Telephone No. 2419677/2419361
Fax: 0821-2419363/2419301

e-mail : registrar@uni-mysore.ac.in
www.uni-mysore.ac.in

# UNIVERSITY OF MYSORE

Estd. 1916

VishwavidyanilayaKaryasoudha
Crawford Hall, Mysuru- 570 005

No.AC2(S)/35/2025-26 Dated: 09.06.2026

## Notification

**Sub**:- Syllabus and Scheme of Examinations of Computer Science (UG) Programme (V & VI Semester) from the Academic year 2026-27.

**Ref:**-1.Decision of Board of Studies in Computer Science (UG) meeting held on 16-03-2026, 17-03-2026 & 20-03-2026.
2. Decision of the Faculty of Science & Technology meeting held on 22-04-2026.
3. Decision of the Academic Council meeting held on 01-06-2026.

*****

The Board of Studies in Computer Science (UG) which met on 16-03-2026, 17-03-2026 & 20-03-2026 has resolved to recommend and approved the V & VI Semester Syllabus and Scheme of Examinations of Computer Science (UG) programme with effect from the Academic year 2026-27.

The Faculty of Science & Technology and Academic Council at their meetings held on 22-04-2026 and 01-06-2026 respectively have also approved the above mentioned Syllabus and Scheme of Examinations, accordingly it is hereby notified.

The Syllabus and Scheme of Examinations content may be download from the University Website www.uni-mysore.ac.in.

Registrar
Registrar
University of Mysore
Mysore

**To;**

1. All the Principals of affiliated Colleges of University of Mysore, Mysuru.
2. The Registrar (Evaluation), University of Mysore, Mysuru.
3. The Chairman, BOS/DOS in Computer Science, Manasagangothri, Mysuru.
4. The Dean, Faculty of Science & Technology, DOS in Botany, MGM.
5. The Director, PMEB, University of Mysore, Mysuru.
6. The Director, College Development Council, Manasagangothri, Mysuru.
7. The Director, ICD, MGM-Request to publish this Notification in university website.
8. The Deputy Registrar/Assistant Registrar/Superintendent, Administrative Branch and Examination Branch, University of Mysore, Mysuru.
9. The PA to Vice-Chancellor/ Registrar/ Registrar (Evaluation), University of Mysore, Mysuru.
10. Office Copy.

# Annexure-C

# B.Sc. (Computer Science)

# I to IV Semesters Modified Syllabus (SEP) 2026-27

# University of Mysore

# Curriculum

## Program: B.Sc.
## Subject: Computer Science

| Semester | Course No. | Theory/ Practical | Credits | L-T-P | No. of Hours | Course Title | Marks | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SEE | CIE |
| I | CSM11T | **Theory** | 03 | 3-0-0 | 03 | Computer Concepts and Problem-Solving using C++ | 80 | 20 |
| | CSM11P | **Practical** | 02 | 0-0-2 | 04 | Computer Basics and Programming in C++ | 40 | 10 |
| II | CSM21T | **Theory** | 03 | 3-0-0 | 03 | Data Structures | 80 | 20 |
| | CSM21P | **Practical** | 02 | 0-0-2 | 04 | Data Structures using C++ | 40 | 10 |
| III | CSM31T | **Theory** | 03 | 3-0-0 | 03 | Database Management System | 80 | 20 |
| | CSM31P | **Practical** | 02 | 0-0-2 | 4 | DBMS Lab | 40 | 10 |
| | CSE31 | **Theory (Elective 1)** | 02 | 2-0-0 | 02 | Software Engineering | 40 | 10 |
| | CSE32 | **Theory (Elective 2)** | 02 | 2-0-0 | 02 | E-Commerce & E-Governance | 40 | 10 |
| IV | CSM41T | **Theory** | 03 | 3-0-0 | 03 | Object Oriented Programming with Java | 80 | 20 |
| | CSM41P | **Practical** | 02 | 0-0-2 | 04 | Java Lab | 40 | 10 |
| | CSE41 | **Theory (Elective 1)** | 02 | 2-0-0 | 02 | Software Testing | 40 | 10 |
| | CSCP41P | **Practical (Compulsory Paper)** | 02 | 0-0-2 | 04 | Digital Marketing Lab | 40 | 10 |

## Semester: I

| | |
|---|---|
| **Course Code:** CSM11T | **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 Learning Outcomes (CLOs):**
**On successful completion of this course, students will be able to:**

**CLO 1:** Understand the basics of computer.
**CLO 2:** Understand problem-solving strategies and techniques.
**CLO 3:** Describe the Object-Oriented Programming principles and concepts.
**CLO 4:** Understand of the syntax and semantics of the C++.

**Course Pedagogy**

1. Interactive Lectures on fundamental concepts.
2. Demonstration-Based Learning to illustrate C++ syntax, program structure, and debugging techniques.
3. Problem-Solving Approach.

