

JSS MAHAVIDYAPEETHA
JSS COLLEGE OF ARTS, COMMERCE AND SCIENCE
(Autonomous)
Ooty Road, Mysuru-570025



Bachelor of Science in Computer Science
(BSc. -PMCs)
Modified CBCS Scheme Syllabus

for
2024-25, 2025-26 & 2026-27 on words
(Revised)

{BoS Meeting Held on 22 - 08 - 2026}
DEPARTMENT OF COMPUTER SCIENCE

The objectives of the B.Sc. - Computer Science Program

1. The primary objective of this program is to provide a foundation of computing
2. It helps students analyze the requirements for system programming and exposes
3. This programme provides students with options to specialize in various software
4. To produce outstanding Computer Scientists who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves.
5. To provide an opportunity for the study of modern methods of information processing and its applications.
6. To develop among students programming techniques and problem-solving skills through programming
7. To prepare students who wish to go on to further studies in computer science and related subjects.
8. To acquaint students working effectively with a range of current, standard Office Productivity software applications.

Program Outcomes: BSc. (Computer Science) Degree

1. **Discipline knowledge:** Acquiring knowledge of the basics of Computer Science and the ability to apply design principles in the development of solutions for problems of varying complexity.
2. **Problem-Solving:** Improved reasoning with a strong mathematical ability to identify, formulate, and analyze problems related to computer science and exhibiting a sound knowledge of data structures and algorithms.
3. **Design and Development of Solutions:** Ability to design and development of algorithmic solutions to real-world problems.
4. **Programming a computer:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day scientific applications.
5. **Application Systems Knowledge:** Possessing a minimum knowledge to practice existing computer application software.
6. **Communication:** Must have reasonably good communication knowledge, both orally and in writing.
7. **Ethics on Profession, Environment, and Society:** Exhibiting professional ethics to maintain integrity in a working environment and also having concern for societal impacts due to computer-based solutions for problems.
8. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.
9. **Motivation to take up Higher Studies:** Inspiration to continue education towards advanced studies in Computer Science.

Program Structures for the Under-Graduate Program Bachelor of Science in Computer Science for 2025-26 onwards						
Semester	Coerces	Type	Credits	L: T: P	Marks	
I	Major 1to 3	Theory	09	3: 0: 0	300	
	Major 1 to 3	Practical	06	0: 0: 2	150	
	Language 1	Theory	03	3: 0: 0	100	
	Language 2	Theory	03	3: 0: 0	100	
	Constitutional Values	Theory	02	2: 0: 0	50	
II	Major 1to 3	Theory	09	3: 0: 0	300	
	Major 1 to 3	Practical	06	0: 0: 2	150	
	Language 1	Theory	03	3: 0: 0	100	
	Language 2	Theory	03	3: 0: 0	100	
	Constitutional Values	Theory	02	2: 0: 0	50	
III	Major 1to 3	Theory	09	3: 0: 0	300	
	Major 1 to 3	Practical	06	0: 0: 2	150	
	Language 1	Theory	03	3: 0: 0	100	
	Language 2	Theory	03	3: 0: 0	100	
	Elective 1	Theory	03	2: 0: 0	50	
	Elective 2	Theory	03	2: 0: 0	50	
IV	Major 1to 3	Theory	09	3: 0: 0	300	
	Major 1 to 3	Practical	06	0: 0: 2	150	
	Language 1	Theory	03	3: 0: 0	100	
	Language 2	Theory	03	3: 0: 0	100	
	Elective 3	Theory	03	2: 0: 0	50	
	Compulsory Paper	Practical/Skill	02	0: 0: 2	50	
for 2024-25 batch only						
Semester	Coerces	Type	Credits	L: T: P	Marks	
V	Major 1to 6	Theory	18	3: 0: 0	600	
	Major 1 to 3	Practical	06	0: 0: 2	150	
	Compulsory Paper	Practical/Skill	02	0: 0: 2	50	
VI	Major 1to 5	Theory	15	3: 0: 0	500	
	Major 1 to 3	Practical	06	0: 0: 2	150	
	Compulsory Paper	Practical/Skill	02	0: 0: 2	50	
Note: Only one Elective and one Compulsory Papers are offering (in any Semesters)						

Modified CBCS Syllabus – BSc Computer Science, for 2024-25 onwards

Year	Sem	Course Code	Title	Credits			Hours / Week			Maximum Marks						Exam Duration	Total Marks
				L	T	P	L	T	P	Th. IA		Pr. IA		Exam			
										C1	C2	C1	C2	Th.	Pr.		
I	I	GCS 103 (Theory)	Computer Concepts and Problem-Solving using C++	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 102 (Practical)	Computer Basics and Programming in C++	0	0	2	0	0	4	-	-	05	05	-	40	3 Hours	50
	II	GCS 203 (Theory)	Data Structures	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 202 (Practical)	Data Structures using C++	0	0	2	0	0	4	-	-	05	05	-	40	3 Hours	50
II	III	GCS 301 (Theory)	Database Management System	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 302 (Practical)	DBMS Lab	0	0	2	0	0	4	-	-	05	05	-	40	3 Hours	50
	IV	GCS 401 (Theory)	Object Oriented Programming with Java	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 402 (Practical)	Java Lab	0	0	2	0	0	4	-	-	05	05	-	40	3 Hours	50
		GCS *** (Theory)	*Elective:	2	0	0	2	0	0	05	05	-	-	40	-	2½ Hours	50
III	V	GCS 501 (Theory)	Operating Systems	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 503 (Theory)	Web Technology	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 504 (Practical)	Web Technology Lab	0	0	2	0	0	4	-	-	05	05	-	40	3 Hours	50
	VI	GCS 601 (Theory)	Computer Networks	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 603 (Theory)	Python Programming	3	0	0	3	0	0	10	10	-	-	80	-	3 Hours	100
		GCS 604 (Practical)	Python Programming Lab	0	0	2	0	0	4	-	-	05	05	-	40	3 Hours	50
		GCS*** (Practical)	#Compulsory Paper	0	0	2	0	0	4	-	-	05	05	40	-	3 Hours	50
Note: all (Practical) Courses experiments are Example list/templets it may modified & approval from the Department Council Meeting																	
*Elective: Software Engineering (2Cr. 413)/ E-Commerce & E-Governance / Software Testing (3Cr. 411)																	
#Compulsory Paper: Digital Marketing / AI Tools and its applications / Power BI (612)																	

Modified CBCS Course Content for BSc. in Computer Science

Semester: I

Course Code: GCS103	Course Title: Computer Concepts and Problem-Solving Using C++
Course Credits: 03 (3-0-0)	Hours/Week: 03
Total Contact Hours: 44	Formative Assessment Marks: 20
Exam Marks: 80	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Understand the basics of computers.

CO 2: Understand problem-solving strategies and techniques.

CO 3: Describe the Object-Oriented Programming principles and concepts.

CO 4: Understanding of the syntax and semantics of C++.

Unit – 1	Course Contents	Hours
	Introduction To Computers: Definition And Characteristics of Computers, Brief History of Computers, Generation of a Computers, Components of a Computer, Classification of Computers Based on Size and Processing Ability. Applications Of Computers. Computer Architecture: CPU, ALU, Control Unit, Registers, Cache Memory, RAM, ROM, Input/Output Components, Buses, Ports. Hardware And Software: Types of Software - System Software [Operating System, Interpreters], Application Software and Utility Programs. Languages of Computer, ASCII And EBCDIC, Introduction to Number System, Conversion from One System to another Problem-Solving Techniques: Problem, Definition, Analysis, Design Tools [Algorithm & Flow Chart], Coding, Testing, Maintenance. Basics Of Algorithm. Basic Programming Concepts: Tokens of Programming Language, Identifiers, Constants, Variables, Data Types, And Operators. Introduction To Programming Concepts.	11
Unit-2		
	Introduction To C++: Overview Of Programming Languages, History and Features of C++, Structure of C++ Program, Data Types in C++. Control Structures: Decision Making: If, If-Else, Nested If, Switch Statement. Looping: While, Do-While, For. Unconditional Statements: Break And Continue Statements.	11

Unit-3	
Arrays: Introduction, 1D-2D Declaration, Initialization, Accessing Elements, Use of arrays in solving problems Functions: Introduction, Advantage, Classification Based on-Origin, Arguments & Return Value, Scope Pointers: Introduction to Pointers, variable through variable Pointers In C++. Introduction To OOPS with C++: Objects, Classes, Data Encapsulation, Data Abstraction, data members, member functions, Friend Function, Constructors and Destructors.	11
Unit-4	
Access Specifiers: Public, Private, Protected. Polymorphism: Function Overloading, Operator Overloading, Virtual Functions. Inheritance: Single, Multiple, Multilevel, Hierarchical	11

Text Book:

1. "Object-Oriented Programming With C++" by Somashekara, M. T., Guru, D. S., Nagendra swamy, H. S., Manjunatha, K. S. Edin. Second Edition

Reference:

2. "Programming: Principles and Practice Using C++" by Bjarne Stroustrup.
3. "C++ Primer" by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo.
4. "Object-Oriented Programming with C++" by E. Balagurusamy
5. "Problem Solving and Object-Oriented Programming with C++" by Tanenbaum and Mukherjee:

CO-PO Matrix

PO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	--	-	-	-	-	-
CO2	2	2	2	-	-	-	-	-	-
CO3	2	-	-	2	-	-	-	-	-
CO4	2	-	-	2	-	-	-	-	-

CO-Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓			
II		✓		
III			✓	
IV				✓

Course Code: GCS102	Course Title: Computer Basics and Programming in C++
Course Credits: 02 (0-0-2)	Hours/Week: 04
Total Contact Hours: 60	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Illustrate the hardware components of a computer.

