Ashton Larkin
May 17, 2022
IT FDN 110 A Spring 2022: Foundations of Programming: Python
Assignment05: "To Do List"
https://github.com/AshtonUniverse/IntroToProg-Python

Introduction:

The objective of assignment 05 is to update a provided starter script adding code to the script sections of each step with a comment "# *TODO: Add Code Here*" and to maintain the integrity of the existing logic and formatting of the original programmer.

The "ToDoList" program (Assignment05_Starter.py) is divided into three different sections of concerns; Data, Processing, and Presentation (Input-Output) and performs the following steps (pseudocode) as scripted by the original programmer "RRoot":

Step 1 - When the program starts, load any data that you have in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2).
Step 2 - Display a menu of choices to the user.
Step 3 - Show the current items in the table.
Step 4 - Add a new item to the list/Table.
Step 5 - Remove a new item from the list/Table.
Step 6 - Save tasks to the ToDoToDoList.txt file.
Step 7 - Exit program.

This document is a breakdown of the logic I used for assignment 05. I will list each step including the original code from the starter script and the code I added to complete the program.

Added my name and date to the change log in the script header:

```
# ------------------------------------------------------------------------ #
# 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 the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# ALarkin,5.17.2022,Added code to complete assignment 5
# ------------------------------------------------------------------------ #
```

Replaced:

```
# <YOUR NAME HERE>,<DATE>,Added code to complete assignment 5
```

Step 1 - *When the program starts, load any data that you have in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2).*

```
# -- Data -- #
# declare variables and constants
# objFile = "ToDoList.txt"   # An object that represents a file
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
```

I changed the object file variable name from objFile to strFile and commented out the original variable.

```
# -- Processing -- #
# Step 1 - When the program starts, load any data that you have in a text file
# called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)
# Process the data
# TODO: Add Code Here
try:
    objFile = open("ToDoList.txt", "r")
    for row in objFile:
        strData = row.split(",")
        dicRow = {"Task": strData[0], "Priority": strData[1].strip()}
        lstTable.append(dicRow)
objFile.close()
except:
    pass # continue
```

When the ToDoList program runs, user input data is collected into a dictionary, placed into a list table, and written to a text file. When the program begins, Step 1 reads data from the file, if the file exists, and writes any existing data to a list table. The logic I added is wrapped in a try/except which will allow the program to continue if the “ToDoList.txt” file does not exist. If the file does not exist, the following error occurs: FileNotFoundError: [Errno 2] No such file or directory: 'ToDoList.txt'. The except:pass construct essentially silences any and all exceptional conditions that come up while the code covered in the try: block is being run and allows the program to continue.

Step 2 - *Display a menu of choices to the user.*

```
# -- 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
```

There are no changes to the original script in Step 2.

Step 3 - *Show the current items in the table.*

```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    # TODO: Add Code Here
    for row in lstTable:
        print(row["Task"] + ' | ' + row["Priority"])
    continue
```

The Step 3 code I added uses a for loop to print each rows task and priority.

Step 4 - *Add a new item to the list/Table.*

```
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    # TODO: Add Code Here
    strTask = input("Enter a Task: ")
    strTask = (((strTask.strip()).lower()).title())
    strPriority = input("Enter a Priority: [High, Medium, Low] - ")
    strPriority = (((strPriority.strip()).lower()).title())
    dicRow = {"Task":strTask, "Priority":strPriority}
    lstTable.append(dicRow)
    continue
```

The Step 4 code I added asks the user to enter a new task and its priority. I used the *strip()* method to remove whitespace and characters from the beginning and the end of the strings, the *lower()* method to return strings where all characters are lower case, and the title() method to return a string where the first character in every word is upper case. This ensures consistency in

how the data stored, presented, and removed in other steps. The code sets the user input data to a dictionary row and then appends that row to the list table.

Step 5 - *Remove a new item from the list/Table.*

```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    # TODO: Add Code Here
    strRemove = input("Enter a task to remove from ToDoList: ")
    strRemove = (((strRemove.strip()).lower()).title())
    for row in lstTable:
        if (row["Task"] == strRemove):
            lstTable.remove(row)
            print("\n" + strRemove + " removed from ToDoList ")
    continue
```

The Step 5 code I added asks the user to input a task to remove. If the task entered by the user exists in the list table, the logic removes the row and prints that the task that was removed.

