**Camden Fergen**

**Section 2**

**Lab 05: Pre-Lab**

**Part 1: Introduction to Conditionals**

Start by reading through the entire lab document prior to coming to lab so you know are able to have a better idea of what we are doing this week. In addition to that, feel free to start on the lab or bring any questions that you may have so the TAs can go over them. Next, answer the question below based on having read the lab document.

**Question 1:** Based on the lab document, how many functions (not including the main) do you need in your program? List function stubs for those possible functions. A function stub is the following: **type function\_name(arg1, arg2, arg3)**

3 or more

**type close\_to(tolerance, point, value)**

**type end(bool buttonPress)**

**type movement(char cur[50], char past[50])**

**Part 2: “Close To” Function**

When writing your code, it can be useful to develop functions outside of what is required if you are reusing the same code over and over. This allows you to get rid of redundant code and have a more concise program. Thus, it is suggested that you develop a function for this lab called close\_to().

**Question 2:** Based on the description of what the close\_to() is supposed to do, come up with a visual representation of it. Hint: Think of using a graph or number line to represent each part of it.

3 in the graph would be “close to”

**Question 3:** Now that you should hopefully have a generalized idea of how the close\_to()function works, write the code for it. You can either write the actual C code or pseudocode for it.

If (value >= point-tolerance && value <= point+tolerance)

Return true

Else

Return false

**Question 4:** Lastly, when developing your code, it can be useful to have variables that act as “flags”. A flag in the code is essentially a variable that holds a single character or integer that you check in your code. What is one part of the code that it may be useful to have a flag? How would you use it?

Instead of using a flag, use a bool for something such as if it has moved. So a boolean that would be false if it has moved and a function that checks if it has moved and if it has then print out the new movment

**TA Check Off:**

**Pre-lab and Attendance TA Signature:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Question 1:** | **/5** |
| **Question 2:** | **/3** |
| **Question 3:** | **/4** |
| **Question 4:** | **/3** |
| **Total:** | **/15** |