Lab 1 - Simple program to calculate area of rectangle by creating a class – Note: **get** method we simply call ***accessors*** and **set** method we call ***mutators***

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

// Rectangle class declaration.
class Rectangle
{
   private:
      double width;
      double length;
   public:
      void setWidth(double);
      void setLength(double);
      double getWidth() const;
      double getLength() const;
      double getArea() const;
};

//****************************************************
// setWidth assigns a value to the width member.    *
//****************************************************

void Rectangle::setWidth(double w)
{
   width = w;
}
```

```
//****************************************************
// setLength assigns a value to the length member. *
//****************************************************

void Rectangle::setLength(double len)
{
   length = len;
}

//****************************************************
// getWidth returns the value in the width member. *
//****************************************************

double Rectangle::getWidth() const
{
   return width;
}
```

```
//*****************************************************
// getLength returns the value in the length member. *
//*****************************************************

double Rectangle::getLength() const
{
   return length;
}

//*****************************************************
// getArea returns the product of width times length. *
//*****************************************************
double Rectangle::getArea() const
{
   return width * length;
}

int main()
{
   Rectangle box;      // Define an instance of the Rectangle class
   double rectWidth;   // Local variable for width
   double rectLength; // Local variable for length

   // Get the rectangle's width and length from the user.
   cout << "This program will calculate the area of a\n";
   cout << "rectangle. What is the width? ";
   cin >> rectWidth;
   cout << "What is the length? ";
   cin >> rectLength;

   // Store the width and length of the rectangle
   // in the box object.
   box.setWidth(rectWidth);
   box.setLength(rectLength);

   // Display the rectangle's data.
   cout << "Here is the rectangle's data:\n";
   cout << "Width: " << box.getWidth() << endl;
   cout << "Length: " << box.getLength() << endl;
   cout << "Area: " << box.getArea() << endl;
   system ("pause");
   return 0;
}
```

```
This program will calculate the area of a
rectangle. What is the width? 3
What is the length? 2
Here is the rectangle's data:
Width: 3
Length: 2
Area: 6
Press any key to continue . . .
```

Lab 2 - Use **dynamically allocate the objects** with class

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

// Rectangle class declaration.
class Rectangle
{
   private:
      double width;
      double length;
   public:
      void setWidth(double);
      void setLength(double);
      double getWidth() const;
      double getLength() const;
      double getArea() const;
};

//****************************************************
// setWidth assigns a value to the width member.    *
//****************************************************

void Rectangle::setWidth(double w)
{
   width = w;
}
```

```
//****************************************************
// setLength assigns a value to the length member.  *
//****************************************************

void Rectangle::setLength(double len)
{
   length = len;
}

//****************************************************
// getWidth returns the value in the width member.  *
//****************************************************

double Rectangle::getWidth() const
{
   return width;
}
```

```
//*****************************************************
// getLength returns the value in the length member. *
//*****************************************************

double Rectangle::getLength() const
{
   return length;
}

//*****************************************************
// getArea returns the product of width times length. *
//*****************************************************
double Rectangle::getArea() const
{
   return width * length;
}
```

```
int main()
{
   double number;        // To hold a number
   double totalArea;     // The total area
   Rectangle *kitchen;   // To point to kitchen dimensions
   Rectangle *bedroom;   // To point to bedroom dimensions
   Rectangle *den;       // To point to den dimensions

   // Dynamically allocate the objects.
   kitchen = new Rectangle;
   bedroom = new Rectangle;
   den = new Rectangle;

   // Get the kitchen dimensions.
   cout << "What is the kitchen's length? ";
   cin >> number;                              // Get the length
   kitchen->setLength(number);                  // Store in kitchen object
   cout << "What is the kitchen's width? ";
   cin >> number;                              // Get the width
   kitchen->setWidth(number);                   // Store in kitchen object

   // Get the bedroom dimensions.
   cout << "What is the bedroom's length? ";
   cin >> number;                              // Get the length
   bedroom->setLength(number);                  // Store in bedroom object
   cout << "What is the bedroom's width? ";
   cin >> number;                              // Get the width
   bedroom->setWidth(number);                   // Store in bedroom object
```

```
   // Calculate the total area of the three rooms.
   totalArea = kitchen->getArea() + bedroom->getArea()
             + den->getArea();

   // Display the total area of the three rooms.
   cout << "The total area of the three rooms is "
        << totalArea << endl;

   // Delete the objects from memory.
   delete kitchen;
   delete bedroom;
   delete den;
   kitchen = 0;    // Make kitchen point to null.
   bedroom = 0;    // Make bedroom point to null.
   den = 0;        // Make den point to null.

   return 0;
}
```

```
What is the kitchen's length? 32
What is the kitchen's width? 45
What is the bedroom's length? 43
What is the bedroom's width? 65
What is the den's length? 43
What is the den's width? 43
The total area of the three rooms is 6084
Press any key to continue . . .
```

Lab 3 – Using ***constructor***

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

// Demo class declaration.

class Demo
{
public:
    Demo();      // Constructor
};

Demo::Demo()
{
    cout << "Welcome to the constructor!\n";
}

//****************************************
// Function main.                        *
//****************************************

int main()
{
    cout << "This is displayed before the object is created\n\n";
    Demo demoObject;        // Define a Demo object.
    cout << "\nThis is displayed after the object is created.\n";
    return 0;
}
```

```
This is displayed before the object is created

Welcome to the constructor!

This is displayed after the object is created.
Press any key to continue . . .
```

Lab 4 – Using ***destructor***

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

class Demo
{
public:
   Demo();      // Constructor
   ~Demo();     // Destructor
};

Demo::Demo()
{
   cout << "Welcome to the constructor!\n";
}

Demo::~Demo()
{
   cout << "The destructor is now running.\n";
}

//*********************************************
// Function main.                             *
//*********************************************

int main()
{
   Demo demoObject;  // Define a demo object;

   cout << "This program demonstrates an object\n";
   cout << "with a constructor and destructor.\n";
   system("pause");
   return 0;
}
```

```
Welcome to the constructor!
This program demonstrates an object
with a constructor and destructor.
Press any key to continue . . .
```