CO 2: Use Open-Source Office tools.

CO 3: Demonstrate simple programming skills through the C++ programming language.

Laboratory Program List

Part A

1. Demonstration of Desktop computer Hardware components.
2. Demonstration of Word Processor software [Open-Office].
3. Demonstration of Spreadsheet software [Open-Office].
4. Demonstration of Presentation software [Open-Office].
5. Write a C++ program to print the user's name.
6. Write a C++ program to swap 2 numbers.
7. Write a C++ Program to Find the Largest Among 3 Numbers
8. Write a C++ program to perform arithmetic operations.

Part B

9. Write a C++ Program to Print the Multiplication Table of a Number
10. Write a C++ Program to Reverse a Number
11. Write a C++ Program to Check Whether a Number is a Palindrome or Not.
12. Write a C++ Program to Create a Class and an Object
13. Write a C++ program for single inheritance.
14. Write a C++ program to demonstrate multi-level inheritance.
15. Write a C++ program to demonstrate operator overloading.
16. Write a C++ Program by using Pointers

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	-	-	-	3	-	-	-	-
CO2	-	-	-	-	3	-	-	3	-
CO3	3	-	-	3	-	-	-	-	-

Semester: II

Course Code: GCS203	Course Title: Data Structures
Course Credits: 03 (3-0-0)	Hours/Week: 03
Total Contact Hours: 44	Formative Assessment Marks: 20
Exam Marks: 80	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Understand the basics of Data Structures.

CO 2: Identify the appropriate data structures and algorithms for solving real-world problems.

CO 3: Understand the practical applications of Trees and Graphs.

Course Contents	Hours
Unit – 1	
Introduction: Data Structure Definition, Basic Terminology and Concepts, Importance of Data Structures in Programming. Classification of Data Structures. Primitive Data Structures, Non-Primitive Data Structures. C++ Strings, String Handling Functions, Applications of Arrays and Strings. Stack: Definition, Memory Representation, Algorithms for Stack Operations (Push, Pop), Applications of Stack-Recursion. Expression Conversion- Infix, Prefix, Postfix, Evaluation of Prefix, Postfix	11
Unit-2	
Searching Techniques: Linear And Binary Search Sort [Algorithm, Time & Space Complexity]. Sorting Techniques: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort [Algorithm, Time & Space Complexity]. Queue: Definition, Memory Representation & Operations of Queues: Linear Queue, Circular Queue. Applications Of Queue.	11
Unit-3	
Dynamic Memory Allocation- Malloc, Calloc & Realloc. Linked Lists: Definition, Types. Singly Linked List: Memory Representation of Singly Linked List, Implementation - Insertion, Deletion [At the Beginning, at the End]. Doubly Linked List: Memory Representation of Doubly Linked List. Implementation - Insertion, Deletion [At the Beginning, at the End]. Applications of Linked List. Tree: Definition, Terminologies, Memory Representation Using Array and Linked List.	11

Unit-4	
Binary Tree: Definition, Traversal Algorithms [Pre-Order, In-Order, Post-Order], Construction of Tree from In-Order and Pre-Order, In-Order and Post-Order. Binary Search Trees: Insertion of a Node, Deletion of a Node. Heap: Heap Operations and Applications. Graph: Definition, Memory Representation of Graph. Adjacency Matrix, Adjacency List. Graph Traversal Algorithms: Breadth-First Search (BFS), Depth-First Search (DFS).	11

Reference Books:

1. Data Structures Through C++ (4th Edition) Yashvant Kanetkar.
2. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss.
3. Data Structure and Algorithms using C++ by Sachi Nandan Mohanty, Pabitra Kumar Tripathy.
4. Data Structures and Algorithms in C++, Second Edition by Adam Drozdek.

CO–PO Matrix

PO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	2	-	-	-	-	-	-	-
CO2		3	3	-	-	-	-	-	-
CO3	2	2	-	-	-	-	-	-	-

CO–Unit Mapping

Unit	CO1	CO2	CO3
I	✓		
II		✓	
III			✓
IV			✓

Course Code: GCS202	Course Title: Data Structures using C++
Course Credits: 02 (0-0-2)	Hours/Week: 04
Total Contact Hours: 60	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Implement data structures using C++.

CO 2: Demonstrate searching and sorting techniques using ++.

CO 3: Demonstrate simple programming skills through the C++ programming language.

Laboratory Program List

Part -A

1. C++ program to find GCD of two numbers.
2. C++ program to implement Tower of Hanoi.
3. C++ program to print Fibonacci series.
4. C++ program to find largest and smallest element in an array.
5. C++ program to perform stack operations.
6. C++ program to perform Linear queue operations
7. C++ program to insert a node at the beginning of a singly linked list.
8. C++ program to delete a node at the end of a singly linked list.

Part B:

1. C++ program to construct a binary search tree
2. C++ program for Binary Tree traversal.
3. C++ program to implement DFS
4. C++ program to implement BFS
5. C++ Program to Sort an Array (Selection Sort)
6. C++ Program to Sort an Array (Bubble Sort)
7. C++ Program to perform Linear Search of an Element in an Array.
8. C++ Program to perform Binary Search of an Element in an Array.

CO-PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	-	-	3	-	-	-	-	3	3	-
CO2	-	3	3	3	-	-	-	-	-	-	3
CO3	3	-	-	3	-	-	-	-	3	3	-

Semester: III

Course Code: GCA301	Course Title: Database Management System
Course Credits: 03 (3-0-0)	Hours/Week: 03
Total Contact Hours: 44	Formative Assessment Marks: 20
Exam Marks: 80	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Understand basic concepts of databases, data models, and ER diagrams.

CO 2: Apply relational algebra and SQL to query and manipulate data.

CO 3: Analyze database normalization to improve design and remove anomalies.

CO 4: Evaluate transaction management and database security features.

Content	Hours
Unit-1	
Introduction to Databases: Definition of Data, Database, and DBMS, Overview of Database, Applications, Advantages and Disadvantages of DBMS, Roles of Database Users and Administrators. Data Models: Introduction to Data Models, Types of Data Models (Hierarchical, Network, Relational, Object-oriented), Importance of Data Models in DBMS. Database Design: Keys: Primary Key, Candidate Key, Super Key, Foreign Key, Composite Key, Alternate Key, Unique Key, Surrogate Key, Constraints in a table: Primary Key, Foreign Key, Unique Key, NOT NULL, CHECK. Entity-Relationship (ER) Model, Entities and Entity Sets, Attributes and Relationships, ER Diagrams, Key Constraints and Weak Entity Sets, Extended ER Features, Introduction to the Relational Model and Relational Schema.	11
Unit-2	
Relational Algebra: Introduction to Relational Algebra, Operations: Selection, Projection, Set Operations, Join Operations, Division. Structured Query Language (SQL): SQL Basics: DDL and DML, Aggregate Functions (Min (), Max(), Sum(), Avg(), Count ()), Logical operators (AND, OR, NOT), Predicates (Like, Between, Alias, Distinct), Clauses (Group By, Having, Order by, top/limit).	11
Unit-3	
SQL Joins and Views: Inner Join, Natural Join, Full Outer Join, Left Outer Join, right outer Join, Equi Join, Definition of View, creating a View, Managing Views (Listing, Updating, Deleting). Normalization: Anomalies in relational database design. Functional dependencies - Axioms. Decomposition, Transitive Dependency. Data Normalizations: First normal form, Second normal form, Third normal form. Boyce-Codd normal form.	11

Unit-4	
Query Processing Transaction Management: Introduction Transaction Processing, Single user & multiuser systems. Transactions: read & write operations. Need of concurrency control: The lost update problem, Dirty read problem. Types of failures. Transaction states. Desirable properties (ACID properties) of Transactions. Storage of Database, File Operations, and Database Security.	11

Text Books:

1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7th Edition, Pearson.