**Course Contents**

| | |
|---|---|
| **Unit-1** | |
| **Introduction To Computers:** Definition And Characteristics of Computers, Brief History of Computers, 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: System Software [Operating System, Interpreters], Application Software. Languages of Computer, ASCII And EBCDIC, Computer Threats and Safety Measurements. | 11 Hours |
| **Unit-2** | |
| **Problem Solving Techniques:** Problem, Definition, Analysis, Design Tools [Algorithm & Flow Chart], Coding, Testing, Maintenance. Basics of Algorithm Analysis: Time Complexity, Space Complexity, Asymptotic Notations.<br>**Basic Programming Concepts:** Tokens of Programming Language, Identifier, Constant, Variable, Data Types, And Operators. Introduction to Programming Concepts. Comparison of POP And OOP. | 11 Hours |
| **Unit-3** | |

| | |
|---|---|
| **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.<br>**Access Specifiers:** Public, Private, Protected. Pointers In C++. | 11 Hours |
| **Unit-4** | |
| **Objects, Classes:** Base and Derived Classes, Data Encapsulation, Data Abstraction, Friend Function, Inheritance: Single, Multiple, Multilevel, Hierarchical. **Polymorphism:** Function Overloading, Operator Overloading, Virtual Functions. **Constructors And Destructors:** Exception Handling. | 11 Hours |

**Learning Resources:**


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

| | |
|---|---|
| **Course Code:** CSM11P | **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 Learning Outcomes (CLOs):**

On successful completion of this course, students will be able to:

**CLO 1:** Illustrate the hardware components of Computer.
**CLO 2:** Use Open-Source Office tools.
**CLO 3:** Demonstrate simple programming skills through C++ programming language.

**Course Pedagogy**

1. Students write, compile, and execute programs during lab sessions.
2. Step-by-step explanation of program execution.
3. Students first design algorithms/flowcharts. Then implement solutions using C++.

**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. C++ program to print user name.
6. C++ program to swap 2 numbers.
7. C++ Program to Find Largest Among 3 Numbers
8. C++ program to perform arithmetic operations.

**PART-B**

1. C++ Program to Print Multiplication Table of a Number
2. C++ Program to Reverse a Number
3. C++ Program to Check Whether a Number is a Palindrome or Not
4. C++ Program to Create a Class and Object
5. C++ program for single inheritance.
6. C++ program to demonstrate multi-level inheritance.
7. C++ program to demonstrate operator overloading.
8. C++ Program by using Pointers

**Evaluation Scheme for Lab Examination [Marks: 40]**

| Assessment Criteria | | Marks |
|---|---|---|
| Writing | One Program from Part A | 15 |
| | One Program from Part B | 15 |
| Execution | Any one of the written Program | 5 |
| Viva Voce | | 5 |
| **Total** | | **40** |

## Semester: II

| | |
|---|---|
| **Course Code:** CSM21T | **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 (CLOs):**

On successful completion of this course, students will be able to:

**CLO 1:** Understand the basics of Data Structures.
**CLO 2:** Identify the appropriate data structures and algorithms for solving real world problems.
**CLO 3:** Understand the practical applications of Tree and Graph.

**Course Pedagogy:**

1. Interactive Lectures and Conceptual Discussions to introduce various Data Structure concepts.
2. Student Assignments and tests related to various data structures and its implementation.

| | |
|---|---|
| **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. **Arrays:** Declaration, Initialization, Accessing Elements, Multidimensional Arrays. C++ Strings, String Handling Functions, Applications of Arrays and Strings. **Stack:** Definition, Memory Representation, Algorithms for Stack Operations (Push, Pop), Applications of Stack. | 11 |
| **Unit-2** | |
| **Queue:** Definition, Memory Representation, Linear Queue, Circular Queue, Enqueue, Dequeue. Applications Of Queue. **Linked Lists**: Definition, Types. **Singly Linked List:** Implementation, Insertion [At the Beginning], Deletion [At the End]. **Doubly Linked List:** Memory Representation of Singly Linked List and Doubly Linked Lists. Applications of Linked List. | 11 |
| **Unit-3** | |
| **Tree**: Definition, Memory Representation Using Array and Linked List. **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. **Advanced Tree Structures AVL And B-Trees:** Definition and Applications. | 11 |

| **Unit-4** | |
|---|---|
| **Graph:** Definition, Memory Representation of Graph. Adjacency Matrix, Adjacency List. Graph Traversal Algorithms: Breadth-First Search (BFS), Depth-First Search (DFS). **Sorting Techniques:** Bubble Sort, Selection Sort [Algorithm, Time & Space Complexity]. **Searching Techniques:** Linear And Binary Search Sort [Algorithm, Time & Space Complexity]. **Heap:** Heap Operations and Applications. | 11 |

**Learning Resources:**


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.

## Semester: II

| | |
|---|---|
| **Course Code:** CSM21P | **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 Learning Outcomes (CLOs):**

**On successful completion of this course, students will be able to:**

**CLO 1:** Implement data structures using C++.
**CLO 2:** Demonstrate searching and sorting techniques using ++.
**CLO 3:** Demonstrate simple programming skills through C++ programming language.

**Course Pedagogy:**

1. Demonstration-based teaching of the various data structures.
2. Hands-on laboratory sessions on the implementation of the data structures and its application.
3. Problem-solving and programming exercises.

