JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST-II (EXAMINATION - OCT 2023)

B.Tech. - III Semester (CSE & IT)

COURSE CODE (CREDITS): 18B11CI311 (3) MAX. MARKS: 25

COURSE NAME: Object-Oriented Systems and Programming

COURSE INSTRUCTORS: A. Sharma, A. Kumar, D. Gupta, H.S. Rawat & M. Singh MAX. TIME: 1:30 Hr.

*Note: 1) All questions are compulsory. Marks and COs are indicated against each question. 2) Attempt **questions in the given sequence.** 3) Be precise in your answers. 4) Write neatly.*

Q 1. Create a C++ class called Employee to represent an employee's information, including their name, employee ID and salary. Implement the following requirements: [3] CO3

a) A default constructor with no parameters that initializes the name to "Unknown" employee ID to 0, and salary to 0.0.

b) A parameterized constructor that takes the name, employee ID, and salary as arguments.

c) Overload the assignment operator to allow assigning one Employee object to another.

d) Implement a copy constructor for the Employee class.

In the main () function, create two Employee objects using default and parameterized constructors. Then, create a third Employee object by copying one of the existing objects. Finally, demonstrate the use of the assignment operator by assigning the values of one object to another.

Q 2. Define two user-defined classes - Celsius and Fahrenheit, to represent temperatures in celsius and fahrenheit respectively. Write a program that demonstrates how to convert an object of one user-defined type class Celsius to an object of another user-defined type class Fahrenheit using a constructor and a conversion function. Note: $F = (C * 9/5) + 32$ [3] CO3

Q 3. Write a C++ program to create a text file named "data.txt" with the following content: [3] CO4

```
10
20
30
40
50
```

Assume that the above file has gone through multiple changes after its creation but it still contains at least two numbers. Write another C++ program that reads the "data.txt" file and displays the second last number from the file.

Q 4. Write a base class CBase and its derived class CDerived to accomplish the following: [3]

a) CBase is an abstract class having a pure virtual function vFunction ().

b) CDerived is derived in public mode from CBase and overrides vFunction () to display "*No legacy is so rich as honesty*". CO5

c) In main () function, implement dynamic binding to invoke vFunction () of CDerived.

Finally, elaborate the role of virtual table (vtable) and vtable pointer in the aforementioned scenario.

Q 5. Describe the following (*max. 8-10 sentences*): [2*5 = 10]

a) C++ cannot overload .*, :: and ?: operators. Why?

b) File pointers and modes: i) seekg (n, ios:cur) ii) tellp () iii) ios::ate iv) ios::trunc CO3-5

c) C++ supports virtual destructor, but not virtual constructor. Why?

d) Diamond problem leads to ambiguity in multiple inheritance. Why?

e) What are different ways to prevent object slicing in C++?

Q 6. Mention the **output** of each of following program and also give **brief explanation** (*2-3 sentences*) in support of your answer. Assume the following statements are already there: [1*3 = 3]

#include <iostream> CO3-5

using namespace std;

a)

```
class CTest {
private:
   int iCount;
public:
   CTest(int iCount)
   {
      this -> iCount = iCount;
      cout << iCount;
   }
   CTest(int iCount, int iTemp = 1)
   {
     this -> iCount = iCount * iTemp;
      cout << this -> iCount;
   }
};

int main ()
{
   CTest (2, 3);
   CTest (4);
   return 0;
}
```

b)

```
class CBase {
   public:
   virtual void vTemp (int) = 0;
   void vTemp () {
      cout << "Inside CBase" << endl;
   }
};
class CDerived1 : public CBase {
   void vTemp () {
      cout << "Inside CDerived1"<< endl; }
};
class CDerived2 : public CDerived1 {
   void vTemp (int iCount) {
      cout << "Inside CDerived2"<< endl; }
};
int main () {
      CBase *ptr;
      CDerived1 obj1;
      CDerived2 obj2;
      ptr = &obj1;
      ptr -> vTemp ();
      ptr = &obj2;
      ptr -> vTemp (2);
      return 0;
}
```

c)

```
class CBase {
   private:
   int iCount;
   public:
   CBase () {
      cout << "Inside Constructor" << endl;
   }
   ~CBase () {
      cout << "Inside Destructor" << endl;
   }
};
class CDerived : public CBase {
   public:
   CDerived () {
      cout << "Inside Constructor 1" << endl;
   }
   ~ CDerived () {
      cout << "Inside Destructor 1";
   }
};
   int main () {
      CBase *ptrBase = new CDerived;
      delete ptrBase;
      return 0;
}
```