B.SC. (I. T.) SEM. - II (2011 COURSE) : SUMMER - 2018

SUBJECT: OOP USING C++

Day : **Tuesday**
Date: **22/05/2018**

S-2018-0974

Time: **10.00 am to 12.00 Noon**
Max Marks. 40

N.B.

1) All questions are **COMPULSORY**.
2) Figures to the right indicate **FULL** marks.

Q.1 **Select proper choice**: **(06)**

a) If there is one class Temp which has one static member variable, then that static variable will belong to,

i) Every instance of class Temp will share the same copy of static variable.
ii) Every instance of class Temp will have its own copy of static variable.
iii) Compilation error
iv) Don't remember.

b) A class derived from another derived class is an example of

i) Single inheritance
ii) Multilevel inheritance
iii) Multiple inheritance
iv) Hierarchical inheritance

c) Select a true statement

i) A friend function can directly access only protected members of the class in which it is declared as friend.
ii) A friend function can directly access only private members of the class in which it is declared as friend.
iii) A friend function can directly access only public members of the class in which it is declared as friend.
iv) A friend function can directly access both protected and private members of the class in which it is declared as friend.

d) Select a true statement about operator overloading.

i) Operator overloading is an example of dynamic binding.
ii) All operators in C++ can be overloaded.
iii) Operator overloading changes complete meaning of the operator.
iv) Operator overloading gives special meaning to the operators by extending semantics of the operator but does not allow to change its syntax.

e) Select a true statement.

i) A destructor can have a parameter and a return value.
ii) Destructor needs to be invoked explicitly in the program before exit.
iii) Destructor can be overloaded
iv) Compilers invoked destructors implicitly upon exit from the program to clean up the storage.

f)

```
class basex
{
    int x;
    public:
        void setx (int y)
        {
            x = y;
        }
```

P.T.O.

```
};
class derived : basex
{

};
```

What is the access level for the member function "setx" in the class "derived" above?

i) private ii) local iii) public iv) protected

Q.2 **Find the output of the following:** **(06)**

a)

```
# include < iostream.h >
# include < conio.h >
int main ()
{
 int a = 10 ;
 int &r = a ;
 r %= 3;
 int *p = &r;
 *p * = 25 ;
  cout << " \n a = "<< a ;
  getch () ;
  return 0 ;
}
```

b)

```
# include < iostream.h >
#include <conio.h>
class test
  {
     static int val;
     public:
            static void show()
              {
                val++;
                cout<<"\nValue="<<val;
              }
  };
int test:: val=200;
int main ()
{
   test::show();
   cout<<"\n Value =" <<val;
   return 0 ;
}
```

Q.3 **Find out errors in the following code:** **(06)**

a)

```
# include < iostream.h >
class Hi
{
    int k;
    float m;
    public:
        Hi();
        Hi(int a, int b): k(a), m(b)
        {

        }
       Hi(Hi r)
       {
           k = r.k;
```

P.T.O

```
        m = r.m;
    }
    void show()
    {
        cout<< "\n k = " <<k;
        cout << "\n m =" << m;
    }
    Hi()
    {

    }
int main ()
{
  Hi h1;
   h1 = Hi(30, 60);
   Hi h2(h1);
   h1.show();
   h2.show();
   return 0 ;
}
```

b)

```
#include <iostream.h>
 class Demo
  {
    int num;
    static int count =0;
    public:
          Demo()
            {
              count++;
              num=count;
            }
        static void show()
          {
            cout<<"\n Object number =" << num;
            cout<< "\n Total number of objects = " << count;
          }
   }; // End of class Demo
int main()
  {
    Demo D1, D2;
    Demo:: show();
    return 0;
  }
```

Q.4 **Explain with example:** **(06)**

a) Explain features of static data members and static member functions with example.

OR

b) Explain Multiple inheritance, with example?

Q.5 Explain Any **FOUR** with examples: **(16)**

a) When a class is derived from the base class in private mode, then how visibility modes of private, protected and public members of the base class get affected in the derived class?

b) What is a destructor? List properties of destructor? Is it mandatory to write destructor in each class? When destructors are useful?

c) The effect of default arguments can be alternatively achieved by function overloading. Discuss with example. (No need to write complete program).

P.T.O

d) What is operator overloading? What is an operator function? Describe syntax of operator functions in overloading unary minus(-) operator and *(binary operator of multiplication).

e) What is inline function? Explain with example.

f) What are reference variables? What is the main advantage of passing arguments by reference?

* * *