Islington College

Information System

CC4002NA

Coursework 2

Submitted By:

Summit Shakya

NP01CP4A170019

Group: L1C9

Date: Jan 12, 2018

Submitted To:

Mr. Sukrit Shakya

Module Leader

Information System

Proposal

This proposal is written to address the coursework of Information System which is an individual task. The coursework is about developing codes in python that demonstrates the billing system of an electronic store which sells electronic appliances and maintains the stock in a text file. The coursework was given in 7th week and is required to be fulfilled by 11th week.

- **Purpose**

  The purpose of the coursework is to write algorithm and pseudocode for the codes in python which provides billing system and also prepare flowchart of the program.

- **Problem statement**

  Any retail store would need a system to store their sales and their inventory. The store must have all the records about the appliances that they have sold and also would require record of the appliances that they have in stock.

- **Aims and objectives**

  The main aim of this project is to fulfil all the assigned tasks in the coursework in the best way possible. Various researches on file handling will be done to fulfil the project and much efforts will be made on it.

- **Proposed approach**

  Various journals and books regarding file handling in python will be used as research materials for the project. Few suitable data structures will be chosen to store different data. Then algorithm, pseudocodes and flowchart will be prepared for proper organisation of the program.

- **Target audience**

  The main target audience for this project are the retailers. Anyone interested to learn about file handling in python and billing systems through

python can utilize this project. It shows the way in which the billing system and file handling can be performed in python.

- **Hardware and software requirements**

There are no any special hardware requirements for the program to execute. For the software, python 3.6 should be available.

- **Activity description and timeline**

The given project is definitely not an easy task. It is required to finish it in 5weeks. It will be completed in a planned and systematic manner.

  - In the first week, various research about file handling will be done.
  - In the second week, various modules will be created and planning for coding will start.
  - In the third week, algorithms flowcharts and pseudocodes will be prepared which will be very helpful for program development.
  - In the fourth week, the coding part will be completed after suitable data structures are found.
  - In the final week, testing of the code will occur so that there are no bugs in the program.

Table of Contents

Table of Figure

Table of Tables

1. Introduction

This is a report for the coursework which had asked us to develop a system in python for an electronic store. There, we had to write a program which would store the inventory of the shop and record the transactions and generate invoices of the appliances sold. The inventory would be stored in a text file where the quantity of products remaining and the price of the product is stored. The program will read the text file and take name of customer and required product name as input from the user and then process as per assigned condition. Then the program will generate the invoice as per the products bought by customer and it will also update the inventory.

The assessment was definitely not an easy one. We have to start preparing an algorithm, then flowchart then pseudocode and then develop the codes in python which was lengthy to some extent. Finding out a lot of errors and debugging them was a difficult task. Building a system for retailer is not a simple ask because the system that we have designed will directly effect the business. If we create a small mistake during the development of the system, it will directly harm the retailer's business. The main aim of the project was to fulfil the assessment in the given time period in the best way without any bugs in the project.

1.1 Features of the program:

The features of the program are:

- ✓ The program will be able to read and display the text file through python.
- ✓ It can take the name of product from user and search for it in the inventory.
- ✓ It allows user to choose product and can generate the invoice for the products purchased.
- ✓ It can also calculate grand total after adjusting discount.
- ✓ Along with generating the invoice after the purchase of the product, the program can also update the inventory along with the products sold.

2. Discussion and Analysis

In this coursework we had to develop an application for an electronic store, where the program would record the transactions of the shop and also would update the inventory on the basis of the number of products sold. We also had to prepare and present the algorithm, flowchart and pseudocode for the program. It was not as easy as I seemed to be. With proper effort and proper guidance, all the assessment was completed.

Some tools were backbone for the development of this project which are listed below:

2.1 Python 3.6

Python is an object-oriented programming language. Python was used to develop all of the codes of this project. Since the electronic store stored their inventory in a text file and needed it to be updated every time goods are sold, python was the best option to develop the codes for the project as it could easily read text files, easily write and also manipulate the data in text file. Without python, developing the system would have been a big challenge.

