

Final Assessment Test(FAT) - Apr/May 2025

Programme	B.Tech.	Semester	Winter Semester 2024-25
Course Code	BCSE102L	Faculty Name	Prof. Hemalatha K
Course Title	Structured and Object-Oriented Programming	Slot	E1
		Class Nbr	CH2024250501541
Time	3 hours	Max. Marks	100

Instructions To Candidates

- Write only your registration number in the designated box on the question paper. Writing anything elsewhere on the question paper will be considered a violation.

Course Outcomes

CO1: Understand different programming language constructs and decision-making statements; manipulate data as group.

CO2: Recognize the application of the modular programming approach; create user-defined data types and idealize the role of pointers.

CO3: Comprehend various elements of object-oriented programming paradigm; propose solutions through inheritance and polymorphism; identify the appropriate data structure for the given problem and devise a solution using generic programming techniques.

Section - I

Answer all Questions (7 × 10 Marks)

01. #include <stdio.h>
#include <string.h>

union Data {

int i;
float f;
char str[20];

};

int main() {

union Data data;

data.i = 10;

printf("data.i = %d\n", data.i);

data.f = 220.5;

printf("data.f = %.2f\n", data.f);

strcpy(data.str, "Hello, Union!");

printf("data.str = %s\n", data.str);

printf("After modifying data.str:\n");

data.i = 42;

printf("data.i = %d\n", data.i);

printf("data.f = %.2f\n", data.f);

printf("data.str = %s\n", data.str);

return 0;

}

i) Trace the code and write the output of the given code [6 marks]

ii) Change the given C program to add a **Student** structure with a **member ID** within the **Data**. Assign the value 101 to ID and print the output, or display an error message if the operation is invalid. [4 marks]

[10] (CO2/K3)

02. Given two distinct digits d1 and d2, generate N numbers in increasing order using only these two digits 0 or more times. Your task is to create a function that prints all palindrome numbers from a sequence of N generated numbers.

Input Format	Output Format
d1 d2	N number of elements generated with the combination of d1 and d2
Total number of elements to be generated	List of palindrome numbers identified from the N generated numbers.
Sample Input	Sample Output
4 6	Sequence: 4 6 44 46 64 66 444 446
8	Palindrome: 4 6 44 66 444

Expected Constraint

- Use the function to generate elements and find a palindrome
- Store results in an array

[10] (CO1/K3)

03. Develop a C program for a weather monitoring system to analyze temperature data across multiple stations, using pointer arithmetic and the auto storage class. Define a function *analyzeStationData* that takes a *stationId* (int), a pointer to an array of temperatures, the number of readings for the station, a pointer to a float to store the station's average temperature, a pointer to a float to store the station's maximum temperature, and a pointer to an int to store the count of readings above a threshold (25.0°C). Inside *analyzeStationData*, use pointer arithmetic to calculate the station's average temperature, find the maximum temperature, and count readings above the threshold. In the main program, use auto variables to define the maximum number of stations (set to 5) and the number of stations to process, and an auto variable *stationId* to track the station ID (starting from 1). Use an array to store temperature readings for each station.

The program should:

- Take user input for the number of stations and the number of temperature readings per station.
- Take input for the temperature readings and process each station using *analyzeStationData*.
- Display each station's ID, average temperature, maximum temperature, and number of readings above 25.0°C.

Example

Input: Enter number of stations (max 5): 2 Enter number of readings for station 1: 3 Enter 3 temperature readings: 20.5 26.0 28.5 Enter number of readings for station 2: 2 Enter 2 temperature readings: 24.0 27.0	Output: Station 1: Average Temp = 25.00°C, Max Temp = 28.50°C, Readings above 25.0°C = 2 Station 2: Average Temp = 25.50°C, Max Temp = 27.00°C, Readings above 25.0°C = 1
--	--

Expected Constraint:

- Use auto variables to define the maximum number of stations
- pointer to calculate the station's average, maximum temperature, count of readings for above threshold

[10] (CO2/K3)

04. Mr. Ben, Mr. Xen, and Mr. Yen are instructors for a course titled "A" and are responsible for assessing three students: One, Two, and Three. Each student has received marks represented by a, b, and c, respectively, which are **private** and declared as **static** within their respective classes. The instructors aim to determine the maximum, minimum, and average of these marks. Develop a C++ program that defines three classes: One, Two, and Three, which are mutual friends, allowing them to access each other's private data. Each class should have a static data member for storing marks and static member functions for initializing the values. The computation functions—maximum, minimum, and average—should be implemented as static methods in separate classes, operating on the static data members of all three classes. The main program should create instances of these classes, initialize the marks, and invoke the appropriate functions to compute and display the results.