Reference Books:

1. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
2. Introduction to Database System, C J Date, Pearson, 1999.
3. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6th Edition, McGraw Hill,
4. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition, McGraw Hill.

CO-PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	-	-	-	-	-	-	-	2	2	-
CO2	3	-	-	3	-	-	-	-	3	3	-
CO3	3	3	-	-	-	-	3	-	-	3	3
CO4	4	-	-	-	-	4	4	-	-	4	-

CO-Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓			
II		✓		
III			✓	
IV				✓

Course Code: GCA302	Course Title: DBMS Lab
Course Credits: 02 (0-0-2)	Hours/Week: 04
Total Contact Hours: 60	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Execute single-line SQL queries and apply group functions effectively.

CO 2: Perform database operations using DDL, DML, DCL, and TCL commands.

CO 3: Implement advanced SQL concepts like nested queries and join operations.

CO 4: Create views and apply table-level locking mechanisms for data control.

Laboratory Program List

Part A

Activity 1: Database: Student (DDL, DML Statements)

Table: Student

Name	Reg. No	Class	Major
Smith	17	1	CS
Brown	8	2	CS

Table: Course

Course Name	Course Number	Credit Hours	Department
Introduction to Computer Science	CS1310	4	CS
Data Structure	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database Management System	CS3380	3	CS

Table: Section

Section Identifier	Course Number	Year	Instructor
85	MATH2410	98	King
92	CS1310	98	Andreson
102	CS3320	99	Knuth
112	MATH2410	99	Chang
119	CS1310	99	Andreson
135	CS3380	99	Stone

Table: Grade_Report

Reg. No	Section Identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

Queries

1. Create Table Using create statement.
2. Insert rows into individual Tables using insert statement.
3. Alter table section add new field section and update the records
4. Delete brown's grade report.
5. Drop the table section.

Activity 2: (Select clause, Arithmetic Operators)

Database: Employee

Create the following tables and insert tuples with suitable constraints.

Table: EMPLOYEE

EMPID	FIRSTNAME	LASTNAME	Hire Date	ADDRESS	CITY
1001	George	Smith	11-May-06	83 First Street	Paris
1002	Mary	Jones	25-Feb-08	842 Vine Ave	Losantiville
1012	Sam	Tones	12-Sep-05	33 Elm St.	Paris
1015	Peter	Thompson	19-Dec-06	11 Red Road	Paris
1016	Sarath	Sharma	22-Aug-07	440 MG Road	New Delhi
1020	Monika	Gupta	07-Jun-08	9 Bandra	Mumbai

Table: EMPSALARY

EMPID	SALARY	BENEFITS	DESIGNATION
1001	10000	3000	Manager
002	8000	1200	Salesman
1012	20000	5000	Director
1015	6500	1300	Clerk
1016	6000	1000	Clerk
1020	8000	1200	Salesman

Queries

1. Display FIRSTNAME, LASTNAME, ADDRESS, and CITY of all employees living in PARIS
2. Display the content of the employee table in descending order of FIRSTNAME
Select FIRSTNAME and SALARY of salesmen
3. Display the FIRSTNAME, LASTNAME, and TOTAL SALARY of all employees where
 $TOTAL\ SALARY = SALARY + BENEFITS$
4. Count the number of distinct DESIGNATIONs from EMPLOYEE
5. List the employees whose names have exactly 6 characters
6. Add a new column PHONE_NO to the EMPLOYEE table and update the records
7. List employee names who have joined before 15-Jun-08 and after 16-Jun-07
8. Generate salary slip with Name, Salary, Benefits, HRA=50%, DA=30%, PF=12%, and calculate gross salary. Order by gross salary in descending order.

Activity 3: (Logical, Relational Operators)**Database:** LibraryCreate the following tables and insert tuples with suitable constraints. **Table:****Books**

Book_Id	Book_Name	Author_Name	Publishers	Price	Type	Quantity
C0001	The Klone and I	Lata Kappor	EPP	355	Novel	5
F0001	The Tears	William Hopkins	First Publ	650	Fiction	20
T0001	My First C++	Brain & Brooke	First Publ	350	Text	10
T0002	C++ Brainwork's	A.W. Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Ana Roberts	First Publ	750	Fiction	50

Table: Issued

Book_Id	Quantity_Issued
T0001	4
C0001	5
F0001	2
T0002	5
F0002	8

Queries

1. To show the Book name, Author name and price of books of First Publ. publisher.
2. Display Book id, Book name and publisher of books having quantity more than 8 and price less than 500.
3. Select Book id, book name, author name of books which is published by other than ERP publishers and price between 300 to 700.
4. Generate a Bill with Book_id, Book_name, Publisher, Price, Quantity, 4% of VAT "Total".
5. Display book details with book id's C0001, F0001, T0002, F0002 (Hint: use IN operator).
6. Display Book list other than, type Novel and Fiction.
7. Display book details with author name starts with letter "A".

8. Display book details with author name starts with letter "T" and ends with "S".
9. Select Book_Id, Book_Name, Author Name, Quantity Issued where Books.Books_Id = Issued.Book_Id.
10. List the book_name, Author_name, Price. In ascending order of Book_name and then on descending order of price.

Activity 4: (Date Functions)**Database: Lab**

Create the following table and insert tuples with suitable constraints.

Table: Equipment Details

No.	Item Name	Cost Per Item	Quantity	Date of Purchase	Warranty	Operational
1	Computer	30000	9	21/5/07	2	7
2	Printer	5000	3	21/5/06	4	2
3	Scanner	8000	1	29/8/08	3	1
4	Camera	7000	2	13/6/05	1	2
5	UPS	15000	5	21/5/08	1	4
6	Hub	8000	1	31/10/08	2	1
7	Plotter	25000	2	11/1/09	2	2

Queries

1. To select the ItemName purchase before 31/10/07.
2. Extend the warranty of each item by 6 months.
3. Display ItemName, Dateof purchase and number of months between purchase date and present date.
4. To list the ItemName in ascending order of the date of purchase where quantity is more than 3.
5. To count the number, average of costperitem of items purchased before 1/1/08.
6. To display the minimum warranty, maximum warranty period.
7. To Display the day of the date, month, and year of purchase in characters.
8. To round off the warranty period to a month and year format.
9. To display the next Sunday from the date "07-JUN-96".
10. To list the ItemName, which are within the warranty period till the present date.

Part B:**Activity 5: (Numeric, character functions).****Use Functions for the following.**

1. Find the mod of 165,16.
2. Find Square Root of 5000.
3. Truncate the value 128.3285 to 2 and -1 decimal places
4. Round the value 92.7683 to 2 and -1 decimal places.
5. Convert the string 'Department' to uppercase and lowercase.
6. Display your address convert the first character of each word to uppercase and rest are in lowercase.
7. Combine your first name and last name under the title Full name.
8. Take a string length maximum of 15 displays your name to the left. The remaining space should be filled with '*'.

9. Take a string length maximum of 20 displays your name to the right. The remaining space should be filled with '#'.
10. Find the length of the string 'JSS College, Mysore'.
11. Display substring 'BASE' from 'DATABASE'.
12. Display the position of the first occurrence of character 'o' in Position and Length.
13. Replace string Database with Data type.
14. Display the ASCII value of ' ' (Space).
15. Display the Character equivalent of 42.

Activity 6: Database: subject

Create the following table and insert tuples with suitable constraints.

Table: Physics

Regno	Name	Year	Combination
AJ00325	Ashwin	First	PCM
AJ00225	Swaroop	Second	PMCs
AJ00385	Sarika	Third	PME
AJ00388	Hamsa	First	PMCs

Table: Computer Science

Regno	Name	Year	Combination
AJ00225	Swaroop	Second	PMCs
AJ00296	Tejas	Second	BCA
AJ00112	Geetha	First	BCA
AJ00388	Hamsa	First	PMCs

Queries

1. Select all students from Physics and Computer Science.
2. Select student common in Physics and Computer Science.
3. Display all student details who are studying in the second year.
4. Display students who are studying both Physics and Computer Science in the second year.
5. Display the students studying only Physics.
6. Display the students studying only Computer Science.
7. Select all students having PMCs combination.
8. Select all students having BCA combination.
9. Select all students studying in the third year.
10. Rename table Computer Science to CS.

