪ -Explore various use cases

5	3 4 5	3, 5, 6	**Computer Vision** ▪ Introduction to Computer Vision: AI's role in image processing ▪ Key Applications: Face Recognition, Object Detection, Medical Imaging ▪ Limitations and Ethical Issues in Computer Vision	▪ Use Cases in AI Vision: Experiment with Google Lens and AI Image Classifiers ▪ Demo on OpenCV (GUI - based tool) : Image recognition using AI ▪ -Explore various use cases
6	3 4 5	2, 3, 6	**Natural Language Processing** ▪ Introduction to NLP: AI for understanding human language ▪ Applications: Chatbots, Virtual Assistants, Translation Systems ▪ Speech Recognition and Synthesis ▪ Ethical issues and challenges	▪ Hands-on Activity: Explore AI Assistants (Google Assistant, Alexa,) ▪ Practical on NLP: Try speech-to-text conversion using online AI tools. ▪ Explore various use cases
7	3 4 5	4, 6, 7	**Recommendation Systems** ▪ What are Recommendation Systems? ▪ Importance and their applications ▪ E-commerce (Amazon, Flipkart) ▪ Streaming services (Netflix, Spotify, YouTube) ▪ Social media (Facebook, Instagram, LinkedIn) ▪ Types**:** Collaborative Filtering, Content-Based Filtering, Hybrid model, Knowledge based system**.** ▪ How recommendation system work ▪ Ethical issues and challenges	▪ Explore AI-powered recommendation tools (Google's AI recommendations) ▪ Case Study Discussion: How AI recommends products/movies. ▪ Explore various use cases
8	3 4 5	1, 3, 6	**Reinforcement Learning system** ▪ What is Reinforcement Learning (RL)? ▪ Importance and their applications ▪ Components: Agent, Environment, State, Action, Reward ▪ How RL system works ▪ Applications and challenges ▪ Ethical Issues in RL	▪ Use OpenAI Gym (GUI-based simulation) to explore RL concepts ▪ Simulate a simple RL-based AI in a gaming environment. ▪ Explore various use cases

9	3 4 5	3, 4, 6	**IoT and Edge Computing** ▪ What is Edge AI? ▪ AI Applications in IoT: Smart Devices, Home Automation, Industrial AI ▪ Advantages of Edge AI: Real-time AI processing without cloud dependency ▪ Challenges and ethical issues	▪ Hands-on: Explore Edge Impulse - Build an AI-powered IoT model ▪ Demo: AI-powered IoT examples in real-world applications. ▪ Explore various use cases
10	3 4 5	3, 5, 6	**Robotics and Automation** ▪ AI's Role in Robotics: Industrial Automation, Autonomous Vehicles ▪ AI in Space Exploration and Manufacturing ▪ Challenges and ethical issues	▪ Hands-on: Explore Boston Dynamics and Self-Driving Car Simulations ▪ AI Robotics Demo: How AI powers robotic automation. ▪ -Explore various use cases
11	3 4 5	5, 6, 7	**Future of AI and Generative AI** ▪ AI Creativity: AI-generated art, music, and text ▪ Deepfake Technology and AI Ethics ▪ Limitations and Risks of Generative AI	▪ Explore ChatGPT, Deepseek ▪ Create AI-generated artwork/ music using DALL·E
12	3 4 5	6, 7	**Industry Trends and Careers in AI** ▪ AI Startups and Innovation: Emerging AI-based businesses ▪ Career Paths in AI: Roles, Skill sets, Future Scope	▪ Guest Lecture/Industry Expert Talk: Insights into AI careers and trends
13	3 4	4, 5, 6	**Mini AI Use Case Development** ▪ How AI is transforming industries: Real-world impact ▪ Developing an AI-driven idea for real-world problems	▪ Project Work: Students define and develop an AI-driven solution ▪ Presentation of AI projects using playground tools

4. References:

Sl. No.	Description
1.	https://www.datacamp.com
2.	https://news.microsoft.com/wp-content/uploads/prod/sites/93/2020/04/Student-Guide-Module-1-Fundamentals-of-AI.pdf
3.	Google AI for Beginners – https://ai.google/education/
4.	IBM AI Learning Path – https://www.ibm.com/artificial-intelligence
5.	MIT Introduction to Deep Learning – https://introtodeeplearning.com/

6.	Microsoft Learn AI and ML Path – https://learn.microsoft.com/en-us/training/ai-and-machine-learning
7.	"Artificial Intelligence: A Guide for Thinking Humans" – Melanie Mitchell
8.	"Python Machine Learning" – Sebastian Raschka and Vahid Mirjalili
9.	"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" – Aurélien Géron
10.	"Data Science for Business" – Foster Provost and Tom Fawcett
11.	"AI Ethics" – Mark Coeckelbergh

5. CIE Assessment Methodologies

Sl	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1 TheoryTest	4	90	50	**Average of all CIE=50 Marks**
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 TheoryTest	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(20)re	1-13		50	
Total					**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program:	**Computer Science and Engineering**		**Semester - V**
Course Name:	**Artificial Intelligence Concepts**		**Test: I/III**
Course Code	**25CS51IA**	**Duration: 90 min**	**Marks: 50**
Name of the Course Coordinator:			

Note: - Answer any TWO from I and III (20 + 20 marks).
- Answer any ONE from II and IV (5 +5 marks).

Q.	Questions	CL	CO	Marks
	Section - 1			
I	1. Compare Human Intelligence and Artificial Intelligence. Discuss their differences with real-world examples. 2. Explain the AI Project Cycle and its key phases with an example of an AI project. 3. Explain the importance of data in AI applications.	L2 L3	1 2	10*2=20
II	4. What are the different domains of AI? Explain any two domains with real-life applications. 5. Differentiate between AI, ML and DL.	L2	1	5*1=5
	Section - 2			
III	6. Explain the role of data in AI applications. 7. Explain different types of data with examples. 8. What are predictive systems? Explain its types and importance.	L3 L4	1 2 3	10*2=20
IV	9. What are the key components of predictive systems? 10. List the applications of predictive systems.	L3 L4	3	5*1=5

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**		**Sem: V**
Course Name	**Artificial Intelligence Concepts**		**Test: II/IV**
Course Code	**25CS51IA**	**Duration: 180 min**	**Marks: 50**
Name of the Course Coordinator:			

Questions	**CO**	**Marks**

Q1. You are tasked with developing an AI model to classify different types of fruits using images. Follow the AI Project Cycle to design a structured approach for this task using Teachable Machine (a no-code AI tool). ▪ Problem Identification: a. Define the objective of your AI model. b. Explain why AI is suitable for this task. ▪ Data Acquisition: a. Collect or select images of different fruits (e.g., apples, bananas, oranges). b. Discuss the importance of having a diverse dataset. ▪ Model Training: a. Use Teachable Machine to train an AI model for fruit classification. b. Explain the role of labeled data in supervised learning. ▪ Evaluation: a. Test your model with new images and note its accuracy. b. Identify any biases or errors in predictions. ▪ Deployment and Challenges: a. How can this model be improved for real-world use? b. Discuss challenges like biased data, low-quality images, or incorrect classifications.	2, 3, 4	**50**
Scheme of assessment ▪ Problem Identification ▪ Data Acquisition ▪ Model Training ▪ Evaluation ▪ Deployment and Challenges		**10** **10** **10** **10** **10**
Total Marks		**50**

Signature of the Course Coordinator **Signature of the HOD**

10. Equipment/software list with Specification for a batch of 30 students

Sl.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**Cloud Computing Architecture**	**Type of Course**	Integrated
Course Code	25CS51IB	**Contact Hours**	8 hours/week
Teaching Scheme	L: T:P - 4:0:4	**Credits**	6
CIE Marks	50	**SEE Marks (Theory)**	50

1. Rationale:

Cloud Computing forms the backbone of modern IT infrastructure, driving demand for professionals who can design scalable, cost-efficient, and secure cloud-based systems. This course is designed to equip students with a solid foundation in **Cloud Computing** concepts and progresses into advanced topics such as serverless computing, containerization, and cloud automation. Emphasizing real-world application, the course integrates emerging technologies like AI, DevOps, and Agile practices to ensure learners are prepared for dynamic, fast-paced industry environments.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain the fundamental concepts of **Cloud Computing**, including its architecture, service models, and deployment types.
CO-02	Utilize major cloud services to design and optimize cloud-based solutions.
CO-03	Develop scalable cloud-based systems and set up tools for cloud monitoring.
CO-04	Describe cloud security and compliance practices, along with the architecture of modern cloud solutions.
CO-05	Apply automation solutions, and disaster recovery strategies to develop cloud architectures.

3. Course Content

WEEK	CO	PO	Theory	Practice
1	**1**	**1,2,4,7**	**Introduction to Cloud Computing** What is **Cloud Computing**? Historical background and evolution of **Cloud Computing**. (Distributed and Grid computing) Benefits and challenges of **Cloud Computing**. **Cloud Computing** vs traditional computing. Cloud Deployment Models (Public, Private, Hybrid, Multi-Cloud)	**Introduction** - Exploring AWS Management Console, Azure Portal, and GCP Console - Hands-on: Creating a Free Tier Account on AWS/Azure/GCP

			Key **Cloud Computing** providers (AWS, Azure, Google Cloud, etc.).	
2	**1**	**1,2,3,4,7**	**Cloud Computing Architecture** Cloud Servers, Storage components, Virtualization, and Data Centers. Cloud management platform and Cloud service APIs. Virtual Machines, Hypervisors, Virtualization vs. Physical Machines. Role of Virtualization in resource management and scalability. Examples of Virtual Machines: AWS EC2, Azure VMs, and GCP Compute Engine	Launching a Virtual Machine on AWS/Azure/GCP
3	**1**	**1,2,3,4,7**	**Cloud core services** **Infrastructure as a Service (IaaS)** Virtual Machines, Storage services, Networking, and Cloud Compute. Examples: AWS EC2, Microsoft Azure Virtual Machines. **Platform as a Service (PaaS)** Development frameworks, databases, and runtime environments. Examples: Google App Engine, Microsoft Azure App Services. **Software as a Service (SaaS)** Hosted applications, multi-tenancy, and SaaS management. Examples: Google Workspace, Office 365, Salesforce.	Deploy a simple web application on a cloud platform (AWS, Azure, or GCP).
4	**2**	**1,2,3,4,7**	**Cloud Networking and Security** Virtual Networks, Subnets, and IP Addressing in Cloud. Understanding Security Layers and Best Practices in Cloud. IAM (Identity and Access Management) Concepts. Role-based Access Control (RBAC). Security Groups, Network ACLs, and Firewall Rules	Setting up VPC (Virtual Private Cloud) in AWS Hands-on with NSG (Network Security Groups) in Azure
5	**2**	**1,2,3,4,5,7**	**Cloud Storage and Database services** Cloud Storage Solutions. Object storage vs. Block storage vs. File storage.	Setting up Cloud Storage (S3 in AWS, Blob Storage in Azure)

			Cloud Storage in AWS (S3), Azure (Blob Storage), and GCP (Cloud Storage). Data replication and redundancy in cloud storage. Data Management in Cloud. Data backup and recovery. Cloud-based databases: SQL and NoSQL databases (AWS RDS, Google BigQuery, Azure Cosmos DB). Cloud data migration strategies	Setting Up a Cloud Database using RDS.
6	**3**	**1,2,3,4,5,7**	**High Availability and Scalability in Cloud** Auto-scaling and Load Balancing. Types of Load Balancers. (ALB, NLB, GWLB). Elastic Load Balancing (ELB). Benefits of Auto-Scaling for Cost Management.	Configuring Auto-scaling and Load Balancers on AWS/Azure
7	**3**	**1,2,3,4,7**	**Cloud Monitoring and Management** Cloud Monitoring Tools (AWS CloudWatch, Azure Monitor, GCP Cloud Monitoring) Logging and Auditing (AWS CloudTrail, Azure Log Analytics, GCP Cloud Logging) Cost Management and Optimization in the Cloud	Hands-on: Setting Up Monitoring and Alerts in AWS/Azure/GCP
8	**4**	**1,2,3,4,5,7**	**Cloud Security and Compliance** Identity and Access Management (IAM) in AWS, Azure, and GCP Cloud Security Risks and Threats Encryption Techniques in the Cloud Cloud Security Best Practices	Hands-on: Encryption in AWS/Azure/GCP
9	**4**	**1,2,3,4,7**	**Advanced Cloud Architectures** Multi-Tier Architecture in the Cloud Microservices Architecture Event-Driven Architecture High Availability and Disaster Recovery	Hands-on: Deploying a Multi-Tier Application in AWS/Azure/GCP
10	**4**	**1,2,3,4,5,7**	**Serverless Computing and Container-Based Architectures** Introduction to Serverless Computing (AWS Lambda, Azure Functions). Benefits	Deploying and managing AWS Lambda/ Azure functions/GC functions.

			and Challenges of Serverless Containerization with Docker and Kubernetes AWS Fargate, Azure Container Instances, and GCP Cloud Run.	Creating serverless applications on AWS /Azure/GCP Hands-on: Deploying a Containerized Application in AWS/Azure/GCP.
11	**5**	**1,2,3,4,5,7**	**Cloud Automation and DevOps** Introduction to DevOps in Cloud, Benefits, Lifecyle and DevOps tools. Cloud Automation using Terraform, CloudFormation. Continuous Integration and Continuous Delivery (CI/CD).	Automating infrastructure setup using Terraform or AWS CloudFormation Setting up CI/CD pipelines using AWS with GitHub Actions or Azure DevOps.
12	**5**	**1,2,3,4,5,7**	**Disaster Recovery and Business Continuity in Cloud** Disaster Recovery Planning and Cloud Solutions Backup, Snapshots, and Failover Techniques Business Continuity in Cloud Architecture	Configuring backup and restore strategies for cloud resources Implementing a disaster recovery plan on AWS/Azure/GCP
13	**5**	**1,2,3,4,5,7**	**Emerging Trends and Future of Cloud Computing** Edge Computing and Cloud Integration: Overview of Edge Computing and its relevance to Cloud. Use cases and examples. AI/ML in Cloud: How Artificial Intelligence and Machine Learning are integrated into cloud services (e.g., AWS SageMaker, Azure AI).	Explore Use cases and examples and prepare a case study

4. References:

Sl.No	Description
1	"**Cloud Computing**: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood. https://www.amazon.com/Cloud-Computing-Concepts-Technology-Architecture/dp/0133387526
2	"**Cloud Computing** for machineDummies" by Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Fern Halper. https://www.amazon.com/Cloud-Computing-Dummies-Judith-Hurwitz/dp/1119546658
3	"Cloud Infrastructure and Services" by EMC Education Services. https://www.amazon.com/Cloud-Infrastructure-Services-EMC-Education/dp/0133478310
4	"Architecting **Cloud Computing** Solutions" by Kevin L. Jackson and Scott Goessling. https://www.amazon.com/Architecting-Cloud-Computing-Solutions-Kevin/dp/1788472425

5	"Architecting the Cloud: Design Decisions for **Cloud Computing** Service Models (SaaS, PaaS, and IaaS)" by Michael J. Kavis
6	"Cloud Security and Privacy" by Tim Mather, Subra Kumaraswamy, and Shahed Latif. https://www.amazon.com/Cloud-Security-Privacy-Tim-Mather/dp/0596802765
7	"Cloud Storage for Dummies" by Greg Schulz. https://www.amazon.com/Cloud-Storage-Dummies-Greg-Schulz/dp/111808729X
8	"Cloud Native Patterns" by Cornelia Davis. https://www.amazon.com/Cloud-Native-Patterns-Design-Changing/dp/1617294297
9	"Architecting the Cloud" by Michael J. Kavis https://onlinelibrary.wiley.com/doi/book/10.1002/9781118691779?msockid=03f3dcccf0a16497101ad30cf10c6533
10.	"**Cloud Computing**: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti
11	https://aws.amazon.com/
12	https://azure.microsoft.com/
13	https://cloud.google.com/
14	https://nvlpubs.nist.gov/
15.	https://www.awseducate.com/
16.	https://www.cloudskillsboost.google/
17.	https://learn.microsoft.com/en-us/training/azure/
18.	https://aws.amazon.com/getting-started/hands-on/
19.	https://github.com/cloudcommunity/Free-Books
20.	https://openstax.org/
21.	https://www.bookbub.com/
22	https://scholar.google.com/
23.	https://bookboon.com/
24.	https://www.awseducate.com/

5. Suggestive Online Courses

Components From AWS Academy	**Components From AWS Educate, AWS Skill Builder & Other Material**
Introduction to **Cloud Computing** Security. Module 2 - AWS Academy Cloud Security Foundations (CO1) Identity and Access Management (IAM) in Cloud. Module 3 - AWS Academy	AWS Security Best Practices: Computing (CO2)https://explore.skillbuilder.aws/learn/courses/11263/aws-security-best-practices-computing AWS Security – Encryption Fundamentals (CO2)https://explore.skillbuilder.aws/learn/courses/17895/aws-security-encryption-fundamentals AWS Security Best Practices: Monitoring and Alerting (CO2)https://explore.skillbuilder.aws/learn/courses/11264/aws-security-best-practices-monitoring-and-alerting AWS Shield Getting Started (CO2)https://explore.skillbuilder.aws/learn/courses/21855/aws-shield-getting-started Cloud Security & Compliance (CO3) https://explore.skillbuilder.aws/learn/courses/479/aws-identity-and-access-management-architecture-and-terminology

Cloud Security Foundations (CO1) Data Security in the Cloud - Module 5 - AWS Academy Cloud Security Foundations (CO2) Network Security in the Cloud - Module 4 - AWS Academy Cloud Security Foundations (CO2) Cloud Security Monitoring and Incident Response - Module 6 & 7 - AWS Academy Cloud Security Foundations (CO2)	Digital Sovereignty on AWS (CO3)https://explore.skillbuilder.aws/learn/courses/22162/digital-sovereignty-on-aws Security Learning Plan: Threat Detection and Incident Response (CO4) https://explore.skillbuilder.aws/learn/learning-plans/1803/plan & https://explore.skillbuilder.aws/learn/learning-plans/91/plan AWS Lambda Function Security & Defense in Depth (CO4) - https://explore.skillbuilder.aws/learn/courses/53/aws-lambda-function-security-defense-in-depth Deep Dive on Container Security (CO4)https://explore.skillbuilder.aws/learn/courses/72/deep-dive-on-container-security

6. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max Marks	
1.	CIE-1 Theory Test	4	90	50	
2.	CIE-2 Practice Test	7	180	50	Average of all CIEs
3	CIE-3 Theory Test	10	90	50	

4.	CIE-4 Practice Test	13	180	50	
5.	CIE-5 – ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Springboard /Swayam/ NPTEL/ Any other (30)	1-13		50	
				Total Obtained	**/50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.
Rubrics for the Mini Project (if included) should be defined by the course coordinator.

7. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

8. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -V**	
Course Name	**Cloud Computing Architecture**			**Test**	**I/III**
Course Code	**25CS51IB**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	**Questions**	**Cognitive Level**	**Course Outcome**	**Marks**
	Section - 1			
1	a) List benefits and Challenges in **Cloud Computing** b) Define **Cloud Computing** and explain its key characteristics with examples.	L2	1	10 15

2	a) Define IAM (Identity and Access Management) and explain its importance in cloud security. b) Describe the differences between object storage, block storage, and file storage.	L2	1	10 15
Section - 2				
3	a) Compare and contrast distributed computing and grid computing with **Cloud Computing**. b) Suppose you are deploying a web application in a cloud environment. How would you configure firewall rules to restrict access to the application while allowing authorized users to connect to the database and other services?	L3	2	10 15
4	a) Analyze how IAM (Identity and Access Management) helps ensure the security of cloud environments. b) Compare the advantages of using AWS S3, Azure Blob Storage, and Google Cloud Storage for large-scale data storage.	L3	2	10 15
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

9. CIE Practical Test Model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Cloud Computing Architecture**			**Test**	**II/IV**
Course Code	**25CS51IB**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions **(One Question to be picked from the lot by the students)**	**CO**	**Marks**
1. Suppose that the client wants to migrate on-premises web application into AWS cloud. How do you create an EC2 Instance in a VPC, deploy the application and create the AWS S3 storage as per the clients need? 2. Set up a Virtual Private Cloud (VPC) in AWS with the following requirements: a) Create a VPC with a CIDR block of 10.0.0.0/16 b) Create two subnets: one public and one private c) Configure the public subnet to allow incoming HTTP traffic d) Configure the private subnet to deny all incoming traffic e) Create a Network Access Control List (NACL) to restrict traffic between the subnets f) Launch an EC2 instance in the public subnet and verify its connectivity	1,2,3,4,5	50

3. Configure Network Security Groups (NSGs) in Azure to secure a virtual network with the following requirements: a) Create a virtual network with two subnets: one for a web server and one for a database server b) Create two NSGs: one for the web server subnet and one for the database server subnet c) Configure the web server NSG to allow incoming HTTP traffic from the internet d) Configure the database server NSG to allow incoming traffic only from the web server subnet e) Deploy a virtual machine in each subnet and verify their connectivity 4. Set up a cloud database using Amazon RDS with the following requirements: a) Create an RDS instance with MySQL engine b) Configure the instance with a database name, username, and password c) Create a VPC security group to allow incoming traffic on the MySQL port d) Configure the RDS instance to use the VPC security group e) Connect to the RDS instance using a MySQL client and verify its functionality 5. Configure auto-scaling and load balancing for a web application on AWS/Azure: a) Create an auto-scaling group with 2-5 instances b) Scale out at 70% CPU utilization, scale in at 30% c) Configure a load balancer to distribute traffic across instances d) Test the configuration with a simulated traffic spike		
Scheme of Assessment /Valuation 1. Task Identification 2. Practical Implementation 3. Outcome/Result Assessment 4. Viva-Voce 5. Activity Portfolio (record) Assessment		10 15 10 10 05
Total Marks		50

Signature of the Course Coordinator **Signature of the HOD**

10. **Suggested Activities**:

The listed activities are illustrative and not exhaustive. The course coordinator will design a schedule of lab tasks aligned with syllabus content, ensuring hands-on practice with key concepts. Students must maintain a portfolio documenting completed lab work for evaluation.

Sl. No.	**List of Activities** **(To be prepared by the Course coordinator)**
1	Migrate a client's on-premises web application to AWS by: Launching an EC2 instance in a custom VPC, Deploying the web app, Configuring S3 storage for static assets

2	Build a custom VPC with public/private subnets, configure NACLs, and test connectivity by launching an EC2 instance.
3	Secure an Azure virtual network (VNet) by implementing Network Security Groups (NSGs) to control traffic between web and database subnets, then validate connectivity.
4	Set up a MySQL RDS instance in AWS, configure security groups, and verify connectivity using a MySQL client.
5	Configure auto-scaling and load balancing for a web app on AWS/Azure, then test with simulated traffic.
6	Deploy a "Hello World" web app on AWS EC2, Azure App Service, or GCP Compute Engine.
7	Create a VPC in AWS with public/private subnets and configure route tables.
8	Set up NSGs in Azure to allow HTTP/HTTPS traffic and block SSH/RDP from untrusted IPs.
9	AWS: Create an S3 bucket and enable static website hosting. Azure: Set up Blob Storage with private/public access tiers. GCP: Upload files to Cloud Storage and apply lifecycle policies.
10	Deploy a managed database: AWS: Launch an RDS MySQL instance with a security group. Azure: Provision Azure SQL Database with firewall rules. GCP: Create a Cloud SQL PostgreSQL instance
11	Configure auto-scaling: AWS: Create an ASG (2–5 instances) with CPU-based scaling policies. Azure: Set up a VMSS with scale-out (70% CPU) and scale-in (30% CPU) rules.
12	Deploy a load balancer: AWS: Attach an ALB to the ASG and test traffic distribution. Azure: Configure an Azure Load Balancer for VMSS.
13	Enable encryption: AWS: Encrypt an EBS volume using KMS. Azure: Encrypt a VM disk with Azure Key Vault. GCP: Use Cloud KMS for Compute Engine disk encryption.
14	Set up monitoring and alerts: AWS: Create CloudWatch alarms for high CPU usage. Azure: Configure Azure Monitor alerts for VMSS. GCP: Use Cloud Monitoring for uptime checks.
15	Deploy a multi-tier application: Web tier (public), app tier (private), and database tier (isolated).
16	Build serverless applications: AWS: Lambda function triggered by S3 uploads. Azure: Function App processing Blob Storage events. GCP: Cloud Function responding to Pub/Sub messages.
17	Run containerized apps: AWS: Deploy a Flask app using ECS Fargate. Azure: Host a Docker container on AKS. GCP: Launch a Kubernetes cluster on GKE.
18	Automate infrastructure: AWS: Deploy a VPC and EC2 instance using Terraform. Azure: Provision resources with an ARM template.

19	Create CI/CD pipelines: AWS: Build a pipeline with CodePipeline and GitHub. Azure: Set up Azure DevOps for App Service deployments. GCP: Trigger Cloud Build with GitHub Actions.
20	Implement backup strategies: AWS: Schedule EBS snapshots and RDS automated backups. Azure: Configure a Recovery Services Vault for VMs.
21	a)Solve a business problem: Example: Design a scalable e-commerce platform using: AWS: ALB + ASG + RDS Multi-AZ + S3. Azure: Front Door + VMSS + Cosmos DB. b) Document the solution: Submit a 1-page architecture diagram and cost estimate.

11. Equipment/Software list with Specification for a batch of 30 Students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 4GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka

DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**REST API Development with Spring Boot**	**Type of Course**	Integrated
Course Code	**25CS51IC**	**Contact Hours**	8 hrs per week
Teaching Scheme	4:0:4	**Credits**	6
CIE Marks	**50**	**SEE Marks**	**50 (Theory)**

2. Rationale:

In modern software development, RESTful APIs serve as the backbone for building scalable, efficient, and secure web applications. This course is designed to equip students with the knowledge and hands-on experience required to develop, deploy, and manage RESTful services using Spring Boot, a widely adopted framework in the Java ecosystem.

3. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain the impact of digital transformation and driving technologies.
CO-02	Develop an application using Spring Boot while efficiently handling dependencies and builds with Maven.
CO-03	Design and implement RESTful APIs and connect them to a database with Hibernate/Spring Data JPA for efficient CRUD operations.
CO-04	Implement authentication and authorization using Spring Security, OAuth2, and JWT.
CO-05	Write unit tests using JUnit and Mockito, and validate APIs using DTOs and ModelMapper.

4. Course Content

Week	CO	PO	Theory	Practice
1	**1**	1	**Enterprise Development and Digital Transformation.** ▪ Understanding digital transformation and its impact. ▪ Key Technologies Driving Digital Transformation. ▪ Key Technologies for Enterprise Development: Backend, Frontend, Databases, Cloud and DevOps, Security	▪ Assess a company's digital maturity. ▪ Case Studies of Successful Digital Transformation: ▪ Amazon: AI-driven product recommendations, Cloud-based infrastructure (AWS). ▪ Tesla: IoT-enabled smart cars with OTA updates, AI-powered autonomous driving. ▪ Netflix: AI and big data for personalized recommendations, Cloud migration for global streaming scalability.

