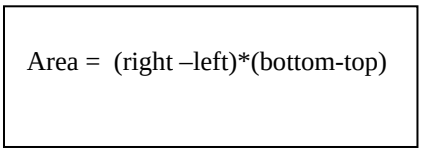
Practice 3

1. Refer to the outline of an incomplete program shown in this question. You need to complete the program by writing the 'box' class definition, create an object called '**button**' and compute the area of 'button'.

(left, top)



(right, bottom)

The class 'box' has the following members:

private data members :

1. **'top'**, an integer variable
2. **'bottom'**, an integer variable
3. **'left'**, an integer variable
4. **'right'**, an integer variable

public member functions:

1. **void setbox(int t, int b, int l, int r);**
   This method is to initialise the data members.
2. **int boxarea(void);**
   This method is to compute and return the area of the 'box'.

To complete the program, you are required to do the following :

a. Write the class definition for '**box**'.

b. Write the member function **setbox** to initialise the data members '**top**', '**bottom**', '**left**' **and** '**right**' with the arguments '**t**', '**b**', '**l**' and '**r**':

c. Write the member function **boxarea()** to compute and return the area of the object created.

The formular is **Area = (right –left)*(bottom-top)**.

d. Fill in the blank in the main program.

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

//Write you class definition and class member functions
// here.

void main()
{
   // create an object call button of class box.

       _____________________________

      button.setbox(5,10,10,20);

      cout << "Area of the box is ";
    // Call the member function boxarea;

      cout << ___________________________________

      cout <<endl;

}
```

2. In the lecture's example, the individual printing of the object's name, ID and GPA data fields get tiresome. Surely we can make use of the computer to do the work? We can create a member function for the SpStudent class, called it studentPrint() to do those work. Think how you should define it: what values does this function return or does it need to return a value at all ? What parameters are needed in this function(i.e. what variables to put between the brackets) to get it to do the work? etc.

   You must then include this new function in the class declaration. Then write the codes of this function in the implementation section:

```
??  SpStudent::studentPrint(??)
{
     //detailed codes to print out the data fields
}
```

   Test the new function in the main program

3. Let's do more practice on member functions. The more practice you have, the more you see how all the syntax, use of parameters all hold together.

   Conceive a member function for SpStudent to compare one object's GPA value with another. Call this compareGPAwith(). Before you jump in, ask the same questions as above: What should be the return type of this function? What parameter should be included between the brackets? Should there be a parameter at all?
   When these questions are answered, start crafting your codes:

```
?? SpStudent:: compareGPAwith(???)
{
//your detailed codes: what they will do? Return
some //values, message codes, do
//some calculations, some comparisons, etc?
}
```

   And of course you need to test it…