

# CS142 Computer Programming II

Kuwait University  
Computer Science Department  
Dr.Maha Alabduljalil

---

## Class Private Members

- **point.h and .cpp:** Create a class **Point** containing:
  - Two private data members representing the coordinates of a point:
    - **x** of type *integer*
    - **y** of type *integer*
  - A parameterized constructor:
    - Note: The coordinates should be set to zeros if no arguments are sent.
  - Public member functions:
    - **void print()** – To print the coordinate of **Point** in the form (**x**, **y**).
    - **int getX()** – To return the **x** coordinate of the point.
    - **int getY()** – To return the **y** coordinate of the point.
    - **double getDistance(Point p)** – To compute and return the distance between the primary **Point** and **Point p**. Use the following distance formula:  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- **rectangle.h and .cpp:** Create a class **Rectangle** that represents a rectangle that is parallel to the axes in a Cartesian coordinate system. The class should contain:
  - Two private data members:
    - **lowLeft** of type *Point* that represents the lower left vertex of the rectangle on the Cartesian coordinate system.
    - **upRight** of type *Point* that represents the upper right vertex of the rectangle on the coordinate system.
  - Constructor(s)
  - Public member functions:
    - **void print()** – To print the vertices of **Rectangle**.
    - **double getPerimeter()** – To compute and return the perimeter of the **Rectangle**. Make use of the **getDistance()** function in class **Point**.
    - **double getArea()** – To compute and return the area of the **Rectangle**. Make use of the **getDistance()** function in class **Point**.
- **main.cpp:** Write a main function to enter two points from the user representing the lower left and upper right vertices of the rectangle. Instantiate an object of class **Rectangle** and store the information input from the user. Print the information stored in your rectangle then print its *area* and *perimeter*.

### Example output:

```
Enter the lower left coordinate point of your rectangle: -1 3
Enter the upper right coordinate point of your rectangle: 5 7

Rectangle has lower left vertex @ Point: (-1, 3) and upper right vertex @ Point: (5, 7)
The area of your rectangle is: 24
The perimeter of your rectangle is: 20
```