Dan Oranski

3/02/20

IT FDN 100 B

Assignment 06

Observations of Assignment 06

# Introduction

This document covers the work through of Assignment 06. In this assignment, students applied the concepts of functions, parameters and arguments to the CD inventory program created in the previous module. Docstrings were introduced and implemented into the coding standard for the course. The concepts were unified by modifying the script from Assignment 05 to utilize functions in place of lists with docstrings.

# Functions

Functions are used to grouped statements conveniently together in the same portion of a script. The grouped statements are then available for use via calling the function and it’s associated arguments. This allows the programmer to call the function repeatedly without need to retype the statements contained within the function. It is also more convenient to use functions as a means to simplify the layout of the script. Functions utilize parameters and arguments in order to pass data both in and out.

# Arguments/Parameters

Parameters, or arguments, are data values that can be used for processing. Parameters can be passed into and out of functions, and can be used in other functions.

## Naming Convention for Arguments and Parameters

It is good programming practice to have an established naming convention for arguments/parameters. Having an established set of naming conventions makes it easier to review the code and prevents squabbles over syntactic preferences. It also assists code quality review tools in focusing their reporting on issues. The standard Python syntax is as follows:

* + UpperCamelCase for class names
  + CAPITALIZED\_WITH\_UNDERSCORES for constants
  + lowercase\_separated\_by\_underscores for other names

The schema utilizing a prefix for variable names is called Hungarian Notation.

# Return Values

Return values provide a means to utilize parameters/arguments outside of the function in which they are defined. Without the use of return values, only the function defining a parameter can call that parameter. In the interest of segregating functions in scripts by the purpose for which they are serving, return values allow programmers to avoid using global variables. This provides a means to avoid overloading user defined variables.

LAB 06-A directed students to convert a provided script to utilize attributes and return values. LAB 06-B directed students to then modify the script from LAB 06-A to utilize tuples to handle the data. The results of these labs are displayed below.