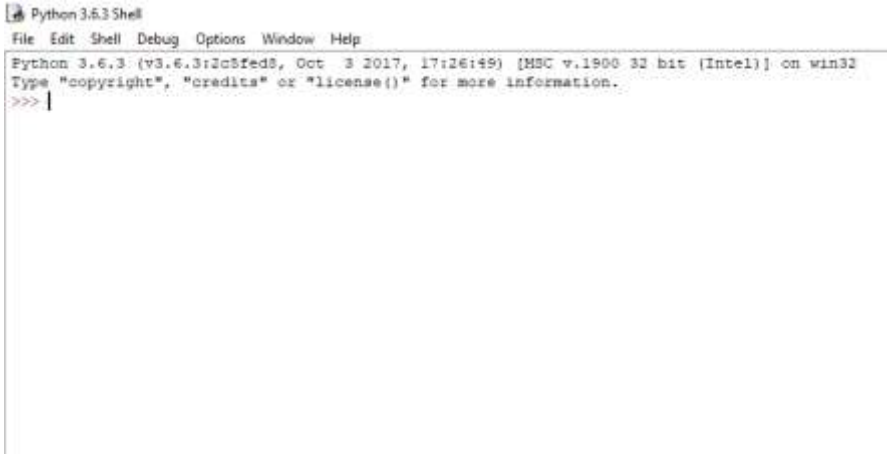

Figure 1 Python Shell

2.2 Draw.io

In one of task in the coursework, a flowchart had to be prepared for the program. Flowchart could also be created in MS-Word but it would be a hectic and time consuming task. So draw.io was chosen to create the flowchart which made preparing a flowchart much more simple and easier.

