Deborah Yenubari

02/016/2021

Fundamentals of Programming (Python)

Assignment 5

## Question

Modify a script that manages a "ToDo list." The "ToDo" file will contain two columns of data, "Task" and "Priority."

Load the columns into a Python Dictionary object. Each dictionary object represents one row of data, and these rows must be added to a Python List object to create a table of data.

A starting template to modify and use for the program is provided.

# ------------------------------------------------------------------------ #  
# 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  
# <YOUR NAME HERE>,<DATE>,Added code to complete assignment 5  
# ------------------------------------------------------------------------ #  
  
# -- Data -- #  
# declare variables and constants  
objFile = "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  
strChoice = "" # A Capture the user option selection  
strMenu = """  
 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  
""" # A menu of user options  
  
  
# -- 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)  
# TODO: Add Code Here  
  
# -- Input/Output -- #  
# Step 2 - Display a menu of choices to the user  
while (True):  
 print(strMenu)  
 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'):  
 # TODO: Add Code Here  
 continue  
 # Step 4 - Add a new item to the list/Table  
 elif (strChoice.strip() == '2'):  
 # TODO: Add Code Here  
 continue  
 # Step 5 - Remove an item from the list/Table based on its name  
 elif (strChoice.strip() == '3'):  
 # TODO: Add Code Here  
 continue  
 # Step 6 - Save tasks to the ToDoToDoList.txt file  
 elif (strChoice.strip() == '4'):  
 # TODO: Add Code Here  
 continue  
 # Step 7 - Exit program  
 elif (strChoice.strip() == '5'):  
 # TODO: Add Code Here  
 break # and Exit the program

Fig 1: Starter Template

**# Data #**

**# Declare Variables and Constants**

The starter template declared Variables and Constants.

objFile = ‘ToDolist.txt’ to save Data

dicRow = {} variable referring to a Dictionary with key ‘Task’ and value ’Priority’. The subscript is the key or value instead of index number in a List.

lstTable = [] Dictionary as rows of Data are added to a List(a collection of rows, creating a table like two-dimensional collection of Data).

strChoice = “”A Menu of user options

“””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”””

**# Processing #**

**# Step 1 When the program starts load any current data in the text file, ‘ToDoList.txt’ into a python list of dictionaries rows.**

dicRow = {"TASK": "Sort Personal Files on PC", "PRIORITY": "Urgent"}  
lstTable.append(dicRow)  
objFile = open("ToDoList.txt", "a")  
objFile.write(dicRow["TASK"] + "," + dicRow["PRIORITY"] + "\n")  
objFile.close()

dicRow = {“TASK”: “Sort Personal Files on PC”, “PRIORITY’: “Urgent”}

lstTable.append{dicRow}

adding subsequent dictionaries to the lstTable.

objFile = open{“ToDoList.txt”, “a”}

objFile.write(dicRow[“TASK”] + “,” dicRow[“PRIORITY’] + “\n)

writing dicRow keys TASK and PRIORITY with their concurrent values into the file ToDoList.txt

objFile.close()

**# Input/ Output #**

**# Step 2 – Display a menu of choices to the User**

while (True):  
 print(strMenu)  
 strChoice = str(input("Which option would you like to perform? [1 to 5] - "))  
 print()

‘while’ loop in the starter template provides string menu of choices as long as True

while (True):

Print(strMenu)

strChoice = str(input”Which option would you like to perform? [1 to 5] – “))

**# Step 3 - Show the current items in the Table**

if strChoice.strip() == "1":  
 for dicRow in lstTable:  
 print(dicRow)