2	**1,2**	**3,4**	**Introduction to Spring Framework** Overview of the Spring Framework and its Need. Dependency Injection (DI) and Inversion of Control (IoC) principles. Configuring Beans using XML and annotations. Bean Lifecycle ApplicationContext vs BeanFactory.	▪ Implement Constructor Injection and Setter Injection in Spring. ▪ Configure beans using annotations
3	**2**	**3,4**	**Understanding Spring Boot and its Core** ▪ Features , Advantages of Spring Boot over Traditional Spring Framework and its Purpose ▪ Understanding Spring Boot auto-configuration. ▪ Creating and running a Spring Boot application. ▪ Spring Boot Annotations.	Modify and Customize Configurations ▪ Setting up a Spring Boot project using Spring Initializer ▪ Use application.properties to configure port and logging levels ▪ Add custom properties and access them in Java classes
4	**2**	**4**	**Maven Project Management** POM.xml, Maven lifecycle, Dependency Management	▪ Install Maven, explore pom.xml, add dependencies, run Maven commands ▪ create a Maven project with multiple dependencies.
5	**2**	**3,4**	**Spring MVC Basics** MVC architecture and its role in web development. Understanding the Model, View, and Controller components	▪ Build a simple MVC app with controllers and views
6	**3**	**3,4**	**Introduction to REST APIs** API, its working, need RESTful principles REST architectural style REST constraints HTTP request and response	Create a simple REST API (GET, POST, DELETE, PUT)
7	**3**	**3,4**	**Building First REST API in Spring Boot** Understanding @RestController, @RequestMapping - Handling HTTP Methods (GET, POST, PUT, DELETE) - Using @PathVariable and @RequestParam - ResponseEntity and Status Codes	▪ Build a CRUD API for a product catalog ▪ Build a CRUD API for a Shopping cart
8	**3**	**3,4**	**API Endpoints and Testing** API endpoint principles, Postman API testing, Mocking APIs	▪ Test APIs using Postman, Write unit tests for REST controllers
9	**3**	**3,4**	**Hibernate and Spring Data JPA** ORM, Hibernate setup, JPA, CRUD operations (PostgreSQL/MySQL)	▪ Connect REST API with MySQL/PostGreSQL, create entity classes and repositories

10	**4**	**3,4**	**Introduction to Spring Security** ▪ Why security is essential for web applications ▪ Authentication – working, types ▪ Authorization – working, types ▪ Spring Security Architecture ▪ Spring Security Filters and Their Roles	▪ Configure Spring Boot with Spring Security for basic authentication ▪ Implement OTP-based two-factor authentication ▪ Implement basic authentication and role-based access control (RBAC)
11	**4**	**3,4**	OAuth2 and JWT Security OAuth2 components, JWT authentication	▪ Implement JWT authentication for REST API
12	**5**	**4**	**Unit Testing and Code Quality** JUnit, Mockito, code coverage	▪ Write unit tests with JUnit and Mockito ▪ Mock an external service dependency using Mockito.
13	**5**	**3,4**	**Validation, DTOs and Mini-Project** - DTOs, ModelMapper, API validation	Complete a **capstone project** integrating all concepts

4. References:

Sl no	Details
1	**Spring Boot in Action** – *Craig Walls*
2	**RESTful Web Services** – *Leonard Richardson and Sam Ruby*
3	**Building RESTful Web Services with Spring 5** – *Packt Publishing*
4	**Spring Security in Action** – *Laurentiu Spilca*
5	**Java Persistence with Hibernate** – *Christian Bauer, Gavin King, Gary Gregory*
6	https://docs.spring.io/spring-boot/docs/current/reference/html/
7	https://www.baeldung.com/rest-with-spring-series
8	https://www.javaguides.net/2021/09/spring-boot-rest-api-tutorial.html
9	https://spring.io/guides/topicals/spring-security-architecture/
10	https://www.baeldung.com/spring-security-oauth-jwt
11	https://www.baeldung.com/mockito-series
12	https://learning.postman.com/docs/getting-started/introduction/
13	https://spring.io/guides/gs/spring-boot-docker/
14	McKinsey Digital
16	Harvard Business Review - Digital Transformation
17	Baeldung - Maven Tutorials
18	Baeldung - DTOs in Spring
19	Coursera - RESTful API with Spring Boot
20	Udemy - Spring and Hibernate for Beginners

5. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	

2.	CIE-2Practice Test	7	180	50	Average of all CIE=50 Marks
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (20)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -**	
Course Name	REST API Development with Spring Boot			**Test**	**I/III**
Course Code	**25CS51IC**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	**Questions**	**Cognitive Level**	**Course Outcome**	**Marks**
	Section - 1			
1	a) List and explain the key technologies driving digital transformation. (5) b) Discuss the role of ApplicationContext in managing the lifecycle of beans. How does it improve performance and ease of configuration compared to BeanFactory? (10) c) Compare XML-based and annotation-based bean configuration. When should each approach be preferred? (10)	L2	1	25
2	a) Discuss how digital transformation is reshaping different industries. Provide examples. (10) b) Explain the core features of Spring Boot. How do these features make it suitable for modern application development? (10) a) c) How do you create and run a Spring Boot application? Explain the steps. (5)	L2	1	
	Section - 2			
3	a) How does Inversion of Control (IoC) work in Spring? Discuss its advantages over traditional object creation in Java. (10) b) Your Spring application is using BeanFactory, but you notice that lazy loading causes delays in bean initialization. How can switching to ApplicationContext help? (10) c) Explain the significance of the <dependencies> section in the pom.xml file. How does Maven resolve transitive dependencies? (5)	L2	2	25
4	a. Explain the principles of Dependency Injection (DI) in Spring. Compare Constructor Injection and Setter Injection with examples. (10) b. Explain the different phases of a Spring bean's lifecycle. How can lifecycle methods be customized? (10) c. Differentiate between ApplicationContext and BeanFactory. When should each be used? (5)	L2	2	

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	
Course Name	REST API Development with Spring Boot			**Test**	**II/IV**
Course Code	**25CS51IC**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**

You are tasked with developing a **RESTful API** for a **Movie Recommendation System**. The API should allow users to search for movies, filter movies by genre, and get personalized movie recommendations based on user preferences.
• **Movie Search and Filtering:**
  ○ Search for movies by title (partial match).
  ○ Filter movies by genre.
  ○ Sort movies by release year (ascending or descending).
• **Movie Recommendations:**
  ○ Get a list of recommended movies based on a user's favorite genre.
  ○ Get a list of top-rated movies (based on average rating).
Movie Endpoints
• GET /api/movies/search?title={title} – Search movies by title (partial match).
• GET /api/movies/filter?genre={genre} – Filter movies by genre.
• GET /api/movies/sort?order={asc/desc} – Sort movies by release year.
Recommendation Endpoints
• GET /api/movies/recommendations?genre={genre} – Get recommended movies based on genre.
• GET /api/movies/top-rated – Get top-rated movies (based on average rating).
Implement fully functional RESTful API using Spring Boot.
		50

Scheme of assessment **a)** Proper setup of Spring Boot project.(Correct use of dependencies) – 5 **b)** Implementation of endpoints. - 25 **c)** **Code Quality (**Clean, readable, and well-structured code. And Proper use of annotations) - 10 **d)** **Viva - 10**		
Total Marks		**50**

Signature of the Course Coordinator **Signature of the HOD**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**Cyber Physical Systems**	**Type of Course**	Integrated
Course Code	**25CS51ID**	**Contact Hours**	8 hrs per week
Teaching Scheme	4:0:4	**Credits**	6
CIE Marks	**50**	**SEE Marks**	**50 (Theory)**

1. Rationale:

In the era of connected computing devices, securing the personal data, application, system, network and organization becomes the challenging task. The Cyber Security specialization prepare students to take up job or to become entrepreneur in the challenging area of Cyber security

2. **Course Outcomes:** At the end of the Course, the student will be able to:

CO-01	Select appropriate devices, technology, protocol, and platform for a cyber physical system application
CO-02	Design, implement, and analyse a simple cyber physical system with security measures for a real world application
CO - 03	Analyze and Mitigate Cyber Threats and Attacks
CO - 04	Perform Incident Response and Security Monitoring
CO-05	Apply concept of Cloud Computing, cryptography, pen testing and Artificial Intelligence in cyber physical systems and security

1. Course Content

Week	CO	PO	Lecture(4HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1	Overview of cybersecurity and network security. CIA Triad (Confidentiality, Integrity, Availability) Security frameworks: NIST, ISO 27001 TCP/IP and OSI Model Security Considerations Understanding Ports and Protocols (TCP, UDP, ICMP) Packet encapsulation and common vulnerabilities Introduction to Cyber Physical Systems (CPS)	Setting Up a Lab Environment ▪ Install Wireshark, VirtualBox, Kali Linux, Metasploit ▪ Create a virtual test environment with Windows/Linux VMs Packet Analysis Basics ▪ Capture and analyze network traffic using Wireshark Identify normal vs. suspicious traffic patterns ▪ Demonstration of Simple components of Cyber Physical systems such as

			What are Cyber-Physical Systems (CPS)? • Definition of CPS • Key characteristics and components of CPS Evolution of Cyber-Physical Systems	sensors, actuators, Arduino/ raspberry pi boards
2	1	1,2	Foundations of Cyber-Physical Systems Sensor Networks Embedded Systems Secure Networked Control Systems Principles of Cyber-Physical Systems Distinguishing Features of Cyber-Physical Systems CPS applications in various industries Manufacturing Healthcare Agriculture Transportation Energy Smart cities	Demonstrate Working of (using Simulator) TCP UDP ICMP Traffic Analysis and Packet Capture Use Wireshark to capture HTTP, DNS, and FTP traffic Analyze packet headers to detect anomalies Port Scanning with Nmap Use Nmap to scan open ports and identify vulnerabilities
3	1, 2, 3	1,2, 3,7	Analyzing a Cyber Attack Types of Malwares Spyware, Malware, Backdoor, Ransomware, Scareware, Rootkit, Virus, Trojan horse, Worms, Symptoms of attack, Methods of Infiltration: Social Engineering, Pretexting, Tailgating, Something for something (quid pro quo), Denial-of-Service and DDoS, Botnet, On the Path attack SEO Poisoning, Wi-Fi Password Cracking, Password Attacks: Password spraying, Dictionary attack, Brute force, Password Cracking Times, Traffic interception, Advanced Persistent Threats: Security Vulnerability and Exploits Hardware Vulnerabilities Meltdown and Spectre Software Vulnerabilities Categorizing Software Vulnerabilities, Software updates	Demonstrate Security attacks using simulator. Firewall Configuration (pfSense/Windows Defender Firewall) • Set up and configure rules in pfSense • Test firewall rules using Nmap Analyzing Malware-Infected Network Traffic Use Wireshark to analyze malicious traffic patterns Phishing Attack Simulation: Set up a phishing test environment using GoPhish

			Defense in depth Fire wall: Types of firewalls: Packet filtering, Stateful, Proxy Email security: SPF, DKIM, DMARC	
4	1, 2, 3	1,2, 3,7	Interplay between Cyber and Physical Components: • Understanding the integration of cyber and physical components in CPS • Importance of communication and coordination between cyber and physical systems Challenges and opportunities in CPS design and operation Overview of CPS Architecture and Design CPS architecture and components Physical Components and architecture Computational Elements Networking Infrastructure Sensors and Actuators Control Systems Data processing Communication Security and privacy Human – Machine interface	Practical Work: Setting up a Cyber-Physical System: • Designing and configuring a small-scale CPS prototype • Selecting appropriate sensors, actuators, and communication components • Integrating cyber and physical components
5	1, 2, 3, 5	1,2, 3,7	Overview of Key Technologies in CPS: Sensors and Actuators Internet of Things (IoT) • Cloud Computing and edge computing • Machine learning and artificial intelligence Real-time systems and control theory	Working with Sensors and Actuators : Interface sensors and actuators with a microcontroller (Arduino/Raspberry Pi, Temperature Sensor- DHT11, LED, Servo Motor) - Write a Python/Arduino script to read temperature and humidity Internet of Things (IoT) – Connecting Devices to the Internet: Send sensor data to the cloud using MQTT.
6	1, 2, 3, 5	1,2, 3	Symmetric vs. Asymmetric Encryption (AES, RSA) SSL/TLS and VPNs (IPSec, OpenVPN) Digital certificates and PKI	▪ implement AES (Symmetric) and RSA (Asymmetric) encryption using Python, OpenSSL, PyCryptodome Library. ▪ Implementing SSL/TLS Encryption ▪ Generate SSL/TLS certificates using OpenSSL ▪ Setting up a VPN ▪ Install and configure WireGuard/ OpenVPN for secure communication
7	1, 2, 3, 5	1,2, 3,7	Security Threats in CPS Security Measures Implementation:	▪ Configuring Active Directory (AD) for Authentication ▪ Set up Windows Server with Active Directory (AD)

			• Implementing secure communication protocols (e.g., Transport Layer Security) • Configuring intrusion detection and prevention systems Applying access control mechanisms (e.g., rolebased access control) Security Mechanisms in CPS Security Solutions for CPS: Secure communication protocols and encryption techniques (TLC, SSH, IPsec, DTLS. WirelessHart, Zigbee security, MQTT, CoAP, OPC UA, DDS, Modbus, EthernetIP, CAN etc) • Intrusion detection and prevention IDS vs. IPS: Snort, Suricata, OSSEC Network Address Translation (NAT) and its security benefits • Access control mechanisms • Risk assessment and management in CPS ▪ Authentication methods: Passwords, MFA, Biometrics ▪ Role-Based Access Control (RBAC) and Least Privilege Principle ▪ Secure authentication protocols: LDAP, Kerberos, OAuth	▪ Implementing Multi-Factor Authentication (MFA) ▪ Configure Google Authenticator or DUO MFA ▪ Setting up Snort IDS ▪ Install Snort and configure rule-based intrusion detection ▪ Configure SSH for secure remote access and enforce key-based authentication ▪ Set up an IPsec-based VPN using StrongSwan for secure communication
8	1, 2, 3	1,2, 3,4, 7	Wireless security protocols: WEP, WPA2, WPA3 Rogue AP detection and Evil Twin attacks Wi-Fi security best practices Network segmentation and VLANs Demilitarized Zone (DMZ) and honeypots Zero Trust Architecture (ZTA)	Securing Wi-Fi Networks ▪ Configure WPA3 encryption on a router ▪ Wireless Penetration Testing ▪ Use Aircrack-ng to test for weak passwords ▪ Perform Wi-Fi deauthentication attack (in a controlled lab) ▪ Configuring VLANs in Cisco Packet Tracer ▪ Set up VLANs and test network isolation Deploying a Honeypot ▪ Install Dionaea or Cowrie to detect attacks

9	1, 4	2,3, 4,5, 6	Incident management introduction and objectives Stages and life cycle of incident management, Tracking incidents Incident remediation, Reporting and documentation, Incident Closure Incident management teams and models, Incident management services and integration tools - Best practices of Incident Management Fundamentals · CIA · Threat Actors · Different kinds of hackers · Different kinds of teams – Blue, Red, Purple · Criminal Groups · Hactivist Groups · APT · Attack Vectors · Protect/Prevent · Detect/Respond · Trust Positive vs False Positive Data · Bits and Bytes Charter Encoding (ASCII, UTF8,Base64) · File Magic Bytes, Hashes · Imphash · Ssdeep	Simulating a Security Incident Response : Learn how to respond to a security incident using a structured approach Analyzing File Signatures and Hashes : Identify malicious files using hashing and magic bytes (Linux CLI, Virus Total)
10	4	2,3, 4,5, 6	Importance of network monitoring SIEM systems (Splunk, ELK Stack) Log correlation for threat detection Disk imaging and memory analysis, Basics of Incident Response · Alert processing · Procedures, runbooks and reference · Response options · Escalations · Incident categories · Incident Resolution Codes Data vs Intelligence · Indicators of compromise (IoCs) · Malware analysis · Accessing IoCs · Contacting threat intelligence Analysis tools · Anomaly · Domain tools · WhoIS · Passive DNS · Virus total · Dynamic File analysis	SIEM Log Analysis with Splunk ▪ Ingest logs into Splunk and create security alerts Analyzing Logs with ELK Stack ▪ Use Kibana to visualize network events ▪ Disk Imaging and Forensic Analysis ▪ Use Autopsy or FTK Imager to analyze evidence ▪ Memory Dump Analysis ▪ Capture and analyze RAM dumps using Volatility

11	5	2,3, 4,5, 7	Cloud and IoT Security Basics of Cloud Computing, Why is Cloud Computing necessary? Introduction to key cloud services (Compute, storage, networking) Cloud delivery models IaaS v/s PaaS v/s SaaS Introduction to cloud vendors(Azure,AWS, GCP) Key Cloud Security Principles - Cloud security risks (AWS, Azure) - Shared responsibility model IoT device vulnerabilities and security solutions	▪ Securing AWS Cloud Resources ▪ Configure IAM policies, Security Groups in AWS ▪ IAM Best Practices, IAM Audit, VPC WAF ▪ Data protection in the cloud ▪ Data protection at rest and at transit ▪ Cloud data storage - AWS EBS, S3 / Azure SAS ▪ Secrets Management IoT Device Security Testing ▪ Scan IoT devices using Shodan
12	5	2,3, 4,5, 7	Penetration Testing and Ethical Hacking Phases of penetration testing (Reconnaissance, Exploitation, Post-Exploitation) Tools: Metasploit, Nmap, Burp Suite Red Team vs. Blue Team operations	▪ Network Scanning and Exploitation ▪ Use Nmap and Metasploit to find and exploit vulnerabilities Web Application Security Testing ▪ Test for SQL injection, XSS using Burp Suite
13	1, 2, 3, 4, 5	7	Capstone Project and Final Review ▪ Review of major security concepts ▪ Career paths in cybersecurity ▪ Certification guidance (CompTIA Security+, CEH, CCNA Security)	Final Capstone Project - Students conduct a full security assessment of a simulated network - Implement defensive measures against simulated attacks

4. References:

Sl no	Details
1	Principles of Cyber-Physical Systems – Rajiv Alur
2	Cyber-Physical Systems – Rajkumar
3	Cyber-Physical Systems: From Theory to Practice – Danda B Rawath
4	Cyber-Physical Systems: A Comprehensive Guide
5	https://www.isc2.org/
6	https://www.netacad.com/
7	http://williamstallings.com/Cryptography/
8	https://purplesec.us/learn/firewall-penetration-testing/
9	https://cse29-iiith.vlabs.ac.in/

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1 Theory Test (Week-1 to week-6)	7	90	50	Average of all CIE=50 Marks

2.	CIE-2 Practice Test(Week-1 to Week-6)	7	180	50	
3	CIE-3 Theory Test (Wee7-Week 12/13)	13	90	50	
4.	CIE-4 Practice Test (Week-7 to Week-11)	12	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(20)	13	-	50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -**	
Course Name	**Cyber Physical Systems**			**Test**	**I/III**
Course Code	**25CS51ID**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	**Questions**		**Cognitive Level**	**Course Outcome**	**Marks**
	Section - 1				

1	a) Explain the CIA Triad (Confidentiality, Integrity, Availability) with real-world examples. (6 Marks) b) Discuss the various types of firewalls and their advantages and disadvantages. (6 Marks) c) What is an Advanced Persistent Threat (APT)? How does it work, and how can it be prevented? (6 Marks) d) Explain Social Engineering attacks with examples of Phishing, Pretexting, and Tailgating. (7 Marks)	L2	1	25
2	a) Compare Symmetric and Asymmetric Encryption and give examples of cryptographic algorithms used in each. (6 Marks) b) Describe the working of SSL/TLS protocols and their importance in securing web communications. (6 Marks) c) Explain DDoS attacks and discuss at least two prevention techniques. (6 Marks) d) What are common password attacks? Explain Brute force, Dictionary attack, and Password spraying. (7 Marks	L2		
Section - 2				
3	a) Define Cyber-Physical Systems (CPS) and describe their key characteristics. (6 Marks) b) Discuss the integration of IoT, Cloud Computing, and AI in CPS. (6 Marks) c) Explain the role of sensors and actuators in CPS, and provide examples. (6 Marks) d) Describe the architecture and networking infrastructure of CPS. (7 Marks)	L2		25
4	a) Explain the security challenges in CPS and suggest solutions for mitigating them. (6 Marks) b) Discuss two real-world applications of CPS in industries like Healthcare, Transportation, Smart Cities, or Energy. (6 Marks) c) Describe Hardware vulnerabilities such as Meltdown and Spectre and their impact on cybersecurity. (6 Marks) d) What are common software vulnerabilities in CPS? Discuss the importance of software updates and patch management. (7 Marks)	L2		

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Signature of the Course Coordinator **Signature of the HOD** **Signature of the IQAC Chairman**

8. CIE Practice Test model question paper

Program	Computer Science and Engineering			Semester	V
Course Name	Cyber Physical Systems			Test	II/IV
Course Code	25CS51ID	Duration	180 min	Marks	50
Name of the Course Coordinator:					

Questions	CO	Marks
You are a cybersecurity expert tasked with **securing a network and implementing a basic Cyber-Physical System (CPS)**. Perform the following steps and document your findings: 1. **Network Security and Traffic Analysis (15 Marks)** o Use **Wireshark** to capture network traffic. o Identify and analyze **one normal packet** and **one suspicious packet**. o Explain potential threats from the suspicious packet. 2. **Port Scanning and Firewall Configuration (10 Marks)** o Use **Nmap** to scan open ports on a test system. o Configure a **firewall rule (pfSense/Windows Defender Firewall)** to block a specific port. o Verify the firewall rule using Nmap. 3. **Encryption and Secure Communication (10 Marks)** o Implement **AES (Symmetric) or RSA (Asymmetric) encryption** in **Python/OpenSSL**. o Encrypt and decrypt a sample message. 4. **Cyber-Physical System (CPS) Prototype (15 Marks)** o Interface a **sensor (e.g., DHT11 for temperature) and actuator (LED/Servo motor)** with an **Arduino/Raspberry Pi**. o Write a **Python or Arduino script** to collect and display sensor data. o Log the data and control the actuator based on sensor readings.		50
Scheme of Evaluation **Task** / **Marks** Wireshark Capture and Packet Analysis — 15 Nmap Scan and Firewall Rule Implementation — 10 AES/RSA Encryption and Decryption — 10 CPS Sensor-Actuator Integration and Data Logging — 15		
	Total Marks	50

Signature of the Course Coordinator **Signature of the HOD**

9. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**Machine Learning**	**Type of Course**	Integrated
Course Code	**25CS52IA**	**Contact Hours**	7 hrs per week
Teaching Scheme	L:T:P - 3:0:4	**Credits**	5
CIE Marks	**50**	**SEE Marks**	**50(Practice)**

1. **Rationale:**

In today's data-driven world, Machine Learning (ML) plays a pivotal role in automating decision-making and driving innovation across industries. As businesses increasingly rely on data-driven insights, there is a growing demand for professionals skilled in ML concepts, data analysis. This course provides a structured approach to understanding ML fundamentals, performing Exploratory Data Analysis (EDA), and applying Feature Engineering to optimize datasets for model training. Learners will build, train, and evaluate ML models using appropriate performance metrics to ensure accuracy and reliability. Additionally, the course covers MLOps principles for model deployment and monitoring, equipping students with the skills needed to integrate ML solutions into real-world applications.

2. **Course Outcomes :At the end of the Course, the student will be able to:**

CO-01	Explain Machine Learning Concepts.
CO-02	Perform Exploratory Data Analysis (EDA) and Feature Engineering
CO-03	Build, train and evaluate machine learning models
CO-04	Analyze and evaluate machine learning models using appropriate performance metrics
CO-05	Apply MLOps Principles for Model Deployment and Monitoring

3. **Course Content**

WEEK	CO	PO	Theory	Practice
1	1	1 2 4	**Introduction to Machine Learning(ML)** AI vs. ML vs. DL Types of ML: Supervised, Unsupervised, Reinforcement ML Workflow Overview Applications of ML in different domains Importance of Data in Machine Learning **Data Collection** - Population and sample	Explore Python libraries used in Machine Learning. Install ML Packages Explore Online data Repositories Use Python libraries like ***requests*** and ***BeautifulSoup*** for web scraping Classify the dataset as **Cross-sectional** or **Time-series**

			- Types of data :type 1 (cross sectional, time series), type 2 (univariate, multivariate) -Variable types (categorical, ordinal, ratio, interval)	Identify the variable types in the dataset as **Categorical, Ordinal, Interval, Ratio**
2	**1,2**	**2,4**	**Math for Machine Learning (***Basic concepts only***.)** **Statistics for ML** ▪ *Descriptive statistics*: Central tendency(mean, median,mode), Dispersion(Min and Max,Standard Deviation,Variance,Skewness,Kurtosis), Covariance & Correlation ▪ *Inferential statistics*: Population vs Sample, Confidence Intervals, Hypothesis Testing (p-value, Type I & II Errors), Z-test / T-test / Chi-square test, ANOVA, Central Limit Theorem **Probability** ▪ Conditional & Joint Probability ▪ Bayes' Theorem ▪ Probability Distributions (Discrete, continuous, central limit theorem) **Linear Algebra** Scalars, Vectors, Matrices, Tensors ▪ Eigenvalues & Eigenvectors ▪ Matrix Operations (Dot Product, Determinant, Inverse)	**Descriptive Statistics**(using pandas, matplotlib, seaborn) - Load a small CSV (e.g., students' marks dataset) - Calculate mean, median, mode, std dev, variance - Plot histograms & boxplots **Inferential Statistics(**using scipy, numpy) - Simulate a hypothesis test (e.g., A/B test for two groups) - Perform a t-test - Interpret p-value & confidence interval **Probability(**using random, numpy, matplotlib) - Simulate coin tosses to explain joint & conditional probability - Plot normal distribution **Linear Algebra(**using numpy) - Create scalars, vectors, matrices,and tensors. - Perform dot product - Compute determinant & inverse - Use numpy.linalg.eig to get eigenvalues/vectors
3	**5**	**3,4 6**	**Data Preparation and Exploration** **Data reduction:** Numerosity data reduction, Dimensionality data reduction **Data Integration:**Adding attributes**,** Adding data objects. **Handling Missing values** **Handling Outliers(IQR, Z-score)** **Data transformation:** - Encoding(Label Encoding, One-Hot Encoding, Ordinal Encoding), - Scaling(StandardScaler, MinMaxScaler, RobustScaler), - Normalization(L1, L2 normalization),	You are given the Titanic dataset. Perform the following operations step by step: ▪ Drop irrelevant columns ▪ Handle missing values in age (median) & embarked (mode) ▪ Create a new column: family_size = sibsp + parch ▪ Detect outliers in fare using IQR and Z-score ▪ Encode sex (Label), embarked (One-Hot), class (Ordinal) ▪ Scale age and fare using StandardScaler

			- Standardization, - Ranking transformation, - Discretization(Binning) **Dealing with imbalanced dataset**	▪ Normalize fare (L1 & L2) ▪ Bin age into 4 equal-frequency groups ▪ Check survived column distribution ▪ Apply upsampling to balance the dataset
4	**1, 2**	**1,2 ,3, 4**	**Exploratory data analysis** **Univariate data analysis -** - Characterizing data with descriptive statistics(Mean, Median, Mode, Std Dev, Variance, Range, Skewness, Kurtosis) - Univariate distribution plots - Univariate comparison plots - Univariate composition plots **Univariate analysis tests:** Hypothesis testing, Error, Test statistic, type, interpreting test statistics. Understanding p-value. **Multivariate data analysis** Finding relationship in data - Covariance - Correlation Multivariate distribution plot Multivariate comparison plot Multivariate relationship plot Multivariate composition plot	Use the Titanic dataset to perform detailed univariate and multivariate data analysis. **Univariate analysis** ▪ Apply descriptive statistics(Age, Income, Spending Score), ▪ Distribution Plots(Histogram of Age, Income) ▪ Comparison Plots(Boxplot of Income by Gender) ▪ Composition Plots(Pie chart of Gender %) ▪ Hypothesis testing(T-test: Male vs Female Income) ▪ **Multivariate analysis** ▪ Relationship: Covariance/Correlation (Income vs Spending Score) ▪ Distribution Plot(Pairplot of all features) ▪ Comparison Plot(Barplot: Region vs Avg Spend) ▪ Relationship Plot(Scatter: Age vs Spend w/ hue) ▪ Composition Plot(Stacked bar: Product × Region)
5	**5**	**2,3 ,4, 6**	**Introduction to Version Control** ▪ What is Version Control? ▪ Types of Version Control: Local, Centralized, Distributed ▪ Introduction to Git and Github ▪ Git Workflow ▪ Git basics : init, add, commit, log, ▪ config ▪ Working with remote repositories: clone, push, pull. ▪ **Branching, and Merging** ▪ Why use branches? ▪ Creating and switching branches	**Exercise :** 1. Install Git and set up a repository. 2. Set up your Git username and email. 3. Initialize a Git repository. Exercise : 1. Create a text file and add it to Git. 2. Modify and commit changes. 3. View commit history. 4. Modify a file and discard the changes. 5. Reset a commit. Exercise:

			▪ Forking a Repository ▪ Merging ▪ Types: Fast-forward Merge, Three-way Merge ▪ Command: branch, checkout, merge ▪ Handling merge conflicts ▪ how to identify, resolve, and commit merge conflicts	a) Clone an existing repository from GitHub. b) Make changes, commit, and push them to GitHub. c) Pull the latest changes from the remote repo. Exercise: a) Create a new branch. b) Make changes and commit. c) Merge the branch into main. Exercise: Handling Merge Conflicts
6	**3**	**1,2,3,4**	**Regression: Simple linear Regression** **-** Introduction to Linear Regression - Understanding the equation of a line: **$y = mx + c$** - Cost Function: MAE, MSE - Optimization: Gradient Descent -Evaluation Metrics: R^2, RMSE, MAE **Overfitting vs underfitting** **Data Splitting:** Training set, Validation set, Testing set	**Dataset:** House Price Prediction **Problem Statement**: Predict the price of a house based on various features -Load and explore the housing dataset -Preprocess – handle missing values, normalize features if needed. -Split the dataset into training and test sets using train_test_split(). -Build a Linear Regression model -Evaluate the model using regression metrics like R^2 Score, RMSE, and MAE. **Visualize** predictions vs. actual prices.
7	**3**	**1,2,3,4**	**Classification: Logistic Regression** - Concept of Sigmoid Function - Decision Boundary & Probability Interpretation - Cost Function: Binary Cross-Entropy loss -Evaluation Metrics: Confusion Matrix, Accuracy, precision, recall, F1-score, AUC-ROC	**Dataset**: Titanic Dataset – Kaggle **Problem Statement**: Build a logistic regression model that predicts whether a passenger survived the Titanic disaster. -Preprocess missing data, encode categorical variables, and split the dataset. -Evaluate the model using metrics like accuracy, precision, recall, F1-score, and AUC-ROC.
8	**3**	**1,2,3,4**	**Classification: Decision Tree** - What is a Decision Tree? - Working of Decision Trees (Splitting, Root & Leaf Nodes) - Cost Function: Gini Impurity, Entropy & Information Gain	**Dataset**: Breast Cancer Wisconsin Diagnostic Dataset – scikit-learn / UCI **Problem Statement**: Build a decision tree classifier to predict whether a tumor is malignant or benign.