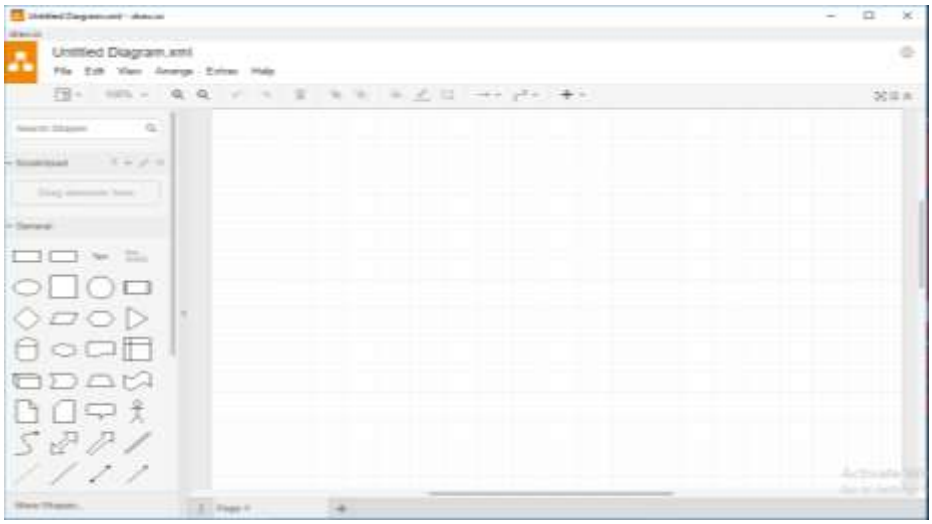

Figure 2 Draw.io

3. Algorithm

3.1 Algorithm

Step-1 Start

Step-2 Read text file

Step-3 Display the inventory

Step-4 Enter customer name

Step-5 Enter name of the product required

Step-6 If product is available

Then goto step 7

Else goto step 5

Step-7 Enter number of products required

Step-8 If user want to buy more products

Then goto step 5

Else goto step 9

Step-9 Name, price and total cost of product is displayed

Step-10 Ask discount percentage

Step-11 Calculate total after deducting discount

Step-12 Display total

Step-13 Updated inventory is displayed

Step-14 If user want to continue with more customer

Then goto step 4

Else goto step 15

Step-15 Stop

3.2 Flowchart

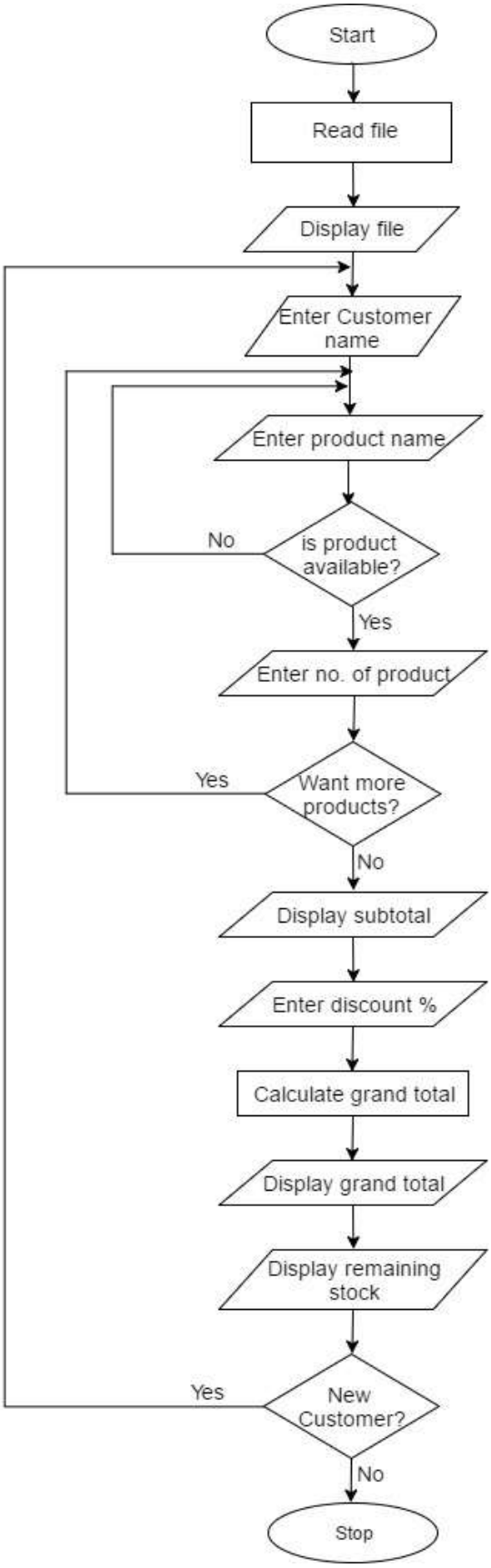

Figure 3: Flowchart

The program starts after the inventory is read and displayed. Then after the name of the customer is asked to input. Then the user has to enter the name of the product that has to be purchased. If the entered product is available, then the quantity of the product is asked or else the user is asked to enter the name of the product again. Once the name and quantity of one product is stored, the user is asked if they wanna buy more appliances. If the user wants to buy more appliances then again the name and quantity of product is recorded and if the user does not want to buy more product, then the subtotal is displayed. The user is asked to enter the amount of discount percentage. Then after the grand total is calculated. The remaining products in inventory is displayed. If the user wants to add a new customer, then the program will again begin from asking the customer's name or else the program stops its execution.

3.2 Pseudo code

Pseudo codes are steps of the program written in simple English language. The pseudo code for this project is attached below. (Gilberg & Forouzan, 2005)

A. Pseudo code for main.py

```
import read
import write
import purchase
newcust="y"
while newcust=="y"
        p=read.stock()
        q=purchase.purchase(p)
        write.inventory(p,q)
        newcust=input("new customer?y/n")
end while
print("Thank you! Please visit again.")
```

B. Pseudo code for read.py

```
Define stock
        file=open
        read=file.readlines
        list_= []
        for x in read:
            list_.append (x.replace (“\n”,””).split (“,”))
        end for
        file.close
        for i in range (len (list_)):
            output products
        end for
end stock
```