**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.

**Evaluation Scheme for Lab Examination [Marks: 40]**

| **Assessment Criteria** | | **Marks** |
|---|---|---|
| Writing | One Program from Part A | 15 |
| | One Program from Part B | 15 |
| Execution | Any one of the written Program | 5 |
| Viva Voce | | 5 |
| **Total** | | **40** |

## Semester: III

| | |
|---|---|
| **Course Code:** CSM31T | **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 Learning Outcomes (CLOs):**

Upon successful completion of this Course, Students will be able to

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

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

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

**CLO 4:** Evaluate transaction management and database security features.

**Course Pedagogy**

1. Lecture Method
2. Visual Learning Approach
3. Problem-Based Learning (PBL) Approach
4. Question–Answer Method
5. Assignment-Based Learning
6.

| **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). | 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 Normalization:** 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 Database Security. | 11 |

**Learning Resources:**


1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7thEdition, Pearson.
2. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
3. Introduction to Database System, C J Date, Pearson, 1999.
4. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6th Edition, McGraw Hill,
5. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition, McGraw Hill.

## Semester: III

| | |
|---|---|
| **Course Code:** CSM31P | **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 Learning Outcomes (CLOs):**

Upon successful completion of this Course, Students will be able to

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

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

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

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

**Course Pedagogy**

1. Experiential / Hands-on Laboratory Learning
2. Activity-Based Learning Approach
3. Demonstration-Based Teaching
4. Project-Based Learning (Mini Database Projects)

**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
3. Select FIRSTNAME and SALARY of salesmen
4. Display the FIRSTNAME, LASTNAME, and TOTAL SALARY of all employees where TOTAL SALARY = SALARY + BENEFITS
5. Count the number of distinct DESIGNATIONs from EMPSALARY
6. List the employees whose names have exactly 6 characters
7. Add a new column PHONE_NO to the EMPLOYEE table and update the records
8. List employee names who have joined before 15-Jun-08 and after 16-Jun-07
9. 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:** Library

Create 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 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 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, year of purchase in characters.
8. To round of the warranty period to 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 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. A) 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 new record.
   b. Update destination='Manglore' where train no='RJD16'.
   c. Delete a record which has train no='KKE55'
2. Create a view **detail** to display train number, train name, and class.
3. Create a view **total_seats** to display train number, start place, use COUNT function on No_of_seats, group by start place and perform the following:
   a. Insert new record.
   b. Update start place='Hubli' where train no='JNS8'.
   c. Delete the last row of the view.
4. Rename view **sleeper** to **class**.
5. 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: Branch**

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

**Table: Depositor**

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

**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.

**Evaluation Scheme for Lab Examination**

| **Assessment Criteria** | | **Marks** |
|---|---|---|
| Writing | One Program from Part A | 15 |
| | One Program from Part B | 15 |
| Execution | Any one of the Written Program | 5 |
| Viva Voce based on Data Base Management System | | 5 |
| **Total** | | **40** |

## Semester: III

| | |
|---|---|
| **Course Code:** CSE31 **(Elective 1)** | **Course Title: Software Engineering** |
| **Course Credits:** 02(2:0:0) | **Hours/Week:** 02 |
| **Total Contact Hours:** 30 Hours | **Formative Assessment Marks:** 10 |
| **Exam Marks: 4**0 | **Exam Duration:** 02 |

**Course Learning Outcomes (CLOs):**

Upon successful completion of this Course, Students will be able to

**CLO 1:** Explain software engineering concepts, process models, and agile practices.

**CLO 2:** Apply techniques to gather and manage software requirements, Model systems using UML diagrams and system modeling techniques.

**CLO 3:** Design software architecture using patterns and UML notations.

**Course Pedagogy**

1. Classroom teaching / Discussion Method
2. Case Study-Based Learning
3. Problem-Based Learning (PBL)
4. Seminar / Presentation Method

| Course Content | Hours |
|---|---|
| **Unit 1:** | |
| **OVERVIEW:** Introduction; Software Engineering Ethics; Software Process Models (Waterfall, V-Model, Incremental, Iterative, Prototyping, Spiral); Process Activities; Coping with Change; Plan- Driven and Agile Development. | 10 |
| **Unit 2:** | |
| **REQUIREMENTS ENGINEERING:** Functional and Non-Functional Requirements; Software Requirements Document; Requirements Engineering Processes; Requirement's Validation; Requirements Management. **SYSTEM MODELING: SYSTEM MODELING:** Context Models; Interaction Models- Use Case Modelling, Sequence Diagrams; Structural Models- Class Diagrams, Behavioral Models- Data-Driven Modelling, Event-Driven Modelling. | 10 |
| **Unit 3:** | |
| **ARCHITECTURAL DESIGN AND IMPLEMENTATION:** Architectural Views; Architectural Patterns- Layered Architecture, Repository Architecture, Client–Server Architecture Pipe and Filter Architecture. Object-Oriented Design Using The UML- System Context and Interactions, Interface Specification; Design Patterns; Implementation Issues. | 10 |

**Learning Resources:**

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

## Semester: III

| | |
|---|---|
| **Course Code:** CSE32 **(Elective 2)** | **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:** 02 |

