# Assignment Steps

I read the assignment, noted the main questions to be answered and watched all the videos and read the book chapter. I wrote this document in parallel to the noting discussion question answers as I went along.

# Discussion

* What is a function?
  + A segment of script with defined inputs and outputs that can be called multiple times, it can have inputs or not.
* What are parameters?
  + Inputs to a function.
* What are arguments?
  + The actual values of the parameters
* What is the difference between parameters and arguments?
  + The parameter is the name of the variable as defined by the functions, the arguments are the actual values passed in.
* What are return values?
  + The output of a function
* What is the difference between a global and a local variable?
  + Local variables exist only inside the function, the global values can be used outside the functions and anywhere in the script.
* How do you use functions to organize your code?
  + Functions are discrete repeatable portions of code with defined inputs and outputs which are repeatable. It’s convenient to make them able to be used in multiple scripts and to work in teams by defining chunks of work to be performed.
* What is the difference between a function and a class?
  + A funcion
* How do functions help you program using the “Separations of Concerns" pattern?
  + A class can be used for each concern and a function for each action in that concern.
* How are the debugging tools use in PyCharm?
  + You can put in break points and step through until you find where the error is, and observe the variables along the way.
* What is a GitHub webpage?

Now that you have reviewed the websites and videos, **modify** a new script that manages a "ToDo list." This project is like the last one, but different enough to be a challenge.

**I have provided a starting template to I want you to modify and use for your program** called "Assigment06\_Starter.py". Currently the code loads data from a file into a Python List of Dictionary objects. However, the code only uses a few functions, and **your job is to add more functions to organize the code.** You may note that it is both easier and harder to work with someone else's code, and that is part of the assignment.

## 5.1 Create a folder

I’m using a somewhat different folder structure because I am working on multiple classes

## 5.2 Create a new project in PyCharm

I’m actually using MS Virtual Studio. I hope that’s ok. I’ve been using my work laptop and they don’t like me to download too many things. I’m confident that I get the idea of the IDEs and how they function.

## 5.3 Add the Starter File

Add code to your script that will perform that assignment’s task. I suggest you start by adding the header and some pseudo-code.

## 5.3 Add code to your Script

# ---------------------------------------------------------------------------- #

# Title: Assignment 06

# Description: Working with functions in a class,

# When the program starts, load each "row" of data

# in "ToDoToDoList.txt" into a python Dictionary.

# Add the each dictionary "row" to a python list "table"

# ChangeLog (Who,When,What):

# RRoot,1.1.2030,Created started script

# RRoot,1.1.2030,Added code to complete assignment 5

# James Roefs, 2020-08-18 ,Modified code to complete assignment 6

# ---------------------------------------------------------------------------- #

# Data ---------------------------------------------------------------------- #

# Declare variables and constants

strFileName = "ToDoFile.txt" # The name of the data file

objFile = None # An object that represents a file

dicRow = {} # A row of data separated into elements of a dictionary {Task,Priority}

lstTable = [] # A list that acts as a 'table' of rows

strChoice = "" # Captures the user option selection

strTask = "" # Captures the user task data

strPriority = "" # Captures the user priority data

strStatus = "" # Captures the status of an processing functions

# Processing --------------------------------------------------------------- #

class Processor:

""" Performs Processing tasks """

@staticmethod

def read\_data\_from\_file(file\_name, list\_of\_rows):

""" Reads data from a file into a list of dictionary rows

:param file\_name: (string) with name of file:

:param list\_of\_rows: (list) you want filled with file data:

:return: (list) of dictionary rows

"""

list\_of\_rows.clear() # clear current data

file = open(file\_name, "r")

for line in file:

task, priority = line.split(",")

row = {"Task": task.strip(), "Priority": priority.strip()}

list\_of\_rows.append(row)

file.close()

return list\_of\_rows, 'Success'

@staticmethod

def add\_data\_to\_list(task, priority, list\_of\_rows):

row={"Task":task,"Priority":priority}

list\_of\_rows.append(row);

return list\_of\_rows, 'Success'