Activity 7: (views)**Database:** Railway Reservation System.

Create the following table and insert tuples with suitable constraints.

Table: Train Details

Train_No	Train_Name	Start_Place	Destination
RJD16	Rajdhani Express	Bangalore	Mumbai
UDE04	Udhyan Express	Chennai	Hyderabad
KKE55	Karnataka Express	Bangalore	Chennai
CSE3	Shivaji Express	Coimbatore	Bangalore
JNS8	Janashatabdi	Bangalore	Salem

Table: Availability

Train_No	Class	Start_Place	Destination	No_of_seats
RJD16	Sleeper Class	Bangalore	Mumbai	15
UDE04	First Class	Chennai	Hyderabad	22
KKE55	First Class AC	Bangalore	Chennai	15
CSE3	Second Class	Coimbatore	Bangalore	8
JNS8	Sleeper Class	Bangalore	Salem	18

Queries

1. Create a view **sleeper** to display train number, start place, destination, which have sleeper class and perform the following: a. Insert a new record.
2. Update destination='Mangalore' where train no='RJD16'.
3. Delete a record which has train no = ' KKE55'
4. Create a view **detail** to display the train number, train name, and class.
5. Create a view **total seat** to display train number, start place, use COUNT function on No_of_seats, group by start place and perform the following: a. Insert a new record.
6. Update start place='Hubli' where train no='JNS8'.
7. Delete the last row of the view.
8. Rename view **sleeper** to **class**.
9. Delete view **details**.

Activity 8: (group by, having clause)

Database: Bank system

Create the following table and insert tuples with suitable constraints.

Table: Account

Account_No	Cust_Name	Branch_ID
AE0012856	Reena	SB002
AE1185698	Akhil	SB001
AE1203996	Daniel	SB004
AE1225889	Roy	SB002
AE8532166	Sowparnika	SB003
AE8552266	Anil	SB003
AE1003996	Saathwik	SB004
AE1100996	Swarna	SB002

Table: Depositor

Account_No	Branch_ID	Balance
AE0012856	SB002	12000
AE1203996	SB004	58900
AE8532166	SB003	40000
AE1225889	SB002	150000

Table: Branch

Branch_ID	Branch_Name	Branch_City
SB001	Malleshwaram	Bangalore
SB002	MG Road	Bangalore
SB003	MG Road	Mysore
SB004	Jayanagar	Mysore

Table: Loan

Account_No	Branch_ID	Balance
AE1185698	SB001	102000
AE8552266	SB003	40000
AE1003996	SB004	15000
AE1100996	SB002	100000

Queries

1. Display the total number of accounts present in each branch.
2. Display the total loan amount in each branch.
3. Display the total deposited amount in each branch in descending order.
4. Display the maximum and minimum loan amount present in each city.
5. Display the average amount deposited in each branch for each city.
6. Display the maximum loan amount in each branch where the balance is more than 25000.
7. Display the total number of accounts present in each city.
8. Display all customer details in ascending order of branch ID.
9. Update the balance to 26000 where Account_No = AE1003996.
10. Display customer names with their branch name.

CO-PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	-	-	3	-	-	-	-	-
CO2	3	3	-	3	-	-	-	-	-
CO3	3	3	-	3	-	3	-	-	-
CO4	-	-	3	-	-	3	-	-	-

Semester IV

Course Code: GCS401	Course Title: Object Oriented Programming with Java
Course Credits: 03 (3-0-0)	Hours/Week: 03
Total Contact Hours: 44	Formative Assessment Marks: 20
Exam Marks: 80	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Understand OOP concepts and basic Java programming structure.

CO 2: Apply classes, objects, constructors, arrays, and strings in Java.

CO 3: Implement multithreading and exception handling in Java programs.

CO 4: Perform file I/O and develop GUI applications using AWT and event handling.

Course Contents	Hours
Unit 1	
Fundamentals of Object-oriented Programming: Object-oriented Paradigm, Basic Principles of Object-oriented Programming, Advantages of Object-Oriented Programming, Applications of Object-Oriented Programming. Introduction to Java Language: Java History, Features, Overview, Difference between C, C++ and Java, Java Environment- JDK, JVM, JRE and API, Java Program Structure, Java Tokens, Implementing a Java Program, Command Line Arguments. Java Programming Fundamentals: Data types, Variables & Constants, Keywords & Naming Conventions, Type Casting, Operators and Expressions, Control Structures, Jumping Statements.	11
Unit 2	
Classes & Objects: Basics of Objects and Classes, Constructors, Access Modifiers, Method Overloading, Overloading Constructors, Static members, this keyword. Arrays: One-dimensional Arrays, Two-dimensional Arrays, Array of Objects. Strings: String Handling functions.	11
Unit-3	
Multithreading in Java: Concepts of Thread, Thread Life Cycle, Creating Threads & Implementing Runnable Interface, Thread Synchronization & Thread Priority. Exception Handling: Concepts of Exception, Different Types of Exceptions, Creating User-Defined Exceptions Using Try-Catch-Finally-Throw Blocks, Nested Try, Catch, Throw, and Throws Blocks.	11

Unit-4	
File Handling: I/O Handling, I/O Streams, Types of Files, Byte Stream, Binary I/O Classes & Its Hierarchy, FileInputStream & FileOutputStream Classes, Object I/O Classes. Event Handling & GUI programming: Event Handling, Event Types, Event Handling Mechanism, Keyboard & Mouse Handling, Introduction to AWT & GUI basics, AWT hierarchy of classes, AWT controls – Frames, Panels, Layout managers & other controls of AWT.	11

Text Books:

1. D.S. Guru, M.T. Somashekara, & K.S. Manjunatha, Object Oriented Programming with Java, PHI Learning, 2017.
2. E Balagurusamy, Programming with JAVA, TMH, 2007

Reference Books:

3. Herbert Schildt, Java 7, The Complete Reference, , 8th Edition, 2009

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	2	-	-	-	-	-
CO2	3	-	-	3	-	3	-	-	-
CO3	3	-	-	3	-	-	-	3	-
CO4	4	-	-	4	-	3	3	-	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓			
II		✓		
III			✓	
IV				✓

Course Code: GCS402	Course Title: Java Lab
Course Credits: 02 (0-0-2)	Hours/Week: 04
Total Contact Hours: 60	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Write Java programs using basic logic and structures.

CO 2: Use OOP concepts like classes, objects, and inheritance.

CO 3: Handle exceptions using Java's try-catch mechanisms.

CO 4: Create GUI and file handling programs using AWT and I/O.

Laboratory Program List

Part A:

1. Program to find whether the given number is Positive, Negative or Zero.
2. Program to list the factorial of the numbers 1 to 10.
3. Program to demonstrate classes & objects.
4. Program to demonstrate method overloading.
5. Program to demonstrate single inheritance (simple calculator – base class, Advanced Calculator – derived class).
6. Program to find Maximum & Minimum element in one-dimensional array of numbers.
7. Program to check whether the given string is palindrome or not.
8. Program to create a 'Student' class with Reg.no., name and marks of 3 subjects. Calculate the total marks of 3 subjects and create an array of 3 student objects & display the results.

Part B:

1. Program to generate negative array size exception
2. Program to generate NullPointerException.
3. Program that reads two integer numbers for the variables a and b. The program should catch NumberFormatException & display the error message.
4. Program to create AWT window with 4 buttons M/A/E/Close. Display M for Good Morning, A for Afternoon, E for evening and Close button to exit the window.
5. Program to demonstrate the various mouse handling events.
6. Program to read and write Binary I/O file.
7. Program to create window with three buttons father, mother and close. Display the respective details of father and mother as name, age and designation using AWT controls.
8. Program to create menu bar and pull-down menus.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	3	-	-	3	-	-	-	-	-
CO2	3	-	-	3	-	-	-	-	-
CO3	3	-	-	3	-	-	-	3	-
CO4	4	-	-	-	-	-	4	-	

Semester: IV

Course Code: GCS413 (Elective)	Course Title: Software Engineering
Course Credits: 02 (2-0-0)	Hours/Week: 02
Total Contact Hours: 30 Hours	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 2½ Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO1: Explain software engineering concepts, process models, and agile practices.

CO2: Apply techniques to gather and manage software requirements.

CO3: Model systems using UML diagrams and system modelling techniques.

CO4: Design software architecture using patterns and UML notations.

