Thivanka Samaranayake

May 26, 2020

Foundations of Programming (Python)

Assignment06

https://github.com/THIVASAM/IntroToProg-Python-Mod06.git

**Using Functions and Classes Build Programs**

# Introduction

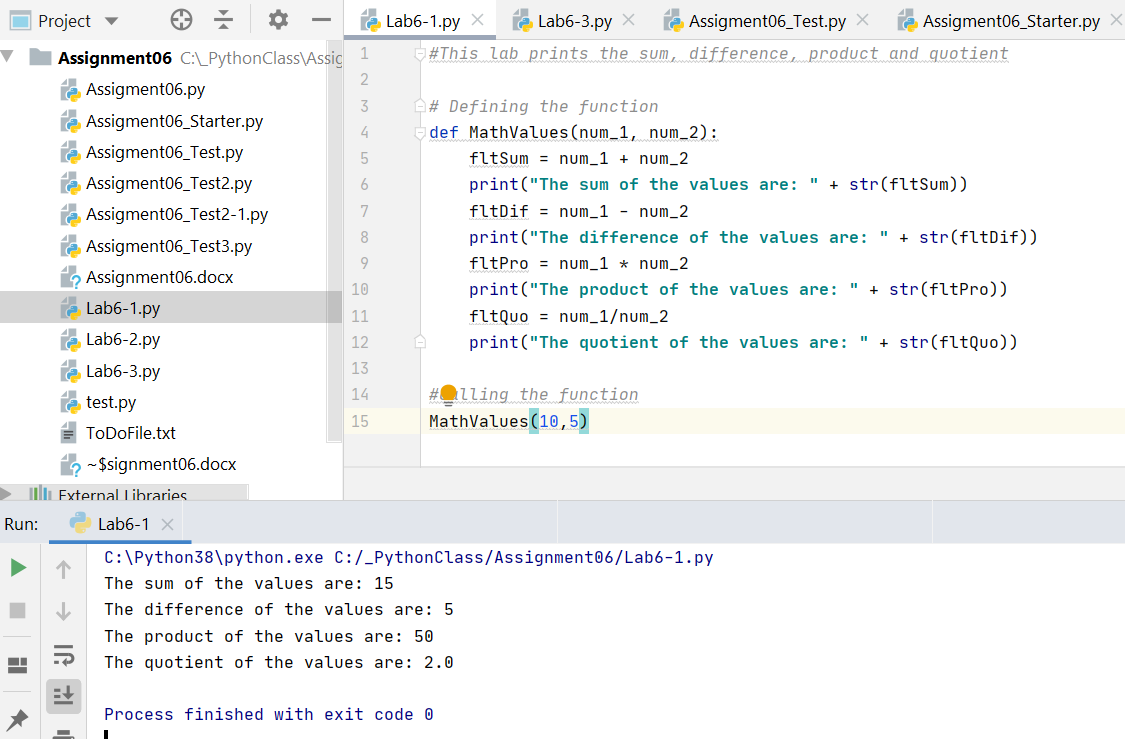
This assignment outlines the steps for using functions and classes to create a script that takes inputs from the user to store information in a text file. The goal of this module was to use two different types of functions: processing and input/output. These two functions were used to create an advanced program that produces a menu of options to the user and saves any information the user provides in a text file. Finally this program also uses GitHub repository to share and manage the assignment.

# What is a Function?

Python has many built in functions like len() and range(). Just like the Python functions, programmed functions perform a task and return the output from that task back into the program. Functions are created by a listing of programming statements that executes a certain goal. In Python you must first define a function and then call upon it later in the body of the program for it to perform the particular function that it was programmed to do.