@staticmethod

def remove\_data\_from\_list(task, list\_of\_rows):

row\_remove = {}

for row in list\_of\_rows:

if (row["Task"] ==task):

row\_remove = row

list\_of\_rows.remove(row\_remove)

return list\_of\_rows, 'Success'

@staticmethod

def write\_data\_to\_file(file\_name, list\_of\_rows):

objFile = open(file\_name,"w")

for row in list\_of\_rows:

objFile.write(row["Task"]+","+row["Priority"]+"\n")

objFile.close()

return list\_of\_rows, 'Success'

# Presentation (Input/Output) -------------------------------------------- #

class IO:

""" Performs Input and Output tasks """

@staticmethod

def print\_menu\_Tasks():

""" Display a menu of choices to the user

:return: nothing

"""

print('''

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

''')

print() # Add an extra line for looks

@staticmethod

def input\_menu\_choice():

""" Gets the menu choice from a user

:return: string

"""

choice = str(input("Which option would you like to perform? [1 to 5] - ")).strip()

print() # Add an extra line for looks

return choice

@staticmethod

def print\_current\_Tasks\_in\_list(list\_of\_rows):

""" Shows the current Tasks in the list of dictionaries rows

:param list\_of\_rows: (list) of rows you want to display

:return: nothing

"""

print("\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*")

for row in list\_of\_rows:

print(row["Task"] + " (" + row["Priority"] + ")")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print() # Add an extra line for looks

@staticmethod

def input\_yes\_no\_choice(message):

""" Gets a yes or no choice from the user

:return: string

"""

return str(input(message)).strip().lower()

@staticmethod

def input\_press\_to\_continue(optional\_message=''):

""" Pause program and show a message before continuing

:param optional\_message: An optional message you want to display

:return: nothing

"""

print(optional\_message)

input('Press the [Enter] key to continue.')

@staticmethod

def input\_new\_task\_and\_priority():

task = input("Enter a task: ")

priority = input("Enter the task priority: ")

return task, priority

@staticmethod

def input\_task\_to\_remove():

task = input("Enter a task to be removed: ")

return task

# Main Body of Script ------------------------------------------------------ #

# Step 1 - When the program starts, Load data from ToDoFile.txt.

Processor.read\_data\_from\_file(strFileName, lstTable) # read file data

# Step 2 - Display a menu of choices to the user

while(True):

# Step 3 Show current data

IO.print\_current\_Tasks\_in\_list(lstTable) # Show current data in the list/table

IO.print\_menu\_Tasks() # Shows menu

strChoice = IO.input\_menu\_choice() # Get menu option

# Step 4 - Process user's menu choice

if strChoice.strip() == '1': # Add a new Task

strTask, strPriority = IO.input\_new\_task\_and\_priority()

Processor.add\_data\_to\_list(strTask,strPriority,lstTable)

IO.input\_press\_to\_continue(strStatus)

continue # to show the menu

elif strChoice == '2': # Remove an existing Task

strTask = IO.input\_task\_to\_remove()

Processor.remove\_data\_from\_list(strTask,lstTable)

IO.input\_press\_to\_continue(strStatus)

continue # to show the menu

elif strChoice == '3': # Save Data to File

strChoice = IO.input\_yes\_no\_choice("Save this data to file? (y/n) - ")

if strChoice.lower() == "y":

Processor.write\_data\_to\_file(strFileName,lstTable)

IO.input\_press\_to\_continue(strStatus)

else:

IO.input\_press\_to\_continue("Save Cancelled!")

continue # to show the menu

elif strChoice == '4': # Reload Data from File

print("Warning: Unsaved Data Will Be Lost!")

strChoice = IO.input\_yes\_no\_choice("Are you sure you want to reload data from file? (y/n) - ")

if strChoice.lower() == 'y':

Processor.read\_data\_from\_file(strFileName,lstTable)

IO.input\_press\_to\_continue(strStatus)

else:

IO.input\_press\_to\_continue("File Reload Cancelled!")