**Course Learning Outcomes (CLOs):**

Upon successful completion of this Course, Students will be able to

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

**CLO 2:** Analyze E-commerce strategies, marketing, and real-world applications.

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

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

**Course Pedagogy**

1. Interactive Lectures and Conceptual Discussions to introduce fundamentals of e-commerce, business models, infrastructure, and e-governance concepts.
2. Case Studies and Real-world Application Analysis to understand practical implementation of e-commerce strategies and e-governance services.
3. Problem-Solving and Scenario-Based Learning to analyze business strategies, supply chain models, and governance challenges.
4. Seminars, Assignments, and Student Presentations focusing on innovations, future trends, and technological advancements in e-commerce and 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 | 10 |
| **Unit-2:** | |
| **E-commerce Strategies and Applications:** E-commerce Marketing and Advertising, Supply Chain Management in E-commerce, Emerging Trends in E-Commerce, Applications of E-Commerce | 10 |
| **Unit-3:** | |
| Introduction to E-Governance, E-Governance Models and Frameworks, Technology and Infrastructure for E-Governance, E- Governance Services and Applications. Challenges in E-Governance, E- Governance in Developing Countries, Emerging Technologies in E-Governance. | 10 |

**Learning Resources:**


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

**Semester - IV**

| | |
|---|---|
| **Course Code:** CSM41T | **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 Learning Outcomes (CLOs):**

**Upon successful completion of this course, students will be able to:**

**CLO 1:** Understand OOP concepts and basic Java programming structure.
**CLO 2:** Apply classes, objects, constructors, arrays, and strings in Java.
**CLO 3:** Implement multithreading and exception handling in Java programs.
**CLO 4:** Perform file I/O and develop GUI applications using AWT and event handling.

**Course Pedagogy**

1. Interactive Classroom lecture method using board and PPT to explain OOP concepts, Java fundamentals, multithreading, exception handling and GUI programming.
2 Demonstration of Java program execution, Live coding sessions to explain program structure, classes, threads and file handling.
3. Problem Solving Approach to solve programming problems related to object-oriented concepts.
4. Assignments, seminars and presentation

| **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.<br>**Arrays:** One dimensional Arrays, Two dimensional Arrays, Array of Objects.<br>**Strings**: String Handling functions. **Inheritance – Types of inheritance, Interface.** | 11 |
| **Unit-3:** | |

| | |
|---|---|
| **Packages** – Built-in and user defined packages. **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 |

**Learning Resources:**


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

## Semester: IV

| | |
|---|---|
| **Course Code:** CSM41P | **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 Learning Outcomes (CLOs):**

Upon successful completion of this course, students will be able to:

**CLO 1:** Write Java programs using basic logic and structures.

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

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

**CLO 4:** Create GUI and file handling programs using AWT and I/O

## Course Pedagogy

1.Demonstrate program execution, debugging techniques, and GUI development in the lab.
2.Hands-on Laboratory Practice
3.Problem Solving Approach to develop logical thinking and coding skills.
4.Interactive Learning, Discussion of errors, exceptions and alternative solutions to improve understanding.

**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 NullPointer Exception.
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.

**Evaluation Scheme for Lab Examination**

| **Assessment Criteria Marks** | | **Marks** |
|---|---|---|
| Writing | One Program from Part A | 15 |
| | One Program from Part B | 15 |
| Execution | Any one of the Written Program | 5 |
| Viva Voce based on Object Oriented with java | | 5 |
| **Total** | | **40** |

**Semester: IV**

| | |
|---|---|
| **Course Code:** CSE41 **(Elective)** | **Course Title: Software Testing** |
| **CourseCredits:02(2:0:0)** | **Hours/Week:02** |
| **Total Contact Hours:30** | **Formative Assessment Marks: 10** |
| **ExamMarks:40** | **ExamDuration:02** |

**Course Learning Outcomes (CLOs):**

**Upon successful completion of this course, students will be able to:**

**CLO1** Understand basics of software testing and test case
**CLO2** Apply decision table and data flow testing method
**CLO3** Analyze integration and system testing techniques.

**Course Pedagogy**

1. Classroom lectures to explain fundamentals of software testing, testing strategies
2. Case Study Approach Study of real software testing scenarios.
3. Group discussions on testing strategies.
4. Assignment Based Learning on test case design.

| **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. | 10 |
| **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, Modeling Issues, Making Appropriate Choices. Integration Testing: Introduction, Decomposition- based, call graph-based, Path-based integrations. | 10 |
| **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. | 10 |

**Learning Resources:**


1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications.
2. Aditya P Mathur: Foundations of Software Testing, Pearson.
3. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, 1st edition, John Wiley & Sons.
4. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices

**Semester: IV**

| | |
|---|---|
| **Course Code:** CSCP41P **(Compulsory Paper)** | **Course Title: Digital Marketing** |
| **Course Credits:** 02 (0:0:2) | **Hours/Week:** 04 |
| **Total Contact Hours:** 60 | **Formative Assessment Marks:** 10 |
| **Exam Marks:** 40 | **Exam Duration:** 03 |

**Course Learning Outcomes (CLOs)**

Upon successful completion of this course, students will be able to:

**CLO 1:** Identify and differentiate various digital marketing channels and tools.
**CLO 2:** Create and optimize social media profiles, content, and promotional Campaigns.
**CLO 3:** Design marketing content for different platforms including email, social media, and mobile.
**CLO 4:** Analyze digital advertisements, campaigns, and metrics to evaluate Effectiveness.
**CLO 5:** Develop and implement a digital marketing plan for real-world business scenarios.
**CLO 6:** Conduct surveys and research to understand digital tool adoption in small businesses.

