CSCA20 Exercise 1

Deadline(s)

There will be a pre-grading run on Wednesday September 19 at 11:59pm. If you submit your work before this deadline, you will receive feedback from the automarker on what grade you would receive on your submission, as well as any errors in your code we've found.

The actual deadline for this exercise will be Friday September 21 at 11:59pm. The latest version of e1.py you submit before this deadline (including possibly anything you've submitted before the pre-grading deadline) will be examined to determine your final grade for this exercise.

UPDATE: there will be another pre-grading run on Friday 11:59pm, and the actual deadline will be **Sunday September 23 at 11:59pm**.

Deadline-related suggestions

I suggest you treat the pre-grading deadline as your actual deadline; ie. try to get all of your work done before then. If you earn a perfect grade, then you're done! And if you happen to make any errors, then you'll have two days to correct them in order to increase your final grade. Not many courses give you this opportunity, so make the best of it.

We will mark the newest version submitted not later than the deadline. Don't submit just once; instead, submit at least once well in advance, so that you know what you're doing, and then keep submitting new versions as you do more of the exercise. That way, if you run out of time on the last function that you just can't do, you'll still get marks for the others.

What you'll be required to do

In this exercise, you'll be practicing the implementation of some simple functions. Within the file e1.py, you will find the (unfinished) definitions and docstrings of four functions, which you will be required to complete by writing code for their bodies. The areas where you are supposed to write your code are clearly denoted with comments.

The docstring for each function describes what each function is supposed to do and/or return. It's up to you to implement them to do what their docstrings require.

What you should also do

In the comments of e1.py, you'll see that I've given suggestions for testing your functions. While we won't be grading you on how well you test your functions, we *will* be grading you on how well your functions *work*. And the best way for you to make sure your functions work is to test them thoroughly.

In general, for any given function argument, you should follow the following guidelines:

- For arguments which are numbers, you should try (where applicable; not every function requires the usage of all of these):
 - A value of 0
 - A positive value
 - A negative value

- Integer values
- Non-integer values
- For arguments which are strings (where applicable; not every function requires the usage of all of these):
 - The empty string (often forgotten)
 - $\circ~$ A string with one character in it
 - $\circ~$ A string with multiple characters in it
 - A string containing letters (a, b, c, etc.)
 - A string containing digits (0, 1, 2, etc.)

For functions which accept multiple arguments, you should generally try all combinations of these for all arguments.

Attention to detail

Remember that Python cannot make any intelligent corrections for you. It will only do exactly what you tell it to. Any typos or logical errors will make your code behave completely differently from what you want it to! So the best way to make sure you haven't made any of these errors is to test! (See above)

Style

The adherence of your modified e1.py to the PEP8 style guide will count towards your grade. After you've implemented and tested your code, run it through the online style-checker provided in the Resources section of the course website and then fix all of the style errors it will no doubt find in your code.

It may seem tedious to write according to the PEP8 style guide when Python will understand non-PEP8-compliant code just as well. But remember that when you're programming, you're programming for humans as well. Writing code which follows style conventions will help humans read it. And as you practice following these conventions, they will become habit and you might eventually find yourself doing it by default.

Thus, the second last thing you should do before you submit is to run your code through the PEP8 style checker. The last thing you should do is the sanity check (see below).

Sanity check

If nothing else, make sure your code runs *without any syntax errors*. In lecture, you've seen how easy it is to forget a bracket and end up with invalid Python code. Even the best programmers make these mistakes regularly. If the code you eventually submit has these kinds of errors, you *will get a 0* on your exercise grade, no matter how well you did everything else! So the last thing you should do before you submit is a final sanity check to make sure you can at least run your code without any errors!

What to hand in

Submit your completed e1.py on MarkUs https://markus.utsc.utoronto.ca/csca20f18/?locale=en

Caution: this is our first exercise too

Watch for announcements about things that had to be fixed. Be prepared to post queries on the discussion board describing troubles you had; it might be our fault, not yours.

Specifically, it may take a while for us to get the pre-marking working. Watch for announcements about that on the discussion board.

The buck stops where?

The terms "our" and "us" in this section refer to the course instructor, who regretfully admits his responsibility for stuff that goes wrong.