

CSCA20 - Lab 1

Input/Output and Variables

Learning Objectives

This lab focuses on interacting with a user through `input` and `print` statements, and also manipulating and using variables, specifically numbers and strings.

Prelab

Hopefully by now, you are comfortable running python files at home and/or in the lab. But just to be sure, you should arrive at your lab with working code that asks a user for their name, and then greets them, as below:

```
What is your name? Brian
Hello Brian
```

Demonstration & Evaluation

Successfully completing this lab will demonstrate mastery of: user input/output.

If you show up with your prelab done and work hard during the tutorial, you will be awarded with 'attempted' on this skill. If you successfully demonstrate working code by the end of the tutorial, you will be awarded with 'practiced'. At the beginning of next week's tutorial, there will be a quiz where you will be asked to modify your working solution to the lab (if you can't get your solution working during the tutorial, you can continue to work throughout the week asking for help on Piazza). If you can successfully complete that quiz, you will be awarded with 'modified' for this skill.

This is our first chance to evaluate a skill, but it won't be our last. If you can't get your code working by the end of the lab, or you can't write the quiz next week, Don't panic. You will have multiple opportunities to demonstrate each level of each component (especially input/output, pretty much every lab from here on in allows you to demonstrate this one).

The Scenario

The Universal Tap-dancing School of Casablanca (UTSC) has asked you to build them a tool to help format their tap-dancing students scores in a way that is compatible with their record keeping system. Eventually, we'll learn how to do this with files and loops... but for now, they just need a system that will prompt the user for the student's name, student number, courses and grades, and then prints out a properly formatted student record with a grade average.

The Input

The system should ask for a student's given name first, then their family name (because that's the way most students will be addressed in person), then their student number, followed by a list of the 4 courses they took this semester ¹. Then, for each of the courses, the system should ask the user for the grade the student got in that course. Because UTSC faculty aren't always super smart², it's important that each time we ask them for input, we remind them which course they're inputting the grade for. (reminder: the courses in the "Enter grade" prompts should match whatever the users entered in the "Enter course code" prompts).

So the prompts for input would look something like this (the parts after the semi-colons are user input)

```
Enter student given name: Brian
Enter student family name: Harrington
Enter the student number for Brian Harrington: 12345
Enter course code #1: CSCA20
Enter course code #2: ENGA01
Enter course code #3: HISB02
Enter coures code #4: XYZC03
Enter grade for CSCA20: 99
Enter grade for ENGA01: 86
Enter grade for HISB02: 75
Enter grade for XYZC03: 58
```

The Output

After all the information is entered, your system should output the student record in the format:

```
Family Name, Given Name, Student Number, Average
```

Where **average** is the average grade (add up the grades and divide by 4). So the output for the input above would be

```
Harrington,Brian,12345,79.5
```

¹fortunately for us: UTSC isn't as flexible as UTSC... every student takes exactly 4 courses so there's no need to worry about students having more/fewer

²I've been told here I should clarify which UTSC I'm referring to in the previous footnote... but I refuse to

Hints

Here are a few hints that might help you with this assignment:

- Even though the student number is a number, do you ever need to store it as a number? The type of a variable isn't about its meaning, it's about what you plan to do with it
- you're going to need to store 2 names, 1 student number, 4 course codes and 4 averages, so start out by planning variables appropriately
- When a user types input into the shell, python will assume it's a string unless you specifically tell it otherwise
- You aren't responsible for your code working if the user doesn't type sensible input, so if you ask for a grade and the user types "BAD", your code is allowed to crash (for now)

Extra Practice

If you get this working and want to move beyond the basics, there are several things you can do:

- Think about your variables: there is a way to do this without actually storing 4 separate grade variables
- Can you make it so that output is always formatted into neat columns that line up regardless of the number of letters in the person's name? (you'll need to look up the documentation for `str`)
- (once you've learned selection) come back to this and see if you can do some error checking: if the user inputs an invalid course code or grade, tell them
- (once you've learned loops + lists) ask the user how many courses the student took, ask them first and then prompt them for that many codes & marks