Code : 100313

B.Tech 3rd Semester Exam., 2019
(New Course)

OBJECT-ORIENTED PROGRAMMING USING C++

Time : 3 hours *Full Marks* : 70

Instructions :

(i) The marks are indicated in the right-hand margin.

(ii) There are ***NINE*** *questions in this paper.*

(iii) Attempt ***FIVE*** *questions in all.*

(iv) Question No. 1 is compulsory.

1. Choose the correct answer (any *seven*) : 2×7=14

(a) Which feature allows open recursion among the following?

[✓ *(i)*] Use of this pointer

(ii) Use of pointers

(iii) Use of pass by value

(iv) Use of parameterized constructor

(b) If same message is passed to objects of several different classes and all of those can respond in a different way, what is this feature called?

(i) Inheritance *(ii)* Overloading

[✓ *(iii)*] Polymorphism *(iv)* Overriding

(c) Which among the following is wrong?

(i) class student{ }; student s;

(ii) abstract class student{ }; student s;

(iii) abstract class student{ } s[50000000];

(iv) abstract class student{ }; class toppers : public student{ }; topper t;

(d) If two classes combine some private data members and provides public member functions to access and manipulate those data members, where is abstraction used?

(i) Using private access specifier for data members

(ii) Using class concept with both data members and member functions

[✓ *(iii)*] Using public member functions to access and manipulate the data members

(iv) Data is not sufficient to decide what is being used

(e) Which class/set of classes can illustrate polymorphism in the following code abstract class student?

```
{
    public : int marks;
    calc_grade( );
}
class topper : public student
{
    public : calc_grade( )
    { return 10;
    }
};
class average : public student
{
    public : calc_grade( )
    { return 20;
    }
};
class failed{int marks;};
```

(i) Only class student can show polymorphism

(ii) Only class student and topper together can show polymorphism

(iii) All class students, topper and average together can show polymorphism

(iv) Class failed should also inherit class student for this code to work for polymorphism

(f) Consider the following code and select the correct option :

```
class student
{
    int marks;
    public : int*fun( )
    {
        return & marks;
    }
};
main( )
{
student s;
int *ptr = c. fun( );
return 0;
}
```

(i) This code is good to go

(ii) This code may result in undesirable conditions

(iii) This code will generate error

(iv) This code violates encapsulation

(g) Which among the following is correct for the class defined below?

```
class student
{
    int marks;
    public: student( ) { }
    student(int x)
```

```
        {
            marks = x;
        }
    };
    main( )
    {
        student s1(100);
        student s2( );
        student s3 = 100;
        return 0;
    }
```

(i) Object s3, syntax error

(ii) Only object s1 and s2 will be created

(iii) Program runs and all objects are created

(iv) Program will give compile-time error

(h) Does constructor overloading include different return types for constructors to be overloaded?

(i) Yes, if return types are different, signature becomes different

(ii) Yes, because return types can differentiate two functions

(iii) No, return type can't differentiate two functions

[✓ *(iv)* No, constructors doesn't have any return type]

(i) Which constructor will be called from the object created in the code below?

```
class A
{
    int i;
    A( )
    {
        i = 0; cout&lt; &lt; i;
    }
    A(int x = 0)
    {
        i = x; cout&lt; &lt; i;
    }
};
A obj1;
```

(i) Default constructor

(ii) Parameterized constructor

[✓ *(iii)* Compile-time error]

(iv) Run-time error

(j) When an object is passed to a function, its copy is made in the function and then

[✓ *(i)* the destructor of the copy is called when function is returned]

(ii) the destructor is never called in this case

(iii) the destructor is called but it is always implicit

(iv) the destructor must be user defined

2. *(a)* What is Object-oriented Programming (OOP)? Write the basic concepts of OOP.

(b) What do you mean by class and object? 7+7=14

3. *(a)* With an example, explain the terms 'constructor' and 'destructor'.

(b) With an example, explain what virtual function is. 7+7=14

4. *(a)* What do you mean by polymorphism?

(b) With an example, differentiate between run-time and compile-time polymorphism. 7+7=14

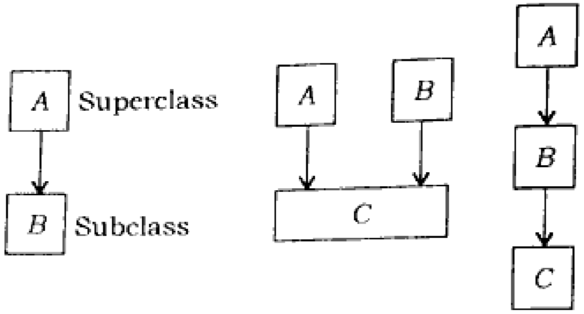
5. *(a)* What is friend function?

(b) What is pure virtual function? 7+7=14

6. *(a)* What is abstract class? Write a program to illustrate. Also outline the advantages of abstract class.

(b) Differentiate between abstract class and interface. 7+7=14

7. What are various types of inheritance, shown in the following figure? 14



(i) *(ii)* *(iii)*

8. *(a)* What is an exception? What do you mean by exception handling?

(b) Explain the keywords : try, catch and throw. 7+7=14

9. With the help of an example program, differentiate between the following : 7+7=14

(a) Overloading *vs.* Overriding

(b) Early binding *vs.* Late binding

★ ★ ★