			Optimization: Pruning (Pre-pruning and Post-pruning)	-Apply feature scaling and split the data into training and testing sets. -Train a decision tree classifier -Evaluate the model using confusion matrix, accuracy, AUC-ROC, and feature importance plot
9	**3**	**1,2,3,4**	**Classification: Ensemble model** What is Ensemble Learning? - Bagging vs. Boosting - Why ensemble methods work **Random Forest** - How Random Forest works (Bootstrap Aggregation) - Advantages of Random Forest over Decision Trees	**Dataset**: Breast Cancer Wisconsin Diagnostic Dataset – scikit-learn / UCI **Problem Statement**: -Build a random forest classifier to predict whether a tumor is malignant or benign. -Apply feature scaling and split the data into training and testing sets. -Train a Random Forest model -Evaluate the model using confusion matrix, accuracy, AUC-ROC, and feature importance plot.
10	**4**	**1,2,3,4**	**Unsupervised Learning: Clustering** **K-Means Algorithm** - What is Clustering? - How K-Means works -Choosing the right value of K (Elbow method, Silhouette Score) Evaluation Metrics - Inertia - Dunn Index **Applications of K-Means**	**Dataset**: Iris Dataset. -Perform basic preprocessing (handling missing values, feature scaling, etc.). -Implement K-Means using **Scikit-learn**. -Visualize the clusters using Matplotlib/Seaborn. -Use the **Elbow Method** to determine the optimal number of clusters. -Use the **Silhouette Score** to validate clustering performance. -Compute **Inertia** and **Dunn Index** for model evaluation. -Analyze and interpret the results
11	**5**	**1,2,3,4,6**	**MLOps Basics** **What is MLOps and Why?** Importance, real-world examples **ML Lifecycle Overview**: Development → Training → Deployment → Monitoring **DevOps vs. MLOps**: Differences, challenges in ML systems	Setup Git & GitHub: Create repo, push ML project Structure ML project: Organize folders for data, models. Track experiments: Use MLflow or CSV to log accuracy, hyperparams Mini Workflow: Train + evaluate model with logs.
12	**5**	**1,2,3,4,6**	**MLOps contd.** : Model Packaging, Deployment, and Basic Monitoring What is Deployment? Local vs. Cloud; Web API-based	Save & Load Model(Serialize using Pickle/Joblib)

			Model Packaging: Using pickle or joblib, basics of Docker **Monitoring:** Need for logs, detecting changes in input/output	Deploy with Streamlit(Build a simple web UI for model predictions) Logging Inputs/Outputs(Log predictions to CSV to simulate monitoring) Final Integration(End-to-end run: Load model → deploy → monitor inputs)
13	**5**	**1,2,3, 4,6**	**Mini Project**:	Train a regression/classification/clustering model on a dataset. Track model performance using Git + MLflow (or manual logs). Deploy the trained model using Streamlit or Flask. Log inputs and outputs for monitoring.

4. References:

Sl no	Description
1	"Python Machine Learning", Sebastian Raschka
2	"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Aurélien Géron
3	"The Hundred-Page Machine Learning Book", Andriy Burkov
4	"An Introduction to Statistical Learning" Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
5	"Foundations of Machine Learning", Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar
6	https://developers.google.com/machine-learning/crash-course
7	https://www.kaggle.com/learn/intro-to-machine-learning
8	https://microsoft.github.io/ML-For-Beginners/
9	https://www.coursera.org/specializations/machine-learning-introduction
10	https://uclaacm.github.io/gradient-descent-visualiser/#playground

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/	1-13		50	

	Swayam/NPTEL/AWS /any other (20)				
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Practice Assessment Methodologies

Sl. No	SEE – Practice Assessment	Duration(minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. CIE Theory Test model question paper

Program	**Computer Science and Engineering**		**Semester - 5**	
Course Name	**MACHINE LEARNING**		**Test**	**I/III**
Course Code	**25CS52IA**	**Duration: 90 min**	**Marks**	**50**
Name of the Course Coordinator:				
Note: Answer any one full question from each section. Each full question carries equal marks.				

Q.No	Questions	CL	CO	Marks
	Section - 1			
1	a) Explain the different types of Machine Learning algorithms. Provide an example for each. b) Discuss at least three real-world applications of Machine Learning across different industries. c) What are some common challenges faced while implementing Machine Learning models?	**L2, L4**	**1,4**	**10** **10** **5**
2	a) Explain Linear Regression with an example. Derive the equation for the line of best fit. b) Describe the working of Decision Trees and how they are used for classification tasks. c) Differentiate between parametric and non-parametric Machine Learning models.	**L2, L3**	**1,3**	**10** **10** **5**
	Section - 2			

3	a) Explain the steps involved in handling missing values in a dataset. Why is it important? b) Discuss how data visualization techniques like histograms, boxplots, and heatmaps help in understanding the dataset. c) What is the significance of detecting and handling outliers in Machine Learning?	**L3**	**2**	**10** **10** **5**
4	a) Explain different encoding techniques for categorical variables in Machine Learning. b) Why is it important to split data into training and testing sets? Discuss the role of stratified sampling in classification problems. c) Compare Standardization and Normalization?	**L3,L4**	**2,3,4**	**10** **10** **5**
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Machine Learning**			**Test**	**II/IV**
Course Code	**25CS52IA**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
Perform an End-to-End Data Preprocessing Workflow for a Machine Learning Dataset. · Install Anaconda and set up a Jupyter Notebook environment. · Load and Explore the Dataset. · Handle Missing Values · Perform Data Visualization and Outlier Detection · Feature Engineering and Data Scaling · Splitting Data for Model Training and Testing				**2,3,4**	**50**
Scheme of assessment **a) Installation and Setup** **b) Exploring datasets** **c) Handle Missing Values** **d) Data Visualization** **e) Feature Engineering and Data Scaling** **f) Model Training and Testing**					 **5** **10** **5** **10** **10** **10**
				Total Marks	**50**

Signature of the Course Coordinator Signature of the HOD

9. SEE- Model Practice Question Paper

Program	Computer Science and Engineering		Semester	5
Course Name	Machine Learning in AI	Course Code	Duration	180 min
Questions			CO	Marks
Question: Build and Evaluate a Predictive Model You are given a dataset containing information about a business domain (e.g., house prices, customer churn, or loan approval). Perform the following tasks: 1. Data Preprocessing (15 Marks) 2. Exploratory Data Analysis (10 Marks) 3. Model Training and Evaluation (15 Marks) 4. Model Optimization (10 Marks)			2,3,4	50
Scheme of assessment **a) Load the dataset and display its structure, Handle missing values and Perform feature encoding.** **b) Visualize, detect ad handle outliers, generate correlation heat map.** **c) Training, testing and evaluation of the model.** **d) Perform Hyper parameter Tuning and apply Feature Scaling**				 **15** **10** **15** **10**
			Total Marks	**50**

1) Signature of the Examiner **2) Signature of the Examiner**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**Cloud Networking**	**Type of Course**	Integrated
Course Code	**25CS52IB**	**Contact Hours**	7 hours/week
Teaching Scheme	L: T:P - 3:0:4	**Credits**	5
CIE Marks	50	**SEE Marks (Practical)**	50

5. Rationale:

This course introduces foundational principles of cloud networking, covering models, data centres, hybrid and multi-cloud setups, and advanced topics. With a focus on practical application, it provides hands-on experience with tools and major platforms like AWS, Azure, and GCP. Students gain skills to design, configure, and manage solutions, preparing them for roles in cloud architecture, networking, and IT, while aligning with trends like SDN, NFV, AI, and serverless computing. It also builds a strong foundation for advanced studies across disciplines.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Configure cloud networks and deploy virtualized infrastructure
CO-02	Deploy secure, scalable cloud networks using VPCs and load balancing
CO-03	Automate hybrid/multi-cloud networks using SDN/NFV and infrastructure-as-code tools.
CO-04	Deploy, troubleshoot, and optimize container networking in cloud environments.
CO-05	Configure and implement cloud-based DNS, CDNs, and serverless networking solutions to support modern application deployment.

3. Course Content

WEEK	CO	PO	Theory (3 Hrs/Week)	Practice (4 Hrs/Week)
1	**1**	**1,2,4,7**	**Introduction to Cloud Computing and Services** Overview of **Cloud Computing**. Overview of Cloud Services. Public Cloud Networking Services: Virtual Networks, VPC, VPNs	▪ Explore the various providers for services like AWS, Azure, GCP, etc ▪ Create EC2 /VN/Compute Engine instances with different OS Images (Free-

			Cloud Infrastructure: Virtualization, Compute, Storage, and Networking	tier, Windows, Linux)
2	**1**	**1,2,3,4,7**	**Introduction to Cloud networking** Types of Networks in Cloud Basic Cloud Networking Terminologies Key Components of Cloud Networking in AWS/Azure /GCP Cloud Network Topologies Cloud Data Centers and their Networking Architecture.	▪ Explore AWS VPC, Azure VNet, and GCP VPC ▪ Setting up basic networks in cloud platforms (AWS / Azure / GCP/etc). ▪ Create a Virtual Private Cloud (VPC) and configure basic networking features.
3	**1**	**1,2,3,4,7**	**Virtual Private Cloud (VPC) and Subnets** Understanding VPC and Subnets. IP Addressing and CIDR Blocks. Public Cloud IP Addressing: Static vs Dynamic IPs Public vs Private Subnets	▪ Create and Configure Subnets in a VPC ▪ Configure Virtual Routers and Gateways for internet access.
4	**2**	**1,2,3,4,7**	**Virtual Private Cloud (VPC) Configuration** Advanced VPC Features: Peering, Transit Gateway, VPC Endpoints VPC Security: Security Groups, Network Access Control Lists (NACLs) VPC Flow Logs and Monitoring	▪ Configure VPC peering and Transit Gateways for inter-VPC communication. ▪ Set up and test security groups and NACLs in AWS or Azure.
5	**2**	**1,2,3,4,7**	**Network Security in the Cloud** Cloud Security Models and Shared Responsibility Security Groups, Firewalls, and VPNs Cloud Security Best Practices: Data Encryption, Access Control, IDS/IPS Network ACLs Firewall Rules in AWS, Azure, and GCP	▪ Configure Security Groups and NACLs ▪ Set up firewalls and configure security rules on VPCs/ VNet in AWS/Azure. ▪ Encrypt network traffic and set up VPNs for secure communication.
6	**2**	**1,2,3,4,5,7**	**Cloud Load Balancing and Auto-Scaling** Introduction to Load Balancing Concepts Types of Load Balancers: Application Load Balancer, Network Load Balancer, Classic Load Balancer Cloud-based Load Balancing Services (AWS Elastic Load Balancer, Azure Load Balancer, etc) Auto-Scaling in the Cloud	▪ Set Up a Load Balancer and Auto-Scaling Group ▪ Set up and configure Load Balancing in AWS or Azure. ▪ Test load balancing with multiple cloud-based virtual machines.
7	**3**	**1,2,3,4,5,7**	**Networking in Hybrid Cloud and Multi-cloud, SDN and NFV**	▪ Set Up a Hybrid Cloud Connection

			Hybrid Cloud Overview and Networking Challenges Connecting On-Premises Data Centers to Cloud Networks (VPN, Direct Connect, ExpressRoute) Hybrid Cloud Architecture and Networking Design Overview of Multi-Cloud Networking Introduction to SDN and its benefits in Cloud Networking Overview of NFV and its role in cloud networks.	▪ Set up a Hybrid Cloud Network using AWS Direct Connect or Azure ExpressRoute. ▪ Configure VPN connections between local network and cloud network.
8	**3**	**1,2,3,4,5,7**	**Network Automation in the Cloud** Introduction to Network Automation Tools for Automation (Terraform, CloudFormation, /Ansible).	▪ Automate Network Configuration ▪ Automate VPC/VNet and subnet setup using Terraform ▪ Automate Network Firewall rules with AWS Cloud Formation
9	**4**	**1,2,3,4,7**	**Cloud Networking for Containers** Networking for Containers (Docker, Kubernetes) AWS ECS/EKS, Azure AKS, and GCP GKE Networking.	▪ Configure Networking for Containers ▪ Deploy a Containerized Application with ECS/AKS.
10	**4**	**1,2,3,4,5,7**	**Troubleshooting Cloud Networks** Network Troubleshooting Tools in Cloud Environments (ping, traceroute, tcpdump). Diagnosing and Resolving Connectivity Issues in Cloud Best Practices for Cloud Network Troubleshooting.	▪ Perform network troubleshooting using AWS or Azure tools. ▪ Simulate network issues and resolve them using troubleshooting techniques.
11	**4**	**1,2,3,4,5,7**	**Cloud Network Monitoring and Performance Optimization** Tools for Cloud Network Monitoring (CloudWatch, Azure Monitor, GCP Operations Suite). Performance Metrics: Latency, Throughput, Packet Loss, etc. Network Optimization Techniques for Cloud Environments.	▪ Set up monitoring for cloud networks using AWS CloudWatch or Azure Monitor. ▪ Analyse network performance and optimize for latency and throughput.
12	**5**	**1,2,3,4,5,7**	**Cloud DNS and Content Delivery Networks (CDNs)**	**Mini Project Work** (Students will work in groups to design and implement a cloud Networking Project)

			Domain Name System (DNS) in the Cloud. Introduction to CDNs (AWS CloudFront, Azure CDN, GCP Cloud CDN) .	
13	**5**	**1,2,3,4,5,7**	**Cloud Networking for Serverless Architectures** Networking for Serverless Functions (AWS Lambda, Azure Functions, GCP Cloud Functions) API Gateway and Endpoint Security.	**Mini Project Work** (Submission of report, Review and Evaluation.)

4. References:

Sl.No	Description
1	"**Cloud Computing**: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood. https://www.amazon.com/Cloud-Computing-Concepts-Technology-Architecture/dp/0133387526
2	"**Cloud Computing** for Dummies" by Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Fern Halper. https://www.amazon.com/Cloud-Computing-Dummies-Judith-Hurwitz/dp/1119546658
3	"AWS Certified Solutions Architect Official Study Guide" by Joe Baron, Hisham Baz, and Tim Bixler. https://www.amazon.com/AWS-Certified-Solutions-Architect-Official/dp/1119138558
4	"Microsoft Azure Networking: The Definitive Guide" by Mohamed Waly. https://www.amazon.com/Microsoft-Azure-Networking-Definitive-Guide/dp/1484264398
5	"Cloud Infrastructure and Services" by EMC Education Services. https://www.amazon.com/Cloud-Infrastructure-Services-EMC-Education/dp/0133478310
6	"Cloud Networking: Understanding Cloud-Based Data Center Networks" by Gary Lee. https://www.amazon.com/Cloud-Networking-Understanding-Cloud-Based-Networks/dp/0128007281
7	"Cloud Security and Privacy" by Tim Mather, Subra Kumaraswamy, and Shahed Latif. https://www.amazon.com/Cloud-Security-Privacy-Tim-Mather/dp/0596802765
8	https://aws.amazon.com/
9	https://azure.microsoft.com/
10	https://cloud.google.com/
11	https://docs.aws.amazon.com/
12	https://learn.microsoft.com/
13	https://aws.amazon.com/getting-started/hands-on/
14	https://learn.microsoft.com/en-us/training/azure/
15	https://www.cloudskillsboost.google/
16	https://www.awseducate.com/
17	https://openstax.org/
18	https://www.bookbub.com/
19	https://scholar.google.com/
20	https://bookboon.com/

5. Suggestive Online Courses

Course	Components From AWS Academy	Components From AWS Skill Builder

Cloud Networking (Course Code: 25CS52IB)	Introduction to **Cloud Computing** and Services | Introduction to Cloud networking. Module 1 and 5 - AWS Academy Cloud Foundations (CO1) Virtual Private Cloud (VPC) and Subnets. Module 7 - AWS Academy Cloud Architecting (CO1) Virtual Private Cloud (VPC) Configuration Module 8 - AWS Academy Cloud Architecting (CO2) Cloud Load Balancing and Auto-Scaling Module 10 - AWS Academy Cloud Foundation & Module 10 - AWS Academy Cloud Architecting (CO2) Networking in Hybrid Cloud and Multi-cloud, SDN and NFV- Module	AWS Networking Basics (CO1) https://explore.skillbuilder.aws/learn/courses/12439/aws-networking-basics Configuring and Deploying VPCs with Multiple Subnets (CO 1 and 2) https://explore.skillbuilder.aws/learn/courses/499/configuring-and-deploying-vpcs-with-multiple-subnets Amazon Virtual Private Cloud (Amazon VPC) – Troubleshooting (CO2) https://explore.skillbuilder.aws/learn/courses/13490/amazon-virtual-private-cloud-amazon-vpc-troubleshooting AWS Security – Encryption Fundamentals (CO2) https://explore.skillbuilder.aws/learn/courses/17895/aws-security-encryption-fundamentals AWS Network Firewall Getting Started (CO2) https://explore.skillbuilder.aws/learn/courses/21952/aws-network-firewall-getting-started Getting Started with AWS CloudFormation (CO3) https://explore.skillbuilder.aws/learn/courses/3627/getting-started-with-aws-cloudformation Amazon Elastic Kubernetes Service (EKS) Getting Started (CO4) https://explore.skillbuilder.aws/learn/courses/22416/amazon-elastic-kubernetes-service-eks-getting-started AWS Network – Monitoring and Troubleshooting (CO4) https://explore.skillbuilder.aws/learn/courses/8187/aws-network-monitoring-and-troubleshooting Amazon CloudFront Getting Started (CO5) https://explore.skillbuilder.aws/learn/courses/22450/amazon-cloudfront-getting-started AWS Lambda Foundations (CO5) https://explore.skillbuilder.aws/learn/courses/99/aws-lambda-foundations

	8 - AWS Academy Cloud Architecting (CO3) Cloud Networking for Containers – Module 8 - AWS Academy Cloud Developing (CO4) Cloud Networking for Serverless Architectures - Module 7 - AWS Academy Cloud Developing (CO5)	

6. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max Marks	
1.	CIE-1 Theory Test	4	90	50	Average of all CIEs
2.	CIE-2 Practice Test	7	180	50	
3.	CIE-3 Theory Test	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5.	CIE-5 – • Portfolio evaluation (10) • Online Course/s of minimum 15 Hrs.in Infosys Springboard / Swayam / AWS / NPTEL / Any other (20) • Mini Project (20)	13	-	50	
				Total Obtained	**/50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

7. SEE – Practical Assessment Methodologies

Sl. No	SEE – Practical Assessment	Duration (minutes)	Max Marks	Min marks to pass
1.	Semester End Examination-Practical	180	50	20

8. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester -V	
Course Name	Cloud Networking			Test	I/III
Course Code	25CS52IB	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section-1			
1	a) Define IaaS, PaaS, and SaaS with examples. b) What is a Virtual Private Cloud (VPC), and why is it important in cloud networking?	L2	1	10 15
2	a) (i) What is the difference between public and private IP addresses in a cloud environment? (ii) Define the role of subnets in a VPC. b) What are the different types of cloud load balancers, and how do they work?	L2	1	10 15
	Section -2			
3	a) How does a VPN establish a secure connection between an on-premises network and the cloud? b) You are tasked with migrating an application to the cloud. How would you choose the appropriate cloud service model (IaaS, PaaS, SaaS) for this application based on its requirements for compute, storage, and scalability?	L3	2	10 15

4	a) Given a scenario where you need to enable private communication between an on-premises data center and a cloud environment, how would you configure VPC endpoints and VPNs to meet security and performance requirements?	L3	2	10
	b) For an e-commerce application hosted on AWS, how would you implement Auto-scaling and Elastic Load Balancing to manage traffic spikes during the holiday season while controlling costs?			15

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Signature of the Course Coordinator HOD/PC IQAC Chairman

9. CIE Practical Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Cloud Networking**			**Test**	**II/IV**
Course Code	**25CS52IB**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions (One Question to be picked from the lot by the students)	**CO**	**Marks**
1. A Client whose on-premises computing infrastructure is nearly becoming obsolete. So, wants to hire AWS infrastructure for her computing and office needs. On premises it has two subnets with 5 computers each. One Computer in a LAN subnet is connected to outside Internet with static website. How do you meet these requirements of the client? 2. Launch and configure EC2 instances on AWS/Virtual Machines on Azure/ Compute Engine instances on GCP with different OS images: a) Launch an instance with a free-tier Linux OS image b) Launch an instance with a Windows OS image c) Launch an instance with a custom Linux OS image d) Configure the instances with the required security groups and networking settings e) Verify that the instances are accessible and function correctly 3. Set up and configure Security Groups in a VPC/VNet to control inbound and outbound traffic for EC2 instances/VMs. Test the rules by allowing SSH/RDP access from specific IP ranges and blocking all other traffic. 4. Create and configure Network Access Control Lists (NACLs) in an AWS VPC to enforce subnet-level traffic rules. Test the NACLs by allowing HTTP/HTTPS traffic from specific subnets and denying traffic from others. 5. Configure Network Security Groups (NSGs) in Azure to secure a VNet. Implement rules to allow traffic only from trusted IP addresses and block all other traffic. Test the setup by attempting to access resources from unauthorized IPs.	1,2,3,4,5	50

Scheme of Assessment/Valuation 6. Task Identification 7. Practical Implementation 8. Outcome/Result Assessment 9. Viva-Voce 10. Activity Portfolio (record) Assessment		 10 15 10 10 05
	Total Marks	50

Signature of the Course Coordinator **HOD/PC** **IQAC Chairman**

10. Suggested Activities:

The listed activities are illustrative and not exhaustive. The course coordinator will design a schedule of lab tasks aligned with syllabus content, ensuring hands-on practice with key concepts. Students must maintain a portfolio documenting completed lab work for evaluation.

Sl.No.	List of Activities (To be prepared by the Course coordinator)
1	Migrate the client's on-premises infrastructure (2 subnets, 10 computers, 1 static website) to AWS while maintaining network architecture, security, and website availability.
2	Launch and configure compute instances in AWS EC2 /Azure VMs/ GCP Compute Engine with different OS images, ensuring proper security and network access.
3	Configure Security Groups (AWS/Azure/GCP) to restrict SSH/RDP access to specific IPs while blocking all other traffic, then validate the rules.
4	Create and test Network ACLs (NACLs) in an AWS VPC to control subnet-level traffic by allowing HTTP/HTTPS from specific subnets while blocking others.
5	Configure AWS/ Azure NSGs to restrict access to trusted IPs only and block all other traffic. Test by simulating unauthorized access attempts.
6	Design and deploy a fault-tolerant web application that automatically scales using AWS Elastic Load Balancing (ELB) and Auto Scaling Groups (ASG).
7	Deploy and configure compute instances in AWS EC2 /Azure VMs while ensuring secure cross-cloud connectivity and compatibility.
8	Deploy a mission-critical application in Azure VNet with tiered subnets and enforce security using Network Security Groups (NSGs) to restrict access to trusted IPs only.
9	Design and deploy a three-tier web application in AWS VPC using Security Groups (SGs) and Network ACLs (NACLs) to enforce strict traffic control between subnets.
10	Establish a secure hybrid cloud connection between on-premises infrastructure and AWS/Azure, then automate network configuration for consistency.
11	Automate VPC creation with Terraform/CloudFormation. Introduce a syntax error and debug the deployment.
12	Deploy a containerized app on ECS/AKS with: Load-balanced services and Restricted ingress/egress rules
13	Simulate a container networking failure (e.g., DNS resolution issues) and resolve using tcpdump/ CloudWatch Logs
14	Set up CloudWatch/Azure Monitor alerts for high latency. Trigger and resolve false positives.
15	Optimize a high-traffic VPC by adjusting MTU sizes or enabling TCP throughput optimizations.
16	Encrypt traffic between instances using VPNs. Test performance impact with/without encryption.

