

Attender Pal Singh

(U01698955)

14th Dec 2019

Project2: - Computer Accessories Shop

CS-632p

Table of Contents

| S.No | Content | Page No. |
|-------------|---------------------------------------|-----------------|
| 1 | Instruction for running the program | 3 |
| 2 | Screen of Running Program | 4 |
| 3 | Flow Charts and explanation of Design | 7 |
| 4 | Testing Screenshots | 8 |
| 5 | Further Enhancements | 8 |
| 6 | Source Code | 8 |

Instructions for running the shop:-

- a) Import Time and Import SYS for the proper execution of Accessories Shop.
- b) The shop is all about the computer accessories inventory and addition and deletion of new items into the shop with a feature of saving a .txt file with new updated data.
- c) There are 8 action buttons which will help you to move forward in the code.
 - 1) Inventory
 - 2) Add a new item
 - 3) Lookup item by number
 - 4) List items that are low on inventory
 - 5) Export to file
 - 6)Exit/Quit
 - 7) Add to inventory
 - 8) Subtract from inventory
 - 9) Return to main menu
- d) With option 1, you can view the whole Inventory with already 10 items present in 3 different dictionaries.
- e) Option 2 will help you to add new item in the Inventory with the predefined rule of setting as Quantity and Description. For example, (5, USB) where 5 is the quantity of item to be added and USB is the desc of the item with a delimiter as comma.
- f) Option 3 will show you the items with respect to the Item Number. And there are 3 options available to 1) Add, 2) Subtract and 3) Return to Main Menu.
- g) Option 4 will show the item which are low in number with respect to Quantity in the Inventory.
- h) Option 5 will save the file in Updated-File.txt on the Folder where we have the Python script.
- i) Option 6 will Exit/Quit the program

Screen-Shots of Computer Accessory Shop:-

```
***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above:
```

```
Enter choice from menu above: 1
#Item Number    #Quantity    #Price    #Item Description
0001             3             $ 20      Wireless Mouse
0002             3             $ 40      Wireless Keyboard
0003             2             $ 80      19" Monitor
0004             2             $ 100     23" Monitor
0005             5             $ 15      HDMI Cable
0006             5             $ 10      VGA Cable
0007            10             $ 5       USB Cable
0008             5             $ 5       Power Cable
0009             3             $ 25      8GB Thumb Drive
0010             4             $ 35      16GB Thumb Drive

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above:
```

```
***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 2

Enter Quantity number and description comma delimited: (e.g, 5,USB) 3,Laptop Cable

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above:
```

```
Enter Quantity number and description comma delimited: (e.g, 5,USB) 3,LaptopCable

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 1
#Item Number    #Quantity    #Price    #Item Description
0001             3             $ 20      Wireless Mouse
0002             3             $ 40      Wireless Keyboard
0003             2             $ 80      19" Monitor
0004             2             $ 100     23" Monitor
0005             5             $ 15      HDMI Cable
0006             5             $ 10      VGA Cable
0007            10             $ 5       USB Cable
0008             5             $ 5       Power Cable
0009             3             $ 25      8GB Thumb Drive
0010             4             $ 35      16GB Thumb Drive
0011             3             $ None    LaptopCable
```

```

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 3
Enter item number: 0001
#Item Number      #Quantity      #Price      #Item Description
0001              3              $ 20        Wireless Mouse

7: Add to inventory
8: Subtract from inventory
9: Return to main menu
Enter Choice from above menu:

```

```

7: Add to inventory
8: Subtract from inventory
9: Return to main menu
Enter Choice from above menu: 7
Please enter the value to amend inventory by: 2

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 1
#Item Number      #Quantity      #Price      #Item Description
0001              5              $ 20        Wireless Mouse
0002              3              $ 40        Wireless Keyboard
0003              2              $ 80        19" Monitor
0004              2              $ 100       23" Monitor
0005              5              $ 15        HDMI Cable
0006              5              $ 10        VGA Cable
0007              10             $ 5         USB Cable
0008              5              $ 5         Power Cable
0009              3              $ 25        8GB Thumb Drive
0010              4              $ 35        16GB Thumb Drive
0011              3              $ None      LaptopCable

