Sergey Timchenko

2/21/2021

Foundations Of Programming: Python

Assignment 06

Functions and Classes

# Introduction

In this assignment the student learns how to use *functions* and how combine them into the *classes*. Also we used the docstrings to describe the concept of *classes* and *functions*. The student modifies the starter file created for Assignment 05 and reorganizes the code to move some operations to *functions* under *classes*.

Functions

Function is a block of code that performs some operation in the program. It is used to make the program more compact and better logically organized. The function can be reused in different parts of the program; it can be modified without compromising the entire program. Different people can work on different functions of the same program in parallel. Functions could be logically combined into classes for better organization of the code.

Arguments /Parameters

Functions allow the option to pass in parameters. These allow you to pass in values for processing. Generally, Parameters are called arguments.

# Return Values

Return values are products of the finished running functions. They can be consumed immediately or be an arguments for other functions. Capturing the results in a variable allows the programmer to use the results multiple times without having to call the function each time. Using a function as an expression (e.g. print out the result) requires calling the function each time.

LAB 06-A directed a student to convert a provided script to utilize attributes and return values.

LAB 06-B directed the student to then modify the script from LAB 06-A to utilize tuples to handle the data.

In LAB 06-C it was created a class that collected several simple functions

The results of these labs are displayed below.

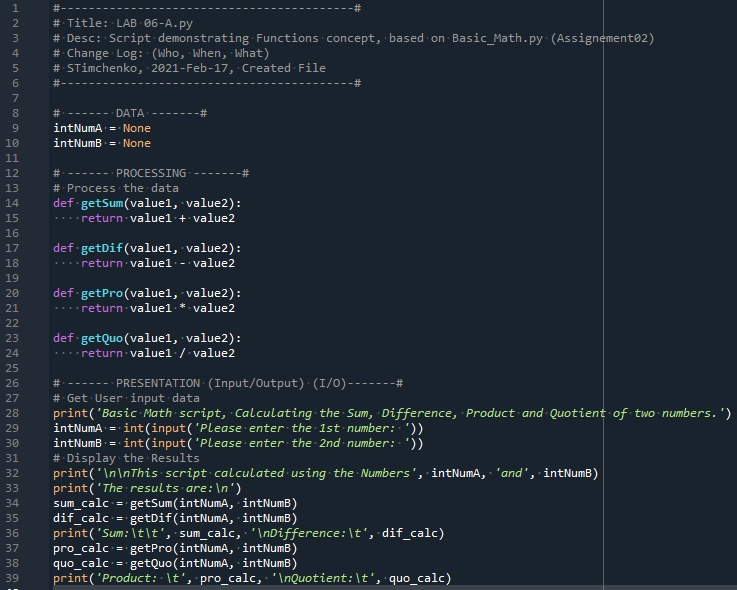


Figure 1 – LAB 06-A results

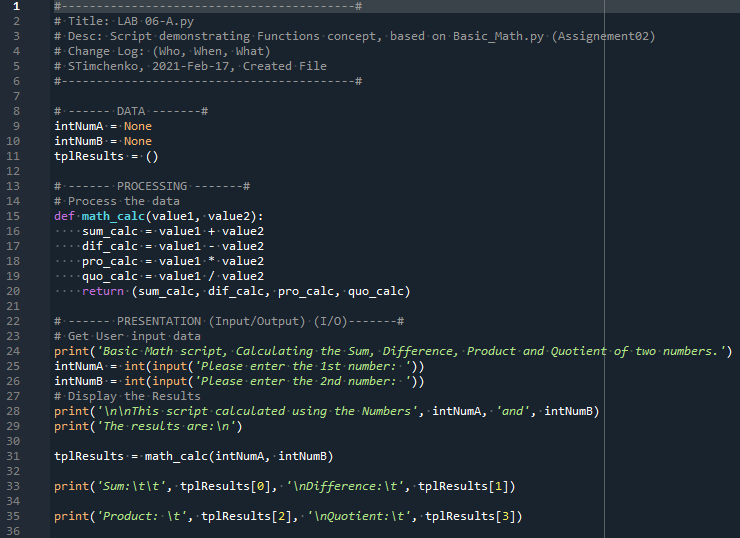


Figure 2 – LAB 06-B results

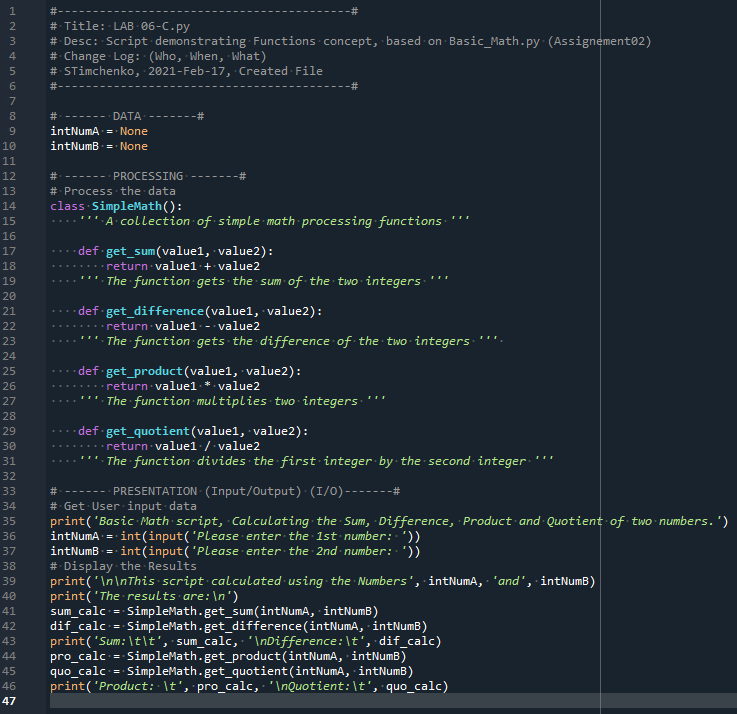


Figure 3 – LAB 06-C results

Positional vs. Named Arguments

When calling a function, it is possible to include the name of the parameter (as defined in the function declaration) and fill it explicitly with a defined argument. Not naming parameters implicitly uses the arguments and assigns them in the same sequence as the parameters are defined in the function declaration. It is possible to mix positional and named arguments (with a few limitations).

# The None Keyword (Again)

The None keyword is a special data type in Python with exactly one value: None. Usually None is 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 describe what function does. When highlighting a class or function name and pressing Ctrl + I, the docstrings populate a popup window with the contents of the docstring the similar way for Python preloaded functions like format(). Docstrings are written inside the triple quotation marks.

# Creating a script

In this assignment I was asked to modify the Assignment 05 starter file script by replacing the regular code text with functions collected in classes.

After opening the starter file with pre populated code we can incorporate TODO tasks. At this time we will be using Spider IDE.

For this program we will need to have a text file to write the user’s data. For this purpose the CDInventory.txt file has been created in the same directory where the python program is saved.