17	Lock down a web server: Restrict SSH/RDP access via Security Groups and Block a specific IP range using NACLs
18	Peer two VPCs (AWS/Azure) and test cross-VPC communication. Break connectivity and diagnose using VPC Flow Logs.
19	Simulate a misconfigured subnet (e.g., no internet access) and troubleshoot using route tables.
20	Mini-Project Challenge: Design a fault-tolerant multi-cloud network with: Cross-region VPC peering, Geo-distributed load balancing and Automated failover
21	Mini-Project Challenge: Build a secure hybrid architecture with: On-premises-to-cloud encryption, Zero-trust security groups, Compliance monitoring (e.g., PCI-DSS rules)

Signature of the Course Coordinator HOD/PC IQAC Chairman

11. SEE Practical Exam model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Cloud Networking**				
Course Code	**25CS52IB**	**Duration**	**180 min**	**Marks**	**50**
Questions (One Question to be picked from the lot by the students)				**CO**	**Marks**
1. Suppose that an organization is launching a new web application expected to experience variable traffic loads. To ensure high availability, scalability, and fault tolerance, you are tasked with designing and deploying a solution using AWS Elastic Load Balancing (ELB) and Auto Scaling services. The application must automatically scale based on traffic demand and remain accessible even during peak loads. 2. Imagine that your organization is undergoing a multi-cloud migration to leverage the strengths of AWS and Azure for different workloads. As part of this initiative, you are tasked with launching and configuring instances across these cloud platforms to ensure compatibility, flexibility, and secure access. 3. Suppose that youur organization is deploying a mission-critical application in an Azure Virtual Network (VNet). The application consists of multiple tiers, including web servers and database servers, each hosted in separate subnets. To ensure the highest level of security, you are tasked with configuring Network Security Groups (NSGs) to restrict access to trusted IP addresses only and prevent unauthorized access. 4. Your company is hosting a multi-tier web application in an AWS Virtual Private Cloud (VPC). The application consists of three tiers: a web tier, an application tier, and a database tier, each hosted in separate subnets. To ensure the security and integrity of the application, you are tasked with configuring Security Groups and Network Access Control Lists (NACLs) to control traffic flow between subnets and instances. 5. Suppose that an organization is transitioning to a hybrid cloud model to integrate on-premises infrastructure with cloud resources. You are tasked with setting up a secure and automated hybrid cloud connection using AWS or Azure. Additionally, you need to automate the network configuration to ensure consistency and efficiency.				1,2,3,4,5	50

Scheme of Assessment/Valuation 1. Task Identification 2. Practical Implementation 3. Outcome/Result Assessment 4. Viva-Voce 5. Activity Portfolio (record) Assessment	 10 15 10 10 05
Total Marks	50

1) Signature of the Examiner-1 2) Signature of the Examiner-2

12. Equipment/Software list with Specification for a batch of 30 Students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 4GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane whiteboard / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	Advanced Frontend Development with React	**Type of Course**	Integrated
Course Code	25CS52IC	**Contact Hours**	7 hrs per week 91 hrs/sem
Teaching Scheme	L: T:P	**Credits**	5
CIE Marks	**50**	**SEE Marks**	**50 (Practice)**

1. Rationale:

This course is designed to equip students with modern front-end development skills using React.js, focusing on state management, API integration, testing, performance optimization, and deployment. By blending theoretical concepts with practical implementation, the course ensures that students develop industry-relevant skills essential for building scalable and high-performance applications.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Design and develop responsive, visually appealing web interfaces using advanced CSS techniques
CO-02	Manage global state efficiently with Context API and Redux.
CO-03	Integrate RESTful APIs into React application handling asynchronous operations.
CO-04	Write and execute unit and component tests for React application
CO-05	Design CI/CD pipeline and deploy the React application to production.

3. Course Content

Week	**CO**	**PO**	**Theory**	**Practice**
1	1	1	**Advanced CSS** ▪ CSS Grid and Flexbox Deep Dive. ▪ CSS Variables, Custom Properties, and Advanced Selectors.	▪ Create a grid-based portfolio layout. ▪ Create a Dark Mode Toggle.
2	1	3,4	**Advanced React Concepts** ▪ Recap of React fundamentals (components, props, state, hooks). ▪ Advanced React Hooks: useReducer, useCallback, useMemo, useRef.	▪ Build a task manager / weather app using useReducer and useContext.

3	2	3,4	**State Management with Context API** ▪ Managing global state with Context API. ▪ Pros and cons of Context API vs. Redux.	▪ Build a theme switcher and user authentication system using Context API.
4	2	4	**State Management with Redux** ▪ Introduction to Redux: Store, Actions, and Reducers. ▪ Connecting Redux with React (react-redux library).	▪ Build a shopping cart with Redux (add/remove items, calculate total). ▪ Convert the **task manager/weather app** to use Redux
5	3	3,4	**Async Operations with Redux Thunk** ▪ Handling asynchronous actions with Redux Thunk. ▪ Integrating APIs with Redux.	▪ Build a TaskManager / weather App app that fetches data from an API (e.g., OpenWeatherMap).
6	3	3,4	**API Integration in React** ▪ Fetching data from APIs using fetch and axios. ▪ Handling loading, error, and success states.	▪ Build a movie search app using a public API (e.g., OMDB API). ▪ Build a TaskManager / weather App using a public API
7	1	3,4	**Advanced Routing in React** ▪ Lazy loading and code splitting with React Router. ▪ Protected routes and role-based access control.	▪ Build a dashboard with protected routes for admin and user roles.
8	1	3,4	**Performance Optimization** ▪ Code splitting and lazy loading. ▪ Memoization with React.memo and useMemo. ▪ Optimizing re-renders and bundle size.	▪ Optimize a React app for performance (e.g., reduce re-renders, lazy load components).
9	3,4	3,4	**Introduction to Next.js** ▪ Server-Side Rendering (SSR) and Static Site Generation (SSG) with Next.js.	▪ Build and deploy a blog using Next.js.

			▪ Building full-stack React apps with Next.js.	
10	4	4	**Testing React Applications** ▪ Introduction to testing in React (Jest and React Testing Library). Writing unit tests for components and hooks.	▪ Write tests for the task manager app built in Week 1.
11	4	4	**Debugging and Error Handling** ▪ Debugging React applications using Chrome DevTools. ▪ Error boundaries and handling runtime errors.	Debug a sample React app with intentional errors
12	5	4	**Deploying React application.** ▪ Deploying React apps (Netlify, Vercel, GitHub Pages). ▪ Setting up CI/CD pipelines for React apps.	▪ Deploy the weather app or movie search app to production.
13	1,2,3,4,5	3,4	**Mini - Project** **Full-Stack E-Commerce App** - User authentication (login/signup) - Product listing and search - Shopping cart and checkout - Admin dashboard for managing products	- **Build and Deploy** a full-stack e-commerce app

4. References:

Sl No	Details
1	Learning React" (5th Edition) – Alex Banks and Eve Porcello
2	React Up and Running" – Stoyan Stefanov
3	Full-Stack React, TypeScript, and Node" – David Choi
4	"The Road to React" – Robin Wieruch
5	**"Redux in Action" – Mark Erikson and Daniel Bugl**
6	"Testing React Applications with Jest and React Testing Library" – Arpad Toth
7	react.dev
8	redux.js.org
9	**Frontend Masters – Advanced React Courses** – frontendmasters.com
10	**Pluralsight – React and Redux Path** – pluralsight.com
11	**Udemy – Modern React with Redux (Stephen Grider)** – udemy.com

12	**CSS Secrets: Better Solutions to Everyday Web Design Problems"** by Lea Verou
13	**"React Explained"** by Zac Gordon
14	**"Fullstack React: The Complete Guide to ReactJS and Friends"** by Accomazzo, Murray, and Lerner Scrimba React Tutorial
16	**"API Design Patterns"** by JJ Geewax
17	**"React Testing Library Tutorial"** by Kent C. Dodds
18	**"Web Performance in Action"** by Jeremy Wagner
19	**"DevOps Handbook"** by Gene Kim, Jez Humble, Patrick Debois, and John Willis
20	**"Testing JavaScript Applications"** by Lucas da Costa

5 CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (20)	1-13		50	
				Total	**51 arks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6 SEE - Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7.CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -**	
Course Name	Advanced Frontend Development with React			**Test**	**I/III**
Course Code	25CS52IC	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	**Questions**	**Cognitive Level**	**Course Outcome**	**Marks**
	Section - 1			
1	a. Whistle blowing-definition-Aspects-Intellectual Property Rights (IPR)–Meaning-Need for protection-Briefly description of concept of patents, Copy right, Trademark b. You are designing a dashboard layout with a header, sidebar, main content area, and footer. Explain how you would use CSS Grid to create this layout. Provide the CSS code for the same. (10) c. Explain how useReducer works in managing complex state logic. Provide an example of a to-do list application using useReducer instead of useState. (10)	L2,l3	1	25
2	a. Describe the grid-template-areas property in CSS Grid. How can it simplify complex layouts? Provide a code example. b. A company wants to create a dark mode feature for their website. Currently, they are using traditional CSS styling without variables. How would you implement dark mode efficiently using CSS Variables? Provide a step-by-step approach along with code snippets. (10) c. How do useCallback and useMemo prevent unnecessary re-renders? Explain with a scenario where these hooks significantly improve performance in a React component tree. (10)	L2,L3	1	

	Section – 2			
3	a. Describe the process of creating and using a global state with Context API and useReducer. Implement a simple user authentication state management system using Context API.(10) b. Explain how Context API avoids prop drilling. Provide an example where Context API simplifies component communication. (05) c. A React-based multi-lingual application allows users to switch between English and Spanish. The selected language should persist and update text dynamically across the app. (10) i. How would you use Context API to store and update the selected language in the app? ii. Write a LanguageContext that holds the current language and provides a function to switch between languages. iii. Show how a component (e.g., Header) can consume the LanguageContext to display translated text dynamically.	L2, L3	2	
4	a. How does the Context API handle re-renders in React? Explain how to optimize performance when using the Context API for global state management. b. What are the advantages and limitations of using Context API for state management? Compare it with useState and useReducer. c. You are developing a React application where users can toggle between light and dark themes. The selected theme should persist across multiple components, such as Navbar, Dashboard, and Settings. i. How would you implement a ThemeContext using Context API to store the theme state globally? ii. Write a React component that provides the theme context and a function to toggle the theme.	L2, L3	2	

	iii. Show how a child component (e.g., Navbar) would consume the theme context and apply styles accordingly.			

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8 CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	
Course Name	Advanced Frontend Development with React			**Test**	**II/IV**
Course Code	25CS52IC	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
Design and develop a **responsive, visually appealing task management application** using **React**, showcasing **global state management** and **API integration**. Tasks Responsive Web Interface State Management Implement **Add**, **Edit**, **Delete**, and **Complete/Incomplete** functionality for tasks. Persist user preferences (theme, task status) across components. API Integration with Asynchronous Operations Handle **asynchronous calls** Implement **search** or **filter** functionality based on task status (completed/pending).					**50**
Scheme of assessment a. Responsive and visually appealing UI - 15 b. Effective state management (Context API/Redux) - 15 c. API Integration and async handling - 15 **d.** Code quality, documentation and innovation / viva - 05					
				Total Marks	**50**

Signature of the Course Coordinator Signature of the HOD

9. **SEE- Model Practice Question Paper**

Program	**Computer Science and Engineering**			**Semester**	
Course Name	Advanced Frontend Development with React	**Course Code** 25CS52IC		**Duration**	**180 min**

Questions	CO	Marks
Design and develop a responsive, feature-rich React application that showcases a product catalog for an online store. The app should handle global state management, integrate RESTful APIs, implement unit/component testing, and be deployed using a CI/CD pipeline. Tasks: a. Build a responsive UI layout for: ▪ Home Page with a list/grid of products. ▪ Product Details Page with product information. ▪ Cart Page to view added products. b. State management c. API integration and asynchronous operations ▪ Fetch product data from a public REST API (e.g., https://fakestoreapi.com/products) or a mock JSON server. ▪ Load product data asynchronously. ▪ Handle loading and error states gracefully. ▪ Implement search/filter functionality based on product categories or price. d. Unit and Component Testing Write and execute unit tests and component tests e. Create a CI/CD pipeline for your application:	**1,2,3,4,5**	**50**
Scheme of assessment a. Responsive UI and CSS techniques – 10 b. Context API/Redux state management – 10 c. API integration and async handling -10 d. Unit and component testing – 10 **e.** CI/CD pipeline and deployment – 10		
	Total Marks	**50**

Signature of the Examiner **2) Signature of the Examiner**

10 Equipment/software list with Specification for a batch of 30students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	32
02	Internet Connection	100 Mbps speed or higher Subscription	1

03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**Cyber Security**	**Type of Course**	Integrated
Course Code	**25CS52ID**	**Contact Hours**	7 hours/week
Teaching Scheme	L: T:P - 3:0:4	**Credits**	5
CIE Marks	**50**	**SEE Marks**	**50 (Practice)**

1. Rationale:

In the era of connected computing devices, securing the personal data, application, system, network and organization becomes the challenging task. The Cyber Security specialization prepare students to take up job or to become entrepreneur in the challenging area of Cyber security

2. **Course Outcomes:** At the end of the Course, the student will be able to:

CO-01	Apply Fundamental Cybersecurity Concepts
CO-02	Analyze and Implement Security Measures
CO-03	Identify and Mitigate Cyber Threats
CO-04	Apply Cryptography and Secure Communication
CO-05	Design, optimize, test, operate and maintain a secure network/system/cloud and data resources for given requirements

3. Course Content

Week	**CO**	**PO**	**Lecture(4HRS) (Knowledge Criteria)**	**Practice(4HRS) (Performance Criteria)**
1	1	1, 5, 7	Introduction and Basic concepts of cyber security What is Cyber security, Security principles CIA - AAA Vulnerability, Threat, Risk, attack and Impact People, Process and Technology McCumbers Cube Cyber Security - Brief history and types - Infrastructure, network, cloud, IOT, application. - Purpose and Importance - Challenges	Setting up a cybersecurity lab (VMware, VirtualBox, Kali Linux, Windows) **•Basic Linux and Windows command-line security commands** • User and Permission management • Network Security and Firewall • System and Process Monitoring • File Integrity and Malware Detection

			Cybersecurity domains (network security, application security, cryptography, etc.) Protecting your personal data - Online identity , - Where is your data ? - Protecting your organization data - Traditional data - Cloud; IoT; Big data - Types of data - Sensitive and non sensitive data - Personal data, PII data - Data classification - Ex: Govt. of India classification of data - Unclassified - Restricted - Confidential - Secret - Top secret	
2	1, 3	1, 2, 7	**Cyber Attack** • **Types of Malwares** Spyware, Malware, Backdoor, Ransomware, Scareware, Rootkit, Virus, Trojan horse, Worms etc Symptoms of attack **Methods of Infiltration:** Social Engineering, Pretexting, Tailgating, Something for something (quid pro quo), Denial-of-Service and DDoS, Botnet, On the Path attack SEO Poisoning, Wi-Fi Password Cracking, **Password Attacks:** Password spraying, Dictionary attack, Brute force, Password Cracking Times, Traffic interception, **Advanced Persistent Threats:** Security Vulnerability and Exploits **Hardware Vulnerabilities** Meltdown and Spectre **Software Vulnerabilities** Categorizing Software Vulnerabilities, Software updates Defense in depth	▪ Simulating Malware Behavior (Safe Environment) ▪ Use a Virtual Machine (Kali Linux, Windows Sandbox) to analyze malware. ▪ Malware Analysis with VirusTotal. ▪ Use Procmon and Wireshark to monitor system activity when executing a benign script simulating malware behavior ▪ Phishing and Social Engineering Simulation ▪ Set up a GoPhish server to create a phishing campaign. ▪ Analyze email headers to detect phishing emails. ▪ Use OSINT tools like Maltego to collect information on a target
3	1,	1, 2,	**Defence in depth –** What is defence in depth?	▪ Check terms of service of the popular application you use on

	2, 3	3, 7	Layers, Needs for Defence in depth, Examples : Host encryption, Anti-virus, Firewall, E-Mail gateway Password management Honeypot Multi Factor Authentication Data Maintenance Using free tools Back Up Your Data How Do You Delete Your Data Permanently? Who owns your data? Terms of service Understand the term; what are you agreeing to The data use policy Privacy settings	your phone and check their data sharing policy, access to device etc. ▪ Safeguarding Your Online Privacy Two Factor Authentication ▪ Open Authorization ▪ Social Sharing ▪ Check if your password is compromised Note :Use Have I been pwned ▪ Configuring Firewalls and Intrusion Detection Systems ▪ Set up iptables or Windows Defender Firewall rules to block specific traffic. ▪ Install and configure Snort as an IDS to detect unauthorized access.
4	4	2, 3	History of cryptography (overview: Caesar cipher, enigma cipher) Introduction (high level overview only) Enc (sym - stream + block ciphers, asym) Hashing Digital signature, MAC - PRNG	Numerical exercises ▪ Hands-on with the Caesar Cipher (Classical Encryption) - Encrypt and decrypt a message using python ▪ Implement PRNG (Pseudo-Random Number Generation) and Secure Key Generation ▪ Inspect digital certificates using a web browser and visiting popular websites
5	4	2, 3	Algebra: groups, rings, fields - definitions + examples AES (SPN structure, rounds, modes of operation - high level overview with diagram) MAC + SHA2/3 (high level + security requirements))	▪ Groups, Rings, and Fields Example problems ▪ Implementation of AES ▪ Implementing MAC ▪ Compute SHA-256 and SHA-3 hash
6	4	2, 3	RSA (with numerical examples) Digital signature (RSA) Number theory - primes, modular arithmetic, gcd, Euler totient function - definitions + examples	▪ Numerical Example problems ▪ Cryptanalysis (brute force over keys, birthday attacks on hash functions, hardness of factoring integers, discrete log problem, sidechannel attacks – high level overview) ▪ Implement RSA ▪ Design a toy crypto algorithm like key generation + encryption + decryption / digital signature /

				hash function
7	2, 3, 5	2, 3, 4, 7	Network Security : Concepts- Firewall, IDS, IPS, VPN Protocols : IPSec, SSL, TLS (versions and vulnerabilities)	▪ Firewall Configuration and Rules Setup using iptables (Linux), Windows Defender Firewall, pfSense (Optional) ▪ Intrusion Detection System (IDS) with Snort (Snort, Wireshark) ▪ Intrusion Prevention System (IPS) with Snort (Snort, iptables) ▪ VPN Setup with OpenVPN using OpenVPN, WireGuard. ▪ Secure Communication with IPSec using StrongSwan (Linux), Wireshark (for packet analysis). ▪ SSL/TLS Security Testing using OpenSSL, SSL Labs (online), Wireshark.
8	2, 3, 5	2, 3, 4, 7	Web Security : Concepts-HTTP, HTML, Frames, browser design HTTP Methods, HTTP Requests and Response, Session management and Cookies Attacks and vulnerabilities: Injection attacks : SQL, HTTP header, OS command	▪ Exploring HTTP Methods, Requests, and Responses analyze request/response headers (Burp Suite, Postman, Wireshark, cURL) ▪ Analyze how web applications store and manage user sessions (Web Browser, Developer Tools, Burp Suite) ▪ Simulate SQL Injection (SQLi) ▪ HTTP Header Injection and Security Headers (Modify HTTP headers to exploit or secure web applications) ▪ Tools required : Burp Suite, curl, Developer Tools. ▪ OS Command Injection Attack Simulation : Execute arbitrary system commands via a vulnerable web application like DVWA. ▪ Tools Required: DVWA, WebGoat, Burp Suite
9	2, 3, 5	2, 3, 4, 7	Windows Security Windows Security Infrastructure Windows Family of Products Windows Workgroups and Accounts Windows Active Directory and Group Policy Windows as a Service	▪ Process observation and analysis with Process Hacker. ▪ Explore Windows security features and security policies (Group Policy Editor, Event Viewer)

			End of Support, Servicing Channels, Windows Update, Windows Server Update Services , Windows Autopilot, Windows Virtual Desktop, Third-Party Patch Management	▪ Configure Windows Workgroup, create users, and manage permissions ▪ Set up Active Directory and apply Group Policy rules ▪ Configure Windows Updates and manage patches using WSUS ▪ Deploy Windows devices using Autopilot and set up a Virtual Desktop
10	2, 3, 5	2, 3, 4, 7	Windows Access Controls NTFS Permissions Shared Folder Permissions Registry Key Permissions Active Directory Permissions Privileges BitLocker Drive Encryption Secure Boot Enforcing Security Policy Applying Security Templates Employing the Security Configuration and Analysis Snap-in Understanding Local Group Policy Objects Understanding Domain Group Policy Objects Administrative Users Privileged Account Management Reduction of Administrative Privileges AppLocker User Account Control Windows Firewall IPsec Authentication and Encryption Remote Desktop Services Recommended GPO Settings. Practice : Auditing and enforcement of system baseline configurations with security templates PowerShell scripting and automation techniques	▪ Configure NTFS permissions to restrict or allow access. ▪ Share a folder and configure access controls. ▪ Registry Key Permissions : Remove unwanted users. ▪ Enable BitLocker for a drive and BitLocker status. ▪ Verify and configure Secure Boot. ▪ Apply security policies through GPO. ▪ Demonstrate AppLocker – Application Control. ▪ Demonstrate Windows Firewall and IPsec Configuration. ▪ Demonstrate Remote Desktop Services (RDS) Security
11	2, 3, 5	2, 3, 4, 7	Linux Security Linux Fundamentals Operating System Comparison Linux Vulnerabilities Linux Operating System Shell	▪ Log File Monitoring and Analysis : Analyze system logs for security threats. ▪ Linux Security Permissions and File System Security : Implement file access controls.

			Kernel Filesystem Linux Unified Key Setup Linux Security Permissions Linux User Accounts Pluggable Authentication Modules Built-in Command-Line Capability Service Hardening Package Management Linux Security Enhancements and Infrastructure Operating System Enhancements ○ SE Linux ○ App Armor Linux Hardening ○ Address Space Layout Randomization ○ Kernel Module Security ○ SSH Hardening ○ Open SCAP ○ CIS Hardening Guides and Utilities Log files	▪ Linux User and Group Management : Manage users and privileges. ▪ Pluggable Authentication Modules (PAM) Security: Strengthen authentication using PAM. ▪ SELinux and AppArmor Security Policies : Enforce mandatory access controls (getenforce, setenforce, aa-status, aa-complain) ▪ Demonstrate Secure SSH access using /etc/ssh/sshd_config ▪ Linux Firewall and IPtables : Configure firewall rules. ▪ Perform security auditing using OpenSCAP
12	2, 3, 5	2, 3, 4, 7	Intro to VAPT Developing a Hacker Mindset • Ethics of Penetration Testing • Goal of Penetration Testing • Thinking like a Hacker • ATTandCK Framework Overview • Introduction to the framework • Deep dive into the key topics ○ Reconnaissance ○ Initial Access ○ Privilege Escalation ○ Lateral Movement ○ Exfiltration Web Application Penetration Testing • Finding common web vulnerabilities (OWASP top 10) • Burp Suite Essentials	▪ Setup Burp Suite on local machine and observe traffic of 1 website and Identify web vulnerabilities using Burp Suite. ▪ Setting Up a Virtual Pentesting Lab : Create a safe environment for penetration testing (Tools Required: VMware/VirtualBox, Kali Linux, Metasploitable, DVWA) ▪ Reconnaissance – Gathering Information About a Target: • Scanning the internet (example: Shodan) • Google dorking • Subdomain enumeration and asset monitoring ▪ Initial Access – Scanning and Enumeration: Identify open ports, services, and vulnerabilities (Nmap, Nikto, Enum4Linux, Gobuster). ▪ Privilege Escalation – Gaining Higher-Level Access: Exploit misconfigurations and weak permissions to escalate privileges (

				Linux/Windows VM, sudo, ps, whoami, systeminfo) ▪ Simulate moving between systems within a compromised network using Netcat, SSH, CrackMapExec
13	1, 2, 3, 4, 5	7	**Capstone Project** Review of cybersecurity concepts	• Capstone project: Conduct a full security assessment of a mock organization • Present findings and mitigation strategies

4. References:

Sl no	Details
1	William Stallings - *Cryptography and Network Security: Principles and Practice*
2	Charles P. Pfleeger, Shari Lawrence Pfleeger - *Security in Computing*
3	W. Richard Stevens - *TCP/IP Illustrated, Volume 1: The Protocols*
4	Michael T. Goodrich, Roberto Tamassia - *Introduction to Computer Security*
5	Ross J. Anderson - *Security Engineering: A Guide to Building Dependable Distributed Systems*
6	Bruce Schneier - *Applied Cryptography: Protocols, Algorithms, and Source Code in C*
7	Dieter Gollmann - *Computer Security*
8	Kevin Mitnick - *The Art of Deception*
9	Mark Stamp - *Information Security: Principles and Practice*
10	Patrick Engebretson - *The Basics of Hacking and Penetration Testing*
11	NIST Cybersecurity Framework – https://www.nist.gov/cyberframework
12	ISO 27001 Standard – https://www.iso.org/isoiec-27001-information-security.html
13	OWASP Top 10 Security Risks – https://owasp.org/www-project-top-ten/
14	MITRE ATTandCK Framework – https://attack.mitre.org/
16	CISCO CyberOps Associate – https://www.cisco.com/site/us/en/learning/certifications/cyberops-associate/
17	CompTIA Security+ Certification Guide – https://www.comptia.org/certifications/security
18	CEH (Certified Ethical Hacker) Study Guide – https://www.eccouncil.org/programs/certified-ethical-hacker-ceh/
19	Kali Linux Documentation – https://www.kali.org/docs/
20	Metasploit Unleashed by Offensive Security – https://www.offensive-security.com/metasploit-unleashed/

5. CIE Assessment Methodologies

Sl.No	**CIE Assessment**	**Test Week**	**Duration (minutes)**	**Max marks**	
1.	CIE-1 Theory Test (Week-1 to week-6)	7	90	50	

2.	CIE-2 Practice Test(Week-1 to Week-6)	7	180	50	Average of all CIE=50 Marks
3	CIE-3 Theory Test (Wee7-Week 12/13)	13	90	50	
4.	CIE-4 Practice Test (Week-7 to Week-11)	12	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(20)	13	-	50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. **CIE Theory Test model question paper**

Program	**Computer Science and Engineering**			**Semester -**	
Course Name	**Cyber Security**			**Test**	**I/III**
Course Code	**25CS52ID**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	**Questions**		**Cognitive Level**	**Course Outcome**	**Marks**

Section - 1				
1	a) Define Cybersecurity and explain its importance in today's digital world. (5 Marks) b) Explain the CIA Triad with relevant real-world examples. (5 Marks) c) Differentiate between Threat, Vulnerability, Risk, and Attack with appropriate examples. (5 Marks) d) Discuss the role of People, Process, and Technology in cybersecurity and how they contribute to an organization's security framework. (10 Marks)	L3		25
2	a) What are different types of malware? Explain how ransomware attacks work with an example. (10 Marks) b) Compare and contrast different types of cyber-attacks, such as Denial-of-Service (DoS), Phishing, and Advanced Persistent Threats (APT). (10 Marks) c) Discuss different password attack techniques and suggest best practices for mitigating such attacks. (5 Marks)	L4		
Section - 2				
3	a) Explain the working of symmetric and asymmetric encryption with examples of AES and RSA. (10 Marks)*)* b) Describe the role of digital signatures in ensuring data integrity and authentication. (5 Marks) c) Explain how VPNs, SSL/TLS, and IPSec contribute to secure communication over networks. (10 Marks)	L3		25
4	a) Design a secure network architecture for an organization by implementing Defense-in-Depth strategies, including firewalls, IDS/IPS, and multi-factor authentication. (10 Marks)*)* b) How does Windows security infrastructure, including Group Policy, BitLocker, and User Account Control (UAC), enhance system security? (10 Marks) c) Explain the importance of security monitoring tools such as SIEM, IDS/IPS, and honeypots in real-world cybersecurity defense. (5 Marks)	L4		
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Cyber Security**			**Test**	**II/IV**
Course Code	**25CS52ID**	**Duration**	**180 min**	**Marks**	**50**

Name of the Course Coordinator:

Questions [One question to be picked from the lot by the students]	CO	Marks
You are a cybersecurity analyst tasked with securing a simulated enterprise network. Your responsibilities include setting up a secure environment, detecting potential threats, analyzing suspicious activities, and implementing cryptographic measures to protect sensitive data. Tasks: 1. **Setting Up a Secure Environment (10 Marks)** o Configure a **virtual lab** using VMware/VirtualBox with Kali Linux and Windows. (5 Marks) o Set up a basic **firewall rule (iptables or Windows Defender)** to restrict unauthorized access. (5 Marks) 2. **Threat Detection and Analysis (15 Marks)** o Use **Wireshark** to monitor network traffic and detect any unusual activity. (5 Marks) o Deploy and configure **Snort IDS** to identify potential cyber threats. (5 Marks) o Run a simulated phishing email analysis using **email headers** to determine legitimacy. (5 Marks) 3. **Cryptography Implementation (15 Marks)** o Implement **AES encryption and decryption** on a given plaintext using Python. (5 Marks) o Compute and verify **SHA-256 hash** for file integrity verification. (5 Marks) o Implement an **RSA encryption-decryption system** with key generation. (5 Marks) 4. **Security Analysis and Report (10 Marks)** o Document the **steps taken, findings, and security improvements** made during the practical. o Provide screenshots/logs supporting your analysis and explain how threats were detected and mitigated.		50

Scheme of Evaluation

Task	Marks
Secure Lab Setup (Firewall and Virtual Machine)	10
Threat Detection (Wireshark, Snort IDS, Phishing Email Analysis)	15
Cryptography Implementation (AES, SHA-256, RSA)	15
Security Report with Findings and Screenshots	10

Total Marks	50

Signature of the Course Coordinator **Signature of the HOD**

9. **SEE- Model Practice Question Paper**

<table>
<tr><td>Program</td><td colspan="2">Computer Science and Engineering</td><td>Semester</td><td></td></tr>
<tr><td>CourseName</td><td>Cyber Security</td><td>CourseCode : 25CS52ID</td><td>Duration</td><td>180 min</td></tr>
<tr><td colspan="3">Questions</td><td>CO</td><td>Marks</td></tr>
<tr><td colspan="3">You are a cybersecurity analyst tasked with securing a simulated enterprise network. Your responsibilities include setting up a secure environment, detecting potential threats, analyzing suspicious activities, and implementing cryptographic measures to protect sensitive data.