# Parameters and Arguments

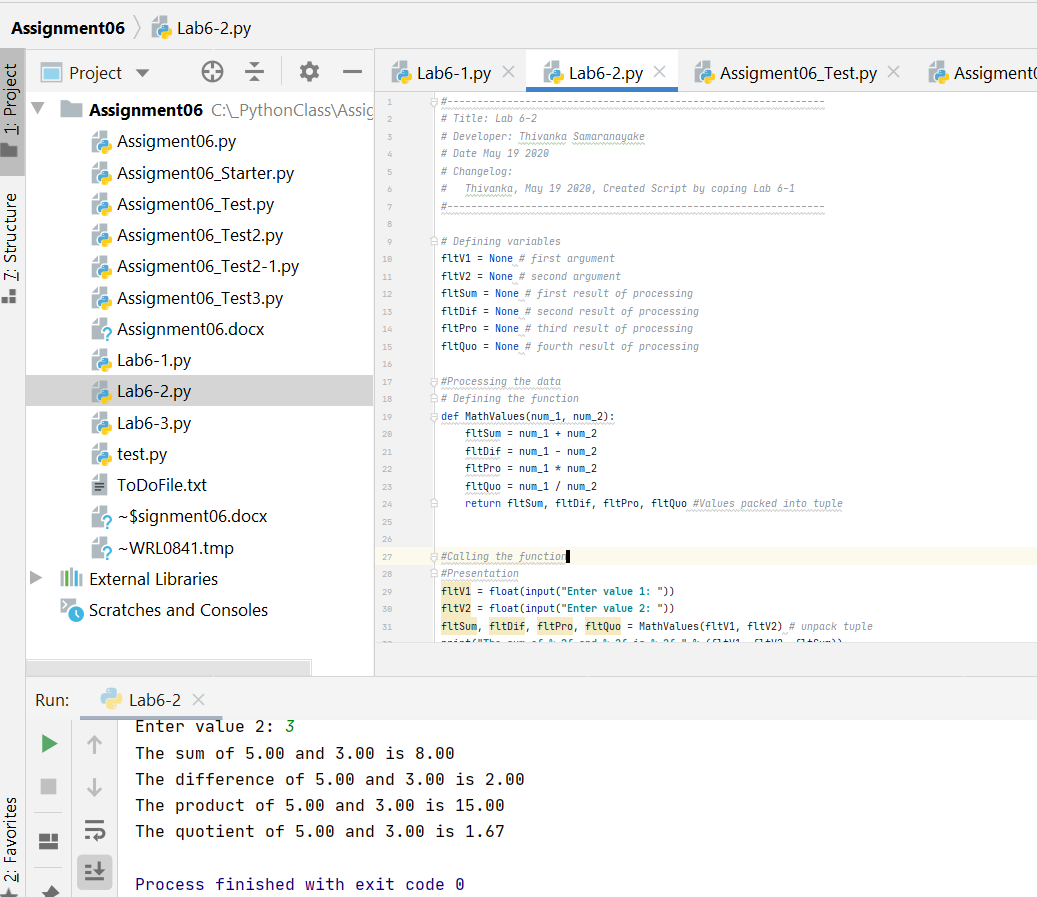
Parameters are values passed into the function for processing. Parameters and arguments are data that is often needed by the function to perform an action. They are usually denoted within parentheses inside of the code. The difference between a parameter and an argument is that arguments are values passed into the parameter. Within Python there is no limit to the number of parameters that can be passed into a function. Figure 1 shows arguments being passed into parameters of a function.



**Figure 1**—Lab 6-1 Math Value Arguments Passed into a Function

# Return Values

Return values are the values that a function returns back into the body of the program. As with parameters, a function can return one or more values. Python uses the word to denote what variables are returned to communicate with the rest of the program. It is return values that make the function an expression that is evaluated. And return values can be captured multiple times without having to call the function again. Figure 2 shows how complex value—tuples—are returned.



