Debugging problem

5.1

```
#include<iostream.h>
struct Room
{
        int width;
        int length;
        void setValue(int w,int l)
        {
                width=w;
                length=l;
        }
};
void main()
{
        Room objRoom;
        objRoom.setValue(12,1,4);      //3 variables are given whether only two varables in the
}                                      //    prototype of the function
```

5.2

```
#include<iostream.h>
class Room
{
        int width,height;
        void setValue(int w,int h)
        {
                width=w;
                height=h;
        }
};
void main()
{
        Room objRoom;
        objRoom.width=12;      //private member declared in class 'Room' can’t be accessed in
}                             //public place
```

5.3

```
#include <iostream.h>
class Item
{
private:
        static int count;
public:
        Item()
        {
                count++;
        }
        int getCount()
        {
                return count;
        }
        int *getCountAddress()
        {
        return count;      //when a function is declared as an array of function ,its return type should
        }                  // also be array
};
int Item::count=0;
void main()
{
        Item objItem1;
        Item objItem2;
        cout<<objItem1.getCount()<<' ';
        cout<<objItem2.getCount()<<' ';
        cout<<objItem1.getCountAddress()<<' ';
        cout<<objItem2.getCountAddress()<<' ';
}
```

5.4

```
#include<iostream.h>
class staticFunction
{
        static int count;
public:
        static void setCount()
        {
                count++;
        }
        void displayCount()
        {
                cout<<count;
        }
};
int staticFunction::count=10;
void main()
{
        staticFunction obj1;
        obj1.setCount(5);     //there is no variable needed in the prototype of the function
        staticFunction::setCount();
        obj1.displayCount();
}
```

5.4

```
#include <iostream.h>
class staticFunction
{
        static int count;
public:
        static void setCount()
        {
                count++;
        }
        void displayCount()
        {
                cout <<count;
        }
};
int staticFunction::count=10;
void main()
{
        staticFunction obj1;
        obj1.setCount(5);
        staticFunction::setCount();
        obj1.displayCount();
}
```

5.5

```
#include <iostream.h>
class Length
{
        int feet;
        float inches;
public:
        Length()
        {
                feet=5;
                inches=6.0;
        }
        Length(int f,float in)
        {
                feet=f;
                inches=in;
        }
        Length addLength(Length l)
        {
                l.inches+=this->inches;
                l.feet+=this->feet;
                if(l.inches>12)
        {
            l.inches-=12;
            l.feet++;
        }
        return l;
    }
    int getFeet()
    {
        return feet;
    }
    float getInches()
    {
        return inches;
    }
};
void main()
{
    Length objLength1;
    Length objLength1(5,6.5);        //objLength1 is redefined
    objLength1=objLength1.addLength(objLength2); //objLength2 is undeclared
    cout<<objLength1.getFeet()<<' ';
    cout<<objLength1.getInches()<<' ';
}
```

5.6

```
#include <iostream.h>
class Room;
void Area()
{
    int width,height;
    class Room
    {
        int width,height;
public:
    void setValue(int w, int h)
    {
        width=w;
        height=h;
    }
    void displayValues()
    {
        cout<<(float)width<<' '<<(float)height;
    }
    };
    Room objRoom1;
```

```
        objRoom1.setValue(12,8);
        objRoom1.displayValues();
}
void main()
{
        Area();
        Room objRoom2;    //the class declared in the function can only be used in the function
}
```

Programming exercise