```

```

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 3
Enter item number: 0001
#Item Number      #Quantity      #Price      #Item Description
0001              5              $ 20        Wireless Mouse

7: Add to inventory
8: Subtract from inventory
9: Return to main menu
Enter Choice from above menu: 8
Please enter the value to amend inventory by: 1

```

```

Enter choice from menu above: 1
#Item Number      #Quantity      #Price      #Item Description
0001              4              $ 20        Wireless Mouse
0002              3              $ 40        Wireless Keyboard
0003              2              $ 80        19" Monitor
0004              2              $ 100       23" Monitor
0005              5              $ 15        HDMI Cable
0006              5              $ 10        VGA Cable
0007              10             $ 5         USB Cable
0008              5              $ 5         Power Cable
0009              3              $ 25        8GB Thumb Drive
0010              4              $ 35        16GB Thumb Drive
0011              3              $ None      LaptopCable

```

```

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above:

```

```

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 5

Enter your name to save file (extension will be added automatically): Atti

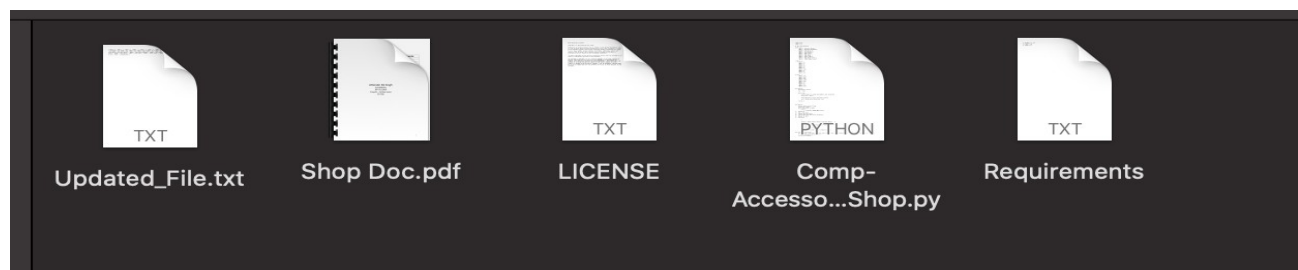
INVENTORY EXPORT SUCCESSFUL. File was saved as 'Updated_File.txt'. [ENTER]

```

```

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above:

```



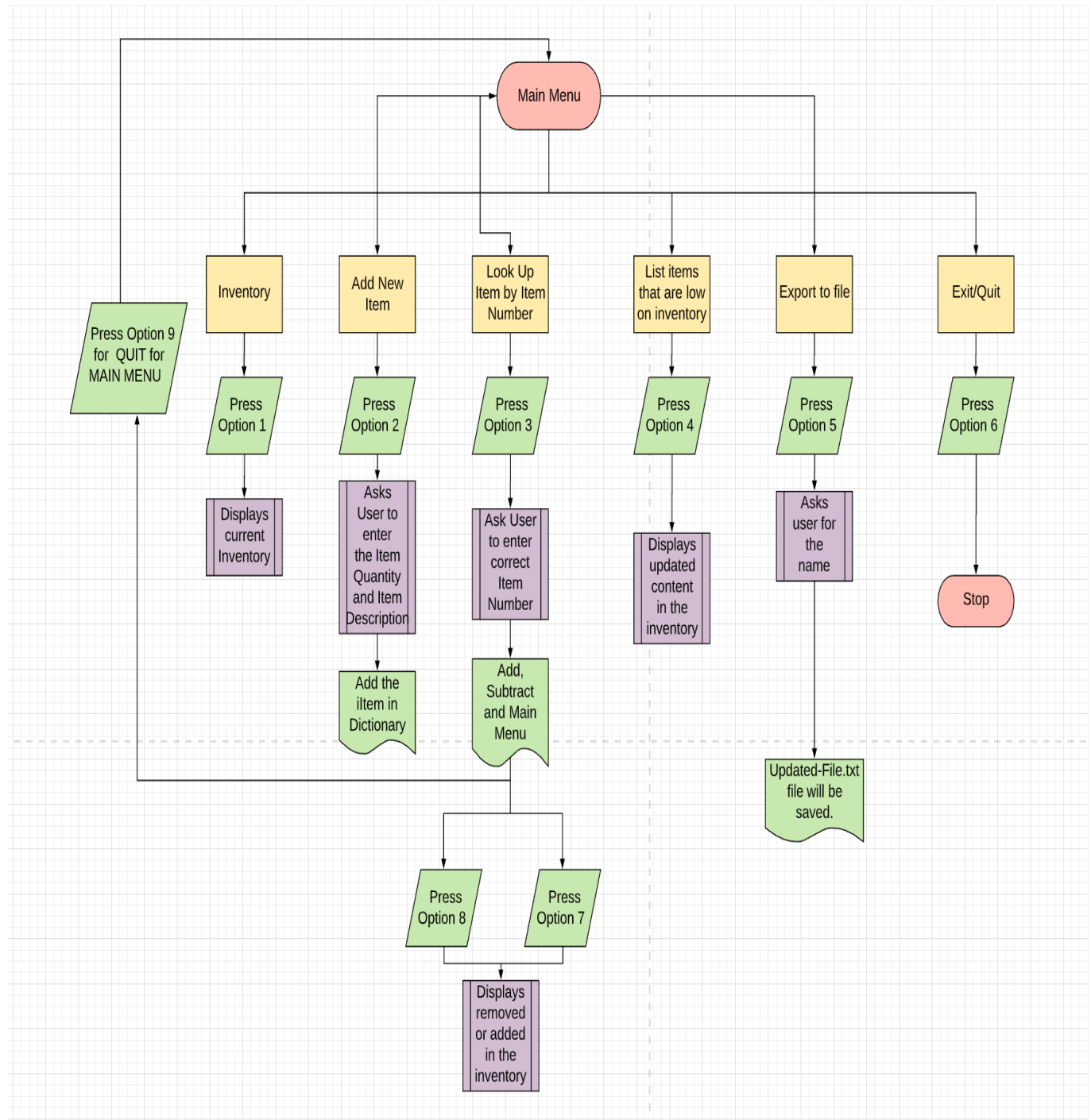
```

***** MAIN MENU *****
1: Inventory
2: Add a new item
3: Lookup item by number
4: List items that are low on inventory
5: Export to file
6: Exit/Quit
Enter choice from menu above: 6

Process finished with exit code 0

```

Flow-Chart:-



Details of Further Enhancement: -

Software can be widely enhanced. We can add parameters to calculate the Retail price, Original price and Profit.

We can increase the complexity while adding and removing items for the inventory.

.txt file can be encrypted with password and only authorized person will be able to view the data.

Source Code:

```
import sys
import re

# Init dictionaries
desc = {
    '0001': 'Wireless Mouse',
    '0002': 'Wireless Keyboard',
    '0003': '19" Monitor',
    '0004': '23" Monitor',
    '0005': 'HDMI Cable',
    '0006': 'VGA Cable',
    '0007': 'USB Cable',
    '0008': 'Power Cable',
    '0009': '8GB Thumb Drive',
    '0010': '16GB Thumb Drive'}

items = {
    '0001': 3,
    '0002': 3,
    '0003': 2,
    '0004': 2,
    '0005': 5,
    '0006': 5,
    '0007': 10,
    '0008': 5,
    '0009': 3,
    '0010': 4}

price = {
    '0001': 20,
    '0002': 40,
    '0003': 80,
    '0004': 100,
    '0005': 15,
    '0006': 10,
    '0007': 5,
    '0008': 5,
    '0009': 25,
    '0010': 35}

def main():
    # do/while control
    run = True
```



```

while run:
    # print menu to screen and gather user selection
    selection = menu()

    # dictswitch for user selection control
    run = defswitcher(selection, run)

return 0

def menu():
    # user input control-loop
    selection = None
    while selection is None:
        print(
            "\n***** MAIN MENU *****"
            1: Inventory
            2: Add a new item
            3: Lookup item by number
            4: List items that are low on inventory
            5: Export to file
            6: Exit/Quit ""

        )

        choose = input("Enter choice from menu above: ")

        # Check validity of user selection, rerun if invalid
        if re.fullmatch("[1|2|3|4|5|6]", choose):
            selection = choose
            return selection
        else:
            print("\nPlease select a valid option")

def export_inventory():
    username = str(input("\nEnter your name to save file (extension will be added automatically): ") + ".txt")
    f= open("Updated_File.txt", "w")
    f.write(str(items))
    f.write(str(price))
    f.write(str(desc))
    f.close()
    input("\nINVENTORY EXPORT SUCCESSFUL. File was saved as 'Updated_File.txt'. [ENTER]")

def defswitcher(selection, run):
    switcher = {
        '1': printall,
        '2': additem,
        '3': lookup,
        '4': lowinv,
        '5': export_inventory,
    }
    # exit control
    if selection is '6':
        run = False
    else:
        # get function name
        process = switcher.get(selection)
        # execute correct function
        process()
    return run

def printall():

