

# Lab 6

## CSE 165: Object Oriented Programming

### Spring 2022

(100 points)

This programming assignment has four tasks, complete each task as instructed. Write a separate file for each of the following tasks. To submit your assignment, please organize your code in the folder "Lab6" by placing your code in its corresponding sub-folder. For example, store your code for task 1 in the following directory "Lab6/1/". Then, submit the compressed version of folder Lab6 to CatCourses. Submissions must arrive by one minute before the lab section of week 8 (3/7 – 3/11). All of the files you need for this programming assignment are available in a ZIP archive file called "Lab6.zip".

#### 1. Vector Class (20 Points)

Design a 2D vector class called `Vec`. Class `Vec` will contain two float data members `x` and `y`, two constructors: one default constructor, and one constructor from two coordinates, an `add` method, and a static member `Vec` object called `null` to represent the (0,0) null vector. Your `Vec` class should allow the `vectors.cpp` file to run without generating any error messages.

#### 2. Rectangle Class (20 Points)

Design a 2D rectangle class called `Rect`. Class `Rect` will need four floats to represent a rectangle which are the upper-left corner `x` and `y` position and the length and the width of the rectangle. Your `Rect` class will at least have a constructor receiving the four floats defining the rectangle and a method called `contains` for classifying if a given `Vec` is inside or outside of the rectangle. The coordinate system is the usual 2D XY Euclidean plane. Your `Rect` class should allow the `rectangles.cpp` file to run without generating any error messages. You may use the file `Vec.h` from the previous exercise.

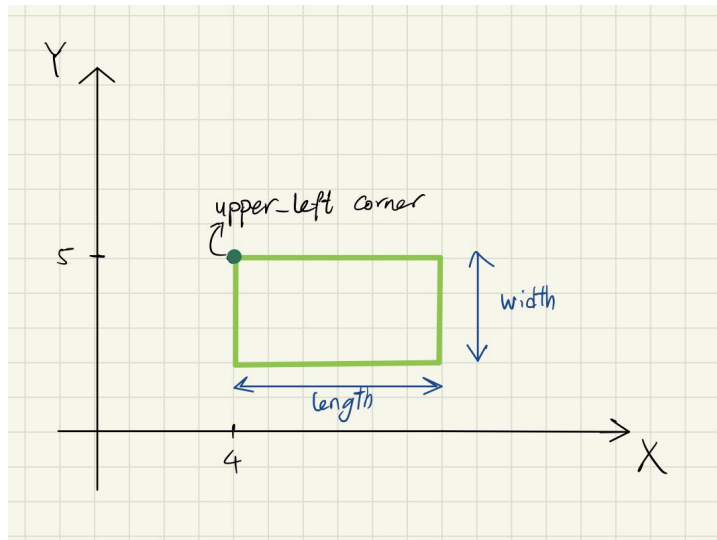


Figure 1: Sample rectangle

### Example:

Consider the following sample rectangle, and two vectors.

rectangle: 4.0, 5.0, 6.0, 3.0

vector1: 6.0,3.0

vector2: 2.0,4.0

The output of method `contains` for first vector is **True** and for second vector is **False**.

### 3. Point-Inside-Polygon (20 Points)

Create a **Stash** class specifically for storing **Rect** objects and call it **RectStash**. Add a default constructor and a destructor to correctly initialize your **RectStash** class. Then write a program that reads several lines as input. Each line will contain four floating point numbers defining a 2D rectangle in the **Rect** format described above. Read the rectangles adding them to a **RectStash** object. Stop reading rectangles when your program reads four negative float values. After this point you will start reading a series of 2D points, and for each 2D point you will print the classification of each point in respect to all previously read rectangles. The classification should print "in" or "out" according to its result. Stop your program when you read vector (-99,-99). You can reuse your **Rect.h** and **Vec.h** files.

### Example:

Input:

2.0 2.0 4.0 6.0

-4.0 -4.0 8.0 6.0

-1.0 -1.0 -1.0 -1.0

3.0 3.0

0.0 -6.0

3.0 0.0

-99 -99

The output for first vector is "**out out**", for second vector is "**out in**", and for the last vector is "**in out**".

#### 4. Qt framework (40 Points)

Starting from this week we will be learning the Qt framework, which is a cross-platform application development framework in C++. This will help you improve your object oriented programming skills and prepare you for the final project.

The task for this week is to install the Qt framework and create your first Hello World GUI app. Qt is a commercial product, however it has a free open source license, so we will be using it in this course. The first task is to install the framework on your machine. Follow this [link](#) to download it for your specific operating system (we tested it on Windows, MacOS, and Ubuntu 20.04.1, and it run without any issues). On that page navigate to **Downloads for open source users**, and at the bottom you can find the link named **Download the Qt Online Installer**. For Windows and MacOS, just open the downloaded installer file, and follow the installation. For Linux machines, first make the downloaded installer file executable and run it. We used the following command for that:

```
chmod +x qt-unified-linux-x64-4.0.1-1-online.run ./qt-unified-linux-x64-4.0.1-1-online.run
```

More instructions for installing on Ubuntu can be found [here](#).

Once you have installed the Qt framework, run the Qt Creator IDE and follow [this Beginner tutorial](#).

The tutorial is quite long, so for this week complete only up to the chapter about Qt class hierarchy. Design your own pretty button from the tutorial, and submit the `main.cpp` file of your pretty button example.