Tasks:

5. Setting Up a Secure Environment (10 Marks)
o Configure a virtual lab using VMware/VirtualBox with Kali Linux and Windows. (5 Marks)
o Set up a basic firewall rule (iptables or Windows Defender) to restrict unauthorized access. (5 Marks)
6. Threat Detection and Analysis (15 Marks)
o Use Wireshark to monitor network traffic and detect any unusual activity. (5 Marks)
o Deploy and configure Snort IDS to identify potential cyber threats. (5 Marks)
o Run a simulated phishing email analysis using email headers to determine legitimacy. (5 Marks)
7. Cryptography Implementation (15 Marks)
o Implement AES encryption and decryption on a given plaintext using Python. (5 Marks)
o Compute and verify SHA-256 hash for file integrity verification. (5 Marks)
o Implement an RSA encryption-decryption system with key generation. (5 Marks)
8. Security Analysis and Report (10 Marks)
o Document the steps taken, findings, and security improvements made during the practical.
o Provide screenshots/logs supporting your analysis and explain how threats were detected and mitigated.</td><td></td><td>50</td></tr>
<tr><td colspan="3"><u>Scheme of Evaluation</u>
<table><tr><th>Task</th><th>Marks</th></tr><tr><td>Secure Lab Setup (Firewall and Virtual Machine)</td><td>10</td></tr><tr><td>Threat Detection (Wireshark, Snort IDS, Phishing Email Analysis)</td><td>15</td></tr><tr><td>Cryptography Implementation (AES, SHA-256, RSA)</td><td>15</td></tr><tr><td>Security Report with Findings and Screenshots</td><td>10</td></tr></table></td><td></td><td></td></tr>
</table>

TotalMarks	**50**

1) Signature of the Examiner **2) Signature of the Examiner**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science And Engineering**	**Semester**	**V**
Course Name	**Deep Learning**	**Type of Course**	**Integrated**
Course Code	**25CS53IA**	**Contact Hours**	**7 hrs per week**
Teaching Scheme	**L: T:P : 3:0:4**	**Credits**	**5**
CIE Marks	**50**	**SEE Marks**	**50(Practice)**

1. Rationale:

In today's AI-driven world, Deep Learning (DL) plays a crucial role in powering image recognition, natural language processing (NLP), autonomous systems, and intelligent decision-making across industries. This course is designed to provide beginners with a structured, hands-on approach to mastering fundamental Deep Learning techniques using Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and NLP models, with an emphasis on hyperparameter tuning, model optimization, and deployment. By the end of the course, students will train, fine-tune, and deploy deep learning models, ensuring they gain the practical skills required in AI-based applications.

2. **Course Outcomes :**At the end of the Course, the student will be able to:

CO-01	Explain deep learning fundamentals, architectures (ANN, CNN, RNN), and tools like TensorFlow, Keras, and TensorBoard.
CO-02	Preprocess structured and unstructured data using normalization, augmentation, encoding, and dataset splitting techniques.
CO-03	Build and train deep learning models with various activation functions, optimizers, loss functions, and regularization methods.
CO-04	Evaluate and tune models using TensorBoard, KerasTuner, and performance metrics like accuracy, precision, recall, and F1-score.
CO-05	Apply model explainability (XAI), deploy models, monitor performance, and present results through a mini-project.

3. Course Content

WEEK	CO	PO	Theory	Practice
1	**1**	**1, 2, 4**	**Introduction to Deep Learning:** ML Vs DL, and real world use cases **Key Concepts:** Biological Vs Artificial neuron, layers, activation functions, and their role in deep learning. **Deep Learning Architectures:** Artificial neural networks(ANN) Convolutional neural networks(CNN), Recurrent neural networks(RNN) **Tools:** TensorFlow, Keras, TensorBoard Importance of Data in Deep Learning	Set up environment (TensorFlow, Keras, TensorBoard) Load, visualize and analyze different types of data **Structured** -Titanic Dataset, Iris Dataset **Unstructured**-CIFAR-10, MNIST, IMDB Movie Reviews **Hands-on**: Building a simple neural network using Keras.

			Types of Data: Structured (e.g., tabular), Unstructured (e.g., images, text, audio),	**TensorBoard Visualization**: Visualizing the neural network architecture using TensorBoard
2	**1**	**2, 4**	**Data Preprocessing** Importance of data preprocessing, Handling missing values and Noise, Handling (Denoise, Augment, Normalize), Feature Scaling & Normalization: - Min-Max Scaling, Z-score Standardization - Normalization (scaling pixel values from 0–255 to 0–1) Augmentation(rotation, flipping, zooming) Encoding(one-hot encoding) **Splitting dataset:** Train/Validation/Test Split	Preprocess data (MNIST/CIFAR-10) for image classification. -Discard corrupted/missing images or replace with average pixel -Normalize Fashion MNIST images (divide by 255.0) -Apply one hot encoding for labels in MNIST dataset **TensorBoard Visualization:** - visualize noisy vs clean images - Log histogram of raw vs scaled data. - compare augmented vs original images
3	**2**	**3, 4 , 6**	**Artificial neural networks(ANN)** Perceptron, Feedforward neural network: Input layer, Hidden layer(s), Output layer • Weights(Bias), • Activation Functions(Sigmoid, Tanh, ReLU, Softmax) • Loss Functions: MSE, CrossEntropy • Optimizers(SGD, Adam, RMSprop) and Backpropagation	Build ANN with Keras **Dataset:** MNIST/CIFAR-10 Train ANN with different activations/losses/optimisers **TensorBoard Visualization:** Log training metrics & visual model graph Compare scalar logs for each setup
4	**2**	**2, 3, 4**	**Hyper parameter tuning** • Epochs, batch size, learning rate • Overfitting, Underfitting • Regularization: Dropout, L1, L2 • Early stopping, validation splits Hyperparameter tuning methods: KerasTuner, Grid Search, Random Search **Model evaluation** Confusion Matrix, Accuracy, Precision, Recall, F1	**Dataset:** MNIST/CIFAR-10 Train a deep neural network with different combinations of **epochs**, **batch sizes**, and **learning rates**. Add Dropout and L2 regularization. Apply **early stopping** to prevent overfitting Evaluate model performance **TensorBoard Visualization**: Monitor training and validation losses. Log all experiments and interpret the results.
5	**3**	**3,, 4**	**Introduction to CNNs** Inspiration behind CNN,	**Dataset:** MNIST/CIFAR-10 Build and train CNN Architecture

			Applications of CNN in deep learning Key component of CNN • Convolution Operation, Filters, Padding • MaxPooling, Stride, Feature Extraction	Visualize Filters, Feature Maps, and Stride Effects **TensorBoard Visualization**: Visualize input images and feature maps Scalar metrics: accuracy, loss Filters and activations using image summaries.
6	**4**	**2, 3, 4**	**Transfer Learning in CNNs** • Concept of Transfer Learning • Pretrained Networks (VGG, ResNet) • Feature Extraction vs Fine-Tuning	Fine-tune a pre-trained model (e.g., VGG16/ResNet) on CIFAR-10, **TensorBoard Visualization:** visualize training metrics and compare feature extraction vs fine-tuning.
7	**4**	**2, 3, 4**	**Recurrent Neural Networks (RNN)** • Need for sequence modeling • Vanishing/exploding gradients, Long Short-Term Memory (LSTM) units • Applications in text and time series	Build a simple RNN for time-series prediction, upgrade to LSTM, **TensorBoard Visualization:** Compare training dynamics. Log text embedding and performance.
8	**4**	**2, 3, 4**	**Generative Models** • Autoencoders and use cases • Variational Autoencoders (VAEs) • Introduction to GANs	• Build basic Autoencoder for MNIST • Compare original vs. reconstructed images • Use **TensorBoard** to visualize image reconstruction
9	**4**	**3, 4, 6**	**Explainable AI (XAI)** Need for Explainability in AI Challenges of Deep Learning Models Explainability tools: Grad-CAM (for CNNs), SHAP, LIME	**Model Explainability** Train a CNN (e.g., on CIFAR-10). Apply **Grad-CAM** to visualize important regions for classification Log visualizations using **TensorBoard**
10	**5**	**3, 4, 6**	Functional API in Keras, custom layers/loss functions, **model registration** (e.g., MLflow, model versioning)	Build a multi-input/output model with Functional API, register a model using MLflow, visualize training in TensorBoard
11	**5**	**4, 5, 6**	**Model deployment**: saving/loading models, deployment options (e.g., Flask, TensorFlow Serving), best practices	Deploy a trained model (e.g., Flask API), test endpoints, set up TensorBoard logging for deployment monitoring
12	**5**	**4, 5, 6, 7**	**Model monitoring**: performance tracking, drift detection, presenting results, future directions	Monitor a deployed model (e.g., log predictions), analyze performance /drift with TensorBoard, prepare final report

13	**2, 3, 4, 5**	**3, 4, 5, 6**	**Mini project** Project planning: problem definition, dataset selection. Best practices for model evaluation and debugging Presenting results: metrics, visualizations, and insights. Future directions in deep learning	Choose a project (e.g., image classification, text generation, time-series forecasting). Start data collection and preprocessing Finalize and train the capstone model. Prepare a short report or presentation summarizing the project

4. References:

Sl. No.	**Books Description**
1.	Neural Networks and Deep Learning – Michael Nielsen
2.	Deep Learning – Ian Goodfellow, Yoshua Bengio, Aaron Courville
3.	Deep Learning with Python – François Chollet
4.	Dive into Deep Learning (D2L) – Aston Zhang, Zachary C. Lipton, Mu Li, Alex J. Smola
5.	Pattern Recognition and Machine Learning – Christopher Bishop
6.	Speech and Language Processing – Daniel Jurafsky & James H. Martin
7.	Natural Language Processing with Python – Bird, Klein & Loper
Website	
9.	TensorFlow Documentation: https://www.tensorflow.org/
10.	Fast.ai Course: https://course.fast.ai/
11.	**DeepLearning.AI** (Andrew Ng): https://www.deeplearning.ai/
12.	OpenAI Blog: https://openai.com/research/

5. CIE Assessment Methodologies

Sl.No	**CIE Assessment**	**Test Week**	**Duration (minutes)**	**Max marks**	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(20)	1-13		50	
				Total	**50 Marks**

Note:
Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.
Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. CIE Theory Test model question paper

Program	**Computer Science And Engineering**			**Semester - V**	
Course Name	**Deep Learning**			**Test**	
Course Code	**25CS53IA**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a)Define AI, ML, and Deep Learning. Explain their differences with real-world applications.**(10 MARKS)** b)How does Deep Learning differ from traditional Machine Learning? Discuss the advantages and challenges of Deep Learning.**(10 MARKS)** c)What are the key applications of Deep Learning? Discuss any two real-world use cases.**(5 MARKS)**	L2, L3	1	**25**
2	a)Explain the significance of data in Deep Learning. Differentiate between structured and unstructured data and discuss their impact on model training.**(10 MARKS)** b)Describe the architecture of an Artificial Neural Network (ANN). Explain the perceptron model and the importance of activation functions (ReLU, Sigmoid, Tanh).**(10 MARKS)**	L2, L3	2	

	c)Explain the role of data collection in Deep Learning. What are common data sources such as web scraping, APIs, and databases?**(5 MARKS)**			
Section - 2				
3	a)Explain Forward and Backpropagation in neural networks. How do loss functions (MSE, Cross-Entropy) contribute to training an ANN?**(10 MARKS)** b)Discuss the role of optimizers like SGD, Adam, and RMSprop in Deep Learning. How do they impact model convergence?**(10 MARKS)** c)Define and compare different model evaluation metrics (Accuracy, Precision, Recall, F1-score). Why are they important?**(5 MARKS)**	L2, L3	2	**25**
4	a)What is overfitting in Deep Learning? Explain various regularization techniques used to address overfitting. **(10 MARKS)** b)Explain hyperparameter tuning in Deep Learning. How do learning rate, batch size, and number of layers affect model performance?**(10 MARKS)** c)What are the key differences between Multi-Layer Perceptron (MLP) and traditional neural networks? Discuss its significance in Deep Learning.**(5 MARKS)**	L2, L3	3	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science And Engineering**			**Semester**	**V**
Course Name	**Deep Learning**			**Test**	
Course Code	**25CS53IA**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
Develop a Deep Learning Model using Artificial Neural Networks (ANN) or Convolutional Neural Networks (CNN) for a classification task. Perform the following tasks Data preprocessing, Hyperparameter tuning, Model evaluation, and Implement Git for version control. **Note: Choose a dataset like MNIST, CIFAR-10, or a custom image dataset**				**1,2,3,4,5**	**50**
Scheme of assessment **a)Dataset Preparation & Preprocessing - 10** **b)Model Implementation (ANN/CNN) - 10**					

c)Hyperparameter Tuning -	**10**	
d)Model Evaluation & Analysis -	**10**	
e)Git & Version Control -	**10**	
	TotalMarks	**50**

Signature of the Course Coordinator **Signature of the HOD**

9. **SEE- Model Practice Question Paper**

Program	**Computer Science And Engineering**		**Semester**	**V**
Course Name	**Deep Learning**	**Course Code 25CS53IA**	**Duration**	**180 min**
Questions			**CO**	**Marks**
Develop and deploy a Recurrent neural network(RNN) model for sentiment analysis using IMDB movie reviews or Twitter sentiment dataset. Perform the following tasks: data preprocessing, Text vectorization, Model training, Hyperparameter tuning, evaluation, and Deployment using Flask.			**1,2,3,4,5**	**50**
Scheme of assessment **a)Dataset Preparation & Preprocessing - 10 marks** **b) RNN model implementation - 10 marks** **c)Hyperparameter Tuning - 10 marks** **d)Model Evaluation & Performance Analysis - 10 marks** **e)Model Deployment - 10 marks**				
			Total Marks	**50**

1) Signature of the Examiner **2) Signature of the Examiner**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**Cloud Security and Compliance**	**Type of Course**	Integrated
Course Code	**25CS53IB**	**Contact Hours**	7 hours/week
Teaching Scheme	L: T:P - 3:0:4	**Credits**	5
CIE Marks	**50**	**SEE Marks (Practical)**	**50 (Practice)**

1. Rationale:

As organizations rapidly adopt cloud technologies, securing infrastructure, ensuring compliance, and managing risks have become critical. This course equips learners with the practical knowledge and tools required to safeguard cloud environments, enforce compliance standards, and respond to security incidents effectively across single, hybrid, and multi-cloud ecosystems.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain cloud security challenges, compare traditional and cloud-based models, manage IAM, and implement threat mitigation strategies
CO-02	Apply data protection techniques, configure secure networks, and develop incident response mechanisms using cloud-native tools.
CO-03	Interpret compliance frameworks, perform security assessments, and execute penetration testing in cloud environments.
CO-04	Implement advanced security practices, secure serverless/container platforms, and integrate DevSecOps into CI/CD pipelines.
CO-05	Design and implement secure solutions for multi-cloud and hybrid environments, integrating key cloud security concepts through real-world mini projects.

3. Course Content

Week	CO	PO	Theory (3 Hrs/Week)	Practice (4 Hrs/Week)
1	**1**	**1,2,4,7**	**Introduction to Cloud Computing Security** Overview of **Cloud Computing** and Security Challenges. (Data Breach, Data Loss, Insecure APIs, etc.) Cloud Security Vs Traditional Security Models **Cloud Computing** Architecture and Security Layers Cloud Security Best Practices .	Overview of Cloud Service Models (IaaS, PaaS, SaaS) and their Security Implications. Lab: Create an account on AWS/Azure/GCP and navigate the security settings.

2	**1**	**1,2,3,4,7**	**Cloud Security Threats and Vulnerabilities** Cloud Security Threats: Data breaches, Insecure interfaces, Denial of Service (DoS), Malicious Insiders Vulnerabilities in Cloud Applications Shared Responsibility Model	Hands-on :Use tools to identify vulnerabilities in cloud environments. Lab: Identify and mitigate security threats in AWS/Azure/GCP.
3	**1**	**1,2,3,4,7**	**Identity and Access Management (IAM) in Cloud** Introduction to AWS Global Infrastructure and Security Features. IAM Fundamentals in AWS/Azure/GCP Users, Groups, Roles, and Policies Multi-Factor Authentication (MFA).	Hands-on: Configuring IAM in AWS Learner Lab /Azure SandBox.
4	**2**	**1,2,3,4,7**	**Data Security in the Cloud** Data Encryption at Rest and in Transit AWS Key Management Service (KMS) AWS Certificate Manager (ACM) Securing Storage Services: AWS: S3 Buckets, EBS Volumes Azure: Blob Storage, Managed Disks	Hands-on: Encryption methods for data in the cloud. Lab: Implement encryption in cloud storage (AWS S3 or Azure Blob Storage).
5	**2**	**1,2,3,4,7**	**Network Security in the Cloud** Virtual Private Cloud (VPC) and Subnets Security Groups and Network ACLs. Cloud Firewalls and VPNs Intrusion Detection and Prevention Systems. AWS Shield for DDoS Protection.	Hands-on: Securing a VPC/VNet Lab: Configure VPCs, security groups, and set up a secure network in AWS or Azure.
6	**2**	**1,2,3,4,5,7**	**Cloud Security Monitoring and Incident Response** Cloud Security Monitoring and Loging Tools (e.g., AWS CloudWatch, Azure Monitor, CloudTrail, AWS Config, etc.) Incident Response in Cloud: Detection, Investigation, and Mitigation. Automated Incident Response with Cloud-native tools. AWS Security Hub for Centralized Security Management.	Hands-on: Setting Up Monitoring and Alerts in AWS/Azure/GCP Lab: Simulate an incident and respond using cloud monitoring tools.
7	**3**	**1,2,3,4,5,7**	**Compliance in Cloud Computing** Introduction to Cloud Compliance Overview of Compliance Frameworks (GDPR, HIPAA, ISO 27001, etc.).	Hands-on: Generating Compliance Reports in AWS and Azure

			AWS: AWS Artifact for Compliance Reports. Azure: Azure Compliance Manager Implementing Governance in AWS and Azure.	Lab: Review a cloud provider's compliance certification (e.g., AWS, Azure). Lab: Implement basic compliance configurations in AWS or Azure.
8	**3**	**1,2,3,4,7**	**Data Sovereignty and Legal Compliance** Data Sovereignty Issues in Cloud Computing. Jurisdiction, Data Localization Laws, and International Regulations. Best Practices for Legal Compliance in Cloud Environments.	Lab: Assess data residency compliance with cloud storage (AWS S3 or Azure Storage). Research and present case studies on data sovereignty issues.
9	**3**	**1,2,3,4,7**	**Securing Cloud-Based Applications** Security of SaaS, PaaS, and IaaS applications. Secure Software Development in the Cloud (SDLC, DevSecOps). Securing Resources , APIs in Cloud-Based Applications.	Hands-on: Secure a cloud-based application using security best practices. Lab: Implement API/resource security using OAuth and /or JWT in a cloud-based app.
10	**4**	**1,2,3,4,5,7**	**Advanced Cloud Security and DevSecOps** Advanced Cloud Security Architectures and Practices. Serverless Security AWS: AWS Lambda. Azure: Azure Functions Container Security. AWS: ECS, EKS Azure: AKS (Azure Kubernetes Service) DevSecOps: Integrating Security into CI/CD Pipelines .	Hands-on: Securing Serverless and Container Workloads Lab: Automate a security configuration in the cloud using Terraform or AWS CloudFormation. Implement a simple DevSecOps pipeline to deploy a secure app.
11	**4**	**1,2,3,4,7**	**Threat Detection and Incident Response** Cloud Security Posture Management (CSPM). Threat Intelligence in the Cloud. AWS: GuardDuty, Inspector. Azure: Microsoft Defender for Cloud, Azure Sentinel . Incident Response in AWS and Azure/GCP.	Hands-on: Simulating and Responding to Security Incidents Lab: Use Cloud-native security tools like AWS GuardDuty or Azure Security Center to monitor for threats.

				Lab: Perform a cloud security audit on a cloud infrastructure.
12	**5**	**1,2, 3,4,5,7**	**Security Challenges in Multi-Cloud and Hybrid Cloud** Challenges of Multi-Cloud Security. Integrating Security in Hybrid Cloud Environments. Tools for Multi-Cloud Security (CloudBolt, Prisma Cloud).	**Mini Project Work** (Students will work in groups to design and implement a secure cloud infrastructure for a real-world scenario. The project may include: Setting up IAM, VPC, and Security Groups Implementing Data Encryption and Monitoring Generating Compliance Reports, etc.) .
13	**5**	**1,2,3,4,5,7**	**Cloud Security Trends and Emerging Technologies** AI and Machine Learning in Cloud Security. Blockchain for Cloud Security Zero Trust Security Models in Cloud.	**Mini Project Work** (Submission of report, Review and Evaluation.)

4. References:

Sl.No	Description
1	**Introduction to Cloud Computing Security. Book**: *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance* by Tim Mather, Subra Kumaraswamy, and Shahed Latif. **Reference Link**: Cloud Security Alliance.
2	**Cloud Security vs Traditional Security Models. Book**: *Cloud Security For Dummies* by Ted Coombs, Patrick P. McDermott. **Reference Link**: NIST **Cloud Computing** Security.
3	**Cloud Service Models (IaaS, PaaS, SaaS) and Security Implications. Book**: *Architecting the Cloud: Design Decisions for **Cloud Computing** Service Models (SaaS, PaaS, and IaaS)* by Michael J. Kavis. **Reference Link**: Microsoft Cloud Adoption Framework.
4	**Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance.** Link: https://www.oreilly.com/library/view/cloud-security-and/9780596802772/.
5	**Cloud Security For Dummies.** Link: https://www.amazon.com/Cloud-Security-Dummies-Computer/dp/1118956572.
6	**The DevSecOps Handbook: Secure Software Development in the Cloud.** Link: https://www.amazon.com/DevSecOps-Handbook-Software-Security-Cloud/dp/1484239811.
7	**AWS Certified Security Specialty Exam Guide.** Link: https://www.amazon.com/AWS-Certified-Security-Specialty-Guide/dp/1801070052.
8	**Cloud Security: A Comprehensive Guide to Secure Cloud Computing.** Link: https://www.amazon.com/Cloud-Security-Comprehensive-Secure-Computing/dp/0470593183

9.	https://aws.amazon.com/
10.	https://azure.microsoft.com/
11.	https://cloud.google.com/
12.	https://docs.aws.amazon.com/
13.	https://learn.microsoft.com/
14	https://aws.amazon.com/getting-started/hands-on/
15	https://learn.microsoft.com/en-us/training/azure/
16	https://www.cloudskillsboost.google/
17	https://www.awseducate.com/
18	https://openstax.org/
19	https://www.bookbub.com/
20	https://scholar.google.com/
21	https://bookboon.com/

5. Suggestive Online Courses

Course	**Components From AWS Academy**	**Components From AWS Educate, AWS Skill Builder & Other Material**
Cloud Security and Compliance (Course Code: 25CS53IB)	Introduction to **Cloud Computing** Security. Module 2 - AWS Academy Cloud Security Foundations (CO1) Identity and Access Management (IAM) in Cloud. Module 3 - AWS Academy Cloud Security Foundations (CO1) Data Security in the Cloud - Module 5 - AWS Academy Cloud Security Foundations (CO2) Network Security in the	AWS Security Best Practices: Computing (CO2) https://explore.skillbuilder.aws/learn/courses/11263/aws-security-best-practices-computing AWS Security – Encryption Fundamentals (CO2) https://explore.skillbuilder.aws/learn/courses/17895/aws-security-encryption-fundamentals AWS Security Best Practices: Monitoring and Alerting (CO2) https://explore.skillbuilder.aws/learn/courses/11264/aws-security-best-practices-monitoring-and-alerting AWS Shield Getting Started (CO2) https://explore.skillbuilder.aws/learn/courses/21855/aws-shield-getting-started Cloud Security & Compliance (CO3) https://explore.skillbuilder.aws/learn/courses/479/aws-identity-and-access-management-architecture-and-terminology Digital Sovereignty on AWS (CO3) https://explore.skillbuilder.aws/learn/courses/22162/digital-sovereignty-on-aws Security Learning Plan: Threat Detection and Incident Response (CO4) https://explore.skillbuilder.aws/learn/learning-plans/1803/plan & https://explore.skillbuilder.aws/learn/learning-plans/91/plan AWS Lambda Function Security & Defense in Depth (CO4) - https://explore.skillbuilder.aws/learn/courses/53/aws-lambda-function-security-defense-in-depth Deep Dive on Container Security (CO4) https://explore.skillbuilder.aws/learn/courses/72/deep-dive-on-container-security

	Cloud - Module 4 - AWS Academy Cloud Security Foundations (CO2) Cloud Security Monitoring and Incident Response - Module 6 & 7 - AWS Academy Cloud Security Foundations (CO2)	

6. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max Marks	
1.	CIE-1 Theory Test	4	90	50	Average of all CIE
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 Theory Test	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(20)	13	-	50	
				Total Obtained	**/50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

7. SEE - Practical Assessment Methodologies

Sl. No	SEE – Practical Assessment	Duration (minutes)	Max Marks	Min marks to pass
1.	Semester End Examination-Practical	90	50	20

8. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester -V	
Course Name	Cloud Security and Compliance			Test	I
Course Code	25CS53IB	Duration	90 Min	Marks	50

Name of the Course Coordinator:

Note: Answer any one full question from each section. Each full question carries equal marks.