Step 6 - *Save tasks to the ToDoToDoList.txt file.ssss*

```
# Step 6 - Save tasks to the ToDoToDoList.txt file
elif (strChoice.strip() == '4'):
    # TODO: Add Code Here
    objFile = open(strFile, "w")
    for row in lstTable:
        objFile.write(row["Task"] + "," + row["Priority"] + "\n")
    objFile.close()
    print("Data saved to file: ", """ + strFile + """)
    continue
```

The Step 6 coded I added uses the *open()* function to write rows from the list table read and appended from the delimited text file from Step 1 and any rows appended to the list table in Step 2 to the file ToDoList.txt.

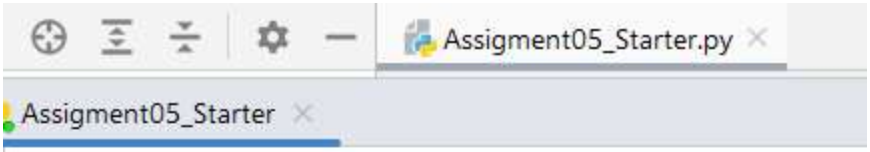
Step 7 - *Exit program.*

```
# Step 7 - Exit program
elif (strChoice.strip() == '5'):
    # TODO: Add Code Here
    print("ToDoList Completed!")
    break
```

The Step 7 code I added prints “ToDoList Completed!” before the program beaks the loop. I also added an *input()* function after the loop breaks asking the user to press the “enter” key to exit and close the program.

```
# Exit the program
input("\nPress the enter key to exit.")
```

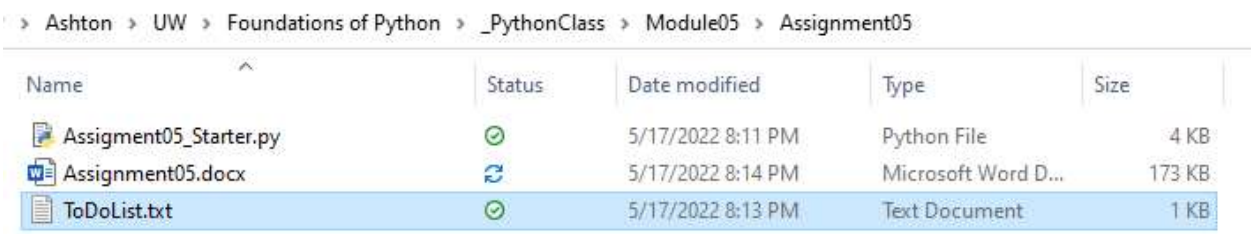
Run the script from PyCharm.

