Kimberly Domingo
Date: 8/2/2023
Class: IT FDN 110A
Assignment 04

## Introduction:

This is a documentation of how I completed Assignment04. Step by step, I will go through how I created a program that asks users to input a household item and its value, just like Assignment 03. Unlike Assignment03, the program will allow users to continuously add in data until they exit the program and see the data collected at any point before they exit the program. Once the users are done, the program will exit and save the data into a text file.

## Writing the Code:

In order for me to tackle this assignment, I need to split what I know how to do and what will be the most difficult thing for me to finish the program. Last week, I learned how to get user input, display the output, and save it into a text file. What will be the most difficult thing for me to do is how to continuously get user input and display them.

I started with creating a menu that users can see which contains the options the program can do. These are adding data, displaying data, and saving the data into a text file.

```
print("""
        Kim's Furniture Shopping List """)
print(
      """
          Menu Options
        1. Add data to list
        2. Display current list
        3. Exit and save to file
        """)
```

**Figure 1: Menu options**

This was easy as I was only using the print() function. Next, I decided to split up the code for each of the options. I decided to create a function for each option. For the first option, adding data to the list, I decided to use list instead of tuples as it is more flexible and I can use the built-in functions on them.

I created an empty list. This list is empty because there's no data yet until the user input data on it. Again, the data will contain the household item and value that the user inputs. Then, I start to write my function. I assigned this function as AddData to make it easier for me to understand later when I combine everything together.

Inside this function, I assigned two variables for the user input on the household item and value. This uses the input function and I nested it inside str(), so the input will be a string data type.

Next, I assigned the variable “item” as a list containing the two inputs. This is one list. This is good if the user input only has one input, but we want the program to be able to handle a list that contains multiple data. This is where the empty list comes in. I want to create a multi dimensional list that contains however many inputs the users put in. The empty list is my multidimensional list. I add data on it using the built-in function that lists has. In case, I will be using the .append() function. This will allow users to add data on the list without creating a new separate one.

```
list = []

def AddData():
    global list
    item_name = str(input("Enter an item: "))
    item_cost = str(input("Enter the estimated value: "))
    item = [item_name, item_cost]
    print(item)
    list.append(item)
    print(list)
```

**Figure 2: Function for the first menu option**

There are a few things to note with this. The first one is the print() functions. This is one way I can make sure that my code is doing exactly what I want it to. The next is the **global variable.** Because the “list” variable is outside my function, this global variable allows me to use the “list” variable inside this specific function, outside this function, or any other functions in my program.

Once I validated this function, I moved on to the next function which is displaying the output. I used the for loop to accomplish this. There are two loops that I used in my program, but the for loop was more suitable for this code. In simplest terms, the for loop repeats a block of code automatically. One thing to note is that the for loop automatically stops when the last value is processed unlike the while loop.

```
def DisplayData():
    for x in list:
        print("Furniture: " + str(x[0]) + "| " "Cost: $" + str(x[1]))
```

**Figure 3: The for loop repeats a specific block of code**

I assigned this function as “DisplayData”. Inside the function, I used the for loop to continuously print out the user’s input. I added extra strings inside the print() function to make it look cleaner. The list contains each individual list containing the item and value. I assigned x to this inside lists. The str(x[0]) will print out the first “column” in the list, which is the household item. As for the value, it is positioned at the second position, which is why it is x[1].

Next is to save the data to the text file. I used the objfile function from last week to do this. First, you have to open the text file with the objfile = open() function. Once it is open, you can use various functions to manipulate what goes in the text file. What I wanted is to print the data the user inputted. I used the objfile.write() function. You can consider this function the same as the print() function. To make sure the same data is being printed out as the output with my second function. I used the same for loop inside to do this.

```
def SaveData():
    objfile = open("Home Inventory.txt", "w")
    for x in list:
        objfile.write("Furniture: " + str(x[0]) + "| " "Cost: $" + str(x[1]) + '\n')
    objfile.close()
```

**Figure 4: .write() function can be considered as a print() function**

If I don't use the for loop and just use objfile.write() function, it will cause an error since x is not defined using I define it in my loop. I have also tried to use objfile.write(str(list)). This works, but it will just print it out in a list form. I wanted to clean it up and make it look prettier, so I decided to use the same for loop I used on the displaydata function.

Now that I have all the functions working, I need to combine all three together. I used the while loop to combine all the functions together. In my while loop, I need to assign a condition. I used while(True): in this case. This means that every time a statement is true inside my while, it will continuously loop without an end. This is what I needed, because I want my program to run as if the user picks 1, 2, or 3, it will execute the specific functions for each. This will also cause an error because there's no end to the loop. So, I need to make it go back to the beginning of the loop when the user picks 1 or 2, which is the main menu where the users get to choose what they want to do with the program. Another one is to break when the user picks 3, so this will break the loop which is what we want when the user decides to exit and save the program.

```
while(True):

    print(color.BOLD  + color.DARKCYAN + """
          Kim's Furniture Shopping List """ + color.END)
    print(
        """
                Menu Options
            1. Add data to list
            2. Display current list
            3. Exit and save to file
          """)

    user_choice = str(input("Please select an option [1 to 3]: "))
    if user_choice == "1":
        AddData()

    elif user_choice == "2":
        DisplayData()

    elif user_choice == "3":
        SaveData()
        print(color.BOLD + color.GREEN + "You have saved this data" + color.END)
        DisplayData()
        break
    else:
        print(color.BOLD + color.RED + "Please only choose from option [1, 2, or 3]" + color.END)
```

**Figure 5: Functions inside the while loop**

This is a working program that executes what the Assignment is intended to do. I added extra features to clean up the program more.

## Summary:

Overall, I went over the steps that I did to accomplish Assignment04. This is a working program with extra features to challenge myself.