C. Pseudo code for write.py

```
Define inventory (x,y)
list_=x
dist=y
for k in dist.keys():
    if k=="phone":
        list_[0][2]=str(int(list_[0][2])-dist["phone"])
    elif k=="LAPTOP"
            list_[1][2]=str(int(list_[1][2])-dist["laptop"])
    else:
            list_[2][2]=str(int(list_[2][2])-dist["hdd"])
    end if
end for
print(list_)
file=open
for row in x:
    file.write(str(",".join(row)))
    file.write("\n")
end for
```

```
file.close
end inventory
```

D. Pseudo code for purchase.py

```
Define purchase(p):
   list_=p
   Input name of customer
   items={}
   condition="y"
   while ans=="y":
      Input name of product, prod
      prod1=prod.lower()
      if prod1==list_0][0].lower() or prod1==list_[1][0].lower() or
prod1==list_[2][0].lower():
                numprod=int (input ("enter the number of products: "))
                 if prod1==list_[0][0].lower() and numprod<=int(list_[0][2]):
                 items [goods1]=numprod
                 elif prod1==list_[1][0].lower() and numprod<=int(list_[1][2]):
                 items[goods1]=numprod
                 elif prod1==list_[2][0].lower() and numprod<=int(list_[2][2]):
                 items[goods1]= numprod
                 else print ("Sorry! We dont have adequate number of products")
                 condiiton=input("Do you want to buy more? ")
       else:
                print ("Sorry! We do not have the product that you are searching
for.")
          End if
       End while
        total =0
        for k in items.keys():
        if keys=="phone ":
                phoneprice = int(list_[0][1])
```

```
            phoneqty=int (items["phone"])
            phonecost= (phoneprice*phoneqty)
             total += (phoneprice*phoneqty)
             print("price of the phone is: ", phonecost)
elif keys=="laptop ":
            laptopprice=int(list_[1][1])
            laptopprice=int(items["laptop"])
            laptopcost=(laptopprice*laptopqty)
            total += (lapttopprice*laptopqty)
            print("price of the laptop is: ", laptopcost)
else:
            hddprice = int([list_1][2])
            hddqty=int (items["hdd"])
            hddcost = (hddprice*hddqty)
            ttotal +=(hddprice*hddqty)
             print ("Price of HDD is; ", hddcost)
            print ("Total price of the items: ", total)
             dis = float(input("any discounts, if available(%): "))
             total1=float (value-(discnt*value)/100)
             print ("Grand total=: ",total1)
```

End If

End For

Import date and time

Generate Invoice

Display Inventory

4. Data Structures

4.1 Integer

Integer data type is generally used to store integer values. Here in the project, integer data type has been used in places where suitable such as at times when the quantity of the product has to be stored. Obviously the quantity of products is not a string or it cannot be in float, for example we cannot say there are 2.3 number of phones in inventory. So, the integer data type has been used to store and manipulate number of products in write.py file.

4.2 String

String data type is typically used to store characters in most of the conditions. But at times, it can also be used to convert integer into string for easier concatenation. Here in the project we have to store customers name, product name and various other string messages where the string data type has been used. String value cannot be stored in an integer data type or float or any other data type. So the only alternative was to store it in a string data type. It has been mostly used in purchase.py file.

4.3 List

Lists can store numerous data in a single variable. In this project, list has been used to store and display the products that are available in the inventory of the store in read.py file. In write.py file, the 2D list has been used to update the quantity of products in the inventory. List has also been used in purchase.py file to sort out if the product input by user is available or not.

4.4 Dictionary

Dictionary has a special feature to store data in a key: value format which allows easy storing of name or product and its price while the customer choses to

buy any appliance from the store. There would have been no better option than dictionary to choose to store name and price of product side by side.

5. Program

The program in this project contains four modules which are listed below:

5.1 Main.py

Main.py is the main file in this project which inherits purchase.py, read.py and write.py. The keyword ‘import’ is used to inherit. The keyword ‘import’ allows us to call the function of inherited modules. All the modules of the project are executed from this module ‘main.py’

The program is executed and firstly calls the function from read.py. Then the function from purchase.py is called with the parameter and then the function from write.py is called with two parameters. After completion of one customer, the program asks if there are more customers. If yes, then the program is executed again or else the program closes with a thank you message.