#if strChoice is 1 then…if not continue…

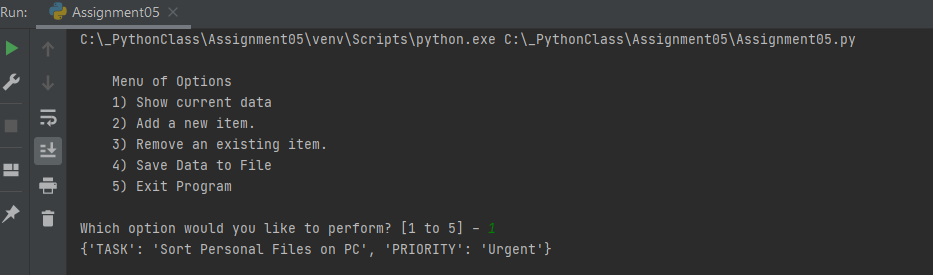
if strChoice.strip() == “1”:

for dicRow in lstTable:

print(dicRow)

continue

if strChoice ==1, print each dicRow from listTable



**Output for # Step 2 and # Step 3 Menu of Options and Show current Data**

**# Step 4 - Add a new item to the List/Table**

elif strChoice.strip() == "2":  
 strTask = input("ENTER A TASK: ").strip()  
 strPriority = input("ENTER ITS PRIORITY LEVEL: ").strip()  
 dicRow = {"TASK": strTask, "PRIORITY": strPriority}  
 lstTable.append(dicRow)  
 continue

# else if strChoice is 2, then…if not continue…

elif strChoice.strip() == ”2”:

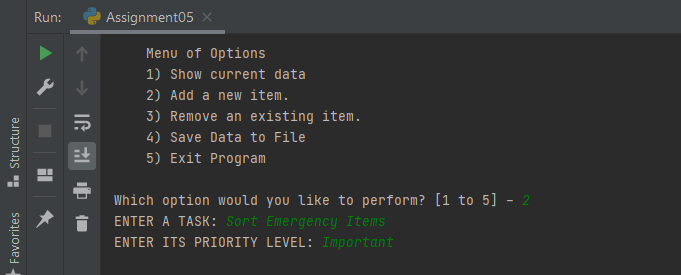
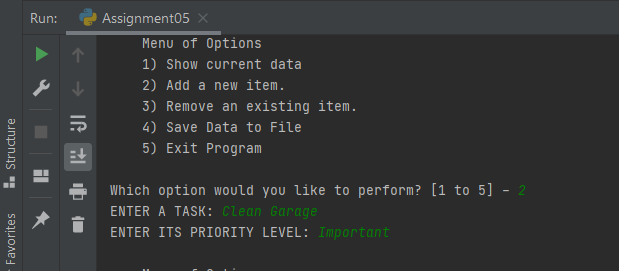
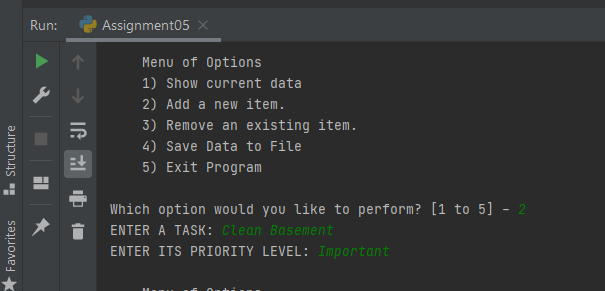
strTask = input(“ENTER A TASK: “).strip() # strip to eliminate unwanted spaces in user input

strPriority = input(“ENTER ITS PRIORITY LEVEL: “).strip()

dicRow = {“TASK: strTask, “PRIORITY”: strPRIORITY)

lstTable.append(dicRow)

continue

**Output for # Step 4 Add a new item to the list/ Table**

**# Step 5 Remove an item from the List / Table by referring to the key ‘Task’**

elif strChoice.strip() == "3":  
 strTask = input("ENTER A TASK TO DELETE: ").strip()  
 for dicRow in lstTable:  
 if dicRow["TASK"] == strTask:  
 lstTable.remove(dicRow)  
 print(strTask, "HAS BEEN DELETED!")  
 continue

else if strChoice == “3” then, look for each dicRow in lstTable, if dicRow TASK in lstTable == strTask, remove that dicRow…if not continue…

elifstrChoice.strip() == “3”:

strTask = input(“ENTER A TASK TO DELETE: “).strip()

