CSC108 Lab 2: Functions & Logic

September 15th – 19th 2025

1 Overview

Welcome to your second lab in CSC108. You will do a bit more coding than the previous lab, and you will be evaluated on both coding style and correctness. You are submitting this code to MarkUs, and we will be checking for plagiarism. You may show your code to your TA in a breakout room or through a private Piazza post.

The focus of this lab will be on writing individual functions that do not depend on one another, although you are free to use helpers if you deem them necessary.

2 Running Doctests

Download `lab2.py` from the labs section on Quercus and take a look at the file. You'll notice that there are *doctests* within the *docstrings* of each function. *Doctests* are snippets of code within the *docstring* of a function which can be used to test that the function works as intended. They are not usually a comprehensive suite of tests, but are generally enough to ensure basic functionality. In order to run those tests, you'll need to add a block of code at the bottom of the file (copy-paste or type it in):

```
if __name__ == '__main__':
    import doctest

    doctest.testmod()
```

This block of code just tells Python that if this is the main script that's being run, then execute the code that is inside the **if** statement. The `import doctest` line imports the doctest module, and `doctest.testmod()` parses the docstrings of each function, looking for valid doctests to run. Don't worry if you don't fully understand what it does right now, you'll go through this again later.

If all your tests pass, you should see **no output** (this is good). If some fail, then you should see something like this:

```
**********************************************************************
File "C:/Users/user/lab2.py", line 10, in __main__.is_square
Failed example:
    is_square(9)
Expected:
    True
Got nothing
**********************************************************************
1 items had failures:
   1 of   1 in __main__.is_square
***Test Failed*** 1 failures.
```

Once you have doctests running (you should see a bunch of failures when you run the file), move on to the next section.

3 Lab Tasks

You will have three functions to implement for this lab, and they are detailed in the steps below:

1. Consider the function `my_and(a,b)`, which returns `True` if and only if `a` and `b` are `True`.
   Note: this is essentially recreating the already built-in Python "`and`" operator, but without actually using the keyword.

2. Implement the function `my_and(a,b)`, but only use the two operators: `not`, `or`.
   Hint: consider the expression `not(a and b)`, reason this out; is there an equivalent way to write it without the use of and?
   Implement this function according to the description.

3. Complete the function `exists_triangle`, that takes three floats as input and returns `True` only when there is a proper triangle with these sides. Note that a proper triangle has area > 0. Do not use an "`if`" statement. A doctest is provided for you in the docstring, but you should add more.

4. Complete the function `is_square` according to its docstring. Note that a precondition given is that the input will always be an integer (i.e., ≥ 0). Additionally, please keep the restrictions that the docstring mentions in mind.

4 Final Check!

Once you are finished, submit `lab2.py` to MarkUs. You'll receive a grade once everyone has submitted and automarking has been done. See you next week!