

Denise Albano

2/15/22

Foundations of Programming: Python

Assignment 05

<https://github.com/dla425/IntroToProg-Python>

## “ToDo List” Python Script

### Introduction:

This paper will document the steps I took to modify a python script that manages a to do list. The ToDoList.txt file will contain tasks and priorities and will be loaded using a python dictionary object. In order to complete the assignment, I watched the course video in module 5 by Randall Root, read chapter 5 in text book, as well as reviewed and watched the additional web pages and videos.

### Creating the Python Script:

For this assignment I used PyCharm to modify the assignment05 starter python script. To start, I created a new project in PyCharm that uses the folder \_PythonClass\Assignment05. Within this project, I created a new python script file named Assignment05.py (see figure 1) by copying the information from the assignment05 starter python script.

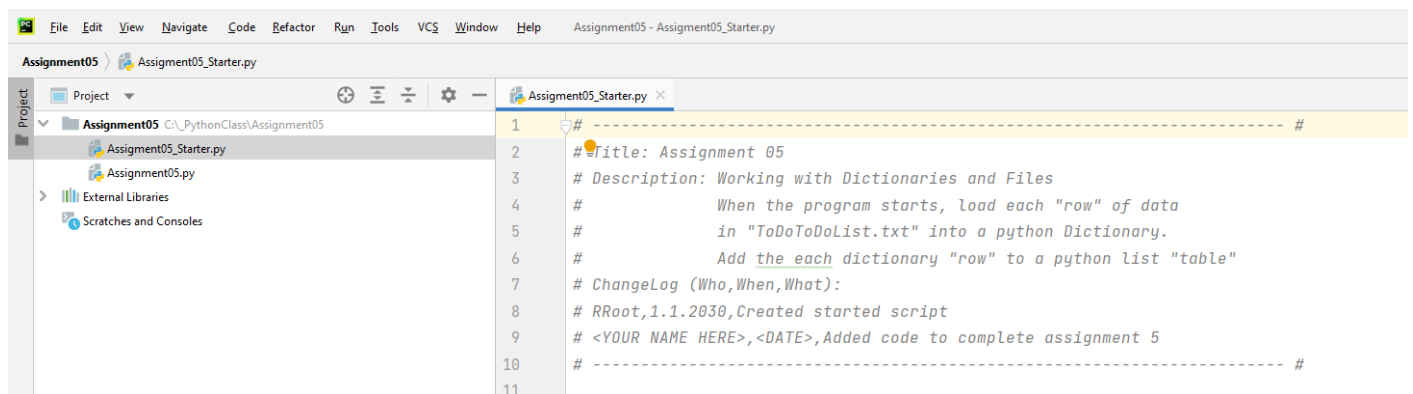


Figure 1: New project and python script in PyCharm

In my assignment05.py script, I updated the header to include my name and date in the change log. Since this assignment needed to read data from a text file, I created the 'ToDoList.txt' file and added two rows of tasks and their associated priority (see Figure 1).

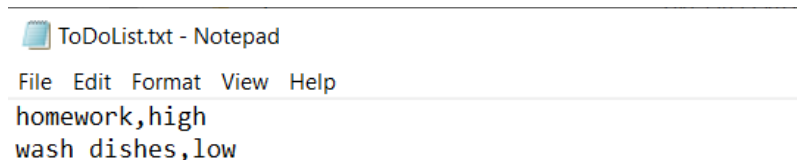


Figure 1: Screenshot of ToDoList.txt file at the start

### Step 1: Load data into a list of dictionary rows

Now that I had a text file to load data from, I used the `open()` function with the `"r"` access mode to open the `"ToDoList"` text file for reading. To loop through and read each line in the file, I used a `'for ... in'` statement. To get the individual items out, I used the `string split()` method using the comma as the separator and then built a dictionary row (`dicRow`). To identify the individual items in the dictionary, I used the keys `'task'` and `'priority'` with the associated index in the row. I also used the `strip()` method to remove any leading or ending characters. I then used the `append()` method to add the items to the list that makes up the table. Lastly, I used the `close()` function to close the file.

### Step 2: Display the menu

The `assignment02 starter.py` script used a while loop to create the menu of choices for the user and a triple-quoted string so that the menu text would be displayed over multiple lines. No additional code was needed in this step.

### Step 3: Show the current items in the table

To display the current items in the table to the user, I used a `'for ... in'` statement to loop through the table by row. Then I used the `print()` function to display each task and its priority, separated by a vertical bar.

