

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST-III (EXAMINATION - DEC 2022)

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

COURSE CODE (CREDITS): 18B11CI311 (3)

MAX. MARKS: 35

COURSE NAME: Object-Oriented Systems and Programming

COURSE INSTRUCTORS: A.Kumar, D.Gupta, H.Jindal & S.Sharma

MAX. TIME: 2 Hrs.

*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. Discuss the key limitations (*at least three each*) of static member functions and friend functions in C++. [1.5] CO2
- Q 2. Create a class CPublication having two data members - Title and Price, and two pure virtual functions - getData and showData to get and display data. Further, derive following classes which override these functions separately: [3] CO4
- a) CBook having data member - PageCount
  - b) CTape having a data member - Time (representing running time in minutes)
- Demonstrate the use of polymorphism to invoke the member functions of derived classes.
- Q 3. Create a class CStudent having following data members: Name, RollNumber and Branch; and methods - getData() and showData() to fetch and display the information. Further, the program implements exceptions with arguments to get detailed information about the exception which causes if: [3] CO7
- a) RollNumber is less than or equal to zero
  - b) Branch is other than CSE or IT
- Q 4. Briefly explain (*1-2 sentences*) the following terms which are used in exception handling: [3] CO7
- a) terminate ()
  - b) unexpected ()
  - c) set\_terminate ()
  - d) set\_unexpected ()
  - e) uncaught\_exception ()
  - f) bad\_exception
- Q 5. Create a polymorphic base class - CEmployee having data members as name and employee code, and derive two classes - CPermanent and CTemporary. Create an object factory to generate a total of 7 objects of derived classes and use RTTI to know the type and count of the objects. [3] CO4
- Q 6. Create a class CSwap having a member function Swapper to swap two integers. Now use pointer to class members to access data members and Swapper function. [3] CO1
- Q 7. Discuss key benefits (*at least three each*) of the following: [1.5]
- a) Function pointers [1.5]
  - b) Smart pointers
- Q 8. Suppose you are Head of the HR Department in your organization. There has been a recent hiring where five candidates would be joining your organization in a week. Prepare an inheritance-based class diagram where you want to present all the details of the organization induction program for the new employees. [3] CO8

Q 9. Explain briefly the following concepts with one example each:

[1.5] CO8

a) Sequence Diagram

[1.5]

b) Association Relationship

Q10. The activity diagram is made to understand the flow of activities and is mainly used by the business users. Design an activity diagram for order management system having four activities associated with conditions. The activities are: Send Order by Customer, Receipt of the Order, Confirm the Order, and Dispatch the Order. After receiving an order request, conditional checks are performed to decide if it is normal or special order. After identifying the type of order, dispatch activity is performed which marks the termination of the whole activity.

[3]

CO8

Q11. What are the outputs of the following programs? Please give brief explanation (2-3 sentences) in support of your answer. Assume that following statements are already there:

[1\*5]

CO2

CO4

CO7

CO8

```
#include <iostream>
using namespace std;
```

a)

```
class CTest {
    static int i;
public:
    CTest(){++i;}
    static int j;
    static void init(int x) {i = x;}
    void show() {cout << i;}
};
int CTest::i = 100;
int main () {
    CTest::init(10);
    CTest obj;
    obj.show();
    return 0;
}
```

b)

```
class CBase{
};
class CDerived:public CBase{
};
int main () {
    CDerived obj;
    try{
        throw obj;
    }
    catch(CBase obj){
        cout<<"Caught CBase";
    }
    catch(CDerived obj){
        cout<<"Caught CDerived";
    }
    catch (...) {
        cout<<"Caught Exception";
    }
    return 0;
}
```

c)

```
void Test (int x, char ch) throw()
{
    if(x == 0) throw x;
    if (ch == 'a') throw ch;
}
int main ()
{
    try{
        Test (0, 'A');
        Test (1, 'a');
    }
    catch (int x) {
        cout << "Exception!\n";
    }
    catch (...) {
        cout << "Wow! Exception?";
    }
    return 0;
}
```

d)

```
void Test (const int *x)
{
    cout << *x++;
}
int main ()
{
    void (*func_ptr)(const int*);
    func_ptr = Test;
    int x = 2;
    func_ptr (&x);
    (*func_ptr)(&x);
    return 0;
}
```

e)

```
class CTest
{
    mutable int x;
public:
    CTest(int x) {
        cout << (this->x = x++);
    }
};
int main ()
{
    const CTest obj (10);
}
```