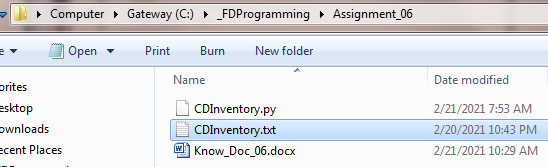


Figure 4 –Creation of CDInventory.txt file

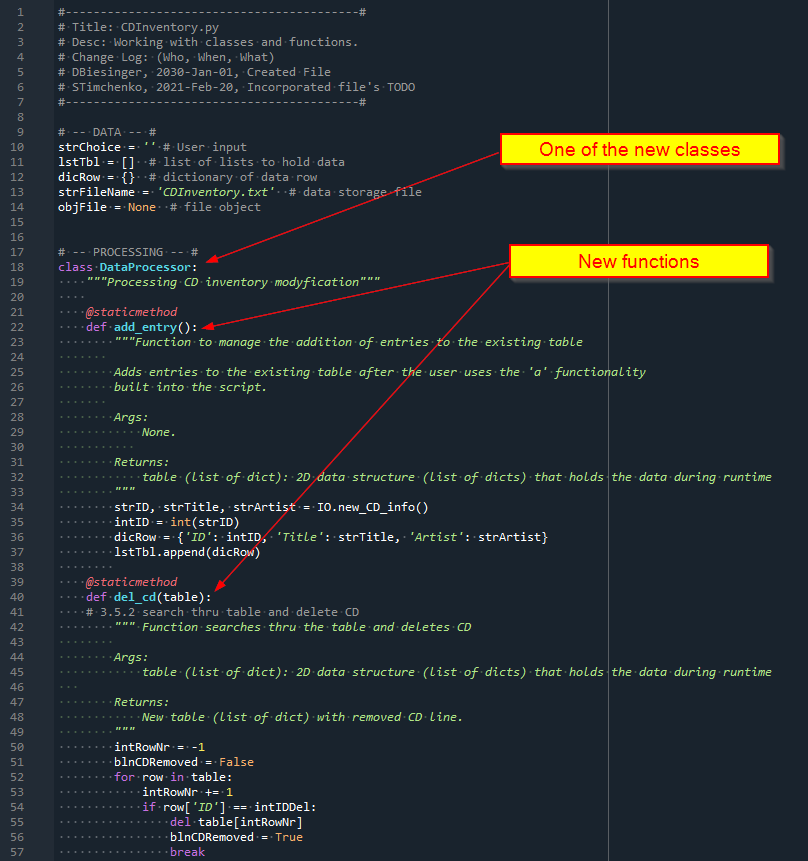


Figure 5 –Part of the scrip with added classes and functions

# The result of the program run

After the code is modified we are testing the program in Spider sub window

1. First we are loading the info from the file and displaying the inventory

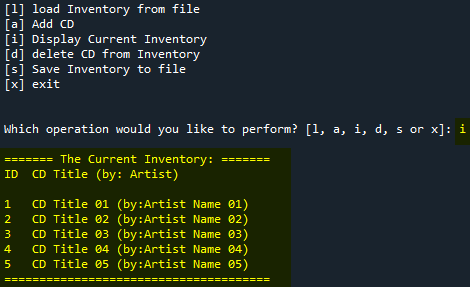


Figure 6 –Choice [l] and [i] demonstration

1. Then loading the new disk info and saving the file

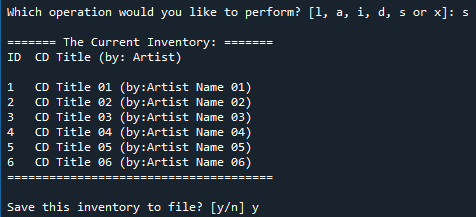
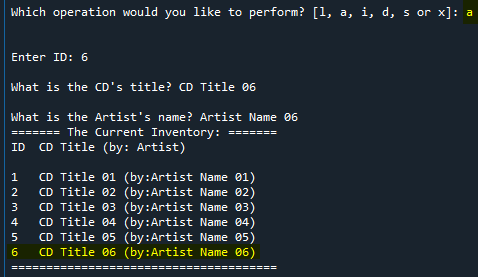


Figure 7 –Choice [a] and [s] demonstration

1. Lastly we are choosing delete CD from the file (choice(d))

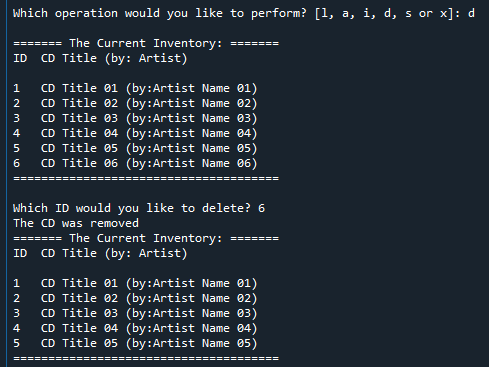


Figure 8 –Choice [d] demonstration

After running the program we are opening the text file and checking the results.

Disk 6 was added to the file and then deleted.