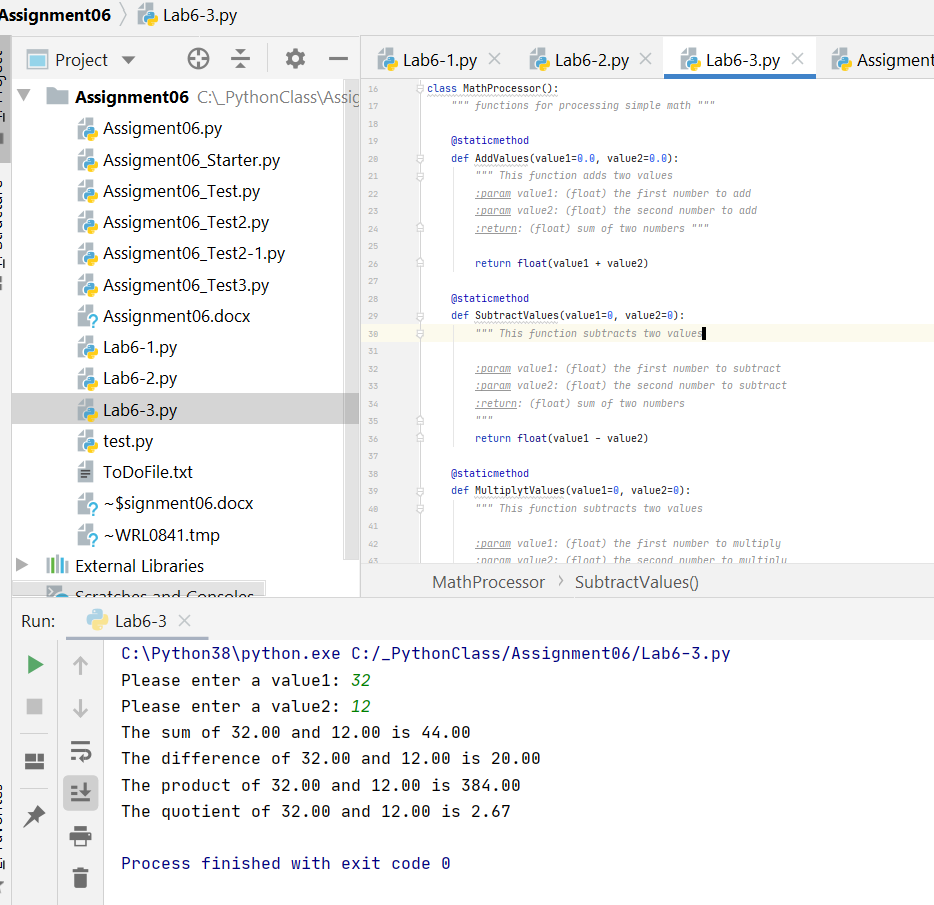
**Figure 2**—Tuples Being Returned from a Function

# Global and Local Variables

Global variables are variables declared in the main body of the script. Thus because of this, these variables can be used anywhere within the script. Local variables are variables that are declared and used in a function. Functions are completely sealed off from the rest of the program. The only communication that occurs between a function and the main script is through parameters and return values. Thus any variables declared inside of a function are not accessible to the main script.

# Organizing Code Using Functions and Classes

Functions help organize code by isolating a particular set of tasks—a function—to a localized area of the script. Then that particular set of code is called when needed. This reduces redundancy in the code and makes it flow smoothly. If functions are a way of organizing statements, a class is a way of organizing functions. Classes store multiple functions that are of the same type. This allows code to be organized even further than with just functions. Specifically functions and classes allows for the **separation of concerns**. A class can be a great way to subdivide the code into general sections. For example, in this module there were two classes: processing and input/output. These two classes contained various functions that either processed data from the user or communicated with the user. Figure 3 shows how a four different functions are organized under one class.