## Course Pedagogy

1. Perform hands-on exercises: social media profile creation, content design, campaign planning, and survey preparation.
2. Demonstrate the use of tools such as Canva, Google Ads, WordPress, and email marketing platforms.
3. Implement a mini-project simulating a digital marketing campaign.
4. Analyze and present campaign results.
5. Group activities for campaign design, content planning, and survey analysis.

**List of Experiments**

**Activity 1: Digital Marketing Channel Identification**

I. List out at least 5 companies (example: Amazon, Flipkart, Myntra).
II. Identify what digital marketing channels they use:
    a. Website
    b. Social media
    c. Email marketing
    d. Mobile apps

**Activity 2: Social Media Profile Creation**

I. Create a sample business profile on any platform (Instagram/Facebook/LinkedIn).
II. Design:
    a. Profile name
    b. Logo
    c. Bio
    d. 3 promotional posts

**Activity 3: Content Creation Activity**

Create the following using canva/power point/any free design tool

a. One promotional post
b. One educational post
c. One product advertisement

**Activity 4: Digital Marketing Strategy Preparation**

Prepare a digital marketing plan for:

a. A college event
b. A new product
c. A startup idea

**Activity 5: Select one advertisement from YouTube or Instagram. and Identify:**

a. Target audience
b. Marketing objective
c. Type of content
d. Strengths

**Activity 6**: Create the Digital Marketing Webpage

**Activity 7:** Digital Marketing Implementation in Business Scenario

**Activity 8:** Advertising through Google Ads

**Activity 9:** Creating Promotional banner through Canva

**Activity 10:** Digital Marketing Implementation in Business Scenario

**Activity 11:** Prepare a survey Report on usage of digital tools in their business in nearby shops

**Evaluation Scheme for Lab Examination**

| **Assessment Criteria** | | **Marks** |
|---|---|---|
| Writing | Any two activities | 15 |
| | | 15 |
| Result | Execution and Implementation | 5 |
| Viva Voce | | 5 |
| **Total** | | **40** |

## CIE, SEE and QP Pattern for Theory Courses (2 Credits):

**Total Lecture hours per paper: 30, No. of Units: 3 (10 Hours each)**
**Internal Assessment C1 - 5 Marks, C2 - 5 Marks**
**Semester End Theory Exam C3 - 40 Marks**

## Question paper pattern for Theory Courses (2 Credits):

**Maximum Marks – 40**
**Part A: Answer any 5 questions out of 6: (5x2=10 Marks)**
**Question No. 1 to 6**
**Part B: Answer the following questions: (10x3=30 Marks)**
**7. a) 7. b) or 7. c) 7. d)**
**8. a) 8. b) or 8. c) 8. d)**
**9. a) 9. b) or 9. c) 9. d)**

## CIE, SEE and QP Pattern for Theory Courses (3 Credits):

**Total Lecture hours per paper: 44**

**No. of Units 4 (11 Hours Each)**
**Internal Assessment: C1 - 10 Marks, C2 - 10 Marks**
**Semester End Theory Exam: C3 - 80 Marks**

## Question paper pattern for Theory Courses (3 Credits):

**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 all the Questions. Each question carries 15 Marks. (4 X 15 =60)**
**(Each question with internal choice and with maximum of 3 sub questions)**

# Programme Learning Outcomes (PLOs)

**PLO 1: Computational Knowledge**

Apply the knowledge of computing fundamentals, mathematics, and domain concepts to solve complex computing problems in areas such as operating systems, computer networks, algorithms, artificial intelligence, and data analytics.

**PLO 2: Problem solving skills**

Identify, formulate, and analyze complex computing problems using logical reasoning, algorithmic thinking, and appropriate computational techniques.

**PLO 3: Modern Tool Usage**

Select and apply modern tools, technologies, and platforms such as AI tools, data analytics software, block chain systems, and visualization tools for effective problem-solving.

**PLO 4: Experimentation & Data Analysis**

Conduct experiments, analyze data, and interpret results using statistical, computational, and visualization techniques to support decision-making.

**PLO 5: Professional Ethics & Security**

Apply ethical principles and commit to professional responsibilities, including data security, privacy, and legal compliance in computing practices.

**PLO 6: Team Work**

Function effectively as a member or leader in multidisciplinary teams to accomplish shared goals.

**PLO 7: Project Management**

Apply project management principles, including planning, scheduling, risk management, and resource allocation in software and IT projects.

# Sustainable Development Goals (SDGs)

**SDG 4:** Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

**SDG 5:** Achieve gender equality and empower all women and girls.

**SDG 9:** Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation.

**SDG 13:** Take urgent action to combat climate change and its impacts.

**PLOs – COURSE COMPLIANCE MATRIX**

| Sl. No. | Course Title | PLO1 | PLO2 | PLO3 | PLO4 | PLO5 | PLO6 | PLO7 |
|---|---|---|---|---|---|---|---|---|
| 1 | Computer concepts and Problem-Solving using C++ | ✓ | ✓ | ✓ | ✓ | ✓ | - | - |
| 2 | Computer Basics and Programming in C++ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 3 | Data Structures | ✓ | ✓ | - | ✓ | - | ✓ | - |
| 4 | Data Structures using C++ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| 5 | Database Management System | ✓ | ✓ | ✓ | ✓ | ✓ | - | - |
| 6 | DBMS Lab | ✓ | ✓ | ✓ | ✓ | - | ✓ | - |
| 7 | Software Engineering | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ |
| 8 | E-Commerce and E-Governance | ✓ | ✓ | ✓ | - | ✓ | - | - |
| 9 | Object Oriented Programming with Java | ✓ | ✓ | ✓ | ✓ | ✓ | - | - |
| 10 | Java Lab | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 11 | Software Testing | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| 12 | Digital Marketing | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## SDGs – COURSE COMPLIANCE MATRIX

