Lab 3.2 Number Cruncher and Function Flow

Instructor Guide

Overview

This lab will begin with the **Control Flow Paper Programming** that is meant to help them more thoroughly understand local variables, and function flow. Following that, students will be able to put their new skills to the test by creating functions that complete some mathematical calculations. If students finish early they can delve into the many more math operators and experiment with different kind of functions.

Learning Goals

- Understand the differences between print and return
- Return ends a function
- Return sends a value back to wherever the function was called from
- The values of parameters inside of functions can change depending on what value is sent in the function call
- A local variable is local to a function, and doesn't affect the global variable

Personal Growth Goals

- Attention: Function flow and the difference between local and global variables is a very complex topic that students need to give all their attention towards.

Skills Required

- Complete understanding of print, input, variables, strings, conditionals, and syntax for all of the previously mentioned skills

- Introduction to the function flow, and the order in which a program follows through the lines of code
- Knowledge of integers, and simple math operators

Resources Required

- Computers for either every student or every pair of students
- Python 3 and a text editor needs to be installed on all the computers
- One mentor per 2-3 students
- A projector to project the central instructor's computer
- Pencils for each student

Instructor Preparation

1. Make sure all the computers students will use have Python and a text editor (right now, we use Pyzo) installed (check to see that students have a way to save/access files)
2. Load the following programming files onto each computer:
    a. 03_02_01_syntax_practice.py
    b. 03_02_02_number_cruncher.py
3. Have one **Functional Flow Paper Programming** printed out for each student.

In Depth Description of Lab Activities

Phase 1: Setup

1. Before the students arrive, open the following files in a text editor on each computer:
    a. 03_02_01_syntax_practice.py
    b. 03_02_02_number_cruncher.py
2. Have one printed out **Functional Flow Paper Programming** for each student on the desks, and pencils available if needed.

Phase 2: Introduction | Review

1. As students are coming into the room, verbally review with them how to create and use functions. Additionally, you can review integers as a type, and the many ways to manipulate them using operators.

Phase 3: Functional Flow Paper Programming

1. The teacher should go through questions a) and b) with the students explaining what the correct solutions are and why, using relatable examples written up on their computers.
2. Go over how return is different from print, and where the return value is sent.
3. Teach what a local variable is, and how it is created when using functions with parameters.

4. Have the students complete the final question on their own, gathering help from mentors if needed.
5. Have the class review over the correct answers, answering any questions along the way.

Phase 4: Functional Fun Syntax Practice Activity

1. Using the syntax guide have the students complete challenges in this activity.
2. Students might not completely understand what return does at this point, have the mentors more thoroughly discuss the purpose, and how it is different from printing.
3. A bad habit that can occur is coping of the template in the paper programming, rather than reproducing or trying to problem solve to develop another solution. It would be ideal if students could think through and come up with a 'unique' to these problems, but if some are struggling, then having them complete these challenges using the paper programming as a template is fine.

Phase 5: Number Cruncher Activity

1. The number cruncher activity is meant to reiterate the wide variety functions can serve, and continue to learn how to use math operators.
2. The students should try to code trace through the code for this activity before running it, then complete the challenges. The syntax guide and mentors will be the main source of help for this activity.

Phase 6: Pack up | Review

1. Mentors should lead a discussion with their students based on the question: What do you think that you can do with these tools now?
2. This question may be useful to use this as a form of review, and can also be used to increase interest in the subject.

Lesson Plan

(:10) means that this part should be done by the tenth minute of the lesson

1. Setup (:0)
2. Introduction | Review (:10)
3. Functional Flow Paper Programming (:20)
4. Functional Fun Syntax Practice Activity (:35)
5. Number Cruncher Activity (:55)
6. Pack up | Review (:End)

Take Away

Students should have a good understanding of the mathematical operators in python, and be able to use them to complete basic computations. Additionally, students should gain a more thorough understanding of integers as their own unique type, and helper functions as serving a specific purpose.