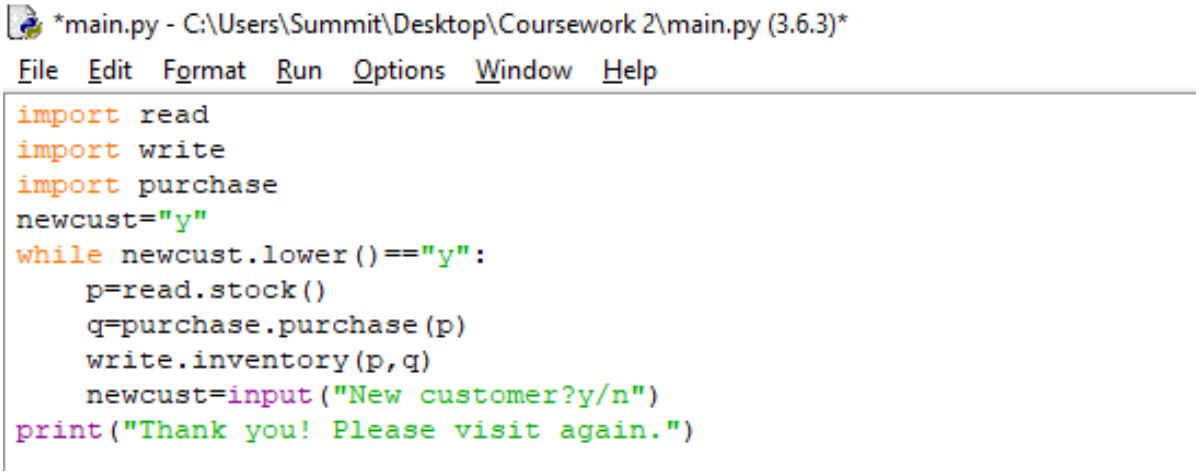

Figure 4: main.py

5.2 Write.py

This is program file through which the inventory of the store is updated which means the quantity of the product decreases in the inventory when some products are sold. After updating the inventory, the program again writes the remaining quantity to the inventory text file.

write.py - C:\Users\Summit\Desktop\Coursework 2\write.py (3.6.3)

File Edit Format Run Options Window Help

```
def inventory(x,y):
    list_=x
    dist=y

    for k in dist.keys():
        if k=="phone":
            list_[0][2]=str(int(list_[0][2])-dist['phone'])
        elif k=="laptop":
            list_[1][2]=str(int(list_[1][2])-dist['laptop'])
        else:
            list_[2][2]=str(int(list_[2][2])-dist['hdd'])
    print (x)

    file=open("stock.txt","w")  #opens same txt file in write mode.
    for row in x:
        file.write(str(",".join(row))) #write to the file
        #file.write("\n")  #break line
    file.close() #close file
    return
```

Figure 5: write.py

5.3 Read.py

Read.py file reads the inventory file and then displays the available product along with its price and quantity remaining to the customer for them to choose the required product. The program extracts the data from the text file and writes it in a 2D list and then displays it in a specified format.

