Atal Bihari Vajpayee Indian Institute of Information Technology and Management (ABV-IIITM), Gwalior

(An Institute of National Importance, Ministry of Education, Government of India)

MAJOR THEORY EXAMINATION-2024

Course Code: CS/IT-103 **Date**: 29-04-2024 (Mon)

Course Name: Object Oriented Programming Systems **Time**: 2:00-5:00 PM

Program & Sem: Batch-A (First Year) **Max Marks: 35**

Instructions:

(i) Read all questions carefully and answer accordingly.

(ii) This Question paper contains Eleven questions.

Part A: Objective Based Questions (5 Marks): **Only one correct answer

Answer all the Questions. Each question carries one mark. **(5Q x 1M = 5M)**

Q.NO. 1.If a base class is inherited in protected access mode, then which among the following is true?

a) Public and Protected members of base class become protected members of derived class
b) Only protected members become protected members of derived class
c) Private, Protected and Public all members of base, become private of derived class
d) Only private members of base, become private of derived class

Q.NO. 2 Which among the following best defines abstraction?
a) Hiding the implementation and showing only the features
b) Showing the important data
c) Hiding the important data
d) Hiding the implementation

Q.NO. 3 What is the size of the object of following class (64-bit system)?

```
class student {  int rollno;  char  name[20];  static int studentno;  };
```

a) 24 b) 22 c) 20 d) 28

Q.NO. 4. Which among the following is not a necessary condition for constructors?
a) It must contain a definition body
b) It must not have any return type
c) Its name must be same as that of class
d) It can contain arguments

Q. No. 5. Which Feature of OOP illustrated the code reusability?
a) Inheritance
b) Abstraction
c) Encapsulation
d) Polymorphism

T.O

Part B: Descriptive Questions (30 Marks)

Answer all the Questions. Each question carries five marks. **(6Q x 5M = 30M)**

Q.NO.6 (a) What is the difference between procedure oriented and object-oriented programming? Explain in four points *(Demonstration program not needed)* **(2M)**

(b) Write a C++ class representing a rectangle with attributes length and width. Implement a copy constructor to create a copy of a rectangle object. *(Demonstration program needed)* **(3M)**

Q.NO.7 In what scenarios would you consider using a static member function instead of a regular member function in a class design? *(Demonstration program not needed)* **(2M)**

(b) Predict the output of the following program? Explain the output also in your own words. **(1+2= 3M)**

```
#include<iostream>
using namespace std;
class Point {
            int x;
            public:
                  Point(int x) { this->x = x; }
                  Point(const Point p) { x = p.x;}
                  int getX() { return x; }
        };
int main()
    {
             Point p1(10);
             Point p2 = p1;
             cout << p2.getX();
             return 0;
    }
```

Q.NO.8 (a) What is difference between compile time and run time polymorphism. **(2M)**

(b) Write a C++ program to **overload addition operator** to add two third order polynomials **(3M)**

Q.NO.9 (a) Could you explain how the choice of **inheritance** mode affects the accessibility of base class members in the derived class. *(Demonstration program not needed)* **(2M)**

(b) Write a C++ program to **overload subtraction operator** to subtract two complex numbers. **(3M)**

Q. NO. 10 (a) How multiple inheritance differs from single and hierarchical inheritance. **(1M)**

(b) Consider a scenario where you are developing a simulation game where different types of characters possess various abilities, such as flying, swimming, and shooting. How would you design the class hierarchy using multiple inheritance to represent these abilities while minimizing code duplication and maintaining flexibility? *(Use demonstration C++ Program to support your answer)* **(4M)**

Q.NO.11 Imagine you are developing a library for handling complex numbers in C++. Define a class named 'Complex' to represent a complex number with real and imaginary components. Implement member functions to perform addition, subtraction, multiplication, and division of complex numbers. Now, to compute the angle between two complex numbers, you need to access their private real and imaginary components. Implement a friend function named 'computeAngle()' outside the 'Complex' class that takes two 'Complex' objects as parameters and calculates the angle between them. Ensure to provide necessary validations, such as checking for zero complex numbers, for the angle calculation. *(Use demonstration C++ Program to support your answer)* **(5M)**