Model Question Paper

Polymorphism-Part I

12th Standard

Computer Science

Reg.No. : | | | | | | |

I.Answer all the questions.

II.Use blue pen only.

Time : 01:00:00 Hrs

Total Marks : 85

Part -I

10 x 1 = 10

1) Which of the following terms means a name having two or more distinct meanings?

(a) Encapsulation (b) Overloading (c) Inheritance (d) Overriding

2) Which of the following operators cannot be overloaded?

(a) : : (b) - - (c) + (d) ++

3) The word morph means

(a) Triangle (b) Shaped (c) Ellipse (d) Circle

4) Which of the following is one of the facets of C++ that supports object oriented programming?

(a) Function overloading (b) Object overloading (c) Method overloading (d) Oriented overloading

5) The ability of a function to process the message or data in more than one form is called as

(a) Function overloading (b) Recursive function (c) Inline function (d) Function type

6) In C++, polymorphism is achieved through the overloading

(a) function (b) object (c) operator (d) both (a) & (c)

7) The term overloading means a name having

(a) Many forms (b) One meaning (c) Many shapes (d) Two or more distinct meaning

8) The return type of overloaded functions

(a) must be different data types (b) may or may not be same data type (c) must be same data type (d) none

9) Which strategy adopted by the compiler when functions invoked in function overloading?

(a) Match (b) Best (c) Best Match (d) Match Best

10) Operator overloading

(a) Overrule original definition (b) Changesoriginal definition (c) Does not overrule the original definition of the operator (d) None

Part-II

5 x 2 = 10

11) List out the operator that can not be overloaded?

12) What are the rules for function overloading?

13) What is polymorphism? How is it be achieved?

14) Define over loading.

15) What are the thing to be provided by operator overloading?

Part-III

5 x 5 = 25

16) Write the rules for function overloading and operator overloading.

17) What is operator overloading? Explain with an example.

18) Look at the following program and answer the following questions.

```
# include
# include
class distance
{
    int feet,inches;
    public:
    void distance_assign(int f,int i)
    {
        feet=f;
        inches=i;
    }
    void display( )
    {
        count<<"\n feet:"<
        count<<"\t inches:"<
    }
    distance operator+(distance d2)      {
        distance d3;
        d3.feet=feet+d2.feet;
        d3.inches=(inches+d2.inches) % 12;
        d3.feet+=(inches+d2.inches) / 12;
        return d3;
    }
};
void main( )
{
    clrscr( );
    distance dist_1,dist_2;
    dist_1.distance_assign(12,11);
    dist_2.distance_assign(24,1);
    distance dist_3=dist_1+dist_2;
    dist_1.display( );
    dist_2.display( );
    dist_3.display( );
    getch( );
}
```

1. Identify the operator that is overloaded
2. Write out the prototype of the overloaded member function.
3. What types of operator are used for overloaded operator?
4. Write out the statement that invokes the overloaded operator?
5. What is the output of the program?

19) a) Find out the following program.

```
// complex numbers addition
#include
#include
class complex_numbers
{
    float x;
    float y:
    public;
    void assign_data (float real, float imaginary)
    {
        x=real;
        y=imaginary;
    }
    void display_data( )
    {
        count<
    }
    complex_numbers operator + (complex_numbers n1)
    {
        complex_numbers c;
        c.x=x+n1.x;
        c.y=y+n1.y;
        return c;
    }
};
void main( )
{
    clrscr( );
    complex_numbers c1,c2,c3;
    c1.assign_data(1.0,2.0);
    c2.assign_data(3.0,4.0);
    c3 = c1+ c2 ;
    cout<<"\n The first complex number is:";
    c1.display_data( );
    cout<<"\n The second complex number is:";
    c2.display_data( );
    cout<<"\n The sum of two complex number is:";
    c3.display_data( );
    getch( );
}
```

(OR)

b) Write the output for the following C++ program?

```
#include
#include
void fun (char a, int times)
{
for(int i =1; i<=times;i++)
cout<
cout< cout<<'\n' ;
}
void fun (int times = 5, char a = '*')
{
for (int i = 1; i<= times; i++)
cout< cout<<'\n';
}
void main( )
{
fun (3, '*');
fun ( );
}
```

****