```

```

print(" #Item Number".ljust(20) + "#Quantity ".ljust(20) + "#Price".ljust(20) + "#Item Description")
# iterate though and print
for key, value in desc.items():
    print("    " + key.ljust(19) + str(items.get(key)).ljust(17) + "$ " + str(price.get(key)).ljust(18) + value)
print()

def additem():

    # user input control-loop
    add = []
    while not add:
        # grab user input and split on comma
        addinput = input("\nEnter Quantity number and description comma delimited: (e.g, 5,USB) ")

        # Check validity of user selection, rerun if invalid
        if re.fullmatch('([\d])+[.]\S+([\w\s])+', addinput):
            add = addinput.split(',')
            itemnum = newitemnum()
            updatedictionaries(itemnum,add)
        else:
            print("\nPlease follow proper formatting")

def newitemnum():
    # sort and grab highest key value, look for cleaner way to do this
    lastknown = sorted(desc.keys())[-1]
    itemnum = int(lastknown)
    itemnum += 1

    return str(itemnum).zfill(4)

def updatedictionaries(itemnum, update):
    global items, desc, price
    # this could be cleaner for inv change/add new, but went this route due to time constraint
    if not update:
        #add to total inventory
        if itemnum[0] == '7':
            addcount = int(itemnum[1])
            newtotal = items.get(itemnum[2])
            newtotal = int(newtotal)
            newtotal += addcount
            items.update({itemnum[2]: newtotal})
        # subtract from inventory
        elif itemnum[0] == '8':
            addcount = int(itemnum[1])
            newtotal = items.get(itemnum[2])
            newtotal = int(newtotal)
            newtotal -= addcount
            items.update({itemnum[2]: newtotal})
    else:
        items.update({itemnum: update[0]})
        desc.update({itemnum: update[1]})

def lookup():
    global s, desc
    # user input control-loop
    itemnum = None
    while not itemnum:
        itemnum = input("Enter item number: ")

        if re.fullmatch('\d{4}', itemnum) and itemnum in items:

```

```

        print(" #Item Number".ljust(20) + "#Quantity ".ljust(20) + "#Price".ljust(20) + "#Item Description")
        print("    " + itemnum.ljust(19) + str(items.get(itemnum)).ljust(17) + "$ " + str(price.get(itemnum)).ljust(18) +
desc.get(itemnum) + "\n")
    else:
        print(itemnum + " does not exist, try again")
        itemnum = None
    changeinv(itemnum)

def changeinv(itemnum):
    getfuncval = []
    while not getfuncval:
        change = input(
            "7: Add to inventory
8: Subtract from inventory
9: Return to main menu
Enter Choice from above menu: "
        )
        if re.fullmatch('[7|8]', change):
            value = None
            while not value:
                value = input("Please enter the value to amend inventory by: ")
                if re.fullmatch('[\d]+', value):
                    getfuncval = []
                    getfuncval.append(change)
                    getfuncval.append(value)
                    getfuncval.append(itemnum)
                else:
                    print("\nPlease follow proper formatting")
            elif re.fullmatch('[9]', change):
                getfuncval = change
                print("\nPlease follow proper formatting")
            update = []
            updatedictionaries(getfuncval, update)

def lowinv():
    #wanted to merge this with one printall function but time constraint
    print(" #Item Number".ljust(20) + "#Quantity ".ljust(20) + "#Price".ljust(20) + "#Item Description")
    # iterate though and print
    for key, value in desc.items():
        if items.get(key) < 3:
            print("    " + key.ljust(19) + str(items.get(key)).ljust(17) + "$ " + str(price.get(key)).ljust(18) + value)
    print()

if __name__ == "__main__":
    sys.exit(main())

```

