Class: BSSE (reg +self) **Day: Wednesday**
Date: 18/03/2020

Lecture time: 9.30 to 12.30

PROGRAM 1:

//** Program to illustrate the working of objects and class in C++ Programming**//

```
#include <iostream>
using namespace std;
class Test
{
   private:
      int data1;
      float data2;

   public:

      void insertIntegerData(int d)
      {
```

```
        data1 = d;
        cout << "Number: " << data1;
      }

     float insertFloatData()
     {
        cout << "\nEnter data: ";
        cin >> data2;
        return data2;
      }
};

 int main()
 {
     Test o1, o2;
     float secondDataOfObject2;
     o1.insertIntegerData(12);
     secondDataOfObject2 = o2.insertFloatData();
```

```
    cout << "You entered " << secondDataOfObject2;
    return 0;
 }
```

PROGRAM 2:

///*private access specifier.cpp***///**

```
#include <iostream>
using namespace std;

class base
{
    private:
      int x;

    protected:
        int y;
```

```
    public:
        int z;

    base() //constructor to initialize data members
    {
      x = 1;
      y = 2;
      z = 3;
    }
};

class derive: private base
{
    //y and z becomes private members of class derive and x remains private
    public:
        void showdata()
```

```
        {
          cout << "x is not accessible" << endl;
           cout << "value of y is " << y << endl;
           cout << "value of z is " << z << endl;
        }
};
int main()
{
    derive a;          //object of derived class
    a.showdata();
     //a.x = 1;   not valid : private member can't be accessed outside of class
     //a.y = 2;   not valid : y is now private member of derived class
   //a.z = 3;   not valid : z is also now a private member of derived class
    return 0;
}
```

PROGRAM 3:

//*** public access specifier.cpp****////**

```
#include <iostream>
using namespace std;
```

```
class base
{
    private:
     int x;

    protected:
        int y;

    public:
        int z;

    base() //constructor to initialize data members
    {
       x = 1;
       y = 2;
       z = 3;
    }
```

```
};

class derive: public base
{
      //y becomes protected and z becomes public members of class derive
      public:
         void showdata()
         {
            cout << "x is not accessible" << endl;
             cout << "value of y is " << y << endl;
             cout << "value of z is " << z << endl;
         }
};
int main()
{
    derive a; //object of derived class
    a.showdata();
   //a.x = 1;   not valid : private member can't be accessed outside of class
    //a.y = 2;   not valid : y is now private member of derived class
```

```
    //a.z = 3;   not valid : z is also now a private member of derived class
    return 0;
}     //end of program
```