**Figure 3**—Four Functions Organized Into a Class

# Debugging with PyCharm

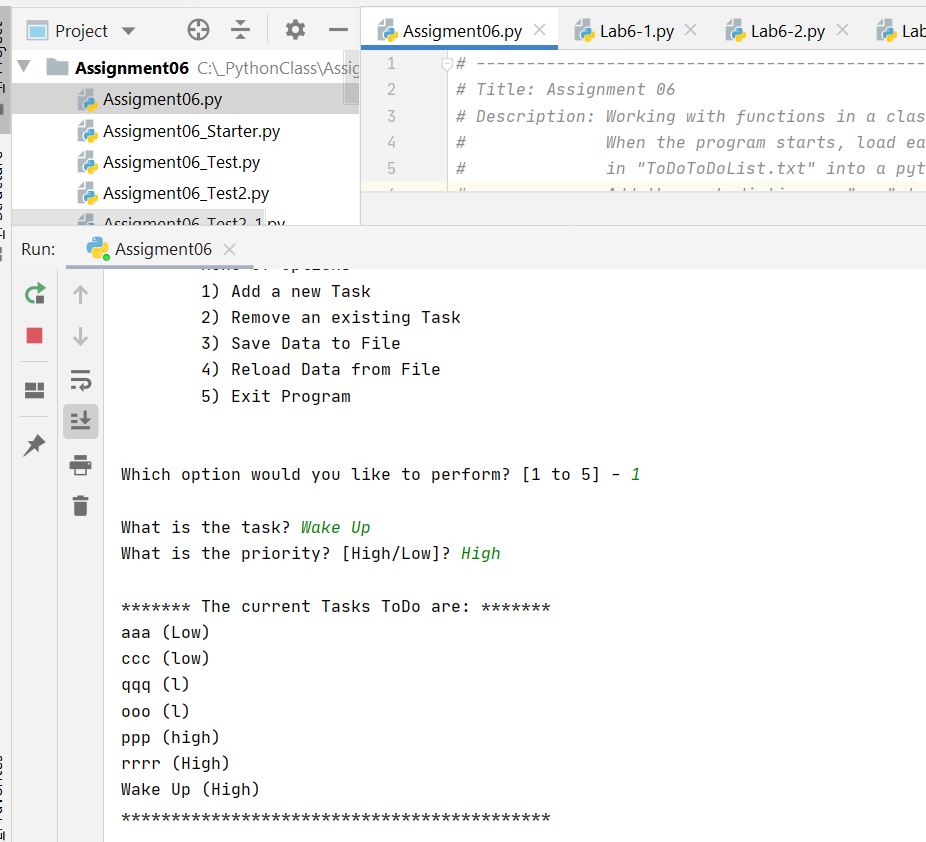
Integrated Development Environments (IDE) like PyCharm have debugging tools to help developers. The most important method of debugging in PyCharm is to use a break and step through the code using the Step Into code button. A variable screen shows the variables in the program and their values as the code is running. The Step Over button allows you to skip seeing the code in a function. The Step Out button allows you to stop showing the code in a function and directly go to the line that called it. These tools, though simple, are a very effective method in debugging even the most complex code.

