

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST-3 EXAMINATION - 2024

B.Tech.- IV Semester (BI)

COURSE CODE (CREDITS): 18B11CI415 (4)

MAX. MARKS: 35

COURSE NAME: OBJECT ORIENTED PROGRAMMING

COURSE INSTRUCTORS: Dr Emjee Puthooran

MAX. TIME: 2 Hours

---

*Note: (a) All questions are compulsory.*

*(b) Marks are indicated against each question in square brackets.*

*(c) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems*

---

- Q1. Differentiate between pass by value and pass by reference in C++, explaining their distinctions through illustrative examples. [CO1, 2M]
- Q2. What is Java Virtual Machine (JVM)? Explain how the implementation of JVM makes Java a platform independent language. [CO1, 2M]
- Q3. What are the differences between an interface and an abstract class in Java? Give the situation where each of them can be used. [CO1, 3M]
- Q4. What is dynamic memory allocation in C++? Discuss its advantages and disadvantages. [CO1, 3M]
- Q5. Explain the concept of exception handling in C++, focusing on the implementation of the try-catch block with an illustrative example. [CO4, 5M]
- Q6. Implement a class template called MyVector to represent a vector (dynamic array) that can store elements of any data type. The class should include methods to add elements, remove elements, get the size of the vector, and access elements by index. Test your class template by creating instances for integers, doubles, and strings, and perform various operations such as adding elements, removing elements, and accessing elements. [CO2, 5M]
- Q7. Design a base class Vehicle with virtual functions accelerate() and brake(). Derive classes Car and Motorcycle from the Vehicle class. Override the accelerate() and brake() functions in each derived class to simulate the acceleration and braking behavior of cars and motorcycles. Write a program to create objects of cars and motorcycles, store them in an array of pointers to the base class, and call the accelerate() and brake() functions using polymorphism to simulate their behavior on the road. [CO3, 5M]
- Q8. Create a C++ program that prompts the user to enter a string and appends the string to a file named "log.txt". Each time the program runs, it should append the entered string as a new line in the file. Additionally, the program should display the contents of the "log.txt" file after appending the string. Ensure that the program handles file opening errors and closes the file properly after use. [CO4, 5M]

Q9. Find the output of the following C++ code snippet. Assume that the code snippet appear inside the `main()` function wherever not given and the header file '`iostream`' is included in the program. [CO3, 5M]

- (a) 

```
char s1[6] = "Hello";
char s2[6] = "World";
char s3[12] = s1 + " " + s2;
cout << s3;
```
- (b) 

```
void square (int *x, int *y) {*x = (*x) * --(*y);}
int main() {
    int number = 30;
    square(&number, &number);
    cout << number;
    return 0; }
```
- (c) 

```
void swap(int &a, int &b) {
    int temp; temp = a; a = b; b = temp;
    cout << "In swap " << a << b; }
int main(){
    int a = 5, b = 10;
    swap(a, b);
    cout << "In main " << a << b;
    return 0; }
```
- (d) 

```
class X {
    int m;
    public:
    X():m(10) {}
    X(int mm): m(mm){}
    int getm(){ return m; } };
class Y : public X {
    int n;
    public:
    Y(int nn) : n(nn) {}
    int getn() { return n; } };
int main() {
    Y obj( 100 );
    cout << obj.getm() << " " << obj.getn() << endl; }
```
- (e) 

```
template<class T>
class A {
    public:
    A(){ cout<<"Created\n"; }
    ~A(){ cout<<"Destroyed\n"; } };
int main(int argc, char const *argv[]) {
    A <int>a1;
    A <char>a2;
    A <float>a3;
    return 0; }
```