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a) What are the primary security challenges faced in **Cloud Computing**, and how do they differ from traditional IT security models? Or b) Explain the concept of Multi-Factor Authentication (MFA) and its importance in securing cloud resources.	L2	1	25
2	a) What is the difference between data encryption at rest and encryption in transit, and why are both important for cloud security? Or b) What is the role of Virtual Private Cloud (VPC) in network security, and how does it help isolate cloud resources from the public internet?	L2	2	25
	Section -2			
3	a) Your company is moving to the cloud and needs to secure sensitive customer data. How would you apply cloud security best practices to prevent data breaches and unauthorized access? Or b) You are tasked with setting up IAM in AWS for a project team. How would you configure IAM users, groups, roles, and	L3	1	25

	policies to ensure that each team member has the necessary permissions while minimizing security risks?			
4	a) You are designing a cloud architecture that stores sensitive customer data. How would you implement data encryption at rest and in transit to ensure data confidentiality in compliance with industry regulations? Or b) Given a scenario where your company's cloud infrastructure needs to be connected securely to an on-premise network, how would you configure a VPN or a cloud firewall solution to ensure safe data transmission?	**L3**	**2**	**25**
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator **HOD /PC** **IQAC Chairman**

9. CIE Practical Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Cloud Security and Compliance**			**Test**	**II**
Course Code	**25CS53IB**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions **[One question to be picked from the lot by the students]**	**CO**	**Marks**
1. Create a VPC with a public subnet and a private subnet. Configure the VPC to allow inbound HTTP traffic from the internet to an EC2 instance in the public subnet, while ensuring that the EC2 instance in the private subnet is not accessible from the internet. 2. Create a secure cloud environment in AWS/Azure/GCP by configuring Identity and Access Management (IAM) settings. Ensure that users and resources have the necessary permissions to perform tasks without compromising security. 3. Implement encryption in cloud storage (AWS S3 or Azure Blob Storage) to protect sensitive data. Ensure that data is encrypted at rest and in transit. 4. Simulate a security incident in a cloud environment (AWS/Azure/GCP) and respond using cloud monitoring and incident response tools. 5. You are a cloud security engineer responsible for implementing basic compliance configurations in AWS. Your task is to implement configurations for data encryption, access controls, and logging.	1,2,3, 4 or 5 (as per Scheme of Assessment/Valuation)	50

Scheme of Assessment/ Valuation	
11. Writing Procedure	10
12. Conduction	15
13. Result	10
14. Viva-Voce	10
15. Portfolio (Record) of activities	05
Total Marks	50

Signature of the Course Coordinator HOD/PC IQAC CHAIRMAN

10. SEE- Model Practical Question Paper

Program	**Computer Science and Engineering**		**Semester**	**V**
Course Name	**Cloud Security and Compliance**	**Course Code 25CS53IB**	**Duration**	**180 min**
Questions [One question to be picked from the lot by the students]			**CO**	**Marks**
1. Suppose that an organization named "XYZ Technologies" is evaluating different cloud service models (IaaS, PaaS, SaaS) for deploying a new application. As an employee you are tasked with understanding the security implications of each model and demonstrating how to secure resources in a cloud environment. How do you achieve this? 2. Imagine that your team has deployed a cloud-based application, but there are concerns about potential vulnerabilities. You are tasked with using tools to identify and mitigate security threats in the cloud environment. How do you solve this? 3. Your team is adopting a DevSecOps approach to ensure secure application deployments. You are tasked with implementing a simple DevSecOps pipeline that integrates security checks into the CI/CD process. How do you do this? 4. You are a cloud security engineer responsible for ensuring compliance with regulatory requirements for a cloud-based application. Your task is to generate compliance reports in AWS/ Azure. How do you achieve this? 5. Your organization is deploying serverless (AWS Lambda) and containerized (Amazon ECS) workloads in the cloud. You are tasked with automating the security configuration for these workloads using infrastructure-as-code tools. How do you perform this?			1,2,3, 4 or 5 (as per Scheme of Assessment/Valuation)	**50**
Scheme of Assessment/ Valuation				
1. Writing Procedure				10

2. Conduction 3. Result 4. Viva-Voce 5. Portfolio (Record) of activities	15 10 10 05
Total Marks	50

Signature of the Examiner **2) Signature of the Examiner**

11. Equipment/Software list with Specification for a batch of 30 Students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 4GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane whiteboard / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	**DevOps: Software Development and Operations**	**Type of Course**	Integrated
Course Code	**25CS53IC**	**Contact Hours**	7 hrs per week
Teaching Scheme	L: T:P - 3:0:4	**Credits**	5
CIE Marks	**50**	**SEE Marks (Practical)**	**50 (Practice)**

1. Rationale:

This course is designed to provide comprehensive knowledge of software development and operations, equipping students with the essential skills and hands-on experience needed to efficiently build, deploy, and manage modern software applications.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain the influence of the DevOps principles and practices on the software development process
CO-02	Use version control systems (Git) and collaborate on open-source platforms.
CO-03	Design and implement a CI/CD pipeline for an application using tools like Jenkins and GitHub Actions.
CO-04	Demonstrate containerization and orchestration
CO-05	Explain the application deployment process, including key concepts and best practices.

3. Course Content

Week	CO	PO	Theory	Practice
1	**1**	**1**	**Introduction to DevOps** ▪ What is DevOps? ▪ DevOps lifecycle and principles. ▪ Core practices ▪ Benefits of DevOps in software development. ▪ Popular DevOps tools and their roles in the DevOps lifecycle.	Set up a DevOps environment on a local machine – by Installing Required Tools case studies of organizations that have successfully implemented DevOps
2	**1**	**1**	**Cloud Computing Basics** What is Cloud Computing? Cloud Service Models Cloud Deployment Models	Set Up a Cloud DevOps Environment Using AWS / Azure Eg. Creating a VM on AWS EC2 Using Cloud Service APIs

			Cloud Providers and Their Offerings Virtualization Security in DevOps (DevSecOps)	case studies of organizations that have successfully implemented DevOps and Cloud Computing.
3	**2**	**1,2,4, 5,7**	**Introduction to Version Control** ▪ What is Version Control? ▪ Types of Version Control: Local, Centralized, Distributed ▪ Introduction to Git and Github ▪ Git Workflow ▪ Git basics : init, add, commit, log, config ▪ Working with remote repositories: clone, push, pull. **Branching, and Merging** ▪ Why use branches? ▪ Creating and switching branches ▪ Forking a Repository ▪ Merging ▪ Types: Fast-forward Merge, Three-way Merge ▪ Command: branch, checkout, merge ▪ Handling merge conflicts ▪ how to **identify, resolve, and commit merge conflicts**	Exercise : 1. Install Git and set up a repository. 2. Set up your Git username and email. 3. Initialize a Git repository. Exercise : 1. Create a text file and add it to Git. 2. Modify and commit changes. 3. View commit history. 4. Modify a file and discard the changes. 5. Reset a commit. Exercise: a) Clone an existing repository from GitHub. b) Make changes, commit, and push them to GitHub. c) Pull the latest changes from the remote repo. Exercise: a) Create a new branch. b) Make changes and commit. c) Merge the branch into main. Exercise: Handling Merge Conflicts
4	**2**	**1,2,4, 5,7**	**Open Source and Collaboration** ▪ Introduction to open source and licenses (MIT, GPL, Apache). ▪ Benefits of contributing to open-source projects. ▪ Building a GitHub portfolio.	**Exercise:** Explore GitHub and find an open-source project to contribute to. **Exercise:** Identify beginner-friendly issues using labels like good first issue **Exercise:** Fork an open-source repository and make a small contribution (e.g., update README, fix a minor bug). **Exercise:** Submit a pull request (PR) and go through the review process. **Exercise:** Design a personal GitHub profile
5	**3**	**1,2,3, 4,5,7**	**CI/CD with Jenkins** ▪ CI/CD Pipeline and stages ▪ Jenkins pipeline syntax and	▪ set up a CI/CD pipeline. ▪ Connect Jenkins with GitHub and trigger builds.

			build triggers. ▪ Integrating Jenkins with GitHub	
6	**3**	**1,2,3, 4,5,7**	**CI/CD with GitHub Actions** ▪ Introduction to GitHub Actions. ▪ Workflow syntax and YAML configuration. ▪ Automating CI/CD with GitHub Actions.	▪ Create a CI pipeline with GitHub Actions. ▪ Implement a full CI/CD pipeline for a sample web app.
7	**4**	**1, 4,5,7**	**Containerization** ▪ Introduction to containerization and Docker. ▪ Docker architecture and basic commands: run, ps, stop, rm. ▪ Creating Dockerfiles and building custom images.	▪ Build a custom Docker image for a sample application.
8	**4**	**1, 4,5,7**	**Orchestration with Kubernetes** ▪ Introduction to Kubernetes and container orchestration. ▪ Kubernetes architecture: Pods, Nodes, Services. ▪ Deploying and scaling applications with Kubernetes.	▪ Set up a Kubernetes cluster (e.g., Minikube). ▪ Deploy a sample application on Kubernetes.
9	**5**	**1, 4,5,7**	**Deployment Strategies** ▪ Deployment phases and strategies: Blue-Green, Canary, Rolling. ▪ Importance of automation in deployment.	▪ Automate deployment using Jenkins or GitHub Actions. ▪ Implement a Blue-Green deployment strategy.
10	**5**	**1, 4,5,7**	**Configuration Management** ▪ Introduction to Infrastructure as Code (IaC). ▪ Tools for configuration management: Ansible, Puppet, Chef.	▪ Write Ansible playbooks to configure a web server. ▪ Automate user account creation with Ansible.
11	**5**	**1, 4,5,7**	**Monitoring and Logging** ▪ Importance of monitoring and logging. ▪ Types of monitoring: Application, Infrastructure. ▪ Logging fundamentals: Sources, levels, formats.	▪ Set up monitoring and logging for a sample application. ▪ Use tools like Prometheus, Grafana, ELK Stack.
12	**5**	**1, 4,5,7**	**Serverless Computing** ▪ Introduction to Serverless computing. ▪ Characteristics, benefits, and challenges. ▪ Use cases of Serverless	▪ Deploy a Serverless function using AWS Lambda or Azure Functions.

			computing (e.g., AWS Lambda, Azure Functions).	
13	**1**	**1, 4,5,7**	**AI in DevOps** ▪ Introduction to AI-powered DevOps tools. ▪ Use cases of AI in DevOps: Predictive monitoring, anomaly detection. ▪ Emerging Trends : Edge Computing	▪ Explore AI tools for predictive monitoring. ▪ Implement a basic AI-powered monitoring solution.

4. References:

Sl No	Description
1	*The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win* - Gene Kim, Kevin Behr, George Spafford
2	*Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* - Nicole Forsgren, Jez Humble, Gene Kim
3	*Cloud Native DevOps with Kubernetes* - John Arundel, Justin Domingus
4	*Pro Git* - Scott Chacon, Ben Straub
5	*GitHub Essentials* - Achilleas Pipinellis
6	*Jenkins 2: Up and Running* - Brent Laster
7	*Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* - Jez Humble, David Farley
8	*Docker Deep Dive* - Nigel Poulton
9	*Kubernetes Up and Running* - Kelsey Hightower, Brendan Burns, Joe Beda
10	***Cloud Computing**: Concepts, Technology and Architecture* - Thomas Erl, Ricardo Puttini, Zaigham Mahmood
11	*AWS Certified Solutions Architect Study Guide* - Ben Piper, David Clinton
12	https://medium.com/@ravipatel.it/understanding-docker-architecture-a-comprehensive-guide-5ce9129df1a4
13	*Introduction to DevOps* - Coursera
14	*Docker Mastery: with Kubernetes + Swarm from a Docker Captain* - Udemy
15	***Cloud Computing** Fundamentals* - Pluralsight
16	Articles on DevOps, CI/CD, and **Cloud Computing**. - Medium DevOps

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1 Theory Test	4	90	50	Average of all CIEs
2.	CIE-2 Practice Test	7	180	50	
3.	CIE-3 Theory Test	10	90	50	
4.	CIE-4 Practice Test	13	180	50	

5.	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/ AWS(30)	13	-	50	
				Total Obtained	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. CIE Theory Test model question paper

Program	**Computer Science and Engineering**			**Semester -**	
Course Name	DevOps: Software Development and Operations			**Test**	**I/III**
Course Code	**25CS53IC**	**Duration**	**90 min**	**Marks**	**50**
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a. Explain the DevOps lifecycle in detail. How does each stage contribute to the overall software delivery process? (10) b. Name three popular DevOps tools and explain their roles in the DevOps lifecycle. (5) c. What are the different cloud deployment models? Explain each with its advantages and disadvantages.	L2	1	25
2	a. Explain the role of virtualization in **Cloud Computing**. How does it enable resource management and scalability? (10)	L2	1	

	b. Discuss the three main cloud service models (IaaS, PaaS, SaaS) with examples. (5) c. Discuss the core practices of DevOps and their impact on software development.(10)			
Section - 2				
3	a) Explain the importance of version control in software development and how Git helps in managing source code effectively. b) A developer is working on a new feature in a separate branch called feature-login. After making multiple commits, they realize they accidentally committed a sensitive API key. How can they remove the API key from Git history while ensuring the branch remains functional? c) Explain the purpose and usage of the following Git commands with examples: git init, git clone, git commit	L2	2	25
4	a) How does branching in Git improve collaboration in software development? Explain with an example. b) A team is working on a project where multiple developers are contributing to the develop branch. One developer tries to merge their feature-cart branch into develop but encounters a merge conflict in the CartService.java file. Explain the steps they should take to resolve the conflict. c) Explain the steps involved in contributing to an open-source project on GitHub.	L2	2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	
Course Name	**DevOps: Software Development and Operations**			**Test**	**II/IV**
Course Code	**25CS53IC**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					
Questions				**CO**	**Marks**
Collaborative Development with Git You are part of a development team working on a web application project. The project is managed using Git, and the team follows a feature-branch workflow. Your task is to simulate a real-world Git workflow by performing various Git					**50**

operations, including branching, merging, handling merge conflicts, and pushing code to a remote repository. The list of tasks include: ▪ Setting Up the Repository ▪ Implementing a Feature in a New Branch ▪ Simulating Collaboration and Resolving Merge Conflicts ▪ Pushing Changes to a Remote Repository		
Scheme of assessment **a)** Repository Initialization and Initial Commit - 10 **b)** Feature Branch Implementation - 10 **c)** Merge Conflict Resolution - 15 **d)** Pushing to Remote Repository – 10 e) Viva - 05		
	Total Marks	**50**

Signature of the Course Coordinator **Signature of the HOD**

9. SEE- Model Practice Question Paper

Program	**Computer Science and Engineering**		**Semester**	
Course Name	**DevOps: Software Development and Operations**	**Course Code 25CS53IC**	**Duration**	**180 min**
Questions			**CO**	**Marks**
You are part of a development team tasked with building and deploying a web application for an online bookstore. The application is built using a Node.js backend and a React frontend. Your team has decided to use GitHub for version control, GitHub Actions for CI/CD. The application must be containerized using Docker. Tasks: a. Set up a GitHub repository for the project. b. CI/CD Pipeline with GitHub Actions c. Containerize the application using Docker. d. Document the entire process and present your work.				**50**
Scheme of assessment a) **Version Control and Collaboration – 10** b) **CI/CD Pipeline – 10** c) **Containerization – 10** **d) Documentation and Presentation - 20**				
			Total Marks	**50**

Signature of the Examiner **Signature of the Examiner**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**5**
Course Name	**Ethical Hacking**	**Type of Course**	**Integrated**
Course Code	**25CS53ID**	**Contact Hours**	**7 hrs per week**
Teaching Scheme	**L: T:P:3:0:4**	**Credits**	**5**
CIE Marks	**50**	**SEE Marks**	**50 (Practice)**

1. Rationale:

Ethical hacking is essential for cybersecurity students because it equips them with practical skills to identify and fix vulnerabilities, understand cyberattack methods, and develop strong defense strategies. By thinking like hackers, they learn to anticipate threats, conduct penetration testing, and use industry-standard tools like Metasploit and Wireshark. Ethical hacking also helps in ensuring compliance with security regulations, enhances problem-solving abilities, and prepares students for high-demand careers such as penetration testing and security analysis. Ultimately, mastering ethical hacking enables cybersecurity professionals to stay ahead of evolving cyber threats and effectively protect systems from malicious attacks.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Describe various penetration testing methodologies and their applications.
CO-02	Perform passive and active reconnaissance to gather critical information about targets.
CO-03	Identify and document vulnerabilities in wired and wireless networks using suitable tools.
CO-04	Analyze and exploit vulnerabilities in web applications, focusing on the OWASP.
CO-05	Document findings from penetration tests in clear and concise reports.

3. Course Content

WEEK	CO	PO	Theory	Practice
1	**1**		**Introduction to Ethical Hacking** ▪ Introduction ▪ Understanding Ethical Hacking and Penetration Testing ▪ Exploring Penetration Testing Methodologies ▪ Building Your Own Lab	Explore different Pentesting methodologies Deploy Kali Linux VM https://www.netacad.com/resources/lab-downloads?courseLang=en-US (Resource Kali) https://www.netacad.com/launch?id=80c156bc-84a4-47c9-a233-5eafe7bdde82andtab=curriculumandview=3174f565-8ce2-5c17-aa59-88a41db574a8
2			**Planning and Scoping a Penetration Testing Assessment**	IT Act 2000 and other cyberlaws in India and across the world

	1		▪ Introduction ▪ Comparing and Contrasting Governance, Risk, and Compliance Concepts ▪ Explaining the Importance of Scoping and Organizational or Customer Requirements ▪ Demonstrating an Ethical Hacking Mindset by Maintaining ▪ Professionalism and Integrity - scenarios in which an ethical hacker (penetration tester) should demonstrate professionalism and integrity	https://www.indiacode.nic.in/bitstream/123456789/13116/1/it_act_2000_updated.pdf https://infosecawareness.in/cyber-laws-of-india https://www.pcisecuritystandards.org/ Section 43A of the IT Act Indian Computer Emergency Response Team (CERT-In) **Industry-Specific Regulations**: RBI IRDAI TRAI **Rules of engagement document** **Strategy: Unknown vs. Known Environment Testing**
3	**2**		**Information Gathering and Vulnerability Scanning** ▪ Introduction ▪ Performing Passive Reconnaissance ▪ Performing Active Reconnaissance ▪ Understanding the Art of Performing Vulnerability Scans ▪ Understanding How to Analyze Vulnerability Scan Results	Passive Reconnaissance ▪ DNS Lookups ▪ Social Media Scraping ▪ Employee Intelligence Gathering ▪ Advanced Searches Active Reconnaissance ▪ Enumeration with Nmap ▪ Packet Crafting with Scapy ▪ Network Sniffing with Wireshark Vulnerability Scans ▪ Vulnerability Scanning with Kali Tools Vulnerability Scan Results ▪ Investigate Vulnerability Information Sources
4	**2**		**Social Engineering Attacks** ▪ Introduction ▪ Pretexting for an Approach and Impersonation ▪ Social Engineering Attacks ▪ Physical Attacks ▪ Social Engineering Tools ▪ Methods of Influence	Pretexting for an Approach and Impersonation ▪ Pretexting and Impersonation Social Engineering Attacks ▪ Phishing ▪ SMS Phishing Physical Attacks ▪ Dumpster Diving ▪ Shoulder Surfing Social Engineering toolkits • Explore Social Engineering toolkit **https://trustedsec.com/resources/tools/the-social-engineer-toolkit-set** Call Spoofing Tools Using the Browser Exploitation Framework (BeEF)

5	**3**		**Exploiting Wired and Wireless Networks** ▪ Introduction ▪ Exploiting Network -Based Vulnerabilities ▪ Exploiting Wireless Vulnerabilities	Network -Based Vulnerabilities ▪ Scanning for SMB Vulnerabilities with enum4linux https://infosecwriteups.com/scanning-for-smb-vulnerabilities-with-enum4linux-896f76d0c078 ▪ DNS Cache Poisoning ▪ SMTP Commands ▪ NAC Bypass and VLAN Hopping ▪ DHCP Starvation and Rogue DHCP Servers ▪ Wireless Vulnerabilities ▪ Unsecured Wireless, KARMA, and Fragmentation Attacks ▪ Wireless Attacks
6	**3**		**Exploiting Application-Based Vulnerabilities** ▪ Introduction ▪ Overview of Web Application-Based Attacks for Security Professionals and the OWASP Top 10 ▪ How to Build Your Own Web Application Lab ▪ Understanding Business Logic Flaws ▪ Understanding Injection-Based Vulnerabilities	Web Application-Based Attacks ▪ Using the GVM Vulnerability Scanner ▪ Injection-Based Vulnerabilities ▪ Command Injection Vulnerabilities ▪ Injection Attacks
7	**3**		▪ Exploiting Authentication-Based Vulnerabilities ▪ Exploiting Authorization-Based Vulnerabilities ▪ Understanding Cross-Site Scripting (XSS) Vulnerabilities ▪ Understanding Cross-Site Request Forgery (CSRF/XSRF) and Server-Side Request Forgery Attacks	Authentication-Based Vulnerabilities ▪ Session Hijacking ▪ Using Password Tools ▪ Authorization-Based Vulnerabilities ▪ Insecure Direct Object Reference Vulnerabilities ▪ Cross-Site Scripting (XSS) Vulnerabilities ▪ Cross Site Scripting ▪ Cross-Site Request Forgery (CSRF/XSRF) and Server-Side Request Forgery Attacks ▪ CSRF/XSRF Attacks
8	**3**		▪ Understanding Clickjacking ▪ Exploiting Security Misconfigurations ▪ Exploiting File Inclusion Vulnerabilities ▪ Exploiting Insecure Code Practices	Security Misconfigurations ▪ Directory Transversal ▪ Insecure Code Practices ▪ Insecure Code ▪ Unprotected APIs ▪ Web Hacking Tools ▪ Use the OWASP Web Security Testing Guide
9	**3**		**Cloud, Mobile, and IoT Security** ▪ Introduction	Attack Vectors and Performing Attacks on Cloud Technologies ▪ Credential Harvesting ▪ Cloud Attack Types

			▪ Researching Attack Vectors and Performing Attacks on Cloud Technologies ▪ Explaining Common Attacks and Vulnerabilities Against Specialized Systems	Common Attacks and Vulnerabilities Against Specialized Systems ▪ Attacking Mobile Devices ▪ Common IoT Vulnerabilities ▪ Management Interface Vulnerabilities
10	**4**		**Performing Post-Exploitation Techniques** ▪ Introduction ▪ Creating a Foothold and Maintaining Persistence After Compromising a System ▪ Understanding How to Perform Lateral Movement, Detection Avoidance, and Enumeration	Creating a Foothold and Maintaining Persistence After Compromising a System ▪ Reverse and Bind Shells ▪ Types of C2 Utilities Understanding How to Perform Lateral Movement, Detection Avoidance, and Enumeration ▪ Post Exploitation Scanning ▪ Post Exploitation Privilege Escalation ▪ Steganography
11	**5**		**Reporting and Communication** ▪ Introduction ▪ Comparing and Contrasting Important Components of Written Reports ▪ Analyzing the Findings and Recommending the Appropriate Remediation Within a Report ▪ Explaining the Importance of Communication During the Penetration Testing Process ▪ Explaining Post-Report Delivery Activities	Comparing and Contrasting Important Components of Written Reports **1)** Penetration Reporting **2)** Control and Distribution of Reports **3)** Common Themes/Root Causes **Explore PenTest Reports** Analyzing the Findings and Recommending the Appropriate Remediation Within a Report ▪ Recommend Remediation Based on Findings ▪ Explaining the Importance of Communication During the Penetration Testing Process ▪ Communication Triggers ▪ Explaining Post-Report Delivery Activities ▪ Post Report Delivery
12	**5**		**Tools and Code Analysis** ▪ Introduction ▪ Understanding the Basic Concepts of Scripting and Software Development ▪ Understanding the Different Use Cases of Penetration Testing Tools and Analyzing Exploit Code	Understanding the Basic Concepts of Scripting and Software Development ▪ Programming Languages ▪ Analyze Exploit Code ▪ Analyze Automation Code Understanding the Different Use Cases of Penetration Testing Tools and Analyzing Exploit Code ▪ Common Tools for Reconnaissance and Enumeration ▪ Common Tools for Forensics ▪ Common Tools for Software Assurance ▪ Cloud Tools
13	**1, 2, 3,**		Mini project work	Conduct complete Penetration testing on a vulnerable website.

	4, 5			

4. References:

Sl. No.	Reference / Link
1	*The Web Application Hacker's Handbook* – Dafydd Stuttard and Marcus Pinto
2	*Hacking: The Art of Exploitation* – Jon Erickson
3	*Metasploit: The Penetration Tester's Guide* – David Kennedy et al.
4	*Ethical Hacking and Penetration Testing Guide* – Rafay Baloch
5	*CEH Certified Ethical Hacker All-in-One Exam Guide* – Matt Walker
6	*The Basics of Hacking and Penetration Testing* – Patrick Engebretson
7	*Practical Malware Analysis* – Michael Sikorski and Andrew Honig
8	Kali Linux Documentation
9	OWASP (Open Web Application Security Project)
10	Offensive Security (OSCP, Kali Linux)
11	MITRE ATTandCK Framework
12	SANS Cyber Security Training
13	Cybrary Ethical Hacking Courses
14	TryHackMe (Hands-on Cybersecurity Training)
15	Hack The Box (Penetration Testing Labs)
16	IT Act 2000 (India)
17	CERT-In (Indian Computer Emergency Response Team)
18	NIST Cybersecurity Framework
19	PCI Security Standards Council
20	Nmap (Network Scanner)
21	Metasploit Framework
22	Burp Suite (Web Application Security Testing)
23	Wireshark (Network Traffic Analysis)
24	Social Engineer Toolkit (SET)
25	BeEF (Browser Exploitation Framework)

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (10) ▪ Mini Project (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (20)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 10 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester –V	
Course Name	Ethical Hacking			Test	I/III
Course Code	25CS53ID	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	a) What are ethical hackers? In your opinion what are the ethics should be followed by an ethical hacker? b) What are the different pen testing methodologies you came across? Explain briefly c) Write a brief plan for penetration testing for a Company A			25
2	a)You are an ethical hacker, When a company XYZ wants to give their pentesting project to you. How can you show them your ethical hacking mindset? b)Mr. Neeraj manager of XYZ Inc. wants to more information **on** Unknown vs. Known Environment Testing. How can you explain the same to him? c) What are the passive reconnaissance techniques you prefer when you are doing reconnaissance on a software company? Justify your answers			
	Section - 2			
3	a)How nmap tool will help you in active reconnaissance? b)What is network sniffing? How can you achieve it using tools? c) How can you analyze Vulnerability Scan Results			25

4	a) What is Pretexting and Impersonation? How you can achieve while doing pentest? b) What are Phishing and Smishing? What are the possible solutions to avoid these attacks? Justify your answers c) What are the social engineering tools you used? Explain briefly			
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	**V**
Course Name	**Ethical Hacking**			**Test**	**II/IV**
Course Code	**25CS53ID**	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
Identify and make a report of any two Web Application-Based Attacks on abc.com website using any tool you learnt.		**50**
Scheme of assessment a) Identification and using of right tool b) Conducting vulnerability scan c) Identifying the issue d) Report e) Explaining Countermeasures/Viva		 10 10 10 10 10
TotalMarks		**50**

Signature of the Course Coordinator Signature of the HOD

9. **SEE- Model Practice Question Paper**

Program	**Computer Science and Engineering**		**Semester**	**V**
Course Name	**Ethical Hacking**	**Course Code**	**Duration**	**180 min**

Questions	**CO**	**Marks**
Conduct Cross-Site Request Forgery (CSRF/XSRF) and Server-Side Request Forgery Attacks on a vulnerable website using right tool. Report your findings along with countermeasures		**50**
Scheme of assessment a) Identification and using of right tool b) Conducting vulnerability scan		 10 10 10

c) Identifying the issue d) Report e) Explaining Countermeasures/Viva	10 10
TotalMarks	**50**

1) **Signature of the Examiner** **2) Signature of the Examiner**

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	**V**
Course Name	Software Engineering and Entrepreneurship	**Type of Course**	Integrated
Course Code	25CS54I	**Contact Hours**	8 hrs per week 104 hrs/sem
Teaching Scheme	L:T:P::4:0:4	**Credits**	6
CIE Marks	**50**	**SEE Marks**	**50 (Theory)**

1. Rationale:

In today's fast-evolving technology landscape, software engineers must not only develop high-quality software solutions but also understand the fundamentals of entrepreneurship to drive innovation and business growth. This course bridges the gap between technical proficiency and entrepreneurial mindset, equipping students with the skills to design, develop, and deploy software solutions while also exploring business strategies, market dynamics, and startup methodologies. By integrating software engineering principles with entrepreneurial thinking, students will learn how to identify market needs, develop scalable software products, and navigate the challenges of launching and managing a technology-driven business. The course fosters creativity, problem-solving, and leadership skills essential for aspiring tech entrepreneurs.

1. Course Outcomes: At the end of the Course project, the student will be able to

CO-01	Explain Software Engineering principles and Software Development Life Cycle
CO-02	Analyze a given problem statement to identify risks and requirements, formulate an effective risk mitigation strategy, and develop user stories aligned with the identified requirements.
CO-03	Implement the given user story and validate its quality through testing.
CO-04	Identify the viable business opportunities through entrepreneur skills.
CO-05	Create comprehensive business plans, including marketing, operations, and financial strategies.

