CSE 165: Object Oriented Programming
Spring 2022
Lab 7 (100 points)

This programming assignment has six tasks, complete each task as instructed. To submit your assignment, please organize your code in the folder "Lab7" by placing your code in it's corresponding sub-folder. For example, store your code (source, header, and assets) for task 1 in the following directory "Lab7/1/". Then, submit the compressed version of folder Lab5 to CatCourses. Submissions must arrive one minute before the lab section of week 8 (3/14 – 3/18).

1 References in C++ (5 points)

Study the file refs.cpp. It makes use of a function named triple which returns a number multiploed by three. Provide this function in a file named refs.h.

```
Sample output from refs.cpp
    >> Enter a number: 5
    << 15
```

2 Polymorphism in C++ (5 points)

Study the files catsDogs.cpp, Cat.h, Dog.h, and Animal.h. The program does not compile in its current form. A piece of code is missing from Animal.h. Add that code and submit the new Animal.h.

```
Sample output from catsDogs.cpp
    >> Woof, woof!
    >> Meow, meow!
```

3 Copy Constructors I (10 points)

Get the file objects.cpp. It instantiates some Object classes and prints out the value of the count variable, which simply keeps track of how many instances of Object have been created.
Your task is to implement the Object class. It only needs to have a static member named count, which should be incremented each time a new instance of the class is created. You should also provide the appropriate constructors. Do not worry about destructors for this exercise. Save your class in a file named Object.h.

```
Sample output from objects.cpp:
>> 3
```

4 Copy Constructors II (10 points)

In the previous questions, where we were counting the objects, there were 3 instances. One of them was created because we passed the object into the function f by value. Modify the function so that this does not happen. Your program should report there are 2 objects, not 3. Upload your modified version of objects.cpp
HINT: Only modify the objects.cpp file.

```
Sample output from objects.cpp:
>> 2
```

5 Virtual Methods (30 points)

You are required to complete the following tasks:

* Create a AppWindow class that will contain as member a rectangle AppRect. AppRect will define it's position and size. AppWindow will have the following two constructors:

```
AppWindow( int x, int y, int w, int h);
AppWindow( const AppRect& r);
```

* Write a method to retrieve the rectangle:

```
const AppRect& AppWindow::rect();
```

* Write the resize() virtual method as described below. Everytime this method is called, your implementation is supposed to change the coordinates of the internal rectangle of the window. (Method resize will be called by the system every time the window is resized)

```
virtual void AppWindow::resize( int w, int h);
```

* Now write two classes deriving the AppWindow: CircleWin and RectWin. These classes will have correct constructors as needed and will override the resize method such that:

a) CircleWin will get the minimum dimension (among w and h) and will print it like this: radius: (min), where (min) is the minimum dimension.

b) RectWin will print the area like this: area: (area)

Upload your three classes as a single header file named Header.h. your code will be tested for correctness using the file virtualMethods.cpp.

```
Sample output from virtualMethods.cpp:
>>radius: 2
>>area: 8
>>radius: 1
>>area: 4
```

6 Qt Continued (40 points)

This week, we will learn more about Qt application development framework. We will follow the same tutorial as last week and learn about Qt class hierarchy, parenting system, QWidget, and the most important features of the Qt framework, i.e. Signals and Slots. We will further learn how Signals and Slots transmit information and how they can be customized based on our needs.
Next your will design a calculator as show below using the Qt framework. Your calculator should the following functionalities:

```
a. Numbers ranging from 0 to 9
b. Allow addition, subtraction, multiplication, and division
c. Clear button to clear digits in the output
d. Should be able to perform arithmetic operation
 when the equal sign is pressed.
e. You are allowed to change the color of the button.
You should make sure that you have some form of style to the button.
For example, you can change the color or font.
```

Link to Qt tutorial: https://wiki.qt.io/Qt_for_Beginner
HINT: You can follow any online lectures or vides like this one: https://www.youtube.com/watch?v=FhV1ZEVNK08