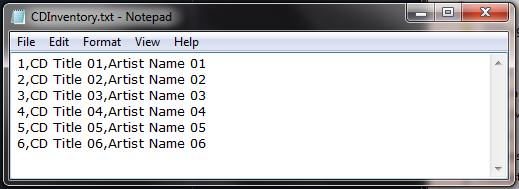


Figure 9 – Addition of the disk 6 from to the text file

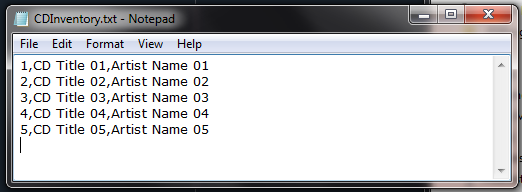


Figure 10 – Deletion of the disk 6 from the text file

Now we want to check if the program runs correctly in Anaconda command line window.

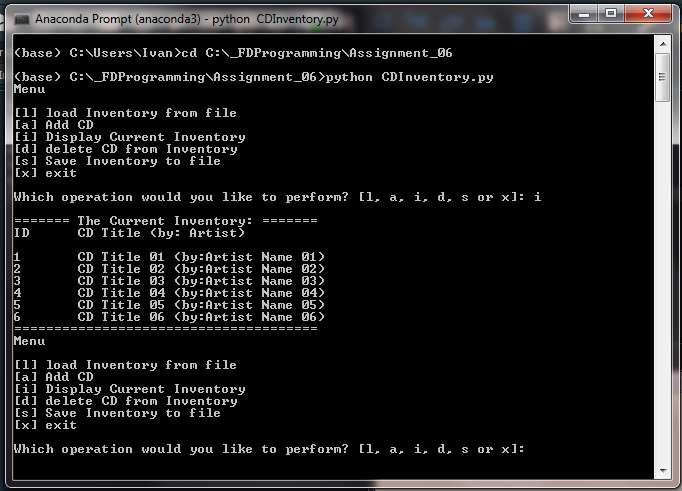


Figure 16 – Running the program in Anaconda Prompt window

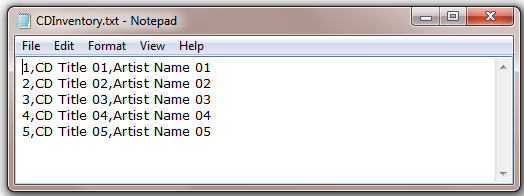


Figure 17 – CDInventory text file after running the script in Anaconda Prompt window

The files for this Assignment were uploaded to GitHub [here](https://github.com/snt2177/Assignment_06-.git).

# Summary

In this Assignment I used function and classes. I fund this assignment easier than the previous one because the basic code was already pre populated.

The use of the function is a great way to make the program more readable and work with the program by smaller manageable peaces.