### Step 4: Add a new item to the table

To add a new task to the table, I created two string variables, `strTask` and `strPriority`, and used the `input()` function to request the user to type in the task to be added and its priority. The new time was added to the dictionary and appended to the table using the `append()` method. I concluded this step with a `print()` function letting the user know the task had been added.

### Step 5: Remove an item from the table

To remove a task from the table, I created a variable, `strItem`, and used the `input()` function to request what task should be removed from the user. I used another `'for ... in'` statement to loop through the table to see if the task was found. If the task was found, I used the `remove()` function to delete the item and then the `print()` function to let the user know that the task had been removed. If the task was not found, I used the `print()` function to tell the user that the task was not found. I found that I also need to include a `break` statement to stop the loop.

### Step 6: Save tasks to the `ToDoList.txt` file

To save the data to the `"ToDoList.txt"` file, I used the `open()` function with the `'w'` access mode to open the `"ToDoList"` text file for writing. Again, I used a `'for ... in'` statement to loop through the table by row. I used the `write()` method to save each row of data to the file. Once that was done, I used a `print()` function to let the user know that the data had been saved. I concluded this step by using the `close()` function to close the file.

### Step 7: Exit the program

To exit the program, I used a `print()` function to display a goodbye message to the user and then used the `break` statement to stop the while loop.

The final code for this assignment is shown below.

```
# ----- #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#             When the program starts, load each "row" of data
#             in "ToDoToDoList.txt" into a python Dictionary.
#             Add each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
```

```

# RRoot,1.1.2030,Created started script
# DALbano,2.15.22,Added code to complete assignment 5
# ----- #

# -- Data -- #
# declare variables and constants
strFile = "ToDoList.txt" # An object that represents a file
strData = "" # A row of text data from the file
# dicRow = {} # A row of data separated into elements of a dictionary
# {Task,Priority}
lstTable = [] # A list that acts as a 'table' of rows
strMenu = "" # A menu of user options
strChoice = "" # A Capture the user option selection
strTask = "" # A capture of the user task entered
strPriority = "" # A capture of the user priority for the task entered
strItem = "" # A capture of the user task to remove

# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like
# Lab 5-2)
objFile = open(strFile, "r")
for row in objFile:
    strData = row.split(",")
    dicRow = {"task": strData[0], "priority": strData[1].strip()}
    lstTable.append(dicRow)
objFile.close()

# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item
    3) Remove an existing item
    4) Save Data to File
    5) Exit Program
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] - "))
    print() # adding a new line for looks

    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        for row in lstTable:
            print("Task: ", row['task'], "|", "Priority: ", row['priority'])
        continue

    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):
        strTask = input("Enter the task: ")
        strPriority = input("Enter the priority: ")
        dicRow = {"task": strTask, "priority": strPriority}
        lstTable.append(dicRow)
        print("\nTask added!")
        continue

    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        strItem = input("Enter task to remove: ")

```

```

for row in lstTable:
    if row["task"].lower() == strItem.lower():
        lstTable.remove(row)
        print("\nTask removed")
    else:
        print("\nTask not found")
    break
continue

# Step 6 - Save tasks to the ToDoToDoList.txt file
elif (strChoice.strip() == '4'):
    objFile = open(strFile, "w")
    for row in lstTable:
        objFile.write(str(row["task"]) + ',' + str(row["priority"]) + '\n')
    objFile.close()
    print("Data saved!")
    continue

# Step 7 - Exit program
elif (strChoice.strip() == '5'):
    print("Goodbye!")
    break # and Exit the program

```

## Running the program:

Before running the program, I check the contents of my ToDoList.txt file (see figure 2).