Course Content	Hours
Unit 1	
Overview: Introduction; Software Engineering Ethics; Software Process Models; Process Activities; Coping with Change; Agile Software Development: Agile Methods; Plan-Driven and Agile Development. Requirements Engineering: Functional and Non-Functional Requirements; Software Requirements Document; Requirements Specification; Requirements Engineering Processes; Requirements Elicitation and Analysis; Requirements Validation; Requirements Management.	15
Unit 2	
System Modeling: Context Models; Interaction Models- Use Case Modelling, Sequence Diagrams; Structural Models- Class Diagrams, Generalization, Aggregation, Behavioral Models- Data-Driven Modelling, Event-Driven Modelling; Model-Driven Engineering. Architectural Design: Architectural Design overview; Design and Implementation: Object-Oriented Design Using The UML- System Context and Interactions, Architectural Design, Object Class Identification, Design Models, Interface Specification; Design Patterns; Implementation Issues.	15

Text Books:

1. Software Engineering by Ian Sommerville, Pearson publications; 10th Edition, 2015.
2. A Concise Introduction to Software Engineering, Pankaj Jalote, 2nd Edition, Springer Cham, 2025.

Reference:

1. Software Engineering, N.S. Gill, Khanna Publishing House, 2023.
2. Software Engineering A Practitioner's Approach, 8th edition, Roger S Pressman, Bruce R. Maxim. McGraw Hill Education, 2015.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	2	-	-	2	2
CO2	3	3	-	-	3	-	-	3	-
CO3	3	-	3	-	3	3	-	3	-
CO4	4	-	4	-	4	4	-	4	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓	✓		
II			✓	✓

Semester: IV

Course Code: GCS*** (Elective)	Course Title: E-commerce and E-Governance
Course Credits: 02 (2-0-0)	Hours/Week: 02
Total Contact Hours: 30	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 2½ Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Understand the fundamentals, models, and infrastructure of E-commerce.

CO 2: Analyze e-commerce strategies, marketing, and real-world applications.

CO 3: Explain E-Governance models, technologies, and service frameworks.

CO 4: Evaluate challenges, innovations, and future trends in E-Governance.

Course Contents	Hours
Unit-1	
Introduction to E-commerce: Fundamentals of E-commerce, E-commerce Business Models, E-commerce Infrastructure, E-commerce Security and Payment Systems E-commerce Strategies and Applications: E-commerce Marketing and 11 Advertising, Supply Chain Management in E-commerce, Emerging Trends in E-commerce, Case Studies and Practical Applications.	15
Unit-2	
Introduction to E-Governance: Fundamentals of E-Governance, E-Governance Models and Frameworks, Technology and Infrastructure for E-Governance, E-Governance Services and Applications. Challenges and Future Trends in E-Governance: Challenges in E-Governance, E-Governance in Developing Countries, Emerging Technologies in E-Governance, Future Directions and Innovations.	15

Textbooks:

1. "E-Commerce 2020: Business, Technology, and Society" by Kenneth C. Laudon and Carol Guercio Traver, Pearson.
2. "Electronic Commerce 2018: A Managerial and Social Networks Perspective" by Efraim Turban, Jon Outland, David King, Jae Lee, Ting-Peng Liang, and Deborrah C. Turban, Springer.

Reference Books:

1. "E-Government: Information, Technology, and Transformation" by Hans J. Scholl, Routledge.
2. "E-Governance: Managing or Governing?" by Jeremy Millard, Routledge.
3. "Public Information Technology and E-Governance: Managing the Virtual State" by G. David Garson, Jones & Bartlett Learning.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	2	-	-	-	2
CO2	4	-	-	-	4	4	-	-	4
CO3	2	-	-	-	2	-	-	-	2
CO4	4	-	-	-	4	-	-	-	4

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓	✓		
II			✓	✓

Semester: IV

Course Code: GCS *** (Elective)	Course Title: Software Testing
Course Credits: 02 (2-0-0)	Hours/Week: 02
Total Contact Hours: 30	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 2½ Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO1: Understand the basics of software testing and test case design.

CO2: Apply decision table and data flow testing methods.

CO3: Analyze integration and system testing techniques.

CO4: Evaluate object-oriented and GUI testing approaches.

Course Content	Hours
Unit 1	
Basics of Software Testing and Examples: Basic definitions, Test cases, Insights from a Venn diagram, identifying test cases, Error and fault taxonomies, Levels of testing, Generalized pseudo code, The triangle problem, The Next Date function, The commission problem, The SATM (Simple Automatic Teller Machine) problem. Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the Next Date function, Test cases for the commission problem. Integration Testing: Introduction, Decomposition-based, call graph-based, Path-based integrations.	15
Unit 2	
System Testing: Definition, Possibilities, Basic concepts for requirements specification, Model-Based Threads, Use Case-Based Threads, ASF (Atomic System Functions). Object-Oriented Testing: Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism, Levels of object-oriented testing, Object-Oriented System Testing: Currency converter UML description, UML-based system testing, State chart-based system testing. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program, case study of windshield wiper.	15

Text Books:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications.

Reference Books:

1. Aditya P Mathur: Foundations of Software Testing, Pearson.
2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, 1st edition, John Wiley & Sons.
3. Srinivasan Desikan, Gopalaswamy Ramesh: Software Testing Principles and Practices.

CO–PO Matrix

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	-	-	2	-	-
CO2	-	3	3	-	-	3	-	-	-
CO3	-	4	4	-	-	-	-	4	3
CO4	4	4	-	4	-	-	-	-	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓	✓		
II			✓	✓

Semester IV (For 2024-25 Batch only)

Course Code: GCS 403 (Elective)	Course Title: Software Testing
Course Credits: 03 (3-0-0)	Hours/Week: 03
Total Contact Hours: 44	Formative Assessment Marks: 20
Exam Marks: 80	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO 1: Understand the basics of software testing and test case design.

CO 2: Apply decision table and data flow testing methods.

CO 3: Analyze integration and system testing techniques.

CO 4: Evaluate object-oriented and GUI testing approaches.

Course Content	Hours
Unit 1	
Basics of Software Testing and Examples: Basic definitions, Test cases, Insights from a Venn diagram, identifying test cases, Error and fault taxonomies, Levels of testing, Generalized pseudo code, The triangle problem, The Next Date function, The commission problem, The SATM (Simple Automatic Teller Machine) problem. Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the Next Date function, Test cases for the commission problem.	11
Unit 2	
Data Flow Testing: Definition- Use testing, Slice-based testing, Guidelines and observations. Life Cycle-Based Testing: Traditional Waterfall Testing, Testing in Iterative Life Cycles, Agile Testing, Model-Based Testing: Testing Based on Models, Peterson's Lattice, Expressive Capabilities of Mainline Models, Modelling Issues, Making Appropriate Choices. Integration Testing: Introduction, Decomposition-based, call graph- based, Path-based integrations.	11
Unit 3	
System Testing: Definition, Possibilities, Basic concepts for requirements specification, Model-Based Threads, Use Case-Based Threads, ASF (Atomic System Functions). Object-Oriented Testing: Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism, Levels of object-oriented testing, GUI testing, Dataflow testing for object-oriented software. Class Testing: Methods as units, Classes as units.	11

Unit-4	
Object-Oriented Integration Testing: UML support for integration testing, MM paths for object-oriented software, A framework for object-oriented dataflow integration testing. Object-Oriented System Testing: Currency converter UML description, UML-based system testing, State chart-based system testing. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program, case study of windshield wiper.	11

Text Books:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Uerbach Publications.
2. Aditya P Mathur: Foundations of Software Testing, Pearson.

Reference Books:

1. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, 1st edition, John Wiley & Sons.
2. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	-	-	2	-	-
CO2	-	3	3	-	-	3	-	-	-
CO3	-	4	4	-	-	-	-	4	3
CO4	4	4	-	4	-	-	-	-	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓			
II		✓		
III			✓	
IV				✓

Semester: V

Course code:	Course title: Operating Systems
Course credits: 03 (3-0-0)	Hours/Week: 3
Total contact class: 44	Formative assessment marks: 20
Exam marks: 80	Exam duration: 3 Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to,

CO 1: Gain knowledge of the basic concepts of Operating System, its functions and services.

CO 2: Apply the various concepts related to Process Management and threads.

CO 3: Evaluate the performance of various algorithms used for CPU scheduling.

CO 4: Explain the concepts of Deadlocks.

CO 5: Understand the concepts of memory management in main memory and virtual memory.

Course Contents	Hours
Unit:1:	
Introduction to Operating Systems, Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Kernel Data Structures, Computing Environments, Open-Source Operating Systems.	11
Unit 2:	
Process Concept, Process Scheduling, Operations on Processes, Inter Process Communication, Examples of IPC Systems, Communication in Client Server Systems. Threads: Overview, Multicore Programming, Multithreading Models, Thread Libraries.	11
Unit:3:	
Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Real-Time CPU Scheduling System Model, Deadlock Characterization, and Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.	11
Unit:4:	
Main Memory: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files.	11

Text Book:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 10th Edition, Wiley, 2018.