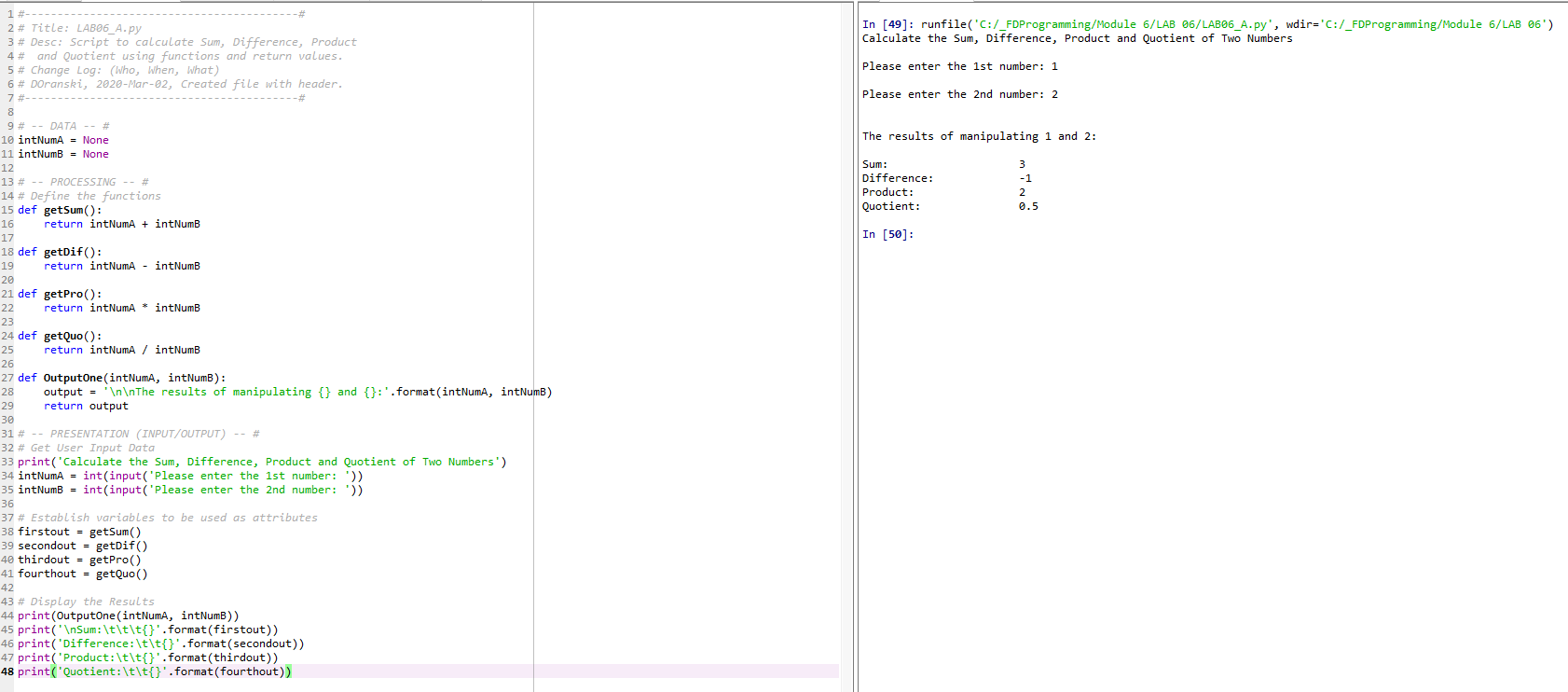


Figure 1 - LAB06\_A

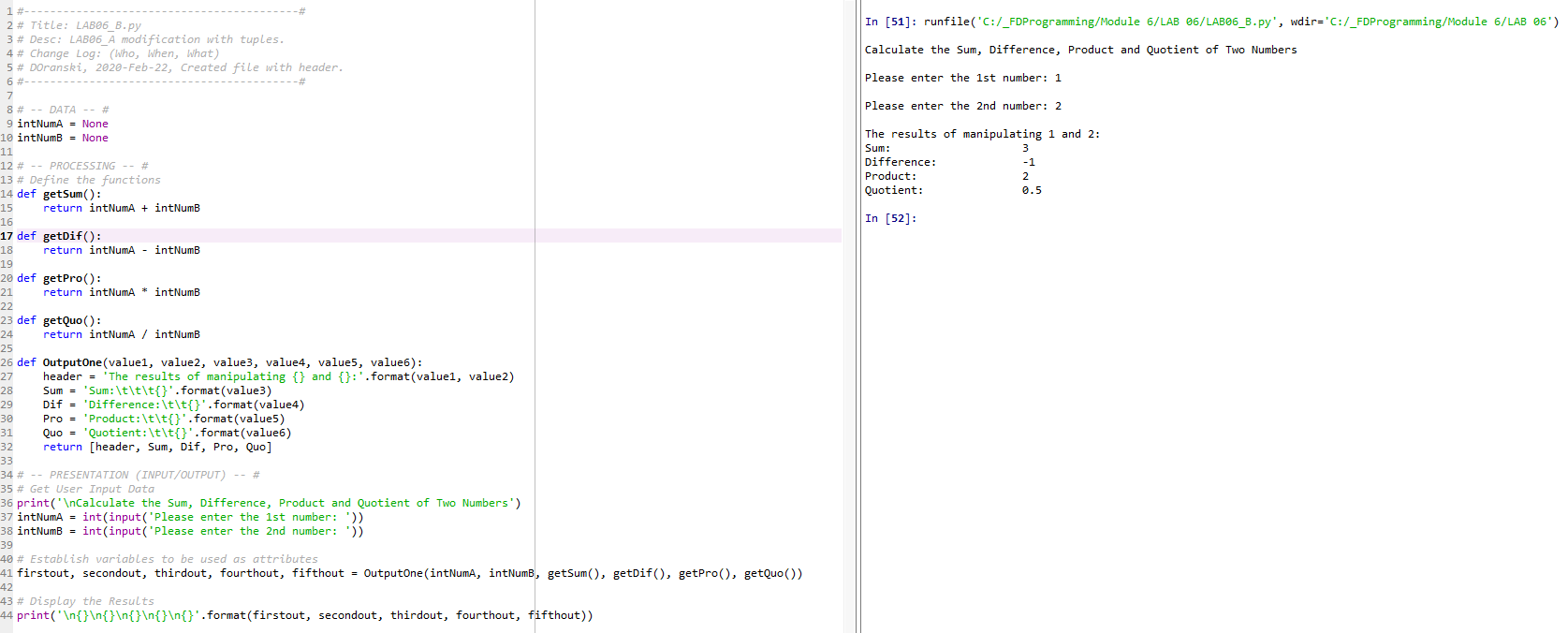


Figure 2 - LAB06\_B

# Positional vs. Named Arguments

When you call a function, you can either include the name of the parameter as it is defined in the function declaration, or allow Python to implicitly use the arguments and assign them a corresponding position based on the sequence of their appearance. It is also possible to mix positional and named arguments.

# The None Keyword (Again)

The None keyword is a special data type in Python with exactly one value: None. It is commonplace to see None used to indicate the absence of values. Using None as the default argument allows for simple checks to change the behavior of functions.

# Docstrings

Docstrings are used to provide additional information regarding the purpose of a function or class. There are automated uses for document docstrings, and when highlighting a class or function name and pressing Ctrl + I, the docstrings populate a popup with the contents of the docstring. This is how tooltips are displayed for the pre-loaded functions in Python such as len(), etc.

LAB 06-C directed the students to recreate the tasks in LAB 06-A using docstrings. The results are displayed below.

