



SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS

CAT-2

WINTER SEMESTER 2025-2026

SLOT: E1

Programme Name & Branch : B.Tech - Computer Science and Engineering (AI & DE)
Course Code and Course Name : BACSE104 - Structured and Object-Oriented Programming
Faculty Name(s) : Dr. K. John Singh, Dr. M. Balasubramani, Dr. P. G. Shynu
Class Number(s) : VL2025260503254, VL2025260503259,
VL2025260503263
Date of Examination : 20-03-2026
Exam Duration : 90 minutes **Maximum Marks: 50**

General instruction(s):

- Answer All Questions
- M - Max mark; CO - Course Outcome; BL - Blooms Taxonomy Level (1 - Remember, 2 - Understand, 3 - Apply, 4 - Analyse, 5 - Evaluate, 6 - Create)
- Course Outcomes
CO2: To apply modular programming techniques by utilizing functions, structures, and files for efficient problem-solving in real-world applications.
CO3: To apply pointers and dynamic memory allocation techniques to manage data efficiently in C/C++ programs.
CO4: To demonstrate object-oriented principles (encapsulation, inheritance, and polymorphism) to design structured and scalable applications.

Q. No	Question	M	CO	BL
1.	A smart agriculture monitoring system collects environmental data from multiple sensors placed in a farm, including temperature, soil moisture, rainfall, and wind speed. The system stores common sensor details such as sensor ID, location, and timestamp using a structure, while the actual sensor reading is stored using a union because only one type of measurement is transmitted at a time to reduce memory usage. The union may store temperature (°C), soil moisture percentage, rainfall level (mm), or wind speed (km/h). After several months of operation, engineers observe inconsistencies in the data because whenever a new reading is stored in the union, the previously stored value is overwritten, resulting in the loss of earlier measurements. In some cases, developers also mistakenly interpret the stored value as a different data type, leading to incorrect analysis of environmental conditions. Write a C program using a structure containing a union to represent the sensor information and readings, and analyze how the use of a union, while memory efficient, introduces drawbacks such as data overwriting, inability to store multiple values simultaneously, and possible misinterpretation of stored data.	10	CO2	BL4
2.	A smart home company is developing a C++ application to manage different electronic devices such as lights, fans, and air conditioners installed in a house. The system should follow basic Object-Oriented Programming concepts such as classes, objects, encapsulation, and member functions. Each device must store details such as device name, power status (ON/OFF), and energy consumption. The company wants the program to allow users to create multiple device objects, switch devices ON or OFF, and display the current status and energy usage of each device. Design a class called Device that encapsulates the device details and provides appropriate member functions to control and display the device information. Write a C++ program to create multiple objects of the class and demonstrate how the system manages different smart devices in the home. Explain how the program uses basic OOP concepts to achieve modularity and data protection.	10	CO3	BL3
3.	A library management system developed using C++ maintains information about books and tracks the total number of books issued in a day. The programmer designs a class Book in which static data members are used to keep track of the total number	10		



SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS

CAT-2

WINTER SEMESTER 2025-2026

SLOT: E1

	of issued books shared by all objects of the class, while different types of constructors such as default constructor, parameterized constructor, and copy constructor are used to initialize book details like book ID, title, and author when objects are created. A destructor is also implemented to display a message when a book object is removed from memory. During system testing, the developers notice certain design limitations: the static member cannot directly access non-static data members of the class, constructors cannot return values or be inherited, and destructors cannot be overloaded or accept parameters. Additionally, they observe that excessive use of static members leads to difficulties in tracking individual object behavior because all objects share the same static variable, and destructors cannot control the exact time when objects are destroyed in some cases. Based on this scenario, write a C++ program that demonstrates the use of static members and different types of constructors in the Book class, and analyze the limitations of using static members, constructors, and destructors in object-oriented program design.		CO4	BL4
4.	A financial auditing company develops a C++ application to manage confidential transaction records between different departments. The system contains a class Account that stores sensitive details such as account balance and transaction history as private members to ensure security. Another class AuditSystem is responsible for verifying and generating audit reports, but it must occasionally access the private transaction data of the Account class without exposing these details to all other parts of the program. Additionally, a separate function compareAccounts() needs to compare the balances of two Account objects to detect suspicious activities. The developers decide to use friend functions and friend classes so that compareAccounts() can directly access private account balances, and the AuditSystem class can examine the private transaction details of Account. However, during a security review, the development team becomes concerned about whether granting friendship might compromise encapsulation and allow excessive access to sensitive data. Explain how friend functions and friend classes can be implemented in C++ for this system and analyze the potential design and security drawbacks of using friendship in object-oriented programming.	10	CO4	BL4
5.	A software development company is creating a C++ application to manage different types of employees working in an organization. The company has a general Employee category that stores common information such as employee ID, name, and basic salary. However, there are different types of employees such as Manager, Engineer, and Intern, each having additional responsibilities and allowances. For example, a manager receives a management allowance, an Engineer receives a technical allowance, and an Intern receives a stipend. The developers decide to implement this system using inheritance in C++ so that the common properties of employees are defined in a base class and the specialized employee types extend it through derived classes. Additionally, the company wants to demonstrate different types of inheritance such as single inheritance (Employee → Manager), hierarchical inheritance (Employee → Engineer and Intern), and multilevel inheritance (Employee → Engineer → SeniorEngineer) to represent the organizational hierarchy. Design appropriate classes using inheritance and write a C++ program to create objects for different employee types, calculate their total salary, and display their details. Explain how different types of inheritance help in code reuse and better organization of the employee management system.	10	CO4	BL3