Reference Books:

1. Operating System Principles, Per Brinch Hansen, Prentice Hall of India (PHI).
2. Operating Systems, Milan Milenkovic, McGraw-Hill.
3. Modern Operating Systems, Andrew S. Tanenbaum, 4th Edition, Prentice Hall of India (PHI).

Course Pedagogy

1. Foundation-oriented Instruction and Interactive Sessions.
2. Illustrative Examples and Concept Demonstrations.
3. Problem-solving Exercises and Algorithm Analysis.
4. Hands-on Activities and Simulation-based Learning.
5. Group Discussions, Assignments, and Technical Presentations.

CO–PO Matrix

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	-	-	-	-	-
CO2	3	3	-	3	-	-	-	-	-
CO3	4	4	4	-	-	-	-	-	-
CO4	3	3	-	-	-	-	-	-	-
CO5	3	3	-	-	-	3	-	-	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4	CO5
I	✓				
II		✓			
III			✓		
IV				✓	✓

Semester: V

Course code: GCS503	Course title: Web Technology
Course credits: 03 (3-0-0)	Hours/Week: 3
Total contact class: 44	Formative assessment marks: 20
Exam marks: 80	Exam duration: 3 Hours

Course Outcome (COs): Upon successful completion of this course, students will be able to,

CO 1: Understand Web Fundamentals

CO 2: Explore the concepts of HTML/XHTML

CO 3: Apply Styling to Web Pages Using CSS

CO 4: Implement Basic Interactivity with JavaScript

CO 5: Demonstrate JavaScript with PHP-MYSQL for Dynamic Web Pages.

Course Contents	Hours
Unit 1:	
Introduction to the Internet, World Wide Web (WWW), Web Browsers and Web Servers, Uniform Resource Locators (URL), Multipurpose Internet Mail Extensions (MIME), Hypertext Transfer Protocol (HTTP/HTTPS), Web Programmer's Toolbox, Origins and Evolution of HTML and XHTML, Basic Syntax and Standard HTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML5 Semantic Elements, Syntactic Differences between HTML and XHTML.	11
Unit 2:	
Introduction to CSS, Levels of Style Sheets, Style Specification Formats, CSS Selectors. Property-Value Forms, Font Properties, Text Properties, Color Properties, Background Images, List Properties, Alignment of Text, The CSS Box Model, Margin, Padding and Border, Positioning and Layout Basics, The <div> and Tags, Introduction to Responsive Web Design. JavaScript Programming: Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Variables and Data Types, Primitives, Operators and Expressions, Screen Output and Keyboard Input, Conditional and Looping Control Statements	11
Unit 3:	
Arrays, Functions, Objects and Constructors, Pattern Matching Using Regular Expressions, Error Handling and Debugging, JavaScript Execution Environment, The Document Object Model (DOM), Element Access in JavaScript, Events and Event Handling, Handling Events from Forms, Buttons and Text Boxes, DOM Tree Traversal and Modification, Navigator Object. PHP and Web Application Development: Introduction to PHP, Installing XAMPP and PHP Environment, PHP Syntax, Variables, Constants and Data Types, Operators and Expressions, conditional and Looping Control Statements	11

Unit 4:	
Functions and Arrays in PHP. Embedding PHP in HTML, Form Handling using GET and POST Methods, Server-side Form Validation Session Handling: Cookies and Sessions Introduction to MySQL: PHP-MySQL Database Connectivity, CRUD Operations (Create, Read, Update and Delete)	11

Text Book:

1. Programming the World Wide Web, Robert W. Sebesta, 10th Edition, Pearson Education, 2018.

Reference Books:

1. Internet and World Wide Web – How to Program, Harvey M. Deitel, Paul J. Deitel, 5th Edition, Pearson Education, 2019.
2. Web Programming: Building Internet Applications, Chris Bates, 3rd Edition, Wiley India, 2020.
3. Introduction to JavaScript Object Notation, Lindsay Bassett, O'Reilly Media, Inc., 2015.

Course Pedagogy

1. Interactive lecture sessions using ppts and chalkboard.
2. Step-by-step Demonstrations and Coding Walkthroughs.
3. Assignments and Seminars

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	2	-	-	-	-
CO2	2	-	-	2	2	-	-	-	-
CO3	3	-	3	3	3	-	-	-	-
CO4	3	3	3	3	-	-	-	-	-
CO5	3	3	3	3	-	-	-	3	3

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4	CO5
I	✓				
II		✓			
III			✓		
IV				✓	✓

Semester: V

Course code:	Course title: Web Technology Lab
Course credits:02(0:0:2)	Hours/Week: 04
Total contact class: 60	Formative assessment marks: 10
Exam marks: 40	Exam duration: 3 Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to,

CO 1: Understand the basic concepts of HTML and explore Hyperlinks, Lists and Images.

CO 2: Demonstrate practical skills, workflows and methods used to create websites using Style sheets.

CO 3: Explore JavaScript and its basic functions and operations.

CO 4: Understand the concepts of Objects and Dynamic HTML.

List of Practical Experiments**Part -A**

1. Personal Portfolio Website:
 - Create a personal portfolio showcasing your skills, projects, and resume.
 - Use HTML5 semantic elements and CSS for styling.
2. Responsive Web Design:
Develop a multi-page website with responsive design using media queries and Bootstrap or another CSS framework.
3. CSS Grid and Flexbox Layout:
Design a layout using CSS Grid and Flexbox to create complex and responsive web layouts.
4. Interactive Form with CSS Validation:
Build a form with custom styling and CSS-based validation (e.g., required fields, pattern matching).
5. Custom Navigation Menu:
Create a navigation menu with dropdowns, responsive behavior, and hover effects using CSS.
6. Personal Blog Design:
Design a personal blog layout with custom styling for posts, comments, and sidebars.
7. Portfolio Website:
Create a portfolio website with sections for projects, skills, and contact information, utilizing advanced CSS layout and styling techniques.

Part -B

8. Restaurant Menu: Develop a stylish restaurant menu with categories, item descriptions, and prices, using CSS for layout and design.
9. Event Page: Design a web page for an event with sections for the event schedule, speakers, and registration, using CSS to style different elements.

10. Write a JavaScript program to
 - Find the square root
 - Calculate the area of triangle
11. Write a simple JavaScript that adds some numbers together, concatenates a couple of strings, and then shows the result in an alert () dialog and on the page.
12. To-Do List Application:
Build a JavaScript application to add, remove, and mark tasks as completed.
13. Write a JavaScript program to create a registration form.
14. Dynamic Content Loading.

Course Pedagogy

1. Practice-oriented Lab Sessions and Guided Implementation.
2. Step-by-step Demonstrations and Coding Walkthroughs.
3. Experiment-based Learning Approach.
4. Application-focused Assignments and Mini Projects.
5. Continuous Evaluation through Lab Performance and Viva Voce.

CO–PO Matrix

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	2	2	-	-	-	-
CO2	3	-	3	3	3	-	-	-	-
CO3	3	3	3	3	-	-	-	-	-
CO4	3	3	3	3	2	2			

Semester VI

Course Code: GCS601	Course Title: Computer Networks
Course Credits: 03 (3:0:0)	Hours/Week: 03
Total Contact Hours: 44	Formative Assessment Marks: 20
Exam Marks: 80	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to,

CO 1: Understand the fundamentals of computer networking

CO 2: Describe the layered architecture of networks

CO 3: Analyze data communication techniques

CO 4: Focus on routing and congestion control mechanisms

Course Contents	Hours
Unit 1:	
Introduction to Computer Network: Introduction, Characteristics, Components, Topology-Star, Mesh, Tree, Bus, Ring and Hybrid topology. Transmission Mode-Simplex, Half Duplex, Full Duplex. Categories of Network-LAN, WAN, MAN. The OSI reference model, The TCP/IP reference model. Digital to Analog Conversion- ASK, FSK, PSK.	11
Unit 2:	
Data Transmission and Analog to Analog conversion: Data Transmission serial and parallel transmission, Analog to Digital transmission, Analog to Analog Conversion- AM, FM, PM. The Physical Layer- Transmission media- Twisted pair, coaxial cable, Optical fiber, Radio, Microwaves and Infrared transmission. Switching- Message Switching, circuit Switching and packet Switching Multiplexing.	11
Unit 3:	
The Data link layer: Data link layer design issues, Error detection- Single Parity checking, Checksum, Polynomial Codes- CRC, error Correction- Hamming Code. The Network layer: Design issues, Routing Algorithm- Dijkstra's Shortest path Routing, Flooding, distance vector routing, Congestion Control Algorithm- Leaky Bucket, Token Bucket.	11
Unit 4:	
The Transport Layer and Application layer: Elements of Transport Service- Internet transport Protocols (TCP and UDP), Application Layer, DNS, Electronic Mailing- Introduction, SMTP, POP3, SNMP, FTP, TELNET and WWW.	11