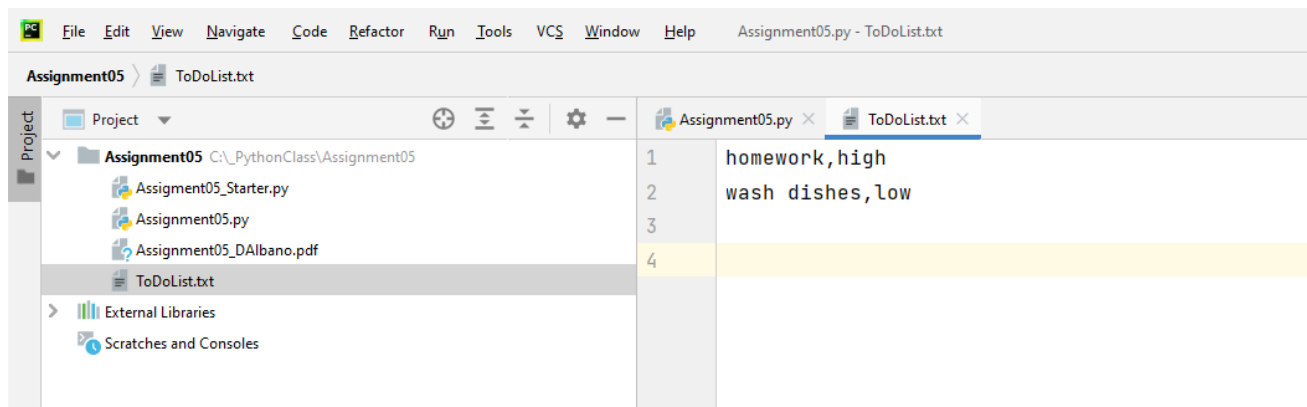


Figure 2: Screenshot of ToDoList.txt file at the beginning of the program.

Next I ran the program in PyCharm and selected each menu item (see figures 3 through 7).

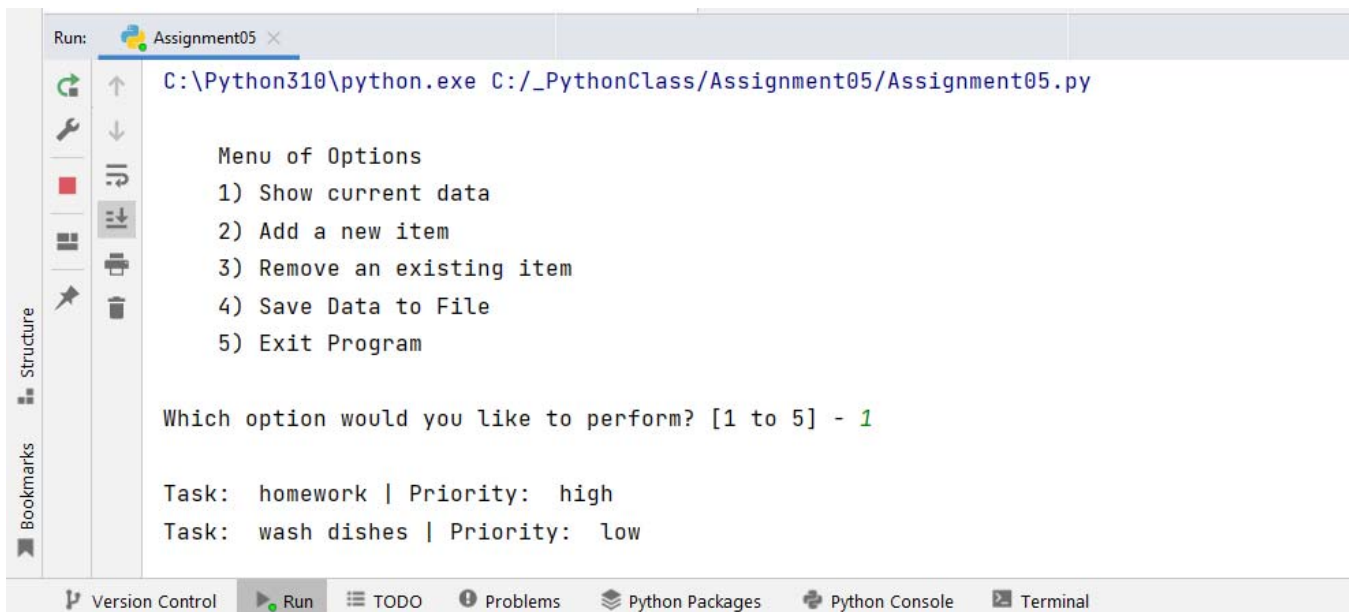


Figure 3: Screenshot of menu item 1.