# GitHub

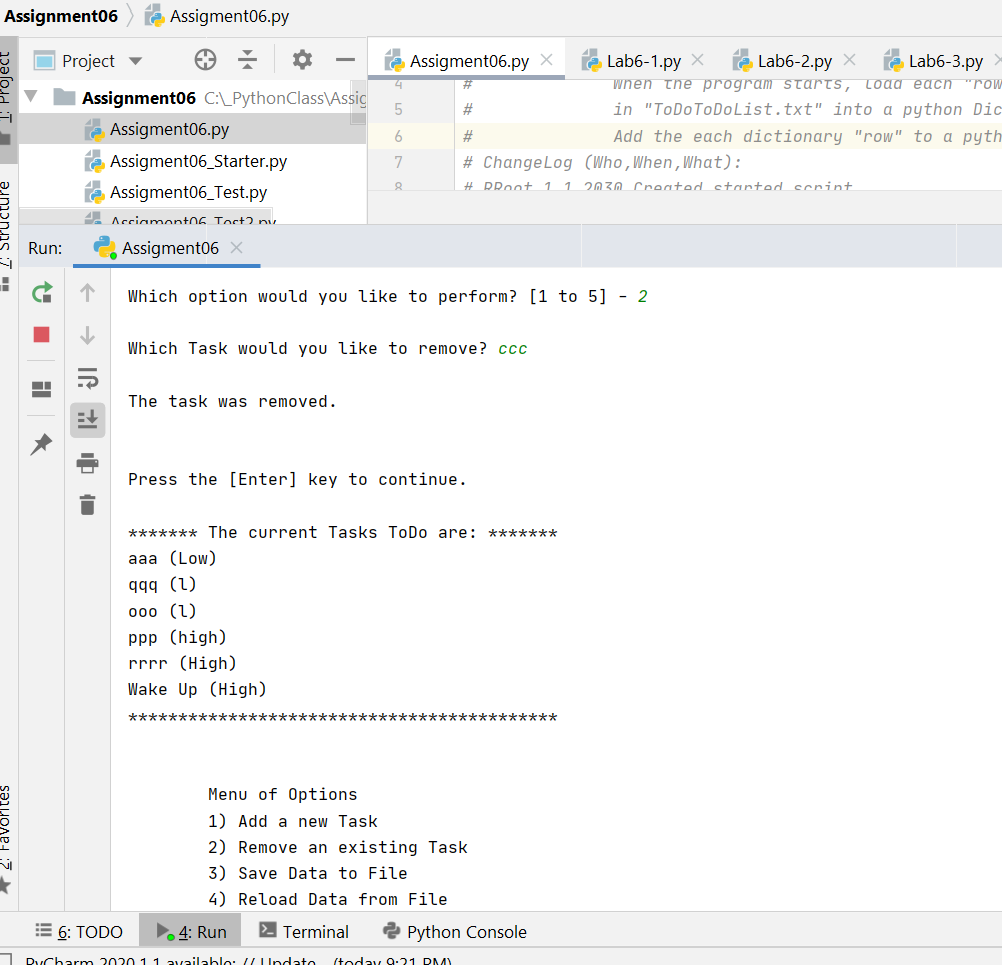
A GitHub webpage is a way to show the contents of a repository in website form. This allows the developer to enhance the presentation of a repository to make it more professional looking. It also allows the developer to present information in a webpage format.

# Assignment06

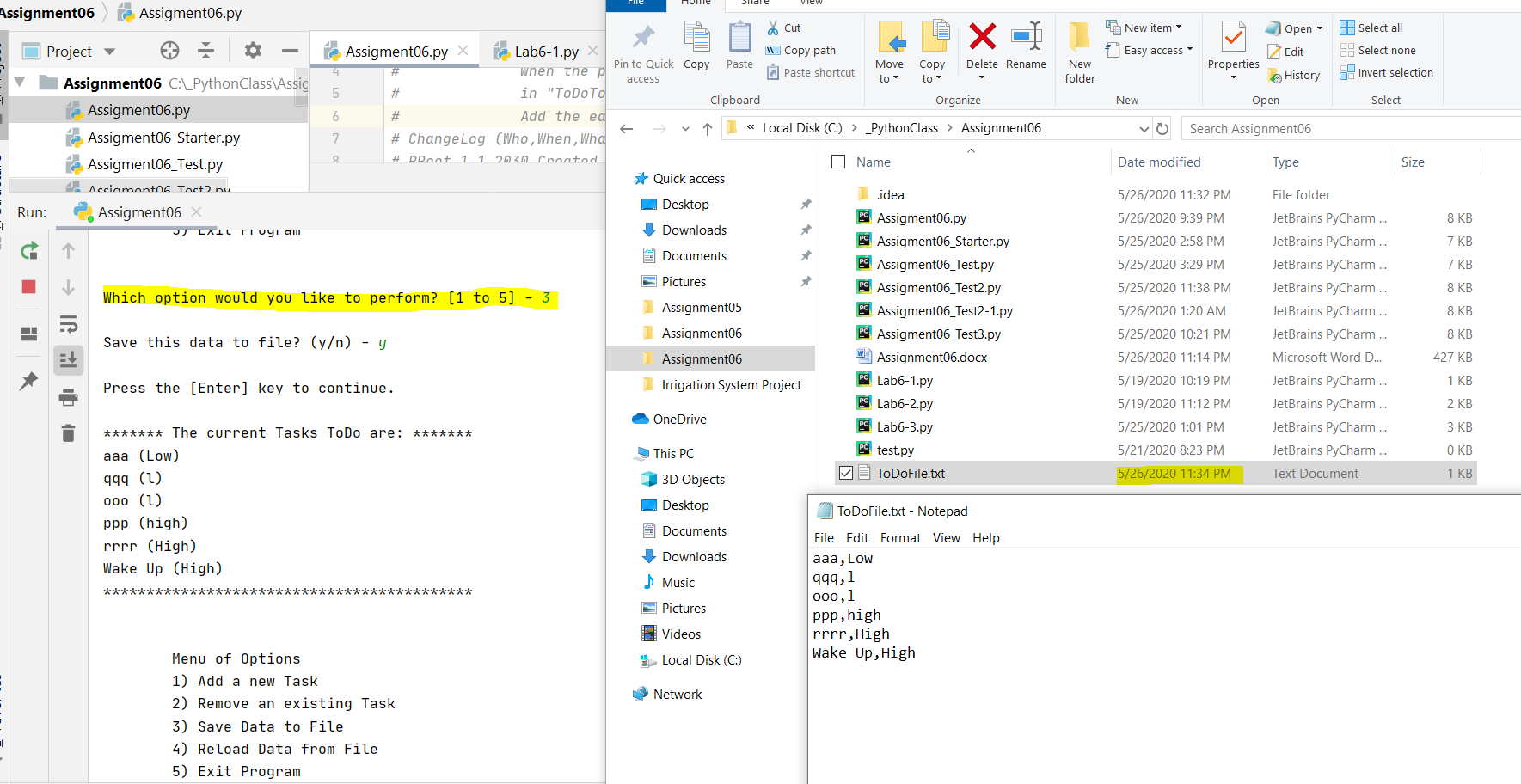
The program shows a menu of items that the user can use to read, add, delete, or store information. Each of the functions is demonstrated in the following figures.