Text Books:

1. Computer Networks – by Andrew S. Tanenbaum, David J. Wetherall, and Nick Feamster
Publisher: Pearson
2. Computer Networking: A Top-Down Approach – by James F. Kurose and Keith W. Ross
3. Data Communications and Networking – by Behrouz A. Forouzan

Reference books:

1. Computer Networks – by Larry L. Peterson and Bruce S. Davie
2. TCP/IP Illustrated Volume 1 – by W. Richard Stevens
3. Internetworking with TCP/IP – by Douglas E. Comer

Course Pedagogy:

1. Classroom lectures to explain fundamental concepts such as network models, protocols, and architectures.
2. Topics include OSI Model, TCP/IP Model, routing, and network layers.
3. Encourages students to actively participate in understanding networking concepts.
4. Demonstrating real network setups, protocols, and communication processes.
5. Visualization of how data travels through different layers.

CO–PO Matrix

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	-	-	-	-	-	-
CO2	2	-	-	-	-	-	-	-	-
CO3	2	2	-	-	-	-	-	-	2
CO4	2	2	2	-	-	2	-	2	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓			
II		✓		
III			✓	
IV				✓

Semester: VI

Course code: GCS603	Course title: Python Programming
Course credits: 02 (2-O-0)	Hours/Week: 02
Total contact class: 30	Formative assessment marks : 10
Exam marks: 40	Exam duration: 3 Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to,

CO 1: Understand the basic concepts of Python Programming.

CO 2: Apply proficiency in the handling of loops and the creation of functions.

CO 3: Identify the methods to create and store strings.

CO 4: Analyze the data using visualization tools.

Course Contents	Hours
Unit 1:	
Python Basics: - Introduction to Features and Applications of Python; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program. Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built-in Functions- Console Input and Console Output, Type Conversions, Python Libraries; Importing Libraries with Examples; Illustrative programs.	10
Unit 2:	
Python Control Flow: Types of Control Flow; Control Flow Statements- if, else, elif, while loop, break, continue statements, for loop Statement; range() and exit () functions; Illustrative programs. Python Functions: Types of Functions; Function Definition- Syntax, Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Illustrative programs.	10
Unit 3:	
Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifiers; Escape Sequences; Raw and Unicode Strings; Python String Methods, Lists, Tuples, Dictionaries and sets. Introduction to NumPy. Data Visualization : Introduction to data visualization Matplotlib library, different types of charts using PYplot,	10

Text Books:

1. Introduction to Machine Learning with Python – Andreas Müller
2. Python Programming: An Introduction to Computer Science – John M. Zelle.

Reference Books:

1. Python Cookbook – David Beazley & Brian K. Jones
2. Python Crash Course – Eric Matthes.
3. A Byte of Python – C.H. Swaroop.

Course pedagogy:

Classroom lectures to explain Python fundamentals such as variables, data types, operators, control structures, functions, and modules.

Demonstrations to show how Python programs are written, executed, and debugged.

Hands-on Practice: Students practice writing Python programs using tools like IDLE, PyCharm, or Jupyter Notebook.

CO–PO Matrix

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	2	-	-	-	-	-
CO2	3	3	3	3	-	-	-	-	-
CO3	3	-	-	3	-	-	-	-	-
CO4	4	4	4	4	4	-	-	4	-

CO–Unit Mapping

Unit	CO1	CO2	CO3	CO4
I	✓			
II		✓		
III			✓	✓

Semester: VI

Course code: GCS604	Course title: Python Programming Lab
Course credits:02(0:0:2)	Hours/Week: 04
Total contact class: 60	Formative assessment:10
Exam marks: 40	Exam duration: 3 Hours

Course Outcomes (CLOs): Upon successful completion of this course, students will be able to,

- CO 1:** Understand the basic concepts of Python programming,
CO 2: Apply control structures such as conditional statements and loops to solve programming problems.
CO 3: Develop modular programs using functions and modules to improve code organization and reusability.
CO 4: Use Python data structures such as lists, tuples, sets, and dictionaries to store and manipulate data.

List of Practical Experiments**Part –A**

1. Python script for checking the given year is leap year or not.
2. Python script to check if a number belongs to the Fibonacci sequence
3. Python script to solve Quadratic Equations
4. Python script to display all numbers which are divisible by 7 but are not a multiple of 5,
5. between given range X and Y.
6. Python script to display Multiplication Tables
7. Python script to create a calculator program
8. Explore string functions
9. Implementation of python script that takes a list of words and returns the length of the longest one.
10. Python script handles multiple errors with one except statement.
11. Python script to check whether password is valid or not. Conditions for a valid password are:
 - Should have at least one number.
 - Should have at least one uppercase and one lowercase character.
 - Should have at least one special symbol.

Part –B

1. Implement python script to remove duplicates from a list.
2. Implement python script to find the repeated items of a tuple.
3. Implement python script to check whether a given key already exists or not in a dictionary.
4. Write a python script to implement method overloading.
5. Drawing Line chart and Bar chart using Matplotlib.
6. Drawing Histogram and Pie chart using Matplotlib.
7. Create Array using NumPy and Perform Operations on Array.

CO–PO Matrix

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	2	-	-	2	-	-	-	-	-
CO2	3	3	3	3	-	-	-	-	-
CO3	4	4	4	4	-	-	-	-	-
CO4	3	3	3	3	-	-	-	-	-

Semester: IV

Course Code: GCS***	Course Title: Digital Marketing (Skill)
Course Credits: 02 (0-0-2)	Hours/Week: 04
Total Contact Hours: 60	Formative Assessment Marks: 10
Exam Marks: 60	Exam Duration: 03:00

Course Outcomes (COs): Upon successful completion of this course, students will be able to:

CO1: Describe the basics, evolution, and channels of digital marketing.

CO2: Apply social media and email marketing strategies effectively.

CO3: Create content and mobile marketing plans with analytics.

Do the Activates:

- To understand the digital landscape and set up a strategic framework.**
 - Set up a free website/blog (WordPress or Wix) for a hypothetical product/service.
- To identify and select appropriate channels for marketing strategy.**
 - Create a digital marketing budget allocation plan for 3 months.
 - Develop a campaign plan combining SEO, social media, and Email.
- To understand how to optimize content for search engines.**
 - Perform Keyword Research using Google Keyword Planner or Uber suggest.
 - Optimize the website/blog created in Lab 1 with meta titles, descriptions, and header tags.
- To understand link building and technical aspects of SEO.**
 - Submit the blog to search engines (Google Search Console).
 - Analyze competitors' backlink profiles and create a link-building strategy.
 - Set up Google Search Console to monitor site performance.
- To effectively establish a brand presence on social media.**
 - Create and optimize Business Profiles on Facebook, Instagram, and LinkedIn.
 - Develop a Content Calendar for one month for a specific platform.
- To curate and schedule posts using management tools.**
 - Create sample posts (Images, Reels, Carousels) using Canva.
 - Schedule posts using Meta Business Suite or Buffer for a week.
- To create and manage paid ads on social platforms.**
 - Create a Facebook/Instagram Ad campaign (Mock or \$5 actual budget) using Ads Manager.
 - Set up targeting (Demographics, Interests, Behaviors).
 - Create A/B testing variants for ad creatives.
- To understand lead generation and email automation.**
 - Set up a Mailchimp or Send in blue account.
 - Create a sign-up form and embed it on the blog.
 - Set up an automated "Welcome Email" sequence for new subscribers.
- To execute effective email campaigns and analyze them.**
 - Design an email newsletter (HTML/Plain Text) promoting a blog post or product.
 - Send a test campaign and analyze open rates, click-through rates (CTR), and bounce rates.
- To create content that attracts and engages target audiences.**
 - Create 3 different content types (Blog, Infographic, Video Script) for a single topic.