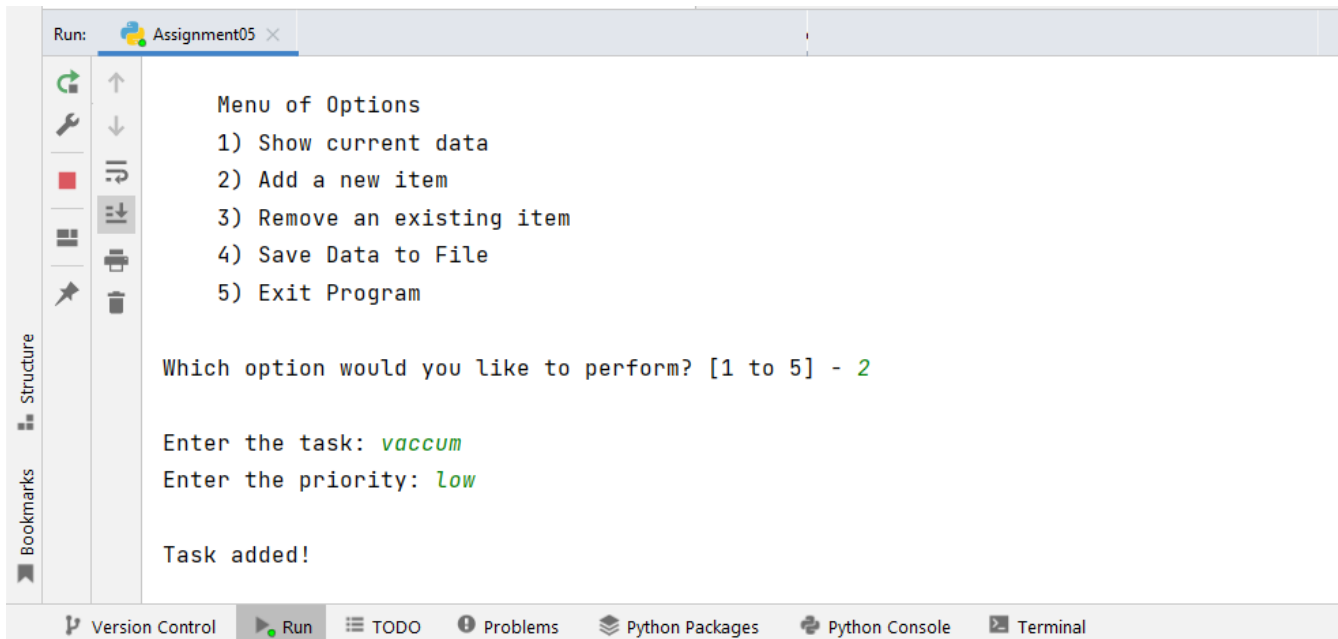


Figure 4: Screenshot of menu item 2.