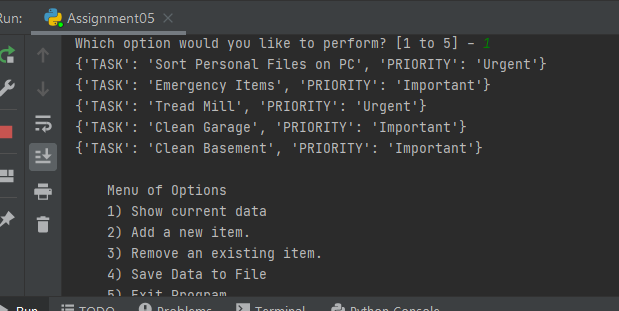
for dicRow in lstTable:

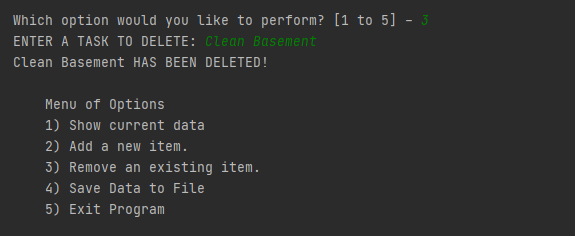
if dicRow[“TASK”] == strTask:

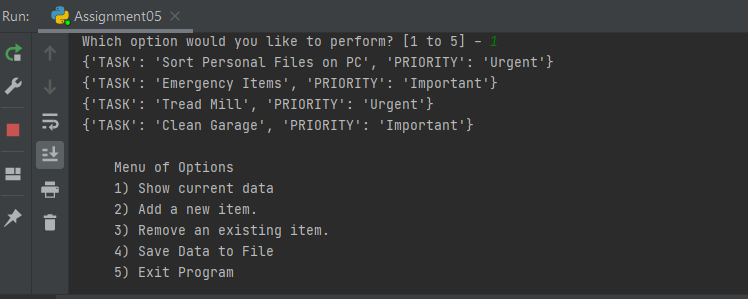
lstTable.remove(dicRow)

print(strTASK, “HAS BEEN DELETED: “)

continue







**Output for # Step 5 ‘Clean Basement’ has been removed from the list Table**

# Step 6 - Save Data to the ToDoList.txt file

elif strChoice.strip() == "4":  
 dicRow = {"TASK": strTask, "PRIORITY": strPriority}  
 objFile = open("ToDoList.txt", "a")  
 objFile.write(dicRow["TASK"] + "," + dicRow["PRIORITY"] + "\n")  
 # lstTable.append(dicRow)  
 objFile.close()  
 print("DATA SAVED TO FILE!")  
 continue

else if strChoice == ‘4” then open objFile “ToDoList,txt”. Stating dicRow with user input key and value. ‘write’ each dicRow TASK concatenate with each DicRow PRIORITY separated by “,”. Close the file. If not continue…

elif strChoice.strip() == “4”

dicRow = {“TASK”: strTask, “PRIORITY”: strPriority}

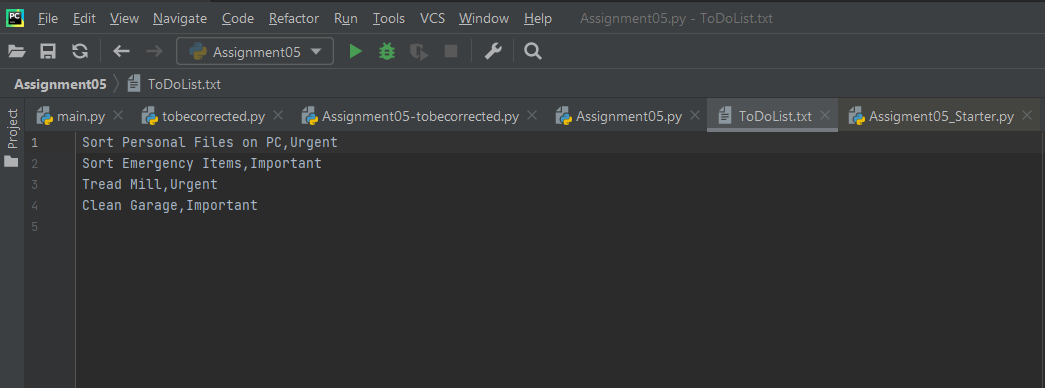
objFile = open(“ToDoList.txt”, “a”)

objFile.write(dicRow[“TASK”] + “,” + dicRow[“PRIORITY”] + “\n”)