- Develop a Content Distribution Plan (Earned, Owned, Paid Media).
- 11 **To understand the use of influencer and guest posting strategies.**
 - Identify 5 influencers in the niche using BuzzSumo or manual search.
 - Write a sample outreach email for guest posting or influencer collaboration.
- 12 **To explore mobile-specific marketing strategies.**
 - Optimize the website for mobile responsiveness.
 - Create a strategy for App Store Optimization (ASO) for a hypothetical app.
 - Set up a mock SMS or WhatsApp marketing campaign using proper consent.
- 13 **To understand the importance of data in marketing.**
 - Integrate Google Analytics 4 (GA4) with the website/blog.
 - Set up Goals in GA4 (e.g., Form submissions, Page views).
 - Understand basic reports (Audience, Acquisition, Behavior).
- 14 **To track, measure, and visualize campaign performance.**
 - Create a Marketing Dashboard using Google Looker Studio.
 - Connect GA4 and Google Ads/Social Media data (manually enter data if API access is limited).
 - Calculate ROI and ROAS for a sample campaign.
- 15 **To integrate all digital marketing skills into a real-world scenario.**
 - **Group Project:** Plan, execute (where possible) or simulate a full Digital Marketing Campaign for the hypothetical product.
 - **Presentation:** Students present their strategy, execution, analytics, and recommendations.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	-	-	-	3	2	2	-	2	-	-
CO2	-	3	-	-	3	3	3	-	3	-	-
CO3	-	-	4	-	3	1	4	2	2	-	-

Semester: V

Course code: GCS***	Course title: AI Tools and its applications
Course credits: 02(0-0-2)	Hours/Week: 4
Total contact class: 60	Formative assessment marks: 10
Exam marks: 40	Exam duration: 3 Hours

Course Outcomes (COs): Upon successful completion of this course, students will be able to,

CO 1: Use AI tools to generate text, images, presentations and reports effectively.

CO 2: Apply AI tools for solving basic computational and programming problems.

CO 3: Analyze data and generate meaningful insights using AI assistance.

CO 4: Compare different AI tools and select appropriate tools for specific tasks.

CO 5: Demonstrate the use of AI tools in academic work such as resume creation, translation and documentation.

List of Practical Experiments**PART A**

1. Create an account and explore features of ChatGPT (text generation, summarization).
2. Use ChatGPT to generate a technical literature review on one of topic (IoT / AI in Healthcare / Smart Grid) and refine it into IEEE abstract format
3. Use Google Gemini to generate a short essay on “AI past and future”.
4. Generate an image of future smart class and analyze
5. Design a project poster using Canva AI tools including smart layout, AI background removal, and presentation enhancement.
6. Create a technical presentation using AI presentation generator on “Future of Edge AI in Engineering Applications”- Gamma.
7. Write a paragraph with mistakes, Upload in Grammarly, Note corrections and Rewrite corrected paragraph
8. Create invitation and flyer for Science day celebration function in your college
9. Develop a no-code, student grievance management prototype using Google AI.

Part B

10. Create AI-generated diagrams for system architecture of a smart campus project using AI diagram tools.

11. Generate an MCQ with 25 questions on Data structure , DBMS and Computer networks with appropriate headings
12. Build a workflow automation using no-code tools for attendance management or report approval system.
13. Compare outputs of two AI chatbots for solving one numerical problem and evaluate response accuracy.
14. Interact with AI Chatbot. Ask 5 academic questions, Note responses, Evaluate correctness
15. Use AI tool to record the answers and observation on Operating system
16. Upload erroneous Python/C++ code into AI tool, debug it, optimize it, and explain performance improvements. Create a resume using Gemini
17. Use Microsoft Excel AI functions to clean a raw dataset, detect anomalies, and generate trend analysis with formulas and charts.

Course Pedagogy

1. Demonstrates the usage of AI tools (e.g., ChatGPT, Canva, Gamma).
2. Hands-On Practical Learning by conducting experiments individually or in small groups.
3. Problem-Based Learning (PBL) - solve real-world or academic problems using AI tools.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
CO1	-	-	3	-	3	3	-	3	-
CO2	3	3	3	3	3	-	-	3	-
CO3	4	4	-	-	4	-	3	4	-
CO4					4	3	3	4	-
CO5	3	-	-	3	3	3	-	3	-

Semester: VI

Course Code: GCS612	Course Title: Power BI
Course Credits: 02 (0-0-2)	Hours/Week: 04
Total Contact Hours: 60	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 03

Course Outcomes (COs): Upon successful completion of this course, students will be able to,

- CO 1:** Import, clean, and transform datasets using tools such as Power Query Editor for effective data preparation.
- CO 2:** Create meaningful data visualizations including bar charts, line charts, pie charts, and donut charts to represent business data.
- CO 3:** Develop interactive reports and dashboards to analyze key performance indicators (KPIs) and business performance.
- CO 4:** CLO 4: Apply data modeling techniques and DAX expressions to establish relationships between tables and generate calculated fields for advanced analysis.

List of Practical Experiments**Part -A**

1. Import a dataset from an Excel file and clean the data by removing duplicates and handling missing values.
2. Use Power Query Editor to transform a dataset, such as splitting columns, merging tables, or changing data types.
3. Create a bar chart to show the total sales by region.
4. Create a line chart to display the trend of sales over time.
5. Build a report that includes various visualizations, such as pie charts, bar charts, and tables, to analyze sales data.

Part -B

6. Create a dashboard that combines several key performance indicators (KPIs) to give an overview of business performance.
7. Create a relationship between tables manually through Power BI.
8. Use the DAX expression to add a new column in the data source.
9. Use the Donut Chart for Profit Analysis.
10. Analyze the more profitable day from the sales analysis table using Bar Chart in Power BI.

Course Pedagogy

1. Hands-on Laboratory Sessions using Power BI for data import, transformation, visualization, and dashboard creation.
2. Demonstration-based Teaching where the instructor explains Power BI tools such as Power Query Editor, data modeling, and DAX functions.
3. Practical Assignments and Exercises requiring students to analyze datasets and build visual reports and dashboards.

CO–PO Matrix

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	3	1	3	-	-	-	2	-	-
CO2	3	3	3	1	3	-	-	-	2	-	-
CO3	2	3	3	3	3	-	-	-	2	-	-
CO4	2	2	2	3	3	-	-	-	2	-	-

Evaluation Scheme

Theory Courses

Evaluation Scheme for 3 / 2 Credits Practicals/ Skill Examination [100 / 50 Marks]

Assessment Criteria		3 Credits	2 Credits
		Marks	Marks
Continues Internal Assessment Evaluation (CIE)	C1	10	05
	C2	10	05
Semester End Examination Evaluation (SEE)	C3	80	40
Total		100	50

Practicals / Skill Courses

Evaluation Scheme for 2Credits Practicals/ Skill Examination [50 Marks]

Assessment Criteria Marks		Marks
Continues Internal Assessment Evaluation (CIE) [10 Marks]	C1	05
	C2	05
Semester End Examination Evaluation (SEE) [40 Marks]	Record	05
	Writing One Program from Part A	10
	Writing One Program from Part B	10
	Execution of the Written Program	05 + 05
	Viva Voce based on Practicals/ Skill Course	05
Total		50

Note: Any two Programs selected from the list of programs (if there is no Part A & Part B)

Scheme of valuation - 50 Marks

1. IA – 10 Marks
 - ☐ Synopsis - 05 Marks
 - ☐ Seminar - 05 Marks
2. Dissertation – 40 Marks
 - ☐ Documentation - 15 Marks
 - ☐ Presentation / Demonstration - 15 Marks
3. Viva- 10 Marks

QP Pattern for Theory Courses:

- Total Lecture hours per paper: 44Hrs. for 3 Credits & 30 Hrs. for 2 Credits
- No. of Units: 4 Units (11 Hours Each) for 3 Credits & 2 Units for 2 Credits

Question Paper Pattern**Instructions: Answer Part-A and Part-B:****Part-A**

**Answer any 10 out of 12 Questions (3 Questions drawn from each unit).
Each question carries 2 Marks. (10 X 2 =20) Q. No. 1 to Q. No. 12.**

Part-B**Answer the following Questions.**

13. a)

 OR
 b)
 14. a)

 OR
 b)
 15. a)

 OR
 b)
 16. a)

 OR
 b)

**Each main Question has with internal choice and carries 15 Marks. (4 X 15 =60)
(Each main question may split into sub-questions like (10 + 5, 8 +7, 5+5+5, ...) and with a maximum of 3 sub-questions)**

Scheme of question paper (2 Credits):**Maximum Marks – 40****Part A: Answer any 5 questions: (5x2=10 Marks)****Question No. 1 to 6****Part B: Answer the following questions: (15x2=30 Marks)**

7. a) 7. b) or 7. c) 7. d)
 8. a) 8. b) or 8. c) 8. d)