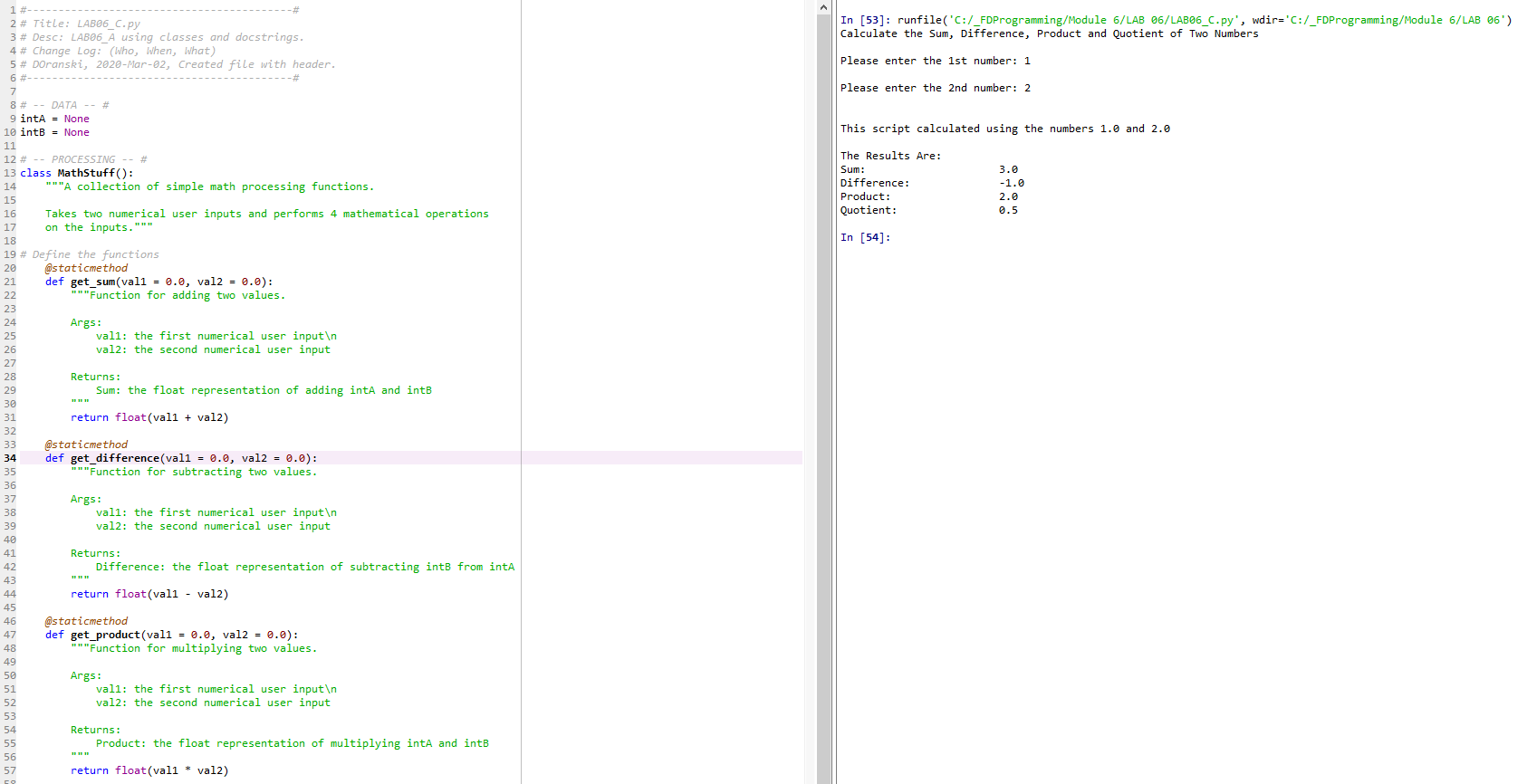


Figure 3 - LAB06\_C

# Application of Learning Objectives

Tying everything together, the student was asked to modify the Assignment 05 Python script by utilizing functions and docstrings. The script was created, troubleshot, and run in the Spyder IDE.