3. Course Content

Week	CO	PO	Lecture (4 Hours per Week)	Practical (CAD) (4 Hours per Week)
1	1	1, 5, 7	**Software engineering** ▪ Need for Software Engineering ▪ Distinguish Product, Project and Process ▪ Characteristics of software ▪ Factors influencing Software development ▪ challenges in a software project ▪ Success – organizational, technical and personal ▪ Software engineering ethics ▪ Software Development life cycle and associated artifacts.	▪ Discuss success and failure stories – software development ▪ Explore and present notable case studies in the software industry to identify the critical factors influencing project outcomes. Through case studies of successful projects and failures analyse key success factors and common pitfalls.
2	1	1, 7	**Process Models** • Traditional models - water fall, incremental • AGILE – values and principles (manifesto) • Agile ceremonies and roles (with their responsibilities in each ceremony) • Agile Frameworks – Scrum and Extreme Programming (concepts and practices) • Process assessment and improvement	• Prepare presentation on XP practices with benefits • Research and analyse case studies from company websites to understand how Agile adoption has enhanced processes, improved productivity, and increased customer satisfaction. Prepare a concise report organized into sections, including an introduction, a summary of findings, and a conclusion with actionable recommendations.
3	2	1, 2, 3, 4, 7	**Risk Management and Requirement Engineering** ▪ **Introduction to Risk Management**: Definition, Importance, Types of Risks – financial, operational, technical, compliance, strategic and reputational [reason, impact and examples for each] ▪ Why Risk management? ▪ Components of risk management: techniques and outcomes ▪ **Introduction to Requirement Engineering:** Definition, Importance, Requirement Engineering Process	▪ Identify risks for a sample IT project and classify them based on severity and probability Eg. Risk Assessment for a New Mobile Banking App Deployment ▪ Identify and categorize requirements for the following projects: a. Online Food Delivery System b. University Complaint Management System c. Smart Parking System

			▪ **Types of Requirements**: Functional vs. Non-Functional, User vs. System, Business vs. Technical	d. Healthcare Appointment Scheduling System e. E-Commerce Website for Electronics
4	2	1, 2, 3, 4, 7	**Requirement Modeling** ▪ Unified Modeling Language ▪ Types of UML Diagrams in Requirement Modeling-Use Case Diagram, Class Diagram, Activity Diagram, Sequence Diagram ▪ Agile Approach to Requirement Modeling – User stories ▪ Structure of a User Story ▪ Acceptance Criteria and Definition of Done (DoD) ▪ INVEST criteria ▪ Story Mapping: Organizing Requirements Visually ▪ Requirement Prioritization Techniques - MoSCoW , Kano Model	**Tools for Requirement Modeling** ▪ Use Lucidchart, Draw.io, or StarUML to create a Use Case Diagram and Class Diagram for an E-commerce Shopping Cart System ▪ Model user login and product purchase flow in an e-commerce system using a sequence diagram ▪ Draw a Use Case Diagram for an online banking system using Lucid chart/Draw.io. ▪ Create **user stories** for a **food delivery app**. ▪ Organize stories using **story mapping**. ▪ Explore JIRA – tool. ▪ Create a product backlog using Jira : Create user stories and define acceptance criteria using JIRA for tracking requirements and risks
5	3	1, 2, 3, 4, 7	**Software Design and Architecture** ▪ Software Design vs. Software Architecture ▪ Importance of Good Design and Architecture ▪ Principles of Software Design ▪ Architectural Patterns - Layered Architecture, Client-Server Architecture, Service Oriented Architecture, Microservices Architecture, ▪ Event-Driven Architecture (concept only) ▪ Design Patterns: Singleton, Factory, Adapter and Observer	▪ Implement design patterns in Java/Python for a real-world scenario ▪ Study and analyze the architecture of popular applications like Netflix or Amazon and discuss trade-offs

6	3	1, 2, 3, 4, 7	**Implementation and Testing** ▪ Clean Code Principles, Secure Coding Practices, best coding practices ▪ Software Testing - Fundamentals ▪ Importance of Testing ▪ Differentiate QA, QC, and Testing ▪ Levels of Testing ▪ White box vs black box testing ▪ Manual vs. Automated Testing, ▪ Static testing vs Dynamic Testing ▪ Functional vs Non-functional testing ▪ Seven Testing Principles ▪ Defect – life cycle	▪ Refactor an unstructured Java/Python codebase following industry coding standards ▪ Write unit test cases using JUnit/ PyTest for a sample application ▪ Write test cases for login page ▪ Defect tracking with Jira
7	3	1, 2, 3, 4, 7	**Operations and Measurement** ▪ What is DevOps? ▪ DevOps lifecycle and principles. ▪ Core practices ▪ Benefits of DevOps in software development. ▪ Popular DevOps tools and their roles in the DevOps lifecycle. **Software Measurement and Metrics** Need for measurement; measurement types Metric types, Agile Metrics	▪ Set up a DevOps environment on a local machine – by Installing Required Tools ▪ Explore tools for software measurement and metrics, evaluate their features, and present a report with recommendations.
8	1, 2, 3	1, 2, 3, 4, 7	**Software Project Management** • What is software project management? • Why is it important? • Key challenges in software project management • Role of a project manager • Project planning essentials: Scope, Schedule, Budget, and Resources • Work Breakdown Structure (WBS) and Task Allocation **Project Scheduling and Estimation** • Agile Estimation Techniques • Agile Project Scheduling with Sprints	Create a project plan for a sample software project • Define the scope, schedule, budget, and resources. • Create a Work Breakdown Structure (WBS) and allocate tasks to team members.

9	4	2, 7	**Introduction to Entrepreneurship and Startup Ecosystem** • Definition and importance of entrepreneurship • Role of startups in the tech industry • Karnataka's startup ecosystem and government support (Startup Karnataka, MSME policies) • Entrepreneurial mindset: risk-taking, resilience, creativity. • Overview of Karnataka startup ecosystem.	**Exploring Tech Business Ideas** **Case Study: Analyze the journey of few successful Indian tech entrepreneurs (e.g., Narayana Murthy, Nandan Nilekani, Kiran Mazumdar-Shaw, Vijay Shekhar Sharma, Nikhil Kamath, Ritesh Agarwal)** **Self-Assessment: Perform SWOT analysis and identify personal entrepreneurial traits.**
10	4	2, 7	**Ideation and Problem-Solving Approach in Startups** • Brainstorming innovative tech business ideas- Market research techniques (surveys, focus groups) • Identifying real-world problems and customer pain points. • Problem-solving models: Design Thinking, Lean Startup, TRIZ • Identifying market gaps and customer needs. • Technology-driven business opportunities • Competitor analysis and unique selling proposition (USP)	**Ideation and Problem-Solving with Design Thinking** ▪ Assign groups to perform brainstorm session and list **5 innovative IT-based business ideas** based on current needs and trends. ▪ **Analyse case studies** of successful start-ups in these domains and challenges faced. ▪ Rapid prototyping (paper sketches, Wireframing) ▪ Conduct a SWOT analysis on the proposed business model to assess viability. ▪ Iterative feedback loop: Validating problem-solution fit
11	4	2, 7	**Minimum Viable Product (MVP) and Lean Startup** ▪ What is an MVP? How to build and test it ▪ Agile methodology and iterative development ▪ Lean Startup principles: Build-Measure-Learn ▪ Pivoting vs. Persevering: How startups evolve based on feedback	Building an MVP- Team works • Develop a basic MVP prototype (low-code/no-code tools) • Conduct customer validation (testing MVP with peers) • A/B Testing and User Feedback Collection - Refining the MVP based on feedback • Draft a Basic Terms of Service and Privacy Policy for an IT Product

				• Case Study: Patent vs. Copyright vs. Trademark in Software Industry • Draft a Software Licensing Agreement for a Product.
12	5	4, 7	**Business Models, Monetization and Legal Compliance** ▪ Business Model Canvas (BMC) for IT startups ▪ Basics of budgeting, costing and break-even analysis ▪ Business model types: Subscription, Freemium, SaaS, On-Demand ▪ Digital marketing, SEO, and social media growth hacking ▪ Monetization strategies: Ads, Affiliate, bootstrapping, angel investors, venture capital, Crowdfunding ▪ Intelligence Property Rights (IPR) – Patents, Copyrights, Trademarks ▪ Startup legal requirements in Karnataka (GST, Company registration)	Monetization Strategy and Digital Business Setup ▪ Develop a Business Model Canvas for any IT related Startup ▪ Conduct a group Activity to Present and Validate BMC and having a Peer Feedback ▪ Calculate the Break-even Point for a Startup Using Cost and Revenue Data. ▪ Setting up a landing page/social media for MVP ▪ Explore Google Analytics, SEO tools for business ▪ Each student/team selects a startup idea (Innovative, Scalable, and easy to explain) that is suitable for crowdfunding. ▪ Assign a case Study to understand IPR and company registration process.
13	5	4, 7	**Startup Pitching, Growth Strategies and Government Support** ▪ How to pitch to investors (Elevator Pitch, Pitch Deck) ▪ Growth hacking strategies for IT startups ▪ Government startup schemes in Karnataka (Elevate 100, Startup Karnataka, Startup India, MSME, SIDBI) ▪ Career options as a Tech entrepreneur	**Final Business Pitch and Roadmap to Implementation** ▪ Students present and deliver a 3-minute elevator pitch. ▪ Live pitch session with peer reviews ▪ Feedback from mentors and industry experts ▪ Invite industry experts and conduct seminars on entrepreneurship and management.

			▪ Overcoming startup failures and risk mitigation	

4. References:

Sl No	Description
1	*Software Engineering: A Practitioner's Approach,* Roger S. Pressman
2	*Clean Code: A Handbook of Agile Software Craftsmanship* Robert C. Martin
3	*The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win* - Gene Kim, Kevin Behr, George Spafford
4	*Scrum: The Art of Doing Twice the Work in Half the Time* - Jeff Sutherland
5	*A Guide to the Project Management Body of Knowledge (PMBOK Guide)*- Project Management Institute
6	*Agile Estimating and Planning* - Mike Cohn
7	*The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* - Eric Ries
8	*Zero to One: Notes on Startups, or How to Build the Future* - Peter Thiel
9	*Software Development Lifecycle Specialization*- Coursera
10	*DevOps Beginners to Advanced with Projects* - Udemy
11	DevOps.com
12	Docker Docs
13	The Art of Agile Development" by James Shore

5. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1 Theory Test	4	90	50	Average of all CIEs
2.	CIE-2 Practice Test	7	180	50	
3.	CIE-3 Theory Test	10	90	50	
4.	CIE-4 Practice Test	12	180	50	
5.	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 15 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	13	-	50	
			Total Obtained		**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

6. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program	Computer Science and Engineering			Semester -V	
Course Name	Software Engineering and Entrepreneurship			Test	I/III
Course Code	25CS54I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
	Section - 1			
1	d) Explain the core values of Agile Manifesto. e) Explain the role of the Product Owner during a Sprint Review. f) A software company successfully delivered a banking application. However, the following observations were made: i. The project met all deadlines but had some minor technical issues. ii. Some team members were dissatisfied due to excessive workload. iii. The application received positive feedback from clients. Assess whether this project is successful from an organizational, technical, and personal perspective. Suggest ways to improve success in all three areas.	L2,L3	1	25
2	b) How does the Scrum Master ensure that Agile ceremonies are effective and valuable? c) A company is developing a software-based hospital management system. The company has a team working on: i. Developing different features such as patient records, billing, and scheduling. ii. Managing the deadlines and requirements for the hospital system.	L2, L3	1	

	iii. Implementing coding standards and testing strategies. iv. Identify which team is involved in product, project, and process aspects. v. Justify your classification with reasoning. d) Explain any Five XP concepts with their significance.			
	Section - 2			
3	a. A Fintech company is developing a new online payment system. During development, they encounter the following risks: ▪ Security vulnerabilities in transactions ▪ Changing government regulations ▪ Competition from other companies Classify these risks into technical, compliance, and strategic categories. Explain how each risk can impact the project and suggest possible mitigation strategies. b. List and briefly explain three key techniques used in risk management. c. A hospital management system is being developed. The following requirements were gathered: ▪ The system should allow doctors to view patient history. ▪ The system should respond within 2 seconds for any query. ▪ The system should comply with medical data privacy laws. Categorize these requirements into functional, non-functional, business, or technical. Justify why each requirement falls into a specific category.	L3	2	25
4	a. A company is developing an e-commerce platform. Initially, they did not document user requirements properly, leading to frequent changes and customer dissatisfaction. i. Identify which Requirement Engineering process steps were missing or improperly handled. ii. Suggest how proper Requirement Engineering could have improved project success. b. Differentiate between functional and non-functional requirements with examples. c. A software development firm is working on a critical healthcare application. They decide to implement risk management strategies to prevent failures.	L2	2	

	Propose a risk identification method they should use. Explain how they can assess and mitigate risks during development.			

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			**Semester**	
Course Name	Software Engineering and Entrepreneurship			**Test**	**II/IV**
Course Code	25CS54I	**Duration**	**180 min**	**Marks**	**50**
Name of the Course Coordinator:					

Questions	**CO**	**Marks**
You have been hired as a Business Analyst at ShopEase, an e-commerce platform planning to introduce a subscription-based shopping model. The company aims to provide customers with: ▪ Monthly grocery deliveries ▪ Premium membership benefits ▪ Personalized product recommendations Your role is to analyze requirements, define user stories, and model the system using UML diagrams. Additionally, you will manage the product backlog and sprint planning using JIRA. **Task** a. Identify at least two epics from the given scenario. b. Write at least four user stories under one chosen epic c. Define Acceptance Criteria for each user story based on the INVEST criteria. d. Model the subscription-based shopping system using Lucidchart, Draw.io, or StarUML. Include the following actors and their interactions: ▪ Customers (subscribe, renew, cancel, receive deliveries) ▪ Admin (manage subscriptions, process payments) ▪ Payment Gateway (handle transactions) e. Create a Product Backlog in JIRA with all user stories and Plan Sprint 1	**2**	**50**
Scheme of assessment Identify epics, Create User Stories and Acceptance Criteria - 15 Draw UML Diagram – 15 Demonstrate Product Backlog and Sprint Planning in JIRA – 15 Viva - 05		
	Total Marks	**50**

Signature of the Course Coordinator Signature of the HOD

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	32
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
Department of Technical Education

C-25 Diploma Curriculum

INTERNSHIP

AND

CAPSTONE PROJECT

GUIDELINES

FOR

FACULTY, STUDENTS AND EXAMINERS

INTERNSHIP

Introduction

The students of Polytechnic Programs will have an opportunity to be part of one of the most challenging educational experiences in the year-3, The students will be trained in the specialization pathways of their interest in fifth semester, followed by 13-week internship or a Capstone Project work in sixth semester.

An internship is a professional learning experience which offers meaningful, practical work relevant to a student's field of study or career interest. It gives the students an opportunity for exploring the various career choices and acquire varied skills. It also offers an opportunity to bring out the innovative, creative ideas and energy into the workplace. This effectively aims at developing talent and potentially builds a pipeline for future Job prospects that may be ready for challenging roles in future. Internship has become very crucial for students to gain on-field experience which acts as an advantage for the students who do not have corporate experience.

Internships allow students to examine new situations, work techniques, problem-solving tactics, interpersonal skills, understanding of timelines and targets which would otherwise not be possible unless they were on board. Companies which plan to offer job placements to students also prefer hiring the interns for a short period as a trial wherein they have an opportunity to assess their ability and select them based on their observations over a considerable amount of time. This alternative gives the recruiter a better understanding of the candidate's worth in comparison to the assessment made in couple of interview sessions. Even for the interns it is a win-win situation as they get an opportunity to learn the corporate work culture in advance and later demonstrate their skills at their workplace

Outcomes

After completing Internship, Interns will be able to,

- Apply the theoretical knowledge and skill during performance of the tasks assigned in internship
- Demonstrate soft skills such as time management, positive attitude and communication skills during performance of the tasks assigned in internship
- Document the Use case on the assigned Task

Facilitating the Interns by an Internship Provider

- Orient intern in the new workplace. Give interns an overview of the organization, Explain the intern's duties and introduce him or her to co-workers
- Develop an internship job description with clear deliverables and timeline
- Allow the interns in meetings and provide information, resources, and opportunities for professional development
- The interns have never done this kind of work before, they want to know that their work is measuring up to organizational expectations, hence provide professional guidance and mentoring to the intern

- Daily progress report of Intern is to be evaluated by industry supervisor. examine what the intern has produced and make suggestions. Weekly supervision meetings can help to monitor the intern's work.

Duties and Responsibilities of the Faculty (Cohort Owner):

- To facilitate the placement of students for the internship
- To liaison between the college and the internship provider
- To assist the Industrial Training Supervisor during assessment

Instructions to the Interns(Students):

- Students shall report to the internship provider on the 1st day as per the internship schedule
- Intern is expected to learn about the organization, its structure, product range, market performance, working philosophy etc
- The interns shall work on live (On Job) projects assigned by the internship provider.
- The Intern shall record all the activities in the daily log book and get the signature of the concerned training supervisor
- Intern shall have 100% attendance during internship programme. In case of un-avoidable circumstances students may avail leave with prior permission from the concerned training supervisor of the respective internship provider. However, the maximum leave permitted during internship shall be as per company norms where they are working and intern shall report the leave sanctioned details to their college cohort owner
- The interns shall abide all the Rules and Regulations of internship provider
- Intern shall follow all the safety Regulations of internship provider.
- On completion of the internship, intern shall report to the college and submit the internship certificate mentioning duration of internship, evaluation of interns by internship provider, Student's Diary, report to the cohort owner.

Assessment

The Internship will be assessed for 100 marks through formative and summative assessment tools, in formative assessment the internship will be evaluated for 50 marks and in summative assessment internship will be evaluated for 50 marks

The Formative Assessment- (Continuous Internal Evaluation- CIE)

The Formative Assessment is conducted for 50 marks throughout the course in three developmental phases as CIE-I, CIE II and CIE-III. Students shall complete CIE-I before taking CIE-II and complete CIE-II before taking CIE-III, otherwise will not be eligible to take Semester End Examination.

Continuous Internal Evaluation- CIE - I conducted at the end of 4th week		
Sl No	**Assessment parameter**	**Marks**
1	Submit a report to the training supervisor and copy to the cohort owner focusing on: • Overview of the organization • Vision and mission of the organization • Organization structure • Roles and Responsibilities of personnel in the organization • Products and market performance	30
2	Give a presentation on the above	20
	Total	**50**

Note: CIE-1 shall be assessed by the Faculty (Cohort owner) for 50 marks using appropriate Rubrics.

Continuous Internal Evaluation- CIE - II conducted at the end of 8th week		
Sl No	**Assessment of On Job Training (OJT)**	**Marks**
1	On select job role of his/her interest in an organization or role assigned by the training supervisor for next Four weeks and submit a report to the training supervisor and copy to cohort owner focusing on: 1. Intern's ability to apply the skill and technical knowledge on OJT 2. Intern's performance on assigned tasks and project 3. Extent of Intern's ability to add value to the organization through internship	30
2	Document a Use case on a task where he is working as intern	20
	Total	**50**

Note:

1. CIE-II shall be assessed by the Industrial Training Supervisor using companies' assessment Tools/Rubrics.

2. Faculty (Cohort owner) shall assist the Industrial Training Supervisor during assessment of CIE-II.

Continuous Internal Evaluation- CIE -III conducted at the end of 13th week		
Sl No	**Assessment of On Job Training (OJT)**	**Marks**
1	On select job role of his/her interest in an organization or role assigned by the training supervisor for next Five weeks and submit a report to the training supervisor and copy to cohort owner focusing on: 1. Intern's ability to apply the skill and technical knowledge on OJT 2. Intern's performance on assigned tasks and project 3. Extent of Intern's ability to add value to the organization through internship	30
2	Documenting of another Use case on a task where he is working as intern	20
	Total	**50**

Note:

1. CIE-III shall be assessed by the Industrial Training Supervisor using companies' assessment Tools/Rubrics
2. Faculty (Cohort owner) shall assist the Industrial Training Supervisor during assessment of CIE-III
3. **Average Marks obtained in the above Three CIE's (CIE-I,CIE-II and CIE-III) shall be recorded as Formative assessment for 50 Marks.**

The Summative assessment- Semester End Examination (SEE)

During the semester end examination, students shall demonstrate the outcomes of their Internship to the Panel of Examiners comprises of a Cohort owner and an external Subject expert The evaluation criteria are as follows

Sl No	SEE Evaluation Criteria	Marks
1	**Presentation shall include**: 1. Overview of the organization, vision, mission, structure, roles and responsibilities of personnel's, products, market Performance etc. - (10marks) 2. The role performed in the organization during OJT and Intern's ability to apply the skill and technical knowledge – (20 marks)	30
2	Evaluation of comprehensive Internship Report with special focus on organization profile, OJT and contribution made to the organization	20
	Total	**50**

Note: Cohort owner and External subject expert shall assess the intern separately using an appropriate rubrics and average marks to be tabulated

FORMATS

Department of Technical Education

FORMAT - 1

Student Internship Application

(Complete and submit to the Training and Placement Officer)

Student Name			
Student e-mail Id			
Mobile			
Name of the Institute			
Name of the Program			
Specify the Specialization Pathway			
Overall CGPA			
Internship Preferences	Location	Core area	Organization
Preferance-1			
Preferance-2			
Preferance-3			
Cohort owner Signature: Date			
Student Signature: Date			

Department of Technical Education

FORMAT - 2

Request Letter from Institute to Internship Provider

(To be forwarded by the Training and Placement Officer/Student)

Date:

To,

.....

.....

Subject: Request for 13 weeks Semester long internship training of Diploma in ---------- engineering Program

Dear Sir/ Madam,

This is to certify your good office to allow Mr/ Ms----------student of our polytechnic perusing sixth semester diploma in ---------- engineering and trained in ---------- specialization pathway in boot camp mode to render on-the-job internship training in your prestigious company

As per the requirement of Diploma in ---------- engineering program, he/she is required to complete 520 hours of internship related to his/her specialization

Your support in this regard is highly appreciated

With warm regards,

Signature of Training and Placement Officer

Department of Technical Education

FORMAT - 3

Agreement

This Agreement is between the student, cohort owner and internship provider. It shall serve to clarify the educational purpose of the internship and to ensure an understanding of the total learning experience among the student, cohort owner and Industrial training supervisor

Part I: Contact Information

Student

Name: _____ Student ID_____Class Year: _____

College Address: _____City, State: _____

Phone: _____ Email: _____

Industrial Supervisor

Name: _____ Designation: _____

Company/Organization: _____

Address: _____City, State ---------- Pin: _____

Phone: _____ Email: _____

Cohort owner

Name: _____ Designation: _____

Phone: _____ Email: _____

College Address: _____

Academic Credit Information

Program: _____

Pathway: ---------- Credits: 16

Beginning Date: _____ Ending Date: _____

Hours per Week: 40

Part II: The Internship

Internship Objectives:
Describe What do the interns intend to learn, acquire skill through this internship? Try to use concrete, measurable terms in listing the learning objectives under each of the following categories:

- Knowledge
- Skills

Job Description:
Describe in as much detail as possible intern's role and responsibilities while on internship. List duties, project to be completed, deadlines, etc. Describe How interns' technical knowledge can be applied at the site of the internship and how they can create value to the organization through internship

Supervision: Describe in as much detail as possible the supervision to be provided/needed at the work site. List what kind of instruction, assistance, consultation interns will receive from whom, etc

Evaluation: Describe How will interns work performance be evaluated? By whom? When?

Part III: Agreement

This Agreement may be terminated or amended by student, cohort owner or industry training supervisor at any time upon 7days written notice, which is received and agreed to by the other parties.

Student _____ Date _____

Cohort owner _____ Date _____

Industry Training Supervisor _____ Date _____

Department of Technical Education

FORMAT 4

Student's Daily Log Book

Day-1	Date:
Time of Arrival	Time of Departure
Dept/Division	Nature of work
Name of the Supervisor With designation and email ID	
Remarks of the Training supervisor:	
Record Main actives of the day (including observation, sketches, discussions, etc) **Signature of Industry Supervisor**	

Note: Prepare a A4 size hard bound Intern work book using this format with college and student details

Department of Technical Education

FORMAT 4

Internship Report template

The student, after the completion of internship should submit a comprehensive Internship report, the contents of the report shall be arranged in the following order:

1. Cover Page
2. Inside Title Page
3. Internship Certificate issued by the organization
4. Acknowledgements
5. Executive Summary
6. Table of Contents
7. List of Figures
8. List of Tables
9. Abbreviations/ Notations/ Nomenclature
10. Text of the Report
    - **Chapter 1:** Company Profile
    - **Chapter 2**: Describe in as much detail as possible intern's role and responsibilities while on OJT. List duties, project completed, etc. Describe How interns' technical knowledge can be applied at the site of the internship and how they can create value to the organization through internship
11. Student Profile/Resume
12. Photo Gallery
13. Appendices

General Guidelines

Report Size - Report may contain maximum of about 50 pages including Proto gallery and appendices.

Paper Size - Use A4 size paper

Paper Quality - White bond paper weighing 85 g/m2 or more should be used. Photographs or images with dense colors may be printed in single side on glossy paper.

Margins - A margin of 40 mm is to be provided on left and 30 mm on right sides, whereas top and bottom margins should be 30 mm. No print matter should appear in the margin except the page numbers. All page numbers should be centered inside the bottom margin, 20mm from the bottom edge of the paper.

Font - Times New Roman (TNR) 12-point font has to be used throughout the running text. The captions for tables and figures should have font size of 11 and foot notes should be set at font size

10. Font sizes for various levels of headings are given in the table below

CHAPTER 3

TITLE PAGE-CENTERED TNR 17-POINT BOLD ALL CAPS

3.1. Section Heading
Left aligned with number, TNR 17 points, bold and leading caps
3.1.1. Second level section heading
Left aligned with number, TNR 14 points, bold and sentence case**.**
3.1.1.1 Third level section heading
Left aligned with number, TNR 12 points, bold and sentence case.
Fourth-level section heading
Numbered subsections beyond third level are not recommended. However, fourth-level subsection headings may be included without numbering, TNR 12-point font, left aligned and italicized
Running text should be set in 12-point TNR and fully justified. First line of paragraph should have indentation of 15 mm**.**

Line Spacing - The line spacing in the main text should be 1.5, for quotations, figure captions, table captions, figure legends, footnotes, equations, tables, figures, and quotations Single line spacing should be given.

Table / Figure/equation Format-

Tables and figures shall be numbered chapter-wise. For example, second figure in Chapter 3 will be numbered Figure 3.2. The figure can be cited in the text as Figure 3.2, Tables shall be numbered similarly (Table 2 in Chapter 3 will be numbered Table 3.2) and shall be cited in the text as Table 3.2. Figure caption shall be located below the figure. Table number and caption shall be located above the table.

Appendices

Include data tables, drawings, background calculations, specification lists for equipment used, details of experimental configuration, and other information needed for completeness,

Page Numbering

Page numbers for the prefacing materials (Inside title page, certificate, acknowledgements, executive summary, table of contents, etc.) of the report shall be in small Roman numerals and should be centered at the bottom of the pages.

The numbering of the prefacing material starts from the Inside Title Page. However, the number is not printed on the Inside Title Page. Each new item of the prefacing materials listed above should start on a fresh paper on right page. If the content of the prefacing material exceeds one page, it has to be printed on both sides of the paper by starting from the right-side page. For example, if the item „Table of Contents" extends for 5 pages, it should be printed in fresh paper on right side page with second page of the „Table of Contents" on the back of the paper and then continued. The page numbers of the prefacing material will be printed in small Roman numerals continuously counting blank pages also. However, the numbers are not printed on the blank pages

The body of the report starting from Chapter 1 should be paginated in Arabic numerals and should be centered at the bottom of the pages. The pagination should start with the first page of Chapter 1 and should continue throughout rest of the report. Each side of a sheet of paper should be counted as a separate page, even if the back side of a sheet of paper is blank. The odd numbered pages are always on the right and even-numbered pages are always on the left. If the end of a chapter is in odd page (right side page) the next chapter should start on odd page i.e., on a fresh paper, and should be numbered as odd only by counting the blank even page also. However, the page number is not printed on the blank pages.