Example

Input	Output
enter a: 50	maximum is 50
enter b: 40	minimum is 23
enter c: 23	average is 37.6667

Expected Constraint:

- Create three mutual classes along with static members and functions
- Marks in private and static declaration in the respective class
- static member function to initialize the marks

[10] (CO3/K3)

05. A banking software company is developing a system to efficiently handle large monetary transactions. As part of this project work, you have been assigned the task of creating a Currency class that facilitates arithmetic and comparison operations on currency values. The class should have two private data members, rupees (integer) and paise (integer), where 1 rupee equals 100 paise. Implement a parameterized constructor to initialize the currency values. Overload the + operator using a member function to allow the addition of two Currency objects, ensuring that any carry-over in paise is correctly handled (i.e., if the total paise exceeds 100, it should be converted into rupees). Additionally, overload the > operator using a member function to compare two Currency objects, returning true if the first currency is greater than the second. Implement a display() function that prints the currency in the format "Rupees: Paise" (e.g., "10:50" for 10 rupees and 50 paise). In the main program function, prompt the user to input values for two Currency objects, perform addition using the overloaded + operator, and compare the two objects using the overloaded > operator. Finally, display the results of both operations, indicating which currency value is greater.

Example

Input

Enter Rupees and Paise for first currency (separated by space): 10 75

Enter Rupees and Paise for second currency (separated by space): 5 50

Output

Sum of both currencies: 16:25

The first currency is greater than the second

Expected Constraint:

- '+' operator - addition of two Currency objects as an overload
- '>' operator -to compare two Currency objects as an overload

[10] (CO2/K3)

06. Create a C++ function template named "Progression" that takes three parameters: total, y, and m. The first two parameters should be of a generic type T, while the third parameter should always be of type **int**. The function should return void and accept total by reference, while y and m by value. The function should compute the total using the formula:

$$\text{total} = 1 + y + 2y + 3y + \dots + my$$

Ensure that the function works for both integer and floating-point values of y. Implement a main() function to demonstrate its usage with different data types.

Expected Constraint:

- Create a function for both integer and floating-point values
- main function with different data types

[10] (CO3/K3)

07. Write a C++ program that reads a list of integers and a list of strings from the user. Implement function overloading to sort both data types in ascending and descending order using STL. The user should be able to choose the order of sorting.

Expected Constraint:

- Read a list of integers and a list of strings
- Function overload to sort using STL (either ascending or descending order)

[10] (CO3/K3)

Section - II

Answer all Questions (2 × 15 Marks)

08. Write a C program to efficiently manage a sports tournament leaderboard using a structure. The program should allow users to input details of multiple teams, including their name, matches played, wins, losses, and points. Additionally, users should be able to update match results by specifying the winning and losing teams, after which the program will automatically update their statistics. The total points calculation should follow the rule: 3 points for a win and 0 points for a loss. Once all match results are recorded, the program should sort the teams in descending order of points, ensuring that teams with the same points are arranged alphabetically. Finally, the leaderboard should be displayed in a structured and well-formatted manner, providing a clear and organized overview of the tournament standings.

Example

Input
Enter number of teams: 3
Enter Team 1 Name: A
Enter Team 2 Name: B
Enter Team 3 Name: C
Enter number of matches played: 3
1 match result
Enter the winning team: A
Enter loser team: C
2 match result
Enter the winning team: C
Enter loser team: B
3 match result
Enter the winning team: A
Enter loser team: C

Output

Tournament Leaderboard:

Rank	Team	Match Played	Win	Loss	Points
1	A	2	2	0	6
2	C	3	1	2	3
3	B	1	0	1	0

Expected Constraint:

- Define a structure to include all data types
- An array of structures for multiple teams
- Score calculations

[15] (CO2/K3)

09. A manufacturing company is implementing an Employee Performance & Bonus System to evaluate employees in two key roles: Production Workers and Quality Inspectors. Production Workers are responsible for assembling products, maintaining machinery, and meeting production targets, while Quality Inspectors focus on inspecting products, identifying defects, and ensuring quality compliance. Salaries are primarily fixed, but performance-based bonuses are provided to encourage productivity and quality assurance. Implement the above using inheritance in C++. A base class, Employee, will store common details such as Employee ID, Name, Age, Base Salary, Role Type, and Years of Experience. The Production_Worker class, derived from the Employee class, will calculate Production scores and bonuses based on units produced and maintenance tasks. Similarly, the Quality_Inspector class, derived from the Employee class, will calculate Quality scores and bonuses based on identified defective items and inspection reports. A final derived class, Salary_Calculator, derived from both Production_Worker and Quality_Inspector, that computes the individual's final salary and displays it along with their details. An array of objects should store information for 'n' employees, compute bonuses, and final salaries. Display employee details and final salaries, and determine the highest-paid employee. The calculation of the Production Score, Quality Score, bonus, and salary is given below:

Formula for bonus and salary calculation:

For Production_Worker

Production Score = (Units Produced * 0.5) + (Maintenance Tasks * 2)

Production Bonus = (Production Score / 50) * (0.1 * Base Salary)

For Quality_Inspector

Quality Score = (Defects Identified * 1.5) + (Inspection Reports * 3)

Quality Bonus = (Quality Score / 50) * (0.1 * Base Salary)

Final Salary:

Final Salary = Base Salary + Production Bonus

(OR)

Final Salary = Base Salary + Quality Bonus

- a. Draw a neat sketch depicting the above inheritance and create an appropriate all-class, data member, and member function (7 Marks) b. Compute bonuses and final salaries. (4 Marks) c. Display employee details along with their final salaries, and determine the highest-paid employee (4 Marks)

Expected Constraint:

- neat sketch
- create a base class, a derived class, and objects
- Function for bonus calculation and salary calculations

[15] (CO3/K3)