| Sl. No. | Course Title | SDG4 | SDG5 | SDG9 | SDG13 |
|---|---|---|---|---|---|
| 1 | Computer Concepts and Problem-Solving using C++ | ✓ | - | ✓ | - |
| 2 | Computer Basics and Programming in C++ | ✓ | - | ✓ | - |
| 3 | Data Structures | ✓ | ✓ | ✓ | - |
| 4 | Data Structures using C++ | ✓ | ✓ | ✓ | - |
| 5 | Database Management System | ✓ | - | ✓ | - |
| 6 | DBMS Lab | ✓ | - | ✓ | - |
| 7 | Software Engineering | ✓ | - | ✓ | - |
| 8 | E-Commerce and E-Governance | ✓ | - | ✓ | - |
| 9 | Object Oriented Programming with Java | ✓ | - | ✓ | - |
| 10 | Java Lab | ✓ | - | ✓ | - |
| 11 | Software Testing | ✓ | - | ✓ | ✓ |
| 12 | Digital Marketing | ✓ | ✓ | ✓ | ✓ |

# Annexure-A

## B.Sc. (Computer Science)

## Fifth and Sixth Semesters Syllabus (SEP) 2026-27

## University of Mysore

# Curriculum

## Program: B.Sc.
## Subject: Computer Science

| Semester | Course No. | Theory/ Practical | Credits | L-T-P | No. of Hours | Course Title | Marks | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SEE | CIE |
| V | CSM51T | **Theory** | 03 | 3-0-0 | 03 | **Operating Systems** | 80 | 20 |
| | CSM52T | **Theory** | 03 | 3-0-0 | 03 | **Web Technology** | 80 | 20 |
| | CSM52P | **Practical** | 02 | 0-0-2 | 04 | **Web Technology Lab** | 40 | 10 |
| | CSCP5P | **Practical (Compulsory Paper)** | 02 | 0-0-2 | 04 | **AI Tools and its applications** | 40 | 10 |
| VI | CSM61T | **Theory** | 03 | 3-0-0 | 03 | **Computer Networks** | 80 | 20 |
| | CSM62T | **Theory** | 02 | 2-0-0 | 02 | **Python Programming** | 40 | 10 |
| | CSM61P | **Practical** | 02 | 0-0-2 | 04 | **Python Programming Lab** | 40 | 10 |
| | CSCP6P | **Practical (Compulsory Paper)** | 02 | 0-0-2 | 04 | **Power BI** | 40 | 10 |

**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 Learning Outcomes (CLOs)**

Upon successful completion of this course, students will be able to,

**CLO 1:** Gain knowledge of the basic concepts of Operating System, its functions and services.
**CLO 2:** Apply the various concepts related to Process Management and threads.
**CLO 3:** Evaluate the performance of various algorithms used for CPU scheduling.
**CLO 4:** Explain the concepts of Deadlocks.
**CLO 5:** Understand the concepts of memory management in main memory and virtual memory.

**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.

| **Course Contents** | **Hours** |
|---|---|
| **Unit:1:**<br>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:**<br>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, | 11 |

| | |
|---|---|
| Thread Libraries. | |
| **Unit:3:**<br>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:**<br>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 |

**Learning Resources:**


1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 10th Edition,Wiley, 2018.
2. Operating System Principles, Per Brinch Hansen, Prentice Hall of India (PHI).
3. Operating Systems, Milan Milenkovic, McGraw-Hill.
4. Modern Operating Systems, Andrew S. Tanenbaum, 4th Edition, Prentice Hall of India (PHI).

**Semester: V**

| **Course code:** | **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 Learning Outcome (CLOs):**
Upon successful completion of this course, students will be able to,

**CLO 1:** Understand Web Fundamentals
**CLO 2:** Explore the concepts of HTML/XHTML
**CLO 3:** Apply Styling to Web Pages Using CSS
**CLO 4:** Implement Basic Interactivity with JavaScript
**CLO 5:** Demonstrate JavaScript with XHTML for Dynamic Web Pages