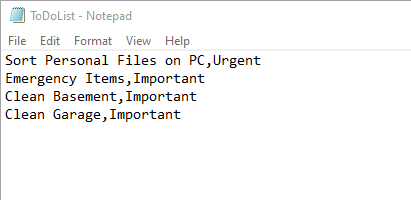
objFile.close()

print(“DATA SAVED TO FILE!”)

continue



**Output for # Step 6, saving data to a text file**



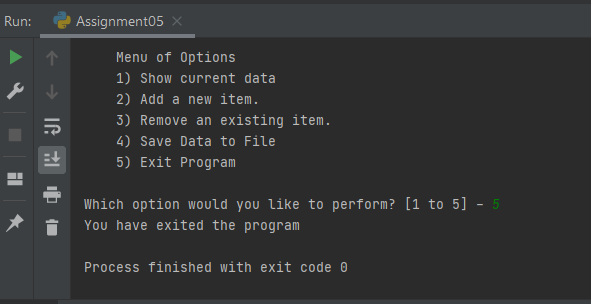
**Output for # Step 6, saving data to a text file**

**# Step 7 – Exit Program**

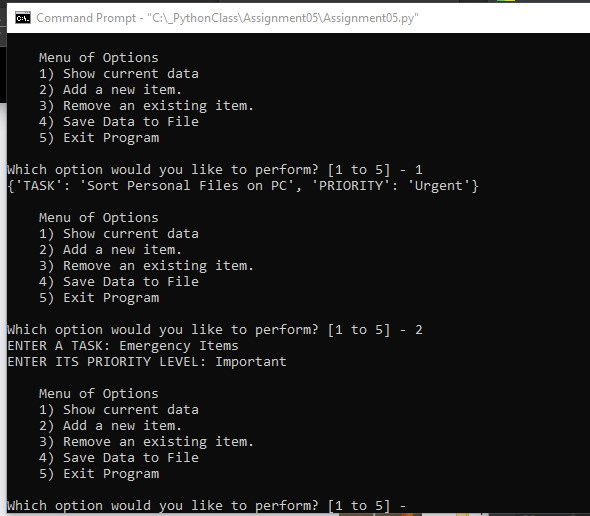
else:  
 print("You have exited the program")  
 break

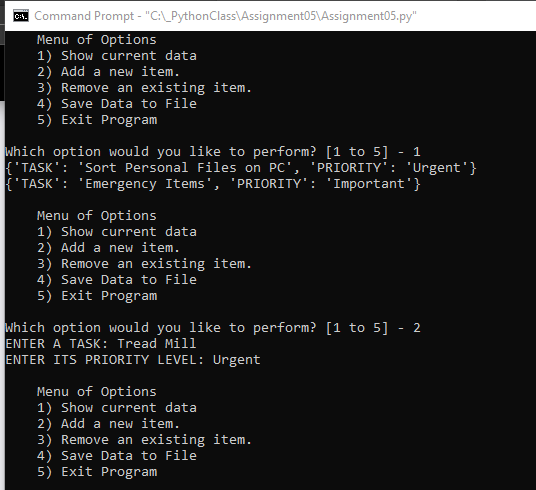
at any point , entering ‘exit’ will do just that. Or 5.

If strChoice == exit or 5.