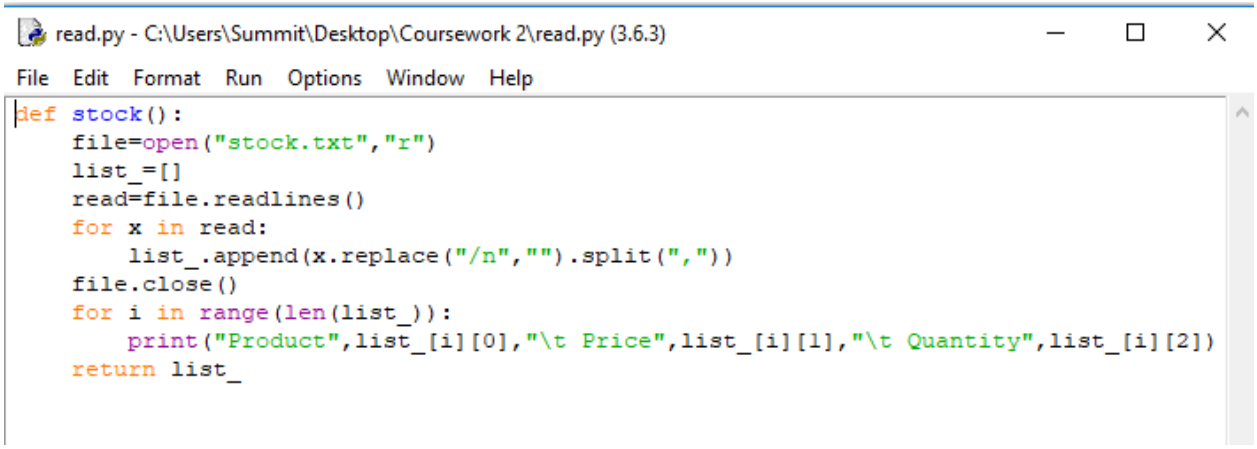

Figure 6: read.py

5.4 Purchase.py

In purchase.py, it takes a list as parameter which contains the available products in the store. Then it takes customer name as input. Then after it asks the name of the product that user want to purchase. If the product is available, then it asks to input the quantity required. If the product is not available, then it asks to enter other product. Then after the name and price of one appliance is stored. It asks if user wants more product. If yes, then it repeats the process else it displays the subtotal. The program then asks user to input discount percentage. Then the grand total is displayed. Then an invoice text file with unique id is written. It asks if there are more customer. If yes, then the whole program repeats again and if no then the program terminates.

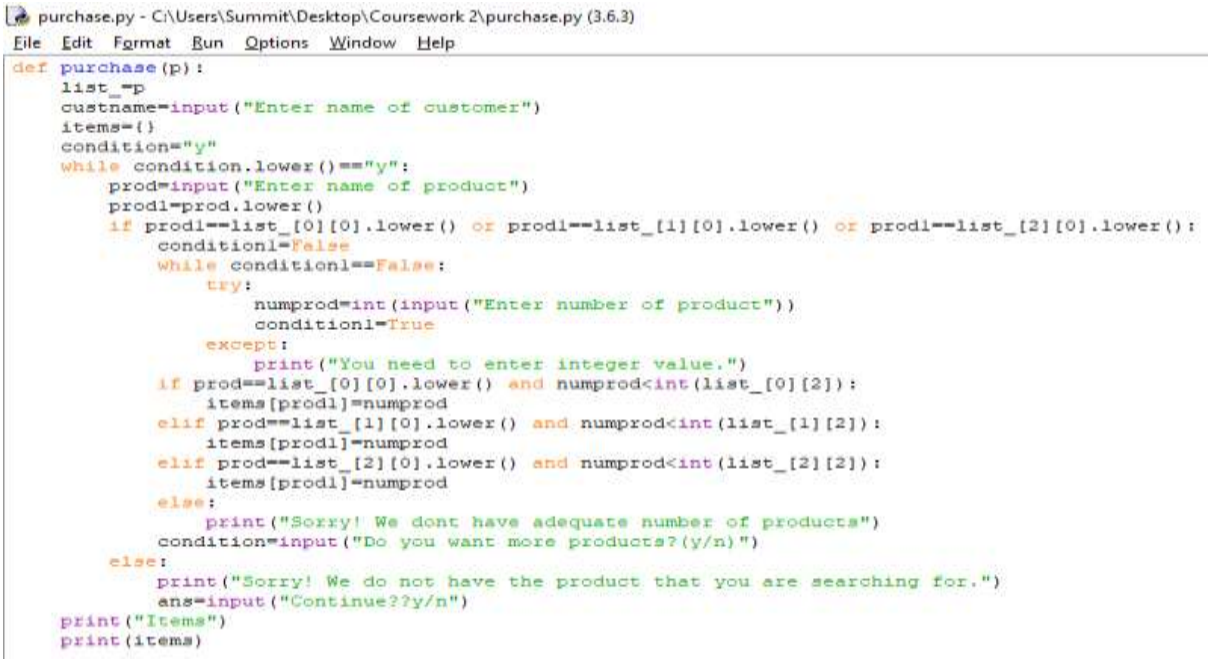

Figure 7: purchase.py(1)

