2E3 Lab 2

1. Write a C++ header file (coordinates.h) for a class Coordinates which is used to hold coordinates in 2-dimensional coordinate system (x, y). Define a default constructor which initialises (x, y) to zero and another constructor which allows (x, y) to be initialised to any integer values.

   Note: you can make everything public in this class.

   Write the C++ implementation file for Coordinate (coordinates.cpp).

2. Write a C++ header file (rect.h) for a class Rect which is used to hold a rectangle represented by four integers:
   topLeft coordinate values: x0, y0, bottomRight coordinate values, x1, y1

   Define the following 3 constructors:
   a. default constructor which initialises both sets of coordinate values to (0, 0)
   b. a constructor with two Coordinates parameters, i.e., pass in two objects of type Coordinates and set the initial values of the rectangle from these.
   c. a constructor with 4 integer parameters (x0, y0, x1, y1), and set the coordinates of the rectangle to these values.

   Define the following methods:
   d. a method that calculates and returns the area of the rectangle
   e. a method findUnion(…) which calculates and returns the union of two Rects (i.e., the minimum Rect that encloses them both)
   f. a method that prints out the coordinates and the area of a triangle.

   Write the C++ implementation file for Rect (rect.cpp).

3. Write a main program that creates two rectangles as follows:

   1. Top left = (1,2), Bottom right = (3,1) – call constructor with 4 ints
   2. Top left = (2,3), Bottom right = (4,1) – call constructor with 2 Coordinates

   The program should print out the coordinates and area of each rectangle, then create its union and print out its details too.

4. Provide a method to allow the user to input the coordinates and change the main program to read in the coordinates of two new rectangles – then do the same as in 3.

You have one week to complete this program, and the marks will be assigned as follows:

This week – Week 2:
0 = did not attend lab
1 = attended the lab session

Max = 1 point

Next week – Week 3:

0 = did not attend lab
1 = attended the second (marking) lab session but have very little working code or the code does not compile. **Your code MUST compile in order to get a mark higher than 1**
1 = correct definition and implementation of the Coordinates class
4 = correct definition and implementation of the Rectangle class
2 = correct implementation of the main program
1 = adding and using the user input method correctly

Max = 9 points

Overall maximum for two labs together= 10 points

The program needs to be DEMONSTRATED for marks to one of the demonstrators before the end of your lab session next week AS WELL AS SUBMITTED ONLINE through Blackboard.

Hints:

- Don't forget `#pragma once` at the start of your include files
- Draw out the two rectangles first you can see what the correct union should be.
- Implement the Coordinates class first and write a main program to create an object of that class, and make sure that works before moving on to the Rectangle class.
- Create simpler versions of each method first to make sure that the parameters are correct and compiling before implementing fully.