



Final Assessment Test (FAT) - May 2024

Programme	B.Tech.	Semester	WINTER SEMESTER 2023 - 24
Course Title	STRUCTURED AND OBJECT-ORIENTED PROGRAMMING	Course Code	BCSE102L
Faculty Name	Prof. Valarmathi Sudhakar	Slot	01
		Class Nbr	CH2023240501321
Time	3 Hours	Max. Marks	100
General Instructions:			
<ul style="list-style-type: none">Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.			

Section - I

Answer all questions (2 X 8 Marks = 16 Marks)

01. In a gymnastics or diving competition, each contestant's score is calculated by dropping the lowest and highest scores and then adding the remaining scores. Write a C program that allows the user to enter eight judges' scores and then outputs the point scored by the contestant. A judge awards point between 1 and 10, with 1 being the lowest and 10 being the highest. For example, if the scores are: 9.2, 9.3, 9.0, 9.9, 9.5, 9.5, 9.6 and 9.8, then the contestant receive a total of 56.9 points. [8]
02. Explain the use of dynamic memory allocation and structures in a C program to manage mobile phone brands and their quantities in a mobile shop. Justify the relevance of dynamic memory allocation for handling an arbitrary number of brands. Use appropriate functions for purchasing and selling phones, including the use of pointers to access and modify data within dynamically allocated memory. [8]
- i. Discuss the advantages of using dynamic memory allocation and structures in a C program to manage mobile phone brands and their quantities in a mobile shop. (2 Marks)
- ii. Provide syntax and code demonstrating the implementation of dynamic memory allocation and functions for purchasing and selling phones.(4 Marks)
- iii. Emphasize the use of pointers to access and modify data within dynamically allocated memory. (2 Marks)

Section - II

Answer all questions (7 X 12 Marks = 84 Marks)

03. Trace the execution step by step and write the output of the following code snippets (4+4+4=12 marks) [12]
- (i)
- ```
#include <stdio.h>
#define display printf
int main()
```

```

{
 int i=0;
 for(display("%d ",++i);display("%d ",++i);display("%d ",++i))
 {
 display("%d ",++i);
 if(i>8)
 break;
 }
 return 0;
}

```

(ii)

```

#include <stdio.h>
#include <math.h>
int main()
{
 int a=4, b=5, c=6, i, d, s = 0;
 s = (b * (2 * a + (b - 1) * c)) / 2;
 d = a + (b - 1) * c;
 for (i = a; i <= d; i = i + c)
 {
 if (i != d)
 printf("%d + ", i);
 else
 printf("%d = %d", i, s);
 }
 return 0;
}

```

(iii)

What will be the output of the program?

```

#include<stdio.h>
int main()
{
 int i=4, j=-1, k=0, w, x, y, z;
 w = i || j || k;
 x = i && j && k;
 y = i || j && k;
 z = i && j || k;
 printf("%d, %d, %d, %d\n", w, x, y, z);
 return 0;
}

```

04. An engineering company keeps track of its employees' names, IDs, designations, years of experience, and salaries. The corporation decides to increase salaries by 12% for employees with more than or equal to 5 years of experience. Create a salary-increment function to increase salaries by 12% for employees with more than or equal to 5 years' experience. [12]

- i. Write a program in C to create a structure Employeeinfo with appropriate members. (5 Marks)
  - ii. Display the employee details after increasing the compensation for the employees based on the aforementioned criteria. (7 marks)
05. In a school management system, there are two classes: Student and Teacher. The Student class contains private data members such as name, age, subjects, and grades (assuming multiple subjects are taught). The Teacher class includes private data members like name, subjects, and teacherID. You need to create a third class called SchoolRecords that is not derived from either Student or Teacher but requires access to their private data members to generate comprehensive reports. [12]
  - i. How would you declare the SchoolRecords class so that it can access private members of both Student and Teacher classes to generate reports? (4 Marks)
  - ii. Write C++ code for the Student and Teacher classes to calculate the average of grades for each student from array of objects across all subjects. If the average is greater than 50, print the result as "pass"; otherwise, print "fail". Sort the student database in ascending order of average marks and print the details of each student and teacher in SchoolRecords. (8 Marks)
06. As a software developer tasked with designing an online shopping system, you are required to create a class hierarchy to represent various products available for purchase. The system should support different types of products, such as electronics and clothing, each with specific attributes and behaviors. [12]
  - i. Design and implement a base class named Product to represent individual products. This class should contain attributes such as name, price, and quantity in stock. Additionally, create intermediate classes for specific product types, such as Electronics and Clothing, which virtually inherit from the Product class. Identify and implement the appropriate inheritance model for these intermediate classes.(6 Marks)
  - ii. Create a derived class named DiscountProduct that virtually inherits from both Electronics and Clothing classes, demonstrating multi-path inheritance.(3 Marks)
  - iii Develop a function named displayAvailableProducts() outside the class hierarchy. This function should be responsible for displaying the names and prices of products that are available for purchase, meaning those with a quantity in stock greater than 0. (3 Marks)
07. Develop a Banking Employee Management System in C++ utilizing operator overloading to streamline various employee management tasks. Create a class named *Employee* to encapsulate banking employee information, including unique identifiers like *Employee ID*, *Name*, *Position/Role* (e.g., Manager, Teller, Loan Officer), and *salary* using necessary constructors, accessor methods, and necessary member functions to manipulate employee data effectively. [4 marks] . Enhance comparison capabilities by overloading the equality (==) and inequality (!=) operators to compare employee objects based on attributes such as ID, name, and salary. Additionally, empower salary increment operations by overloading the plus (+) operator to accommodate percentage-based increments. [4 marks]. Develop essential functions to perform key operations within the system, including adding new employees, displaying all employee details, searching for employees by ID, name, or position, and providing salary increments. [4 marks]. [12]
08. Create a program for managing Teaching Assistants (TAs) at VIT University. This curriculum should concentrate on two major areas: astrobiology and quantum mechanics. Create a configurable 'TA' class with templates that can manage TAs from various departments. Each TA [12]



should contain basic information such as name, ID, course assignment, and role. Use template specialization to change department-specific characteristics, such as showing the department's name. The program should allow users to enter TA details, examine all TAs and their attributes depending on department and role, and quit cleanly when required.

09. Consider a scenario where you've(customer) been diligently saving money for 10 years in a bank account named *AddAmount*. You start with an initial deposit of 50 rupees. During the first year, neither you nor the bank adds any money to your account. However, in the second year, you decide to deposit an additional 200 rupees, but the bank doesn't contribute anything. Starting from the third year onwards, you consistently add 100 rupees each year, while the bank supplements your savings by adding 10% of the previous year's balance as interest. [12]
- Write a C++ program that implements constructors to handle the money for the first, second, and subsequent years. Compute the final amount in your account after 10 years. (6 marks)
  - After 10 years, customer decides to extend the savings scheme for another 5 years with the bank offering the same interest rate. However, the bank imposes a condition: every year, he must add 2 new customers to his bank. If this condition is met, an additional 5 percent interest will be credited to his account annually. Write an inline function named *calculateAdditionalInterest* to calculate the extra interest amount each year. (6 Marks)