Figure 8: purchase.py(2)

```
    for k in items.keys():
        if k=="phone":
            file.write(str("\t"+str(k)+" \t\t "+str(items['phone'])+" \t "+str(list_[0][1])+" \t "+str(phonecost)))
            file.write("\n")
        elif k=="laptop":
            file.write(str("\t"+str(k)+" \t\t "+str(items['laptop'])+" \t "+str(list_[1][1])+" \t "+str(laptopcost)))
            file.write("\n")
        else:
            file.write(str("\t"+str(k)+" \t\t "+str(items['hdd'])+" \t "+str(list_[2][1])+" \t "+str(hddcost)))
            file.write("\n")

    file.write("\n")
    file.write("\t\t\t\t Total:  "+str(total))
    file.write("\n")
    file.write("\n")
    file.write("\t\t\t\t Discount% :"+str(dis))
    file.write("\n")
    file.write("\n")
    file.write("\t\t\t\t Grand Total:"+str(total))
    file.write("\n")
    file.write("\n")
    file.write("**************************THANK YOU!!!**************************")
    file.close()
    return items
```

Figure 9: purchase.py(3)

6. Testing

6.1 Test 1

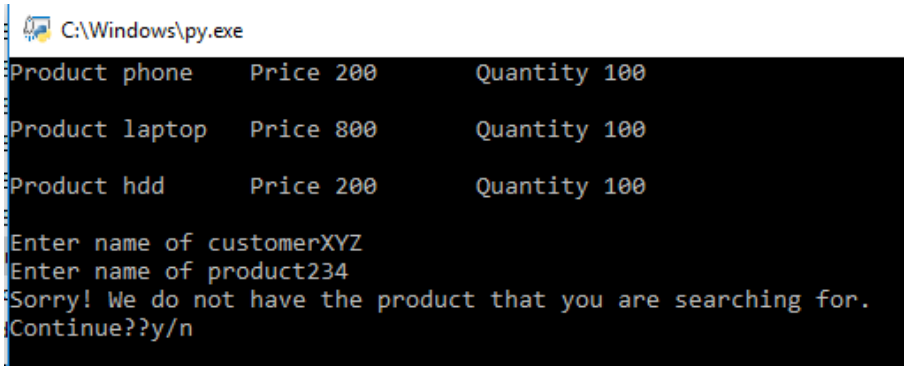

Figure 10 Test 1

Action	Integer value is input instead of product name(string)
Expected result	Error message should be displayed
Actual result	Message that says product not available was displayed
Test result	Test successful

Table 1: Test 1

6.2 Test 2

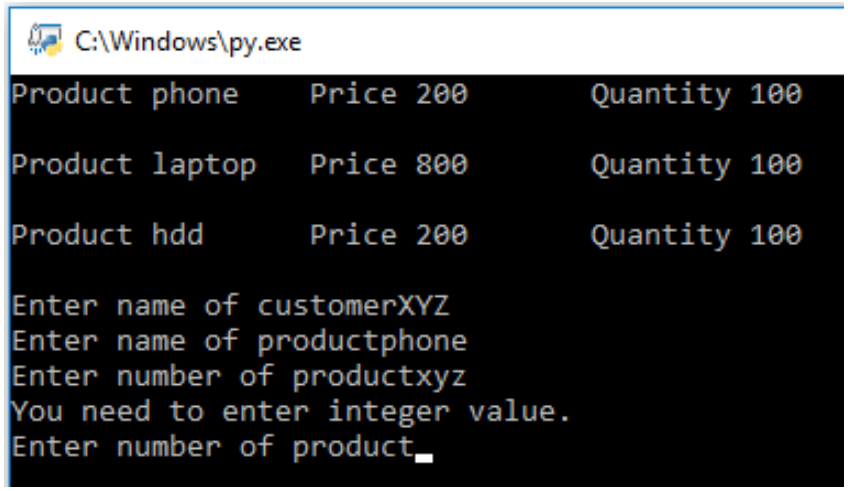

Figure 11: Test 2

Action	String value was entered in place of number of product
Expected result	Message saying enter integer value will be displayed
Actual result	Message saying enter integer value displayed
Test result	Test successful