Each of the items - Inside cover page, Certificate, Acknowledgements, executive summary, Table of Contents, List of Figures, List of Tables, Abbreviations, Notations, Nomenclature, each new Chapter, References, and each new Appendix should start on an odd page i.e., on the right side

Non-Paper Material

A report may contain non-paper material, such as specimen, CDs and DVDs, Pen drive if necessary. They have to be accommodated in a closed pocket in the back cover page of the report. The inclusion of non-paper materials must be indicated in the Table of Contents. All non-paper materials must have a label each clearly indicating the name of the candidate, student Register number and the date of submission**.**

Binding

Two hard bounded copies of the project Report shall be submitted for evaluation; the cover page should be printed on sky blue card of 300 g/m2 or above**.** One copy is used for Semester End Examination and after the exam it should be maintained in the concerned Head of the department and another copy is maintained at cohort owner

Electronic Copy

An electronic version (PDF) of the project report should be submitted to the cohort owner and Head of the department. The file name should contain, student name, Register number and date of submission.

CAPSTONE PROJECT

GUIDELINES
FOR
FACULTY (COHORT OWNER), STUDENTS AND EXAMINERS

CAPSTONE PROJECT

How to design and deliver

The students of Polytechnic Programs will have an opportunity to be part of one of the most challenging educational experiences in the year-3. The students will be trained in the specialization pathways of their interest in fifth semester followed by an internship or a capstone project work in sixth semester. Those students who want to do a capstone project, requires to do developmental work on real-world problems which would motivate them to produce practical solutions. It is an opportunity for the students to use the problem-solving tools and techniques to solve the problems while doing the capstone project. With this approach, the learning process is gained through 'by-doing' experience and the students are expected to apply both the Capstone Project Management Skills and Technical Skills gained in previous years of polytechnic courses, which will enable them to participate and prepare for future employment. Working under the guidance of a Cohort owner, students may shape the direction of what they want to be, as well as gain better understanding of the responsibilities they need to shoulder when they undertake a capstone project. Teamwork will be inculcated with the development of good and professional relationships with their cohort owner and team members. The undertaken capstone project can also be used as a basis for employment or Startup by fully exploiting the learning process they have gone through, the skills they have gathered and the experience they have gained from the capstone project.

The guidelines are prepared for Cohort owner, students and examiners enabling them to execute their respective roles and responsibilities in an effective manner.

Aims of Capstone

1. Promote integration and synthesis within the program of study.
2. Promote meaningful connections between the program of study and career experiences.
3. Improve learner's career preparation and pre professional developments.
4. Demonstrate professional identity as learner's transition from academic to professional World.

Job Alignment and Professional Scenario

While developing a capstone the goal should always to;

1. Use a real world professional scenario- built out with employer engagement where ever possible.
2. Align skills to be assessed to a job.
3. Explicitly and intentionally developed important learner's skills, competencies and perspectives that are tacitly developed in the curriculum and required in the workplace.
4. Give learner's the freedom to showcase their learning though a demonstrable artifact or output e.g. Technical Product, System, Service that resolves a real world problem.

Employer Engagement

Support in capstone development:

- Provide a problem statement
- Provide a case study background
- Review and feedback on case studies/scenarios developed

Support in class

- Mentor learner's during the capstone
- Support cohort owners during class-workshop seminars

Presentation of Capstone

- Sit on presentation panel for learners to give feedback.

Outcomes

On successful completion of the capstone project, students will be able to:

- Write Capstone project scope document
- Prepare a capstone project execution plan
- Manage the capstone project from start to finish meeting stated milestones and timelines
- Test and validate the findings
- Demonstrate interpersonal skills, teamwork, and effective use of appropriate technology required for the capstone project

Responsibilities of the Head of the Department

The Head of the Department shall coordinate in Executing the Capstone projects, their responsibilities can be summarized as follows:

- To ensure that the Capstone project scope document is relevant to the specialization pathway opted by the students in Fifth semester
- To assign Cohort owner to the students
- To maintain a centralized capstone project hub repository to facilitate capstone project management and keeping track of all capstone projects and design changes

Responsibilities of the Faculty (Cohort owner):

Students will be supervised by Cohort owner; their responsibilities can be summarized as follows:

- To guide the students in writing the Capstone project scope document
- To guide the students in preparing capstone project execution plan
- To interact with the students once in a week to review the progress of the capstone project work, these sessions shall reinforce/review the concepts, findings and focus on addressing issues relevant to weekly meetings.

- To guide the students in managing the capstone project from start to finish, meeting the stated milestones and timelines
- To guide the students in preparing the capstone project report
- Develop appropriate Rubrics and evaluate the capstone project work as per assessment criteria
- To oversee the capstone project work until the submission of the final report, and Semester End Examination
- Maintain all the documents related to the capstone project work

Responsibilities of the Students

Students are also required to exercise self-discipline, self-management, job co-ordination, teamwork, and trustworthiness to ensure the success of the capstone project.

The expected responsibilities are:

- To write the Capstone project scope document
- To prepare a capstone project execution plan
- To adhere to the weekly meeting schedule with the cohort owner for the purpose of updating their progress and seeking advice on capstone project matters (Attendance is compulsory as per regulation) and submit weekly report
- To Manage the capstone project from start to finish meeting stated milestones and timelines
- To report immediately to the cohort owner any difficulties encountered that would interrupt the work.
- To submit all reports on time

Group Member Roles and Contributions

The Capstone project groups often function more effectively when group members have designated roles. Each capstone project group shall consist of not more than **four students**. The Three core roles and responsibilities are:

- **Capstone project Lead**: One student in the group shall act as a capstone project lead, who is responsible for keeping the group on task, distributing the workload, meeting deadlines, and ensuring smooth group communication and coordination as well as accountability with the cohort owner and capstone project requirements
- **Documenter Lead**: One student in the group shall act as a documenter lead, who is responsible for recording group discussions and decisions, documenting various aspects of the capstone project's progress, and ensuring well-formed reports and capstone project documents are produced.
- **Development Lead**: Two students in the group shall act as a Development lead, who are responsible for overseeing the collaborative aspects of the capstone project, troubleshooting major technical problems.

The entire capstone project team should be engaged in discussions, documentation, and development of the capstone project. All members are expected to contribute towards the capstone project.

Groups will a have to rotate the roles among members for different stages of the capstone project. This will allow members to gain experience through being responsible in different areas of capstone project management.

Assessment of the capstone project work

This section is addressed to the Faculty (Cohort owner) and examiners. It provides information on assessment criteria for the capstone project work. It also provides guidance to students about what examiners will be looking for in evaluating the capstone projects. The Capstone project work will be assessed for 100 marks through formative and summative assessment tools, in formative assessment the capstone project will be evaluated for 50 marks and in summative assessment capstone project will be evaluated for 50 marks

The Formative Assessment- (Continuous Internal Evaluation- CIE)

The Formative Assessment is conducted for 50 marks throughout the course in three developmental phases as CIE-I, CIE II and CIE-III. Students shall complete CIE-I before taking CIE-II and complete CIE-II before taking CIE-III, otherwise they will not be eligible to take Semester End Examination

Continuous Internal Evaluation- CIE - I conducted at the end of 4th week		
Sl No	**Assessment of parameter**	**Marks**
1	Writing the Capstone project scope document	10
2	**Capstone project Planning**: • Work Breakdown Structure (WBS) - 05 marks • Time-line Schedule - 10 marks • Cost Breakdown Structure (CBS) - 10 marks • Risk Analysis - 10 marks	35
3	Identification of Methodology (Including Literature survey)	05
	Total	**50**

Continuous Internal Evaluation- CIE - II conducted at the end of 8th week		
Sl No	**Assessment of parameter**	**Marks**
1	**Capstone project Details:** • Description of Technology Used • Details of Hardware devices • Details of software products • Programming languages • Descriptions of the components in the system • Component diagrams and required design if any • Construction or Fabrication details • Any other information needed to execute the capstone project	50
	Total	**50**

Continuous Internal Evaluation- CIE - III conducted at the end of 13th week		
Sl No	**Assessment of Parameter**	**Marks**
1	**Testing and validation**: Details of laboratory experiments/programming/modelling/simulations/ analysis/fabrication/construction etc.,	30
2	Results and inference	20
	Total	**50**

Note: Average Marks obtained in the above Three CIE's (CIE-I,CIE-II and CIE-III) shall be recorded as Formative assessment for 50 Marks.

The Summative assessment- Semester End Examination (SEE)

During the Summative assessment, students shall demonstrate the outcomes of their Capstone project work to the Panel of Examiners comprising a cohort owner and an external Subject expert

The evaluation criteria are as follows:

Sl No	Parameters	Marks
1	Power point presentation on outcomes of the Capstone project work	20
2	Demonstration the Capstone project work	20
3	Capstone project Report -Format and Technical writing skill	10
	Total	**50**

Plagiarism

Plagiarism is the act of obtaining or attempting to obtain credit for academic work by representing the work of another as one's own without the necessary and appropriate acknowledgment. If a student is in doubt of the nature of plagiarism, he/she should discuss the matter with the supervisor. If a student is caught committing plagiarism, disciplinary action will be taken against the student

Keeping in view the policy of plagiarism, and avoid piracy of intellectual property, the student needs to follow the citation policy:

- When 10 words are taken together from some established core work, citation becomes essential.
- When the copied content reaches 40 words in accumulation, the fragment needs to be kept under inverted comma ("_") in italic.
- It is necessarily required to cite reference in case of any content adopted from anywhere other than internet open sites. It is also that, even in case of open site internet source or any other source the copied contents if found more than 35 percent in aggregate during plagiarism detection, the work shall not be considered for further process and asked to resubmit the report again for the evaluation

Copyright

The Polytechnic institutions shall be the owner for all findings, designs, patents, and other intellectual property rights.

FORMATS

Department of Technical Education

Capstone project

Format- 1

Capstone project Scope Document

Capstone project Scope Document

The capstone project scope clearly describes what the capstone project will deliver and outlines all the work required for completing the capstone project.

Capstone project Title:

Group Members:

Problem Statement:

Objectives:

Capstone project description:

Capstone project Deliverables:

Key milestones:

Constraints:

Estimated Capstone project Duration:

Estimated Capstone project cost:

Date

Signature of the student

Signature of the cohort owner

Department of Technical Education

Capstone project

Format- 2

Work Breakdown Structure

Capstone project Name: <State the Title of the capstone project >

Capstone project Members: <List of group members>

Capstone project Objective(s): < statements describing the capstone project's objective(s)>

Work Breakdown Structure - Deliverables

1. Identify the deliverables (in the scope statement) to be produced in the capstone project.
   This highlights the work to be done.
2. Decompose each large deliverable into a hierarchy of smaller deliverables. This involves taking a deliverable and breaking it down into lower and lower levels of detail.
3. The lowest level of detail is called a 'work package' which consists of activities and tasks.

Date

Signature of the student

Signature of the cohort owner

Department of Technical Education

Capstone project

Format- 3
Time - line Schedule

Capstone project Name: <State the Title of the capstone project >

Capstone project Members: <List of group members>

1. Identify the activities and tasks needed to produce each work package.
2. Identify resources for each task (e.g., time, knowledge, monetary costs etc.)
3. Estimate how long it will take to complete each task. Consider constraints - resources, time, knowledge
4. Determine which tasks are dependent on other tasks and develop a critical path.
5. Develop a schedule of all activities and tasks - weekly and monthly. Work out when eachtask is scheduled to begin and end. Use a Gantt chart.

Date

Signature of the student

Signature of the cohort owner

Department of Technical Education

Capstone project

Format- 4
Cost Breakdown Structure

Capstone project Name: <State the Title of the capstone project >

Capstone project Members: <List of group members>

A cost breakdown structure (CBS) breaks down cost data into different categories, and helps you manage costs efficiently. It is a crucial part of the capstone project planning and management process, as it allows you to gain better insight into how much you spend and what you spend your capstone project budget on. When you have a solid structure in place, you can have better control of your capstone project costs to avoid going over budget.

1. **Analyze your Work Breakdown Structure**
   - Before you can identify your costs, you must first determine what your capstone project entails.
   - You can do this by looking at your work breakdown structure in detail, and work out the components that will contribute to the capstone project costs.
2. **Estimate the labor cost of work**
   - The next step is to estimate the labor cost of work for each task or activity you have identified in your WBS.
   - The time it takes for your team members to finish each work package in the WBS contribute to your labor costs.
   - Once you have estimated the labor costs of work for all the tasks, you can use them to work out the final cost of labor for your capstone project.
3. **Estimate the cost of materials**

The next step is to look at the cost of the materials needed to complete each task you identified in your WBS. These costs include

- Raw material costs
- Equipment and parts purchased for this capstone project
- Anything rented for the purpose of the capstone project

4. **Overhead costs.**
   - Ensure your CBS also includes an appropriate allocation to overhead costs.
   - Overhead include various costs that aren't related to specific tasks, but are necessary for the capstone project to take place.
5. **Build contingency into your CBS**
   - No matter how accurate your estimates are, you should still allow for some contingency in your cost breakdown structure in the CBS

6. Final-check

- The last step in creating a cost breakdown structure is to check your estimates against your available budget.
- If it your estimate is within the available budget, then you can be confident that the financial aspect of your capstone project will be smooth sailing
- If your CBS comes in higher than the available budget, you can look at ways to control costs.

Date

Signature of the student

Signature of the cohort owner

Department of Technical Education

Capstone project

Format- 5
Capstone project Execution Document

Capstone project Name: <State the Title of the capstone project >

Capstone project Members: <List of group members>

- Main Deliverables -

1) **Design:** descriptions of the components in the system, Component diagrams, and required design if any.
2) **Description of Technology Used:** provide details of Hardware devices, software products, programming languages etc.
3) **Fabrication**: fabrication or construction details
4) **Testing and validation**: provide the details of Methodologies/ laboratory experiments/ computer programming/ modelling/ simulations/ analysis/ findings etc
5) **Results and inference**

Date

Signature of the student

Signature of the cohort owner

Department of Technical Education
Capstone project

Format- 6
Weekly Meeting Record

<For Cohort Owner Use>

Capstone project Title:		
Group Members	1) 2) 3) 4)	□Present □Present □Present □Present
Date		
Meeting venue		□On Time
Documents Submitted	□ Status Report	□On Time
Issues Group Working on		
Assessment of Progress	□Excellent □Good □Satisfactory □Fair □Poor	
Notes/ Concerns/ Comments		

Signature of the Cohort owner

Department of Technical Education

Capstone project

Format- 7
Weekly Status Report

Capstone project Name: <State the Title of the capstone project >

Capstone project Members: <List of group members>

Status:

Briefly describe and illustrate the progress.

Highlights

List any items of note. Breakthroughs, accomplishments, major decisions, or changes in the capstone project plan Are you on schedule, ahead of schedule or behind schedule?

Risks or Issues List

In the following table, list any risk or issue that is critical for the success of the capstone project. This could be anything from "*we need to get data*" to "*how do we ensure that the system is usable*" to "*performance is unacceptable*". This should be a complete historical list that is kept from the beginning of the capstone project until the end.

Status should be one of *New*, *ongoing*, *Closed*.

The resolution column should be filled in if the issue or risk has been taken care of.

A capstone project may be expected to have around 1-3 active issues or risks that are being managed (new or ongoing) at any given time. If you have more than three, then either you have a capstone project in serious trouble or your criteria for what is "critical to success" is too loose.

Date	Risk or Issue	Description	Resolution	Status

Contd..

Tasks in Progress or Completed:

List the tasks that each member of the capstone project worked on up to the present time.

Task Name	Description	Team Member Responsible	Percentage Complete

Upcoming Tasks:

List the tasks that each capstone project member is planning to work on in the upcoming Task.

Task Name	Description	Team Member Responsible

Date:

Signature of the students

Department of Technical Education

Capstone project

Format- 8
Student's Daily Log Book

Capstone project Name: <State the Title of the capstone project >

Capstone project Members: <List of group members>

Day-1	Date:
Capstone project Name:	
Name of the student	
Name of the Cohort owner:	
Remarks of the Cohort owner:	
Record Main actives of the day (including observation, sketches, discussions, etc): Signature of the Cohort owner	

Note: Prepare a A4 size hard bound Student's Diary/ Daily Log book using this format with college and student details

Department of Technical Education

Capstone project

Format- 9

Capstone project Report Template

The contents of the capstone project report shall be arranged in the following order:

1. Cover Page
2. Inside Title Page
3. Certificate signed by the Cohort owner and HOD
4. Declaration signed by the Candidate
5. Acknowledgements
6. Executive Summary
7. Table of Contents
8. List of Figures
9. List of Tables
10. Abbreviations/ Notations/ Nomenclature
11. Text of the Report
    Chapter 1
    Chapter 2
    --
    --
    Chapter... n
12. References
13. Appendices
14. non-paper materials (if any)

The different Chapters in the capstone project report shall have the following content,

Chapter 1

- Introduction
- Scope of the capstone project

Chapter 2

Capstone project planning

- Work breakdown structure (WBS)
- Timeline Development – Schedule
- Cost Breakdown Structure (CBS)
- Capstone project Risks assessment

Requirements Specification

- Functional
- Non-functional (Quality attributes)

- User input
- Technical constraints

Design Specification

- Chosen System Design
- Discussion of Alternative Designs
- Detailed Description of Components/Subsystems
- Component 1- n

Chapter 3

Approach and Methodology

Discuss the Technology/Methodologies/use cases/ programming/ modelling/ simulations/ analysis/ process design/product design/ fabrication/etc used in the capstone project

Chapter 4

Test and validation

i. Test Plan
ii. Test Approach
iii. Features Tested
iv. Features not Tested
v. Findings
vi. inference

Describe what constitute capstone project success and why? Discuss the product/service tests that will confirm the capstone project succeeds in doing what it intended to do.

Chapter 5

Business Aspects

Discuss the novel aspects of this service or product. Address why a company or investors should invest money in this product or service**.**

- Briefly describe the market and economic outlook of the capstone project for the industry
- Highlight the novel features of the product/service.
- How does the product/service fit into the competitive landscape?
- Describe IP or Patent issues, if any?
- Who are the possible capstone projected clients/customers?

Financial Considerations

- Capstone project budget
- Cost capstone projections needed for either for profit/nonprofit options.

Conclusions and Recommendations

- Describe state of completion of capstone project.
- Future Work
- Outline how the capstone project may be extended

General Guidelines

Report Size - Report may contain maximum of about 100 pages including references and appendices.

Paper Size - Use A4 size paper

Paper Quality - White bond paper weighing 85 g/m2 or more should be used. Photographs or images with dense colors may be printed in single side on glossy paper.

Margins - A margin of 40 mm is to be provided on left and 30 mm on right sides, whereas top and bottom margins should be 30 mm. No print matter should appear in the margin except the page numbers. All page numbers should be centered inside the bottom margin, 20mm from the bottom edge of the paper.

Font - Times New Roman (TNR) 12-point font has to be used throughout the running text. The captions for tables and figures should have font size of 11 and foot notes should be set at font size 10. Font sizes for various levels of headings are given in the table below

CHAPTER 3

TITLE PAGE-CENTERED TNR 17-POINT BOLD ALL CAPS

3.1. Section Heading
Left aligned with number, TNR 17 points, bold and leading caps
3.1.1. Second level section heading
Left aligned with number, TNR 14 points, bold and sentence case**.**
3.1.1.1 Third level section heading
Left aligned with number, TNR 12 points, bold and sentence case.
Fourth-level section heading
Numbered subsections beyond third level are not recommended. However, fourth-level subsection headings may be included without numbering, TNR 12-point font, left aligned and italicized
Running text should be set in 12-point TNR and fully justified. First line of paragraph should have indentation of 15 mm**.**

Line Spacing - The line spacing in the main text should be 1.5, for quotations, figure captions, table captions, figure legends, footnotes, and references. The equations, tables, figures, and quotations Single line spacing should be given.

Table / Figure/equation Format-

Tables, figures, and equations shall be numbered chapter-wise. For example, second figure in Chapter 3 will be numbered Figure 3.2. The figure can be cited in the text as Figure 3.2, Tables shall be numbered similarly (Table 2 in Chapter 3 will be numbered Table 3.2) and shall be cited in the text as Table 3.2. Figure caption shall be located below the figure. Table number and caption shall be located above the table.

Listing of the References:

Referencing is a way to give credit to the writers from whom you have borrowed words and ideas. By citing the work of a particular scholar, you acknowledge and respect the intellectual property rights of that researcher. As a student or academic, you can draw on any of the millions of ideas, insights and arguments published by other writers, many of whom have spent years researching and writing. All you need to do is acknowledge their contribution to your assignment.

References are to be listed after last chapter. They are to be listed in alphabetical order and numbered. Within a reference the line spacing should be single. Each reference should be separated by one blank line. The reference number should be left aligned. The text of the reference should have an indentation of 10 mm. The reference format to be followed for journal articles, text books, conference proceedings etc. are given below.

Journals

1. Parkas, K. (2011). Feedback and optimal sensitivity: Model reference transformations, multiplicative semi norms, and approximate inverses. IEEE Transactions on Automatic Control, 26(2): 301–320.

Text books

1. Myers, D. G. (2007). Psychology (1st Canadian ed.). Worth: New York.

Conference proceedings

1. Payne, D.B. and Gunhold, H.G. (1986). Digital sundials and broadband technology, In Proc. IOOC-ECOC, 1986, pp. 557-998.

Reports

1. Milton, M and Robert, L. (2004). Atmospheric carbon emission through genetic algorithm, Environment and Technical Report No.3., Indian Meteorological Department., New Delhi

Online journals with a DOI (Digital Object Identifier)

1. Krebs, D.L. and Denton, K. (2006). Explanatory limitations of cognitive developmental approaches to morality. Psychological Review, 113(3): 672- 675. doi: 10.1037/0033-295X.113.3.672

Online journals without a DOI

1. Vicki, G.T., Thomae, M., Cullen, A. and Fernandez, H. (2007). Modeling the hydrological impact on Tropical Forests. Forest Ecology, 13(10): 122-132. Retrieved from http://www.uiowa.edu/~grpproc/crisp/crisp.html

Online books

1. Perfect, T.J. and Schwartz, B. L. (Eds.) (2002). Applied metacognition. Retrieved from http://www.questia.com/read/107598848 (--If DOI is available, use the DOI instead of a URL

Chapters from a book

1. Krebs, D.L. and Denton, K. (1997). Social illusions and self-deception: The evolution of biases in person perception. In J. A. Simpson and D. T. Kenrick (Eds.), Evolutionary social psychology (pp.21-48). Hillsdale, NJ: Erlbaum

Appendices

Include data tables, drawings, background calculations, specification lists for equipment used, details of experimental configuration, and other information needed for completeness,

Page Numbering

Page numbers for the prefacing materials (Inside title page, dedication, certificate, declaration, acknowledgements, executive summary, table of contents, etc.) of the report shall be in small

Roman numerals and should be centered at the bottom of the pages.

The numbering of the prefacing material starts from the Inside Title Page. However, the number is not printed on the Inside Title Page. Each new item of the prefacing materials listed above should start on a fresh paper on right page. If the content of the prefacing material exceeds one page, it has to be printed on both sides of the paper by starting from the right- side page. For example, if the item „Table of Contents" extends for 5 pages, it should be printed in fresh paper

on right side page with second page of the „Table of Contents‟ on the back of the paper and then continued. The page numbers of the prefacing material will be printed in small Roman numerals continuously counting blank pages also. However, the numbers are not printed on the blank pages

The body of the report starting from Chapter 1 should be paginated in Arabic numerals and should be centered at the bottom of the pages. The pagination should start with the first page of Chapter 1 and should continue throughout rest of the report. Each side of a sheet of paper should be counted as a separate page, even if the back side of a sheet of paper is blank. The odd numbered pages are always on the right and even-numbered pages are always on the left. If the end of a chapter is in odd page (right side page) the next chapter should start on odd page i.e., on a fresh paper, and should be numbered as odd only by counting the blank even page also. However, the page number is not printed on the blank pages.

Each of the items - Inside cover page, Certificate, Acknowledgements, executive summary, Table of Contents, List of Figures, List of Tables, Abbreviations, Notations, Nomenclature, each new Chapter, References, and each new Appendix should start on an odd page i.e., on the right side

Non-Paper Material

A report may contain non-paper material, such as specimen, CDs and DVDs, Pen drive if necessary. They have to be accommodated in a closed pocket in the back cover page of the report. The inclusion of non-paper materials must be indicated in the Table of Contents. All non-paper materials must have a label each clearly indicating the name of the candidate, student Register number and the date of submission**.**

Binding

Two hard bounded copies of the capstone project Report shall be submitted for evaluation; the cover page should be printed on sky blue card of 300 g/m2 or above**.** One copy is used for Semester End Examination and after the exam it should be maintained in the concerned Head of the department and another copy is maintained at cohort owner

Electronic Copy

An electronic version (PDF) of the capstone project report should be submitted to the cohort owner and Head of the department. The file name should contain title of the capstone project, student Register number and date of submission.

Government of Karnataka
Department of Technical Education
Board of Technical Examinations

C-25 SEE Theory Question Paper Pattern

Course Name: Course Code:
Time: 3 Hours Max. Marks :100

Instructions:

For Part-A questions, only the first written answers will be considered for evaluation.

PART A

I. Select the correct answer from the choices given: 15X1 = 15 Marks

1. Multiple Choice Question-1
   A.
   B.
   C.
   D.
2. Multiple Choice Question-2
   A.
   B.
   C.
   D.
3. Multiple Choice Question-3
   A.
   B.
   C.
   D.
4. Multiple Choice Question-4
   A.
   B.
   C.
   D.
5. 5. Multiple Choice Question-5

   A.

   B.

   C.

   D.

6. Multiple Choice Question-6

   A.

   B.

   C.

   D.

7. Multiple Choice Question-7
    A.
    B.
    C.
    D.
8. Multiple Choice Question-8
    A.
    B.
    C.
    D.
9. Multiple Choice Question-9
    A.
    B.
    C.
    D.
10. Multiple Choice Question-10
    A.
    B.
    C.
    D.
11. Multiple Choice Question-11
    A.
    B.
    C.
    D.
12. Multiple Choice Question-12
    A.
    B.
    C.
    D.
13. Multiple Choice Question-13
    A.
    B.
    C.
    D.

14. Multiple Choice Question-14

    A.

    B.

    C.

    D.

15. Multiple Choice Question-15

    A.

    B.

    C.

    D.

II. Fill in the blanks by choosing appropriate answer from those given in the bracket: (Answer-1, Answer-2, Answer-3, Answer-4, Answer-5) 5X1 = 05 Marks

1. Question-1
2. Question-2
3. Question-3
4. Question-4
5. Question-5

PART B

III. Answer any FIVE questions: 5X2 = 10 Marks

1. Question-1
2. Question-2
3. Question-3
4. Question-4
5. Question-5
6. Question-6.
7. Question-7
8. Question-8

PART C

IV. Answer any FIVE questions: 5X3 = 15 Marks

1. Question-1
2. Question-2
3. Question-3
4. Question-4

5. Question-5
6. Question-6
7. Question-7
8. Question-8

PART D (Section I)

V. Answer any FIVE questions: 5X5 = 25 Marks

1. Question-1
2. Question-2
3. Question-3
4. Question-4
5. Question-5
6. Question-6
7. Question-7
8. Question-8

PART D (Section II)

VI. Answer any THREE questions: 10X3 = 30 Marks

1. Question-1
2. Question-2
3. Question-3
4. Question-4
5. Question-5