continue # to show the menu

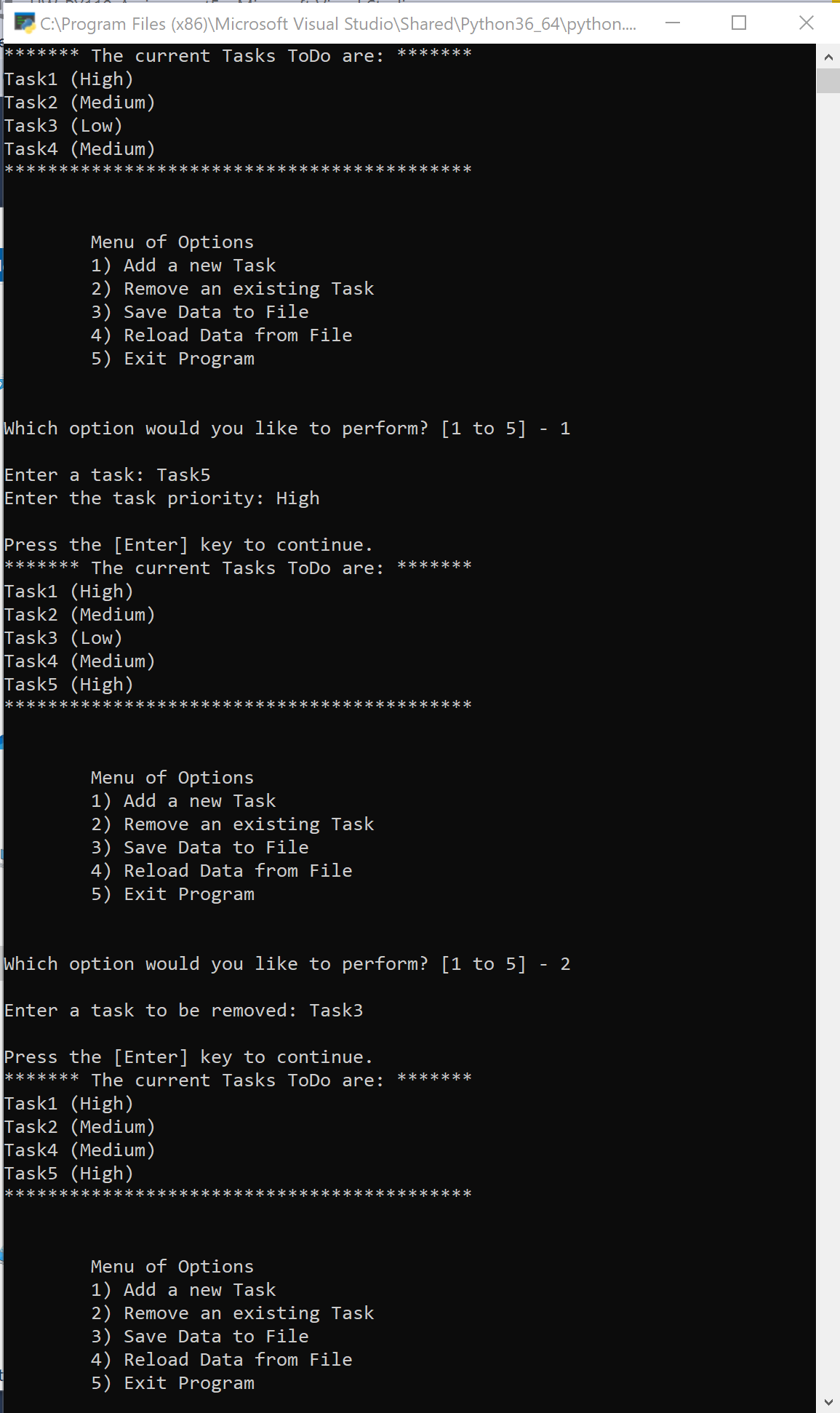
elif strChoice == '5': # Exit Program

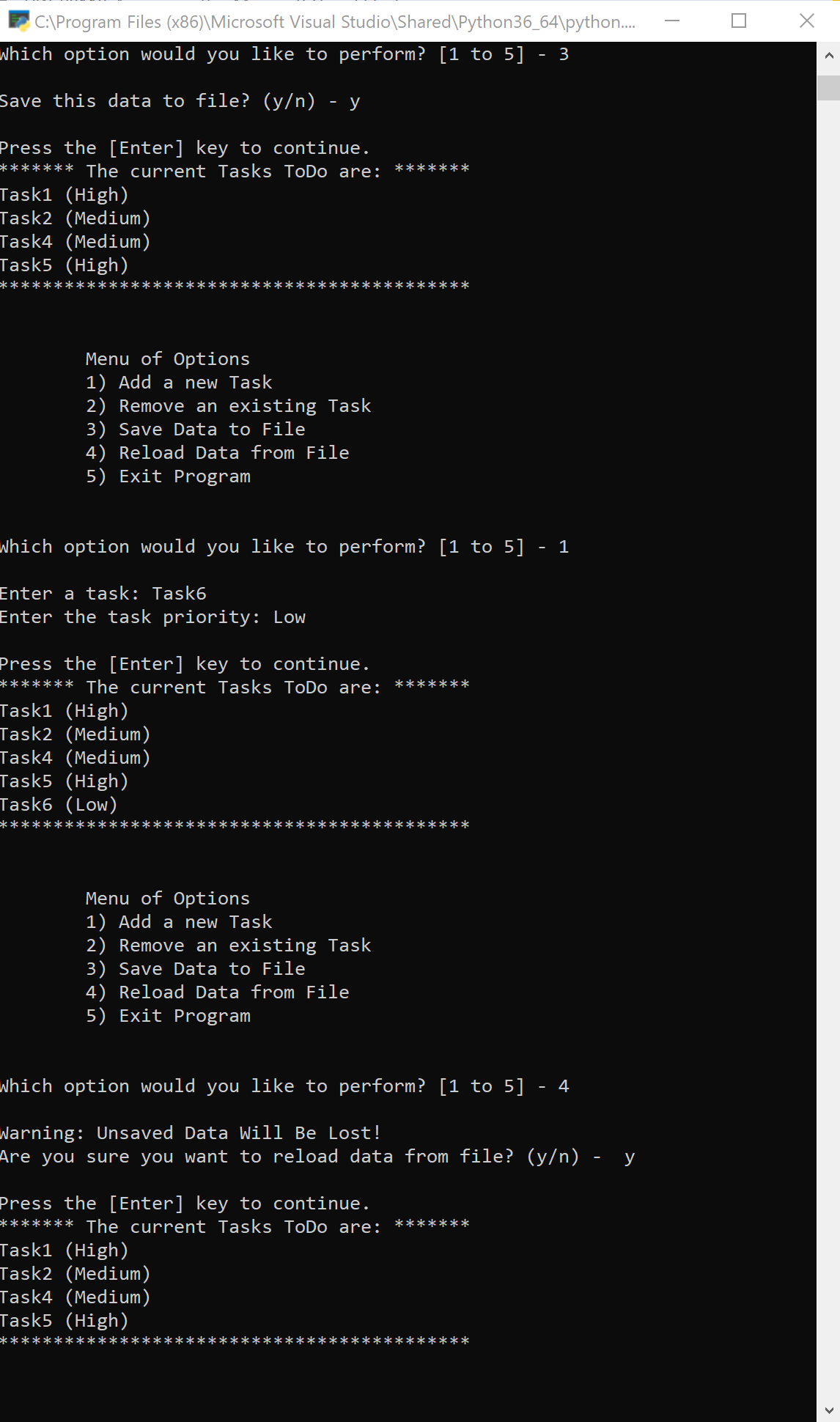
print("Goodbye!")

break # and Exit

## 5.4 Run Your Script

As mentioned, this is run in the MS Visual Studio IDE





It looks the same in the command prompt.

# 6 Document your knowledge

The classes and functions work pretty similarly to c++. Syntax is easier and more forgiving. It looks like a good way to discretize the work and make it easy to break up the work between people.

# 7 Post your Files to GitHub

Done.