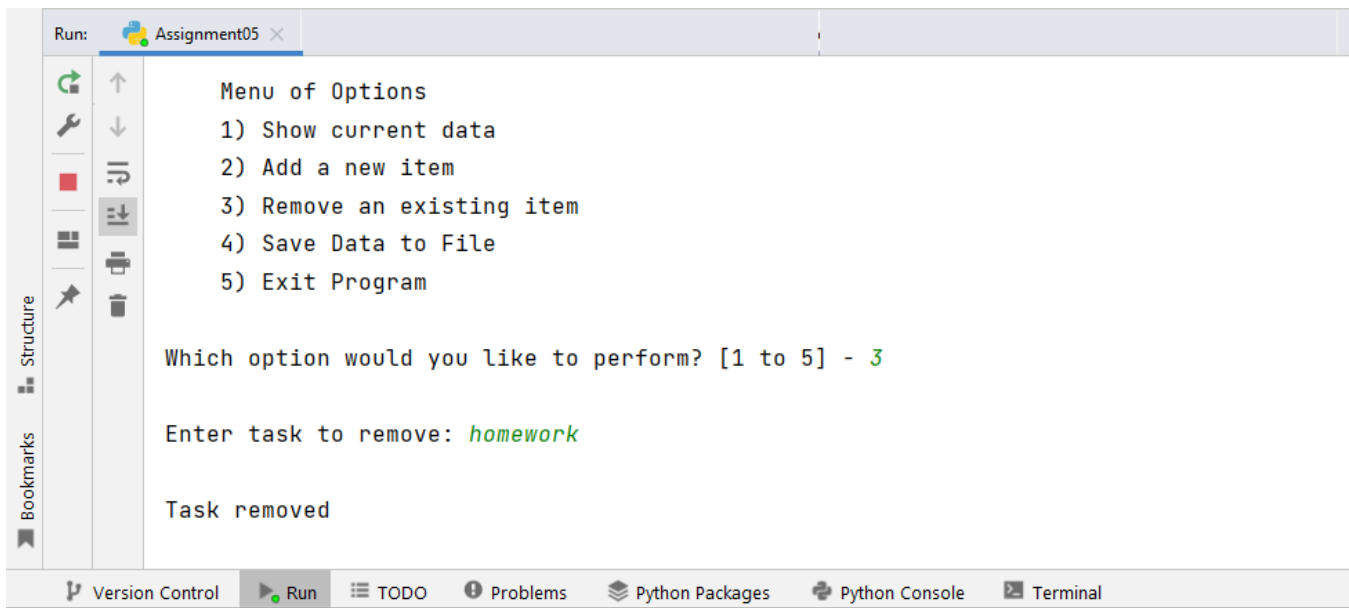


Figure 5: Screenshot of menu item 3.

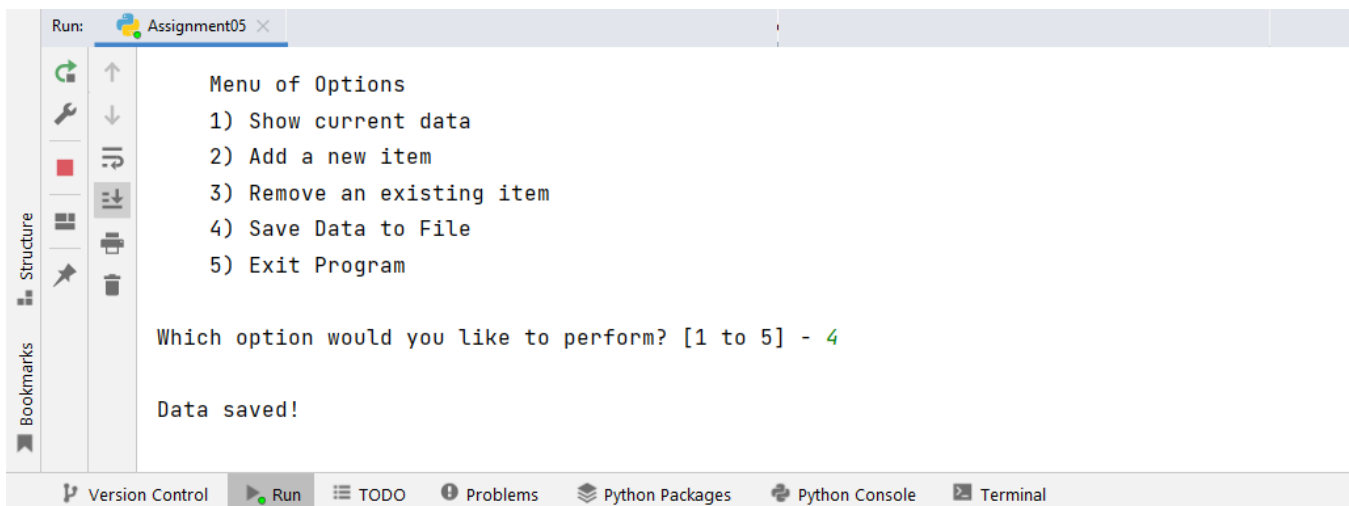
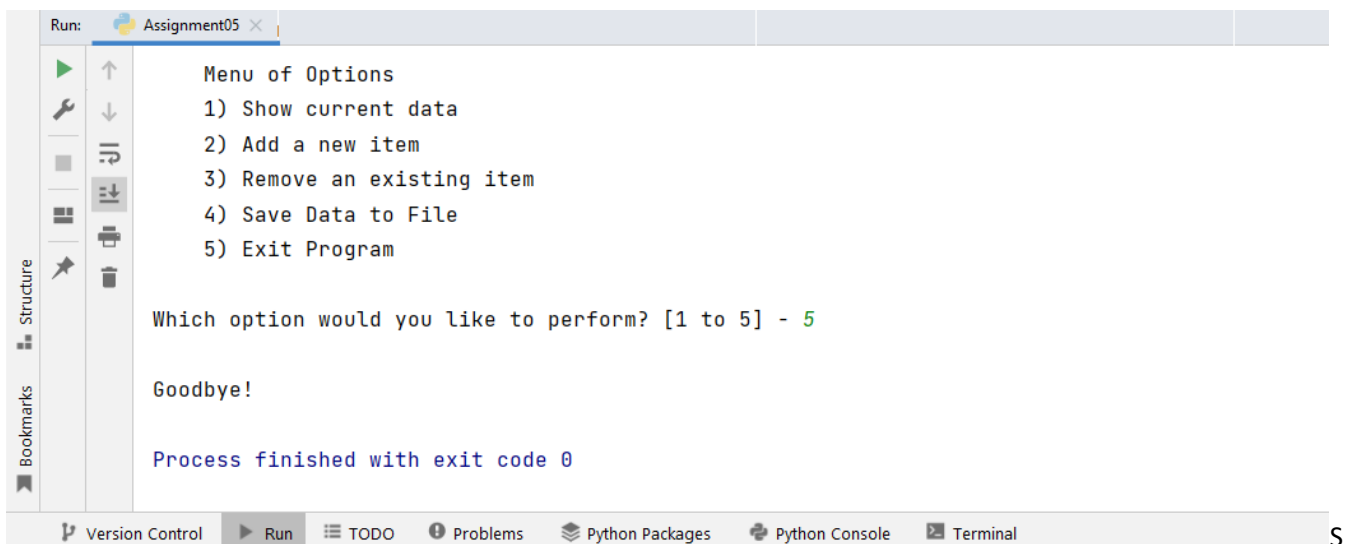


