



Roll no

# National Institute of Technology Goa

## B.Tech-IV Semester - End Semester Examinations

**Course Name:** Object Oriented Programming**Date:** July 05, 2020.**Duration:** 2 Hours**Course Code:** CS252**Time:** 11 A M**Max. Marks:** 60**Instructions:**

1. Write legibly. Unnecessary details attracts penalty
2. You must complement your explanation with the short fragments of C++ code where appropriate
3. Your programs should compile on any standard C++ compiler and be executed.
4. You should assume that appropriate headers and namespace std are included in each program.
5. The question paper is of **four** pages.

**1.**

- a) What are abstract classes? Can you instantiate an abstract class? Justify. Give a scenario wherein you will be using abstract classes. Differentiate function overloading and overriding (4)
- b) How to achieve run-time polymorphism in C++? With an example program explain run-time polymorphism. How compile time polymorphism is realized in C++? (6)
- c) Consider the following program. Correct the program using all possible approaches, if the program is incorrect. If the program is incorrect, reason why it is incorrect. (6)

<pre>#include&lt;iostream&gt; using namespace std;  class base{ public : inti; };</pre>	<pre>class derived1 : public base{ public: int j; };</pre>	<pre>class derived2: public base { public: int k; };</pre>
<pre>class derived3: public derived1,      public derived2{ public: int sum; };</pre>	<pre>int main() { derived3 ob; ob.i=10; ob.j=20; ob.k=30; /* the rest of the code is on right side*/</pre>	<pre>/* continuation of main()*/ ob.sum= ob.i+ob.j+ob.k; cout&lt;&lt;ob.i&lt;&lt; " " &lt;&lt;ob.j&lt;&lt; " "; cout&lt;&lt;ob.k&lt;&lt; " " cout&lt;&lt;ob.sum; return 0; } /*End of main()*/</pre>

**2.**

- a)** Why exception handling is important? How C++ supports exception handling? With an example program explain the concept of exception handling supported by C++. When terminate () and unexpected () functions are invoked and what are their default actions? **(6)**
- b)** How derived class exceptions are handled in C++? Explain with a program example. How to check whether an exception thrown is caught or uncaught? **(4)**

**3.**

- a)** What are the features of Java programming language? How Java achieves platform independence? Write a Java program to multiply two integers. The numbers should be passed as command line arguments. You have to put your program in a package called MyPackage. Also, mention how to run this program? **(6)**
- b)** In Java classes are often organized as packages. The members of a class of a package can have default, public, protected and private access specification. Elaborate on the visibility of various members under various access specifications to the class, classes within a package, and classes outside a package. While elaborating on visibility control, you have to take inheritance also into consideration. For convenience, you can use table to elaborate visibility control.**(6)**
- c)** What is multithreaded programming? How multithread programming can be done in Java. With a suitable Java program explain the same. The Java thread will be in several states during its lifetime. Elaborate on the life cycle of Java thread. **(5)**

4.

a) What is the output of the following program? (2)

```
int main()
{
    int x = -1;
    cout << "Before try \n";
    try {
        cout << "Inside try \n";
        if (x < 0)
        {
            throw x;
            cout << "After throw \n";
        }
    }
    catch (int x) {
        cout << "Exception Caught \n";
        cout << "After catch (Will be executed) \n";
        return 0;
    }
}
```

b) Give the output of the following program. (2)

```
class Top {
public:
    virtual void MyMemory()
    {
        cout << "I forget" << endl;
    }
    void Disk() { cout << "Space" << endl; }
    void Erased() { cout << "For good" << endl; }
    void ThisExam() {
        Erased();
        MyMemory();
    }
    virtual ~Top() {}
};

class Bottom : public Top {
public: void MyMemory() { cout << "Gone" << endl; }
    void Disk() { cout << "Slipped" << endl; };
    void virtual Erased() { cout << "Rubbed out" << endl; }
};

int main() {
    Top* Ob = new Bottom;
    Ob->MyMemory();
    Ob->Disk();
    Ob->ThisExam();
    Top Obj = *(new Bottom);
    Obj.MyMemory();
    Obj.Disk();
    Obj.ThisExam();
    return 0;
}
```

c) What is the output of the program? (2)

```
class Top
{
public:
    Top() { cout << "Start Top\n"; }
    ~Top() { cout << "End Top\n"; }
};

class Bottom : public Top
{
public:
    Bottom() { cout << "Start Bottom\n"; }
    ~Bottom() { cout << "End Bottom\n"; }
};

class Test
{
public:
    Bottom deal;
    Test() { cout << "Start Test\n"; }
};

int main()
{
    Test me;
    return 0;
}
```

d) What is the output of the program? Assume the size of an integer to be 4 bytes.(2)

```
class base
{
    int arr[10];
};

class b1: public base { };

class b2: public base { };

class derived: public b1, public b2 { };

int main(void)
{
    cout << sizeof(derived);
    return 0;
}
```

5. What is the result of compiling and running each program? (Some programs may not compile). Explain your answer. If errors are present, correct the errors and then answer what is the output of the program.

<p><b>a)</b> Correct the error if any, and then write what will be the output of the program.(2)</p> <pre> class Top {     public:     void Foo(char *string) { cout&lt;&lt; "Top";}     void Foo(float a) { cout&lt;&lt; "Top Too";} }; class Bottom : public Top {     public: void Foo(const int a) { cout&lt;&lt; "Bottom";}     void Foo(float a) { cout&lt;&lt; "Bottom Too";} }; int main() {     Bottom* Bot;     Bot-&gt;Foo(5.5);     Bot-&gt;Foo("cat");     return 0; } </pre>	<p><b>b)</b> What is the output of the following program? (2)</p> <pre> class Pot {      public:     int a;     Pot(int x) { a = x; } }; class Tom : public Pot {      public:     int b;     Tom() : Pot(5) { b = 0; a = 0;} };  int main() {     Tom bar;     cout&lt;&lt;bar.a&lt;&lt;"\n" &lt;&lt;bar.b;     return 0; } </pre>
<p><b>c)</b></p> <p><b>i)</b> What output does the following code generate? Why?(2)</p> <p><b>ii)</b> What output does it generate if you make A::Foo() a pure virtual function?(3)</p> <pre> class A {     public:     A()     {         this-&gt;Too();     }     virtual void Too()     {         cout&lt;&lt; "A::Too()" &lt;&lt;endl;     } }; /*rest of the code is on the right side*/ </pre>	<pre> /*Continuation of c. */ class B : public A {     public:     B()     {         this-&gt;Too();     }     virtual void Too()     {         cout&lt;&lt; "B::Too()" &lt;&lt;endl;     } };  int main(int, char**) {     B obB;     return 0; } </pre>

\*\*\*\*\*All the Best\*\*\*\*\*