**CSC 117 – Introduction to Computer Science**

**Lab 14 – Functions (Part 2)**

In this lab, you will cement your understanding of writing and calling functions. Today, we will also be writing functions that will help streamline our work with graphics in future labs.

**For all labs going forward, you’ll be expected to create a main() function and call it in your program. You will have other functions, too.**

Please place all your code in **one file** and submit through Moodle as usual.

**Step 1: Write a function that creates a labeled button.**

Write a function createButton() that creates a rectangular button with a text "label." The function should take **nine arguments**: the graphical window object (GraphWin); the floating point coordinates of the corners of the button, (x1, y1) and (x2, y2); the background color of the button (a string); the text label to be superimposed on the button (also a string); the text size (int); and the text color (string). The function should actually display the button along with its text label on the window. (The text label should have the same dimensions as the rectangle.) It should return two objects: the Rectangle and the Text box that together constitute the "button."

**Remember: Don't overwrite your arguments with assignment statements!**

**Also remember: When calling your function (e.g., from main), you need one argument (object) per parameter in your function definition.**

Write a main() function that calls your function. In it, create a window and call createButton() to place a green button in the center of the window that says "Click Me" in large characters. Test your code before continuing to the next step.

**Step 2: Write a function that detects whether a button (Rectangle object) has been clicked.**

Write a function clicked() that takes a Rectangle and a Point as its two arguments. The function should return True if the Point is on the Rectangle and False otherwise.

Extend your main() function from Step 1 to get a mouse click from the user. Then call your clicked() function to determine if the user clicked the button you created in the previous step. If the user clicked the button, change its color to red. If the user did not click the button, loop back and continue getting mouse clicks until the user clicks the button. (Hint: What pattern do you need for this?)

**Step 3: Write a function that computes the distance between two points.**

Write a function distance() that takes two Point objects and returns the floating point distance between the two points. Specifically, we want the "Euclidian" or "as-the-crow-flies" distance. Recall that we can compute this with the so-called **distance formula**:

You can compute the positive square root with the built-in math.sqrt() function. (For this, you will need to import the math module.)

To test your code, extend your main() function to get two mouse clicks from the user. Then display the distance between the points to the console. When you are confident your program is working correctly, go on to the next step.

**Step 4: Write a function that determines whether a Circle has been clicked.**

Write a function circleClicked() that returns True if a Circle object has been clicked. (The function is analogous to the clicked() function you wrote in a previous step.) The function should take a Circle object and a Point as its two objects. It should return True if the Point is on the Circle and False otherwise.

Finally, extend your main() function to draw a blue circle on the window. Then get a mouse click from the user. Call your function circleClicked() to determine whether the user clicked the circle. If the user clicked the circle, change the color of the circle to red. If the user did not click the circle, get another click and loop back until the user clicks the circle and turns it red.