Figure 6: Screenshot of menu item 4.



```
Run: Assignment05 x
Menu of Options
1) Show current data
2) Add a new item
3) Remove an existing item
4) Save Data to File
5) Exit Program

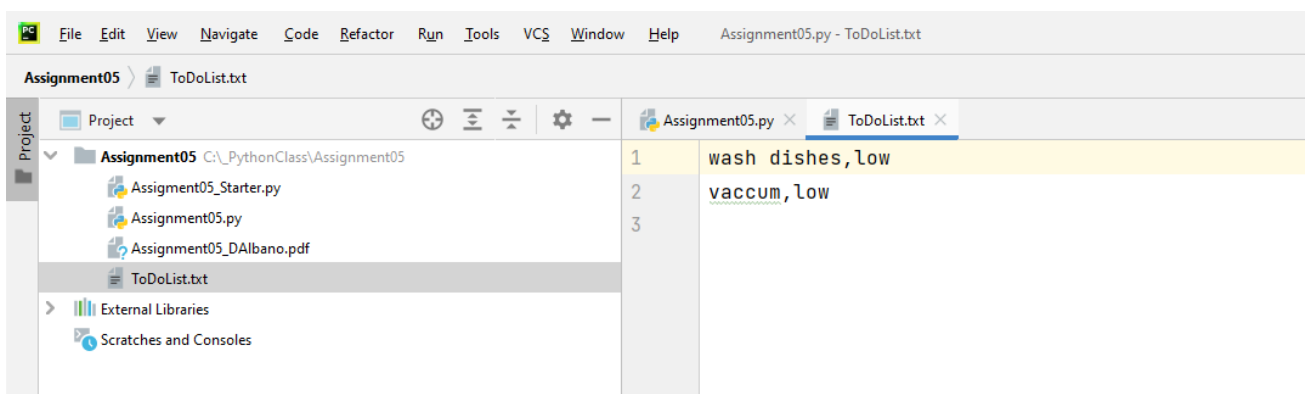
Which option would you like to perform? [1 to 5] - 5

Goodbye!

Process finished with exit code 0
```

Figure 7: Screenshot of menu item 5.

Lastly I check my ToDoList.txt file to see what had been saved (see figure 8).



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Assignment05.py - ToDoList.txt
Assignment05 > ToDoList.txt
Project
  Assignment05 C:\PythonClass\Assignment05
    Assignment05_Starter.py
    Assignment05.py
    Assignment05_DAlbano.pdf
    ToDoList.txt
  External Libraries
  Scratches and Consoles
1 wash dishes, low
2 vacuum, low
3
```

Figure 8: Screenshot of ToDoList.txt. file at the end of the program

## GitHub account creation

After creating a GitHub account, I then created a repository named “IntoToProg-Python.” Within this repository, I uploaded the Assignment05 files and committed the changes. As a last step, I shared the link to my GitHub repository on the canvas discussion board for peer review.

## Summary

To complete this assignment, I needed to understand the difference between lists and dictionaries, as well as understand how keys differed from indexes. It was important to also understand how to read and write data from and to a file. I found the examples in the course video, textbook, and additional materials to be extremely helpful in understanding the concepts needed to create a working program.