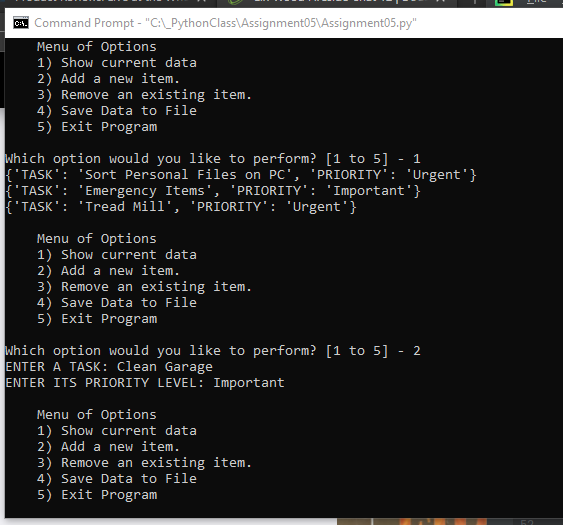


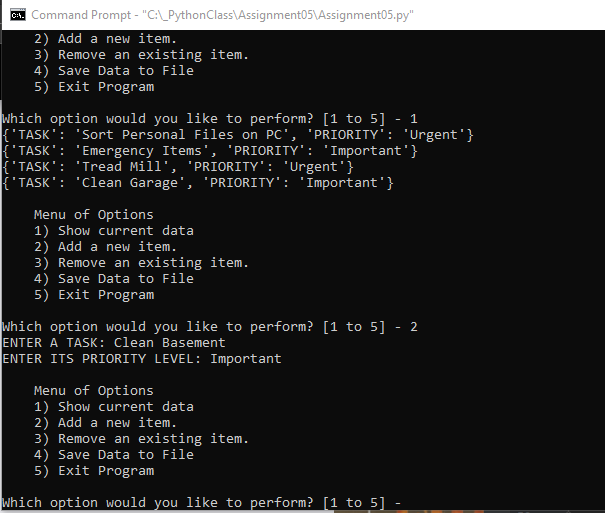
**Output for # Step 7, Exit Program**



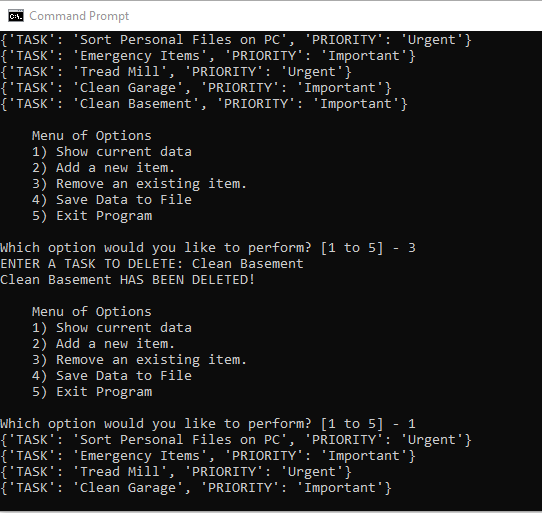


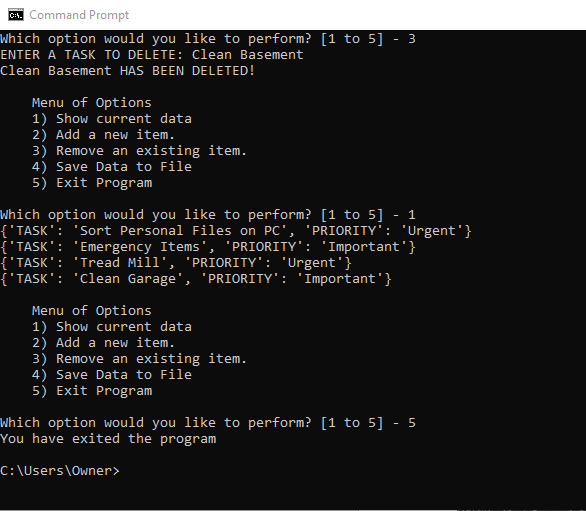
**Output # Step 1, #Step 2 and # Step 4 on Command Prompt Window**





**Output # Step 1, # Step 2 and 4 on Command Prompt Window**





**Output # Step 3 and to # Step 7 on Command Prompt Window**

## Conclusion

The above script used declared variables, including a dictionary row, a list Table of dictionary rows, and a string Menu of options.

Processing included writing current data to a text file, requesting user input using the ‘while’ loop, nested if…elif…continue statement to display current data and request additional user input, elif…continue with a nested ‘for’ loop to remove an user input and else…break to exit the program.