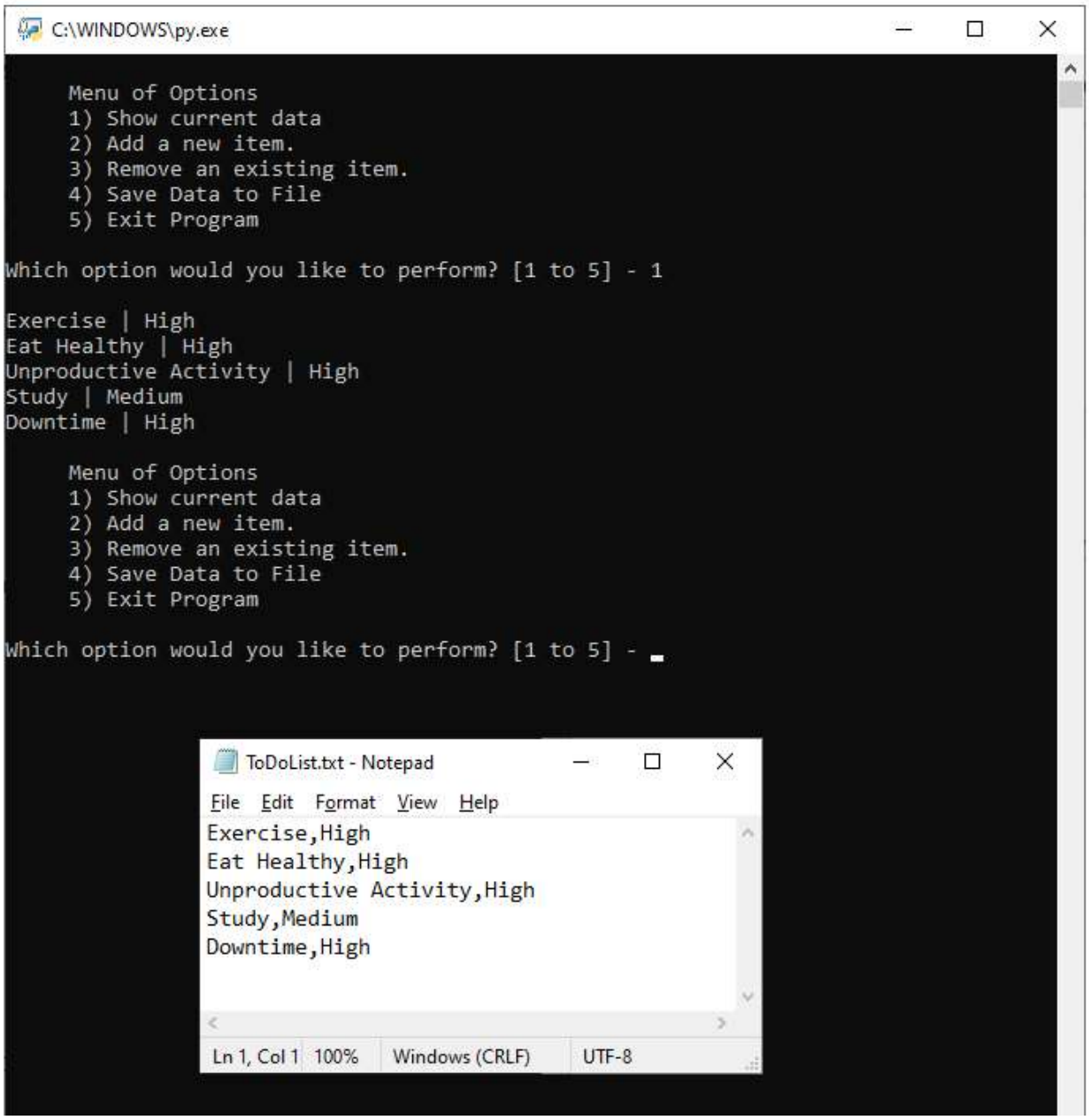


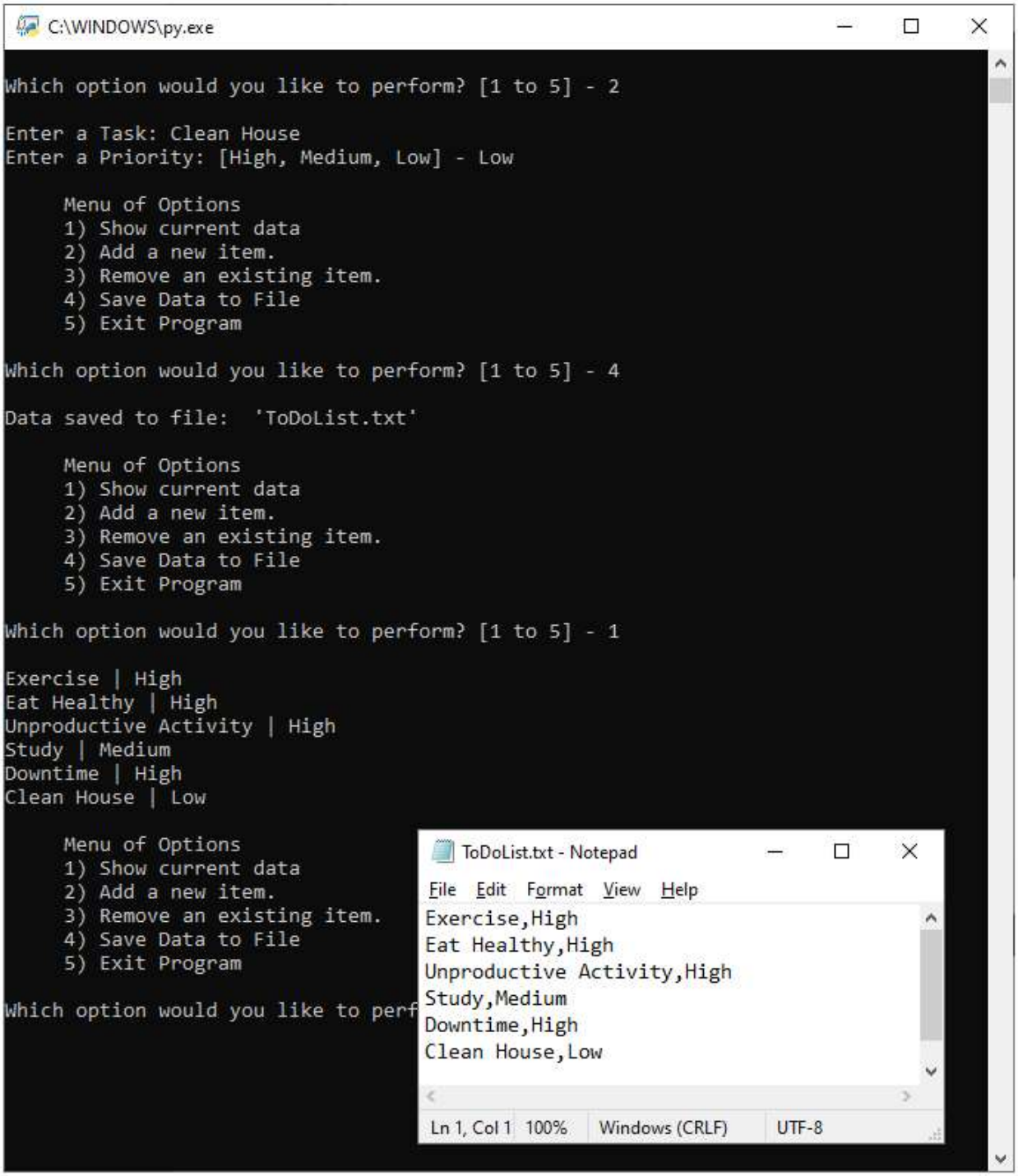

```
     Menu of Options
     1) Show current data
     2) Add a new item.
     3) Remove an existing item.
     4) Save Data to File
     5) Exit Program

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

Exercise | High
Eat Healthy | High
Unproductive Activity | High
Study | Medium
Downtime | High

Which option would you like to perform? [1 to 5] - 2

Enter a Task: Clean House
Enter a Priority: [High, Medium, Low] - Low

     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] - 1

Exercise | High
Eat Healthy | High
Unproductive Activity | High
Study | Medium
Downtime | High
Clean House | Low
```

