



School of Computer Science Engineering and Information Systems
Winter Semester 2024-2025

Slot: C1

Continuous Assessment Test – II

Programme Name & Branch: B. Tech. (IT), B. Tech. – CSE (AI and DE) / (Cyber Security)

Course Name & code: BCSE102L - Structured and Object-Oriented Programming

Class Number (s): VL2024250504093, 4096, 4098, 4100

Faculty Name (s): Dr. Chandra Mouliswaran S./ Dr. John Singh K./ Dr. Thanapal P/ Dr. Shynu P.G.

Exam Duration: 90 Min.

Maximum Marks: 50

Q.No.	Question	Max Marks	CO	BL
1.	You are developing an employee record management system using structures and unions in C. The system should handle multiple employees efficiently, allowing an array of structures to store employee details (int employeeID, char name [50], float salary). Implement a function that accepts this array, sorts employees by salary, and displays the sorted list. Additionally, modify the implementation to use a union, where union Salary {float basicSalary; float totalCompensation;}, and analyze its impact on memory efficiency. Compare the efficiency of using structure vs. union in this context and justify your choice.	10	2	4
2.	Design a student records system in C that stores (int rollNumber, char name[50], float marks[5]). Use dynamic memory allocation to store n students. Implement functions to dynamically accept student details; compute the average marks using a pointer to a structure; display student details and their average; compare dynamic allocation with a fixed-size array and justify its advantages.	10	2	3
3.	Implement a C++ class for a smart home automation system where multiple appliances (like lights, fans, and ACs) are controlled. Define a class Appliance with data members (int applianceID, char type[20], bool status). Use static data members to track the total number of appliances turned ON. Explain how static data members ensure persistence across multiple object instances and analyse their benefits in real-world smart home automation. Demonstrate how <i>this pointer</i> helps differentiate between local and instance variables while updating the status of an appliance.	10	3	3
4.	In a student grading system, a friend function determines the final grade from subject scores. However, it said that utilizing a friend function breaches encapsulation and should be avoided. Implement the grading system with a friend function to calculate the final grade. Rewrite the solution without a friend function and instead use getter methods. Compare both implementations and justify which one is better based on encapsulation, maintainability, security, and memory efficiency.	10	3	4
5.	Implement different types of inheritance in C++ to model a company employee management system and demonstrate its advantages in real-world applications by creating a base class Employee with attributes (int empID, char name[50], float salary). Implement a Single Inheritance where the Manager is derived from the Employee and has additional attributes (int teamSize, float bonus). Implement Multiple Inheritance where TechnicalLead inherits from Manager and another class TechSkills, which contains (char expertise[50]). Implement hierarchical inheritance, where both the intern and full-time employee derive from the employee.	10	3	3