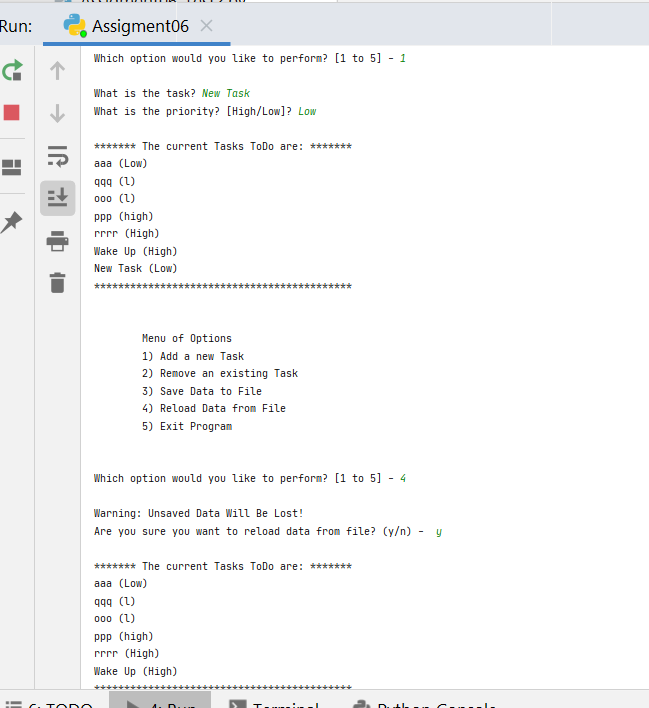
**Figure 4**—Add a New Task



**Figure 5**—Removing a Task



**Figure 6**—Saving a File



**Figure 7**—Reloading a Task (After adding a new task)

# Conclusion

This module was about organizing data through classes and functions. Each of the various functions were organized into processing or communication classes. This method of organizing code coincided with the concept of separation of concerns. The data processing portion of the code was separate from the input-output portion of the code. Functions were used in almost every step of the code to optimize it. Overall the code flows smoothly and can be easily understood by another developer. Github page was created to present the data in the repository.