**Course Pedagogy**

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

| **Course Contents** | **Hours** |
|---|---|
| **Unit 1:**<br>Fundamentals: A Brief Introduction to the Internet, The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, Multipurpose Internet Mail Extensions, The Hypertext Transfer Protocol, the Web Programmer's Toolbox.<br>Introduction to HTML/XHTML: Origins and Evolution of HTML and XHTML, Basic Syntax, Standard HTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML, Syntactic Differences between HTML and XHTML. | 11 |
| **Unit 2:**<br>Cascading Style Sheets: Introduction, Levels of Style Sheets, Style Specification Formats, Selector Forms, Property-Value Forms, Font Properties, List Properties, Alignment of Text, Color, The Box Model, Background Images, The  and Tags | 11 |

| | |
|---|---|
| **Unit 3:**<br>The Basics of JavaScript: Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Primitives, Operations, and Expressions, Screen Output and Keyboard Input, Control Statements, Object Creation and Modification, Arrays, Functions, Constructors, Pattern Matching Using Regular Expressions, Errors in Scripts. | 11 |
| **Unit 4:**<br>JavaScript and XHTML Documents: The JavaScript Execution Environment, The Document Object Model, Element Access in JavaScript, Events and Event Handling, Handling Events from Body Elements, Handling Events from Button Elements, Handling Events from Text Box and Password Elements, The DOM 2 Event Model, The navigator Object, DOM Tree Traversal and Modification. | 11 |

**Learning Resources:**


1. Programming the World Wide Web, Robert W. Sebesta, 10th Edition, Pearson Education, 2018.
2. Internet and World Wide Web – How to Program, Harvey M. Deitel, Paul J. Deitel, 5th Edition, Pearson Education, 2019.
3. Web Programming: Building Internet Applications, Chris Bates, 3rd Edition, Wiley India, 2020.
4. Introduction to JavaScript Object Notation, Lindsay Bassett, O'Reilly Media, Inc., 2015.

**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 Learning Outcomes (CLOs)**

Upon successful completion of this course, students will be able to,

**CLO 1:** Understand the basic concepts of HTML and explore Hyperlinks, Lists and Images.
**CLO 2:** Demonstrate practical skills, workflows and methods used to create websites using Style sheets.
**CLO 3:** Explore JavaScript and its basic functions and operations.
**CLO 4:** Understand the concepts of Objects and Dynamic HTML.

**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.

**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.

**Evaluation Scheme for Practical Examination:**

| **Assessment Criteria** | | **Marks** |
|---|---|---|
| Writing | One Program from Part A | 15 |
| | One Program from Part B | 15 |
| Execution | Any one of the written Program | 5 |
| Viva Voce | | 5 |
| **Total** | | **40** |

**Semester: V**

| | |
|---|---|
| **Course code:** | **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 Learning Outcomes (CLOs)**

Upon successful completion of this course, students will be able to,

**CLO 1:** Use AI tools to generate text, images, presentations and reports effectively.
**CLO 2:** Apply AI tools for solving basic computational and programming problems.
**CLO 3:** Analyze data and generate meaningful insights using AI assistance.
**CLO 4:** Compare different AI tools and select appropriate tools for specific tasks.
**CLO 5:** Demonstrate the use of AI tools in academic work such as resume creation, translation and documentation.

**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.

**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.

**Evaluation scheme for Lab examination:**

| **Assessment criteria** | | **Marks** |
|---|---|---|
| Writing | One program from part A | 15 |
| | One program from part B | 15 |
| Execution | Any one of the written Program | 5 |
| Viva voce | | 5 |
| **Total Marks** | | **40** |

**Semester VI**

| **Course Code:** | **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 Learning Outcomes (CLOs):**

Upon successful completion of this course, students will be able to,

**CLO 1:** Understand the fundamentals of computer networking
**CLO 2:** Describe the layered architecture of networks
**CLO 3:** Analyze data communication techniques
**CLO 4:** Focus on routing and congestion control mechanisms

**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.

| **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 |

**Learning Resources:**


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

**Semester: VI**

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

**Course Learning Outcomes:**

Upon successful completion of this course, students will be able to,

**Course Learning Outcomes (CLOs):**

**CLO 1:** Understand the basic concepts of Python Programming.

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

**CLO 3:** Identify the methods to create and store strings.

**CLO 4:** Analyze the data using visualization tools.

**Course pedagogy:**

1. Classroom lectures to explain Python fundamentals such as variables, data types, operators, control structures, functions, and modules.
2. Demonstrations to show how Python programs are written, executed, and debugged.
3. Hands-on Practice: Students practice writing Python programs using tools like IDLE, PyCharm, or Jupyter Notebook.

| **Course Contents** | **Hours** |
|---|---|
| **Unit 1:**<br>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:**<br>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 | 10 |

| | |
|---|---|
| Definition- Syntax, Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Illustrative programs. | |
| **Unit 3:**<br>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 |

**Learning Resources:**


1. Introduction to Machine Learning with Python – Andreas Müller
2. Python Programming: An Introduction to Computer Science – John M. Zelle.
3. Python Cookbook – David Beazley & Brian K. Jones
4. Python Crash Course – Eric Matthes.
5. A Byte of Python – C.H. Swaroop.

**Semester: VI**

| | |
|---|---|
| **Course code:** | **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 Learning Outcomes (CLOs):**

Upon successful completion of this course, students will be able to,

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

**Course pedagogy:**

1. Demonstrate Python programs and explain syntax, logic, and program execution.
2. Hands-on Programming Practice: Students write and execute Python programs during lab sessions to understand programming concepts.
3. Problem-Based Learning: Students solve computational problems using Python to improve logical thinking and programming ability.
4. Use of Programming Tools: Students practice coding using tools such as IDLE, PyCharm, or Jupyter Notebook.

**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, between given range X and Y.
5. Python script to display Multiplication Tables
6. Python script to create a calculator program
7. Explore string functions

8. Implementation of python script that takes a list of words and returns the length of the longest one.
9. Python script handles multiple errors with one except statement.
10. 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.