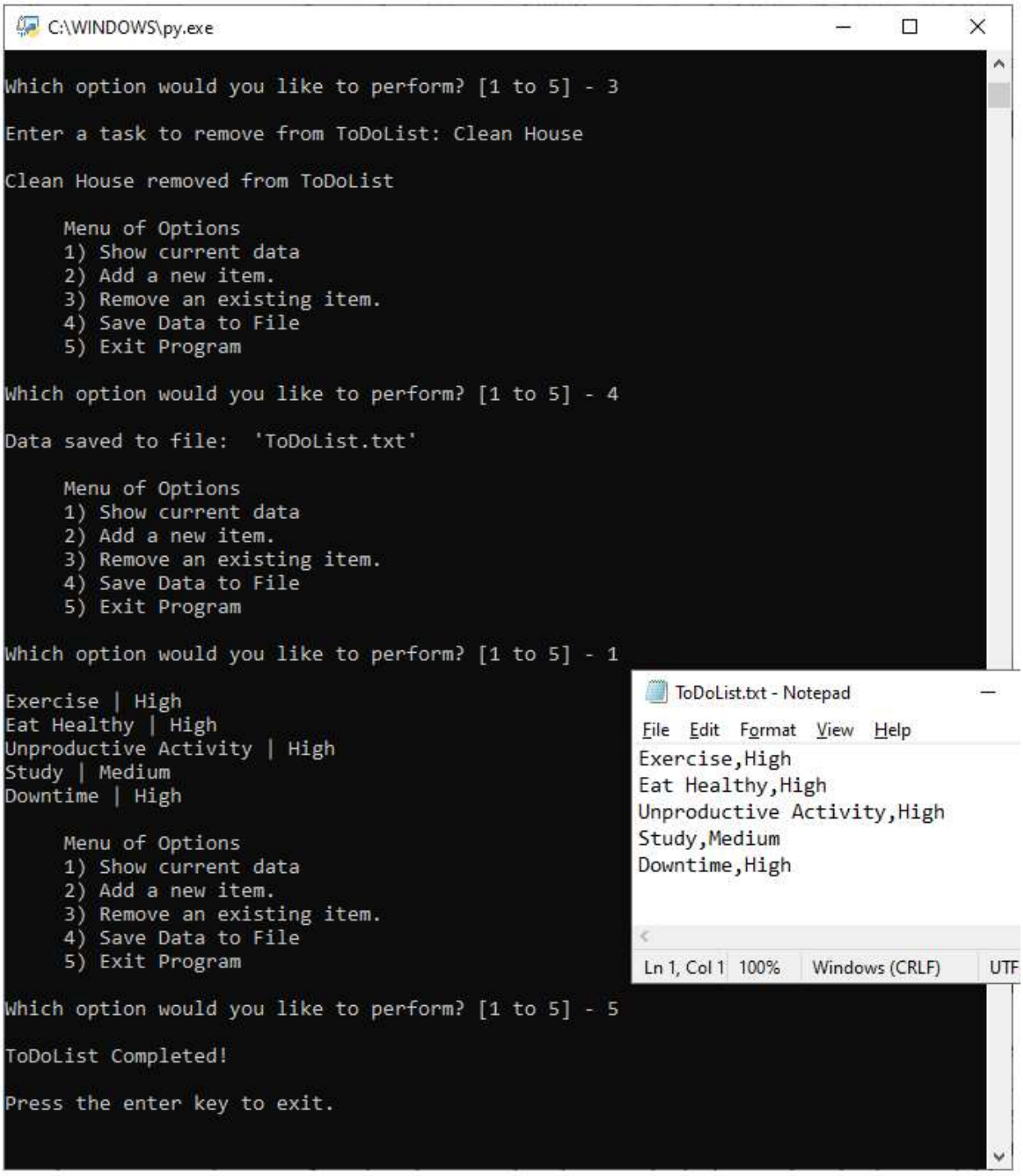
```
Which option would you like to perform? [1 to 5] - 3

Enter a task to remove from ToDoList: Clean House

Clean House removed from ToDoList

     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] - 1

Exercise | High
Eat Healthy | High
Unproductive Activity | High
Study | Medium
Downtime | High
Which option would you like to perform? [1 to 5] - 4

Data saved to file:  'ToDoList.txt'

     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

ToDoList Completed!

Press the enter key to exit.

Process finished with exit code 0
```

Run the Python Script from the Windows OS Command Shell and verify the data in the ToDoList.txt file.






C:\WINDOWS\py.exe
Which option would you like to perform? [1 to 5] - 2
Enter a Task: Clean House
Enter a Priority: [High, Medium, Low] - Low
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] - 4
Data saved to file: 'ToDoList.txt'
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] - 1
Exercise | High
Eat Healthy | High
Unproductive Activity | High
Study | Medium
Downtime | High
Clean House | Low
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 perf
ToDoList.txt - Notepad
File Edit Format View Help
Exercise,High
Eat Healthy,High
Unproductive Activity,High
Study,Medium
Downtime,High
Clean House,Low
Ln 1, Col 1 100% Windows (CRLF) UTF-8


C:\WINDOWS\py.exe
Which option would you like to perform? [1 to 5] - 3
Enter a task to remove from ToDoList: Clean House
Clean House removed from ToDoList
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] - 4
Data saved to file: 'ToDoList.txt'
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] - 1
Exercise | High
Eat Healthy | High
Unproductive Activity | High
Study | Medium
Downtime | High
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
ToDoList Completed!
Press the enter key to exit.
ToDoList.txt - Notepad
File Edit Format View Help
Exercise,High
Eat Healthy,High
Unproductive Activity,High
Study,Medium
Downtime,High
Ln 1, Col 1
100%
Windows (CRLF)
UTF

Summary:

In Assignment05 I modified a provided script that contains starter logic that when completed prompts the user with a menu of options that show current data, add a new item, remove and existing item, saves data to a file and exits the program. This program allows the user to enter a "Tasks" and an associated "Priority" to create a ToDo list. To complete the starter program, I added code to each step in the program at the "TODO" comments. I used a Python dictionary collection to store data in key:value pairs, store dictionary data in a list table, used input/outputs to prompt the user to enter data that appends to and removes data rows from the list table, and writes to and removes data from a delimited text file.