

COURSE CODE (CREDITS): 18B11CI311 (3)

MAX. MARKS: 35

COURSE NAME: Object-Oriented Systems and Programming

COURSE INSTRUCTORS: A. Sharma, A. Kumar, D. Gupta, H.S. Rawat & M. Singh

MAX. TIME: 2 Hr.

Note: 1) All questions are compulsory. 2) Attempt questions in the given sequence. 3) Be precise. 4) Write neatly.

- Q. 1 Design a class Student with private static data member totalStudents. Implement a static member function getTotalStudents() that returns the total number of students. Utilise this function in the main() to display the total number of students. [3] CO2
- Q. 2 Implement an abstract class Shape with pure virtual functions for computing the area and displaying the properties of the shape. Create two derived classes, Circle and Rectangle, to implement above functions. Demonstrate the idea of polymorphism in the main() function using pointers to base class Shape. [3] CO4
- Q. 3 Write a C++ program to implement a Stack class using class template having member functions: createStack(), displayStack(), Push(), Pop(), isFull() and isEmpty(). In the main() function, invoke the member functions for the list of integers and strings. [3]
- Q. 4 What are the key benefits and limitations (at least three each) of function templates? [3]
- Q. 5 Implement a C++ program that demonstrates the use of user-defined custom exceptions, exception throwing, catching mechanisms, and re-throwing an exception. Your program should define a class called TemperatureSensor representing a sensor that measures temperature. It should include a function readTemperature() that simulates reading the temperature. If the temperature exceeds a predefined threshold (say 100 degrees Celsius), throw a custom exception TemperatureOutOfRangeException. In the main(), create an instance of TemperatureSensor and simulate reading the temperature. If the temperature exceeds the threshold, catch the TemperatureOutOfRangeException, display an error message, and re-throw the exception to be handled by a higher level in your program. [3] CO7
- Q. 6 Explain the concept of re-throwing an exception in exception handling. How does it contribute to error propagation and handling in programming? Discuss its significance in the context of user-defined custom exceptions and the role it plays in maintaining program integrity. Provide an example scenario illustrating the use of re-throwing an exception and its impact on error management. [3] CO7
- Q. 7 Discuss key benefits (at least three each) of RTTI and pointers to class members. [3]
- Q. 8 Create a polymorphic base class - Employee having data members as name and employee code, and derive two classes - Permanent and Temporary. Create an object factory to generate a total of 5 objects of derived classes and use RTTI to know the type and count of objects belonging to different classes. [3]
- Q. 9 Draw a UML Class Diagram for the following problem statement: [3] CO8
XYZ company management wants to improve their security measures, both for their building and on site. They would like to prevent people who are not part of the company to use their car park. The company

has decided to issue identity cards to all employees. Each card records the name, department and number of a company staff, and gives them access to the company car park. Employees are asked to wear the cards while on the site. There is a barrier and a card reader placed at the entrance to the car park. When a driver drives his car into the car park, he/she inserts his or her identity card into the card reader. The card reader then verifies the card number to see if it is known to the system. If the number is recognized, the reader sends a signal to trigger the barrier to raise. The driver can then drive his/her car into the car park. There is another barrier at the exit of the car park, which is automatically raised when a car wishes to leave the car park. A sign at the entrance displays "Full" when there is no space in the car park. It is only switched off when a car leaves. There is another type of card for guests, which also permits access to the car park. The card records a number and the current date. Such cards may be sent out in advance, or collected from reception. All guest cards must be returned to reception when the visitor leaves the company

Q. 10 Briefly explain activity diagram and composition giving suitable examples of each of these.

[3]
CO8

Q. 11 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*
5 =
5]

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

a)

```
class CB {
public:
    virtual void show() = 0;
};

class CD1 : public CB { };
class CD2 : public CD1 {
    void show() {
        cout << "Derived02"; }
};

int main(void) {
    CD1 *ptr = new CD2;
    ptr -> show ();
    return 0;
}
```

b)

```
template <class T> class CTest {
public:
    static int iCount;
    CTest() { cout << iCount++; }
};

template <class T>
int CTest <T> :: iCount = 1;

int main() {
    CTest <int> obj1;
    CTest <float> obj2;
    CTest <int> obj3;
    CTest <float> obj4;
    return 0;
}
```

c)

```
int main() {
    int x = -1;
    try {
        if (x < 0) {
            throw x; }
    }
    catch (...) {
        cout << "Exception!" << endl;
        throw "Another Exception!"; }
    catch (const char * str) {
        cout << str << endl; }
    return 0;
}
```

d)

```
void Test (float *x, float y = 1.0) {
    cout << ++*x + y << endl;
}

int main () {
    void (*ptr)(float*, float);
    ptr = Test;
    float x = 1.5, y;
    ptr (&x, y);
    (*ptr)(&x, y);
    return 0;
}
```

e)

```
class CTest {
private:
    mutable int x;
    const int y;
public:
    CTest() : x(4), y(5) {cout << x << y;}
};

int main() {
    const CTest obj1;
    return 0;
}
```