**Evaluation Scheme for Lab Examination:**

| **Assessment Criteria Marks** | | **Marks** |
|---|---|---|
| Writing | One Program from Part A | 15 |
| | One Program from Part B | 15 |
| Execution | Any one of the Written programs | 5 |
| Viva Voce | | 5 |
| **Total** | | **40** |

**Semester: VI**

| | |
|---|---|
| **Course Code:** | **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 Learning Outcomes (CLOs)**

Upon successful completion of this course, students will be able to,

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

**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 modelling, and DAX functions.
3. Practical Assignments and Exercises requiring students to analyze datasets and build visual reports and dashboards.

**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.

**Evaluation Scheme for Lab Examination:**

| **Assessment Criteria** | | **Marks** |
|---|---|---|
| Writing | One program from Part A | 15 |
| | One program from Part B | 15 |
| Execution | Any one of the Written programs | 5 |
| Viva Voce | | 5 |
| **Total** | | **40** |

# CIE, SEE and QP Pattern for 2 credit theory courses:

Total Lecture hours per paper: 30 hours, No. of Units - 3 (10 Hours Each)
Internal Assessment: C1- 5 Marks, C2 - 5 Marks
Semester End Theory Exam: C3- 40 Marks

## Question paper pattern for 2 credits theory courses:

Maximum Marks – 40
**Part A:** Answer any 5 out of 6 questions. Each question carries 2 marks: (5x2=10 Marks)
Question No. 1 to 6
**Part B:** Answer all the Questions. Each question carries 10 Marks. (3 X 10 =30) (Each question with internal choice)

# CIE, SEE and QP Pattern for 3 Credits Theory Courses:

Total Lecture hours per paper: 44
No. of Units- 4 (11 Hours each)
Internal Assessment: C1 - 10 Marks, C2 - 10 Marks
Semester End Theory Exam: C3 - 80 Marks

## Question paper pattern for 3 Credits Theory Courses:

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 all the Questions. Each question carries 15 Marks. (4 X 15 =60) (Each question with internal choice and with maximum of 3 sub questions)

# Programme Learning Outcomes (PLOs)

**PLO 1: Computational Knowledge**

Apply the knowledge of computing fundamentals, mathematics, and domain concepts to solve complex computing problems in areas such as operating systems, computer networks, algorithms, artificial intelligence, and data analytics.

**PLO 2: Problem solving skills**

Identify, formulate, and analyze complex computing problems using logical reasoning, algorithmic thinking, and appropriate computational techniques.

**PLO 3: Modern Tool Usage**

Select and apply modern tools, technologies, and platforms such as AI tools, data analytics software, block chain systems, and visualization tools for effective problem-solving.

**PLO 4: Experimentation & Data Analysis**

Conduct experiments, analyze data, and interpret results using statistical, computational, and visualization techniques to support decision-making.

**PLO 5: Professional Ethics & Security**

Apply ethical principles and commit to professional responsibilities, including data security, privacy, and legal compliance in computing practices.

**PLO 6: Team Work**

Function effectively as a member or leader in multidisciplinary teams to accomplish shared goals.

**PLO 7: Project Management**

Apply project management principles, including planning, scheduling, risk management, and resource allocation in software and IT projects.

# Sustainable Development Goals (SDGs)

**SDG 4:** Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

**SDG 5:** Achieve gender equality and empower all women and girls.

**SDG 9:** Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation.

**SDG 13:** Take urgent action to combat climate change and its impacts.

## PLOs – COURSE COMPLIANCE MATRIX

| Sl. No. | Course Title | PLO1 | PLO2 | PLO3 | PLO4 | PLO5 | PLO6 | PLO7 |
|---|---|---|---|---|---|---|---|---|
| 1 | Operating Systems | ✓ | ✓ | ✓ | ✓ | ✓ | - | - |
| 2 | Web Technology | ✓ | ✓ | ✓ | ✓ | ✓ | - | - |
| 3 | Web Technology Lab | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| 4 | AI Tools and its applications | ✓ | ✓ | ✓ | ✓ | - | - | - |
| 5 | Computer Networks | ✓ | ✓ | - | ✓ | ✓ | - | - |
| 6 | Python Programming | ✓ | ✓ | ✓ | ✓ | - | - | - |
| 7 | Python Programming Lab | ✓ | ✓ | ✓ | ✓ | - | ✓ | - |
| 8 | Power BI | ✓ | ✓ | ✓ | ✓ | - | ✓ | - |

## SDGs – COURSE COMPLIANCE MATRIX

| Sl. No. | Course Title | SDG4 | SDG5 | SDG9 | SDG13 |
|---|---|---|---|---|---|
| 1 | Operating Systems | ✓ | | ✓ | |
| 2 | Web Technology | ✓ | - | ✓ | - |
| 3 | Web Technology Lab | ✓ | | ✓ | |
| 4 | AI Tools and its applications | ✓ | ✓ | - | ✓ |
| 5 | Computer Networks | ✓ | - | ✓ | - |
| 6 | Python Programming | ✓ | - | ✓ | - |
| 7 | Python Programming Lab | ✓ | - | ✓ | - |
| 8 | Power BI | ✓ | - | ✓ | ✓ |