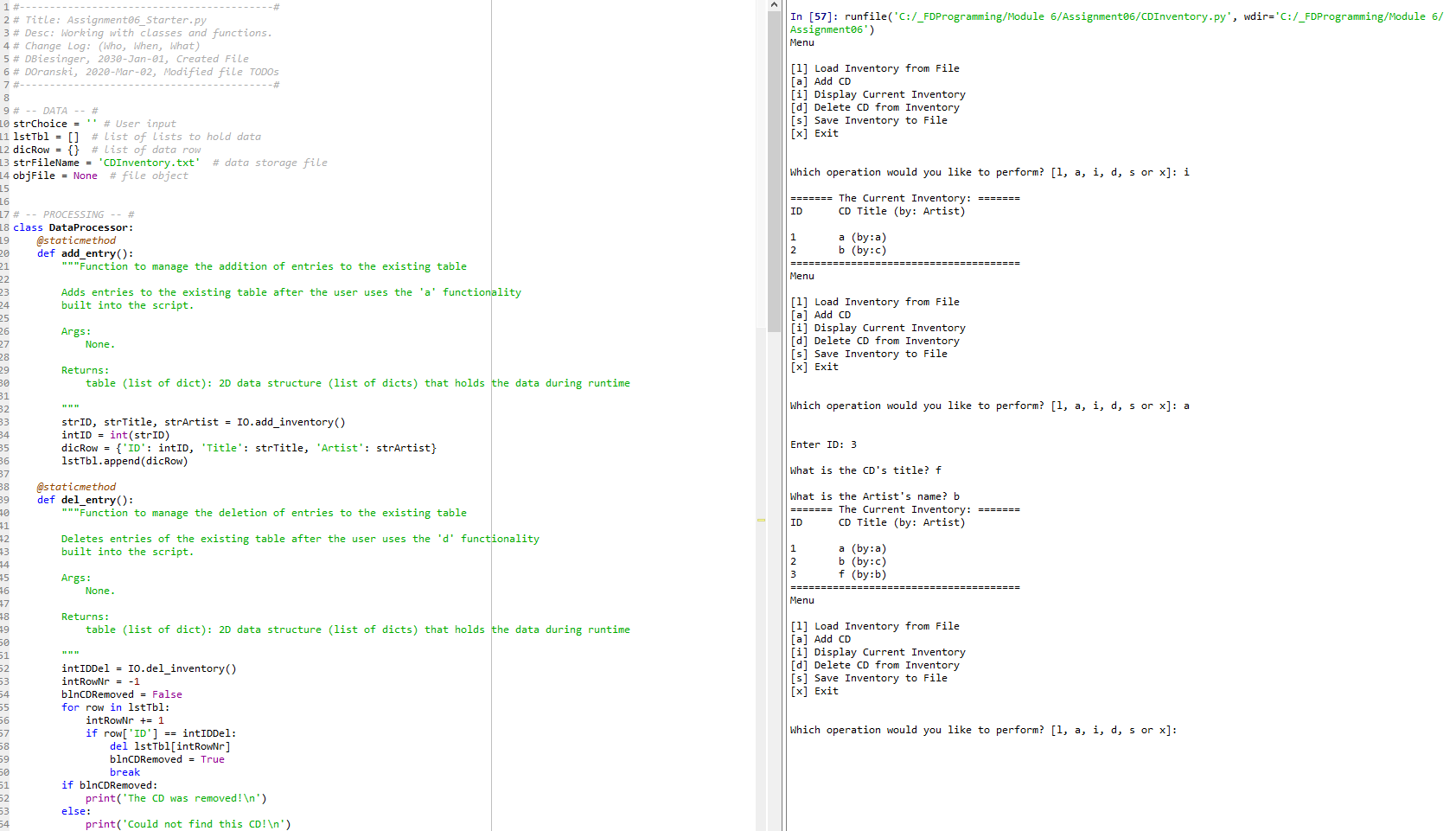


Figure 4 - Assignment06-1

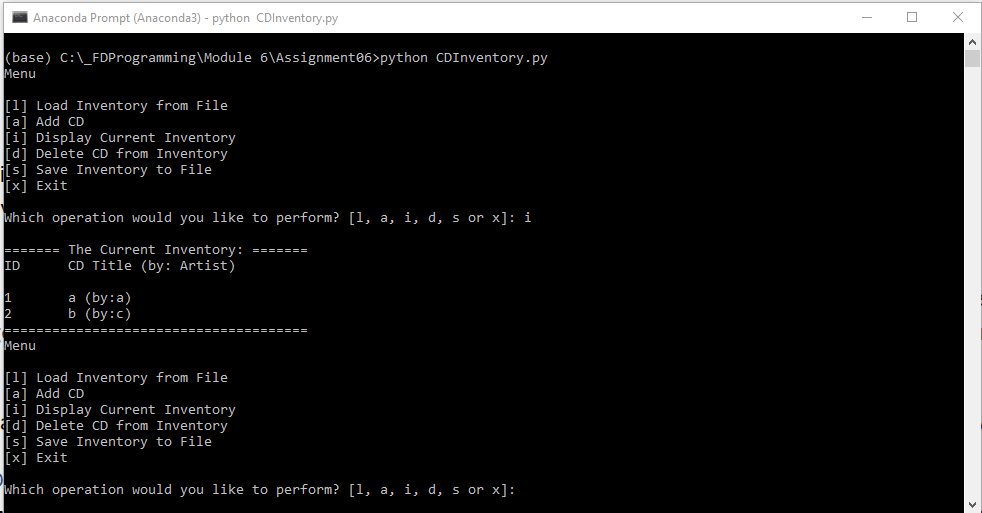


Figure 5 - Assignment06-2

The files for this Assignment were uploaded to GitHub [here](https://github.com/angryeng/Assignment_06).

# Summary

Assignment 06 was less challenging than the previous two, but only because the coding skills attained in the previous modules was so functional in performing the work in this module. Functions and docstrings are great tools for simplifying scripts and making code better human readable.