Table 2: Test 2

6.3 Test 3

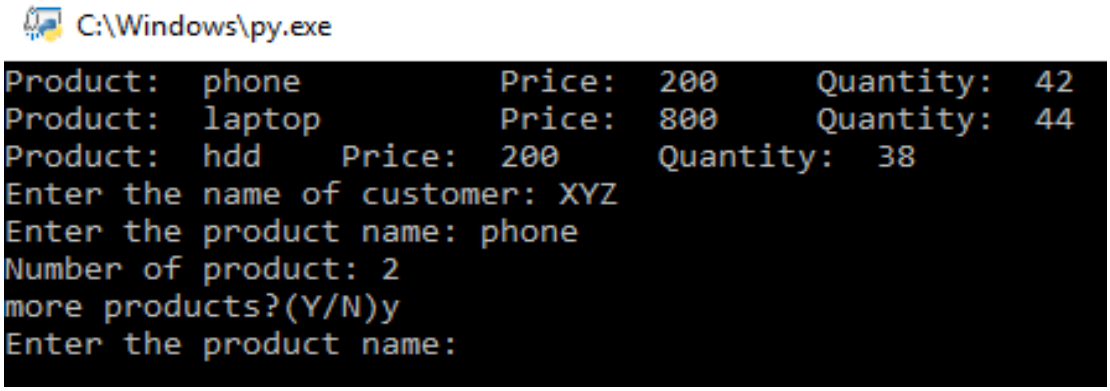

Figure 12: Test 3

Action	‘y’ is entered to buy more appliances
Expected result	Program will ask name of product
Actual result	Program asked name of product
Test result	Test successful

Table 3: Test 3

6.4 Test 4

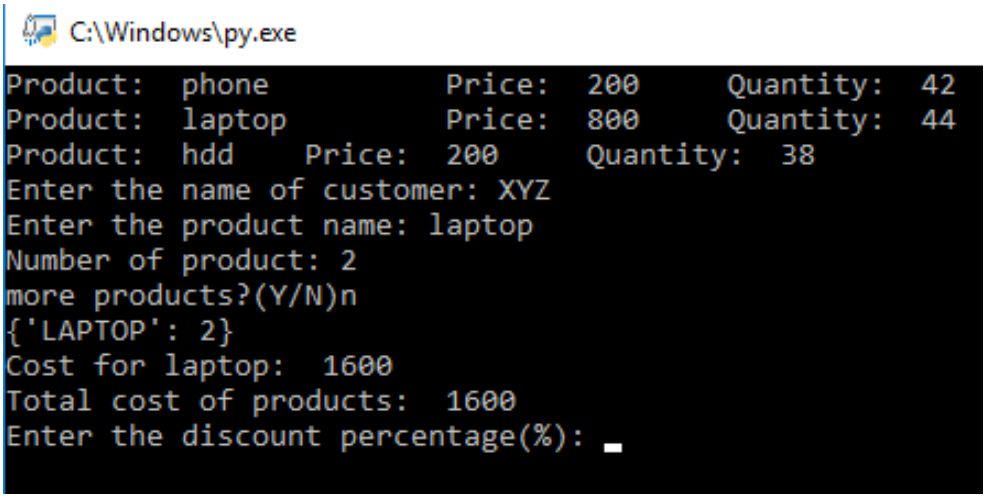

Figure 13: Test 4

Action	when asked more products? ‘n’ was input
Expected result	Subtotal will be displayed
Actual result	Subtotal was displayed
Test result	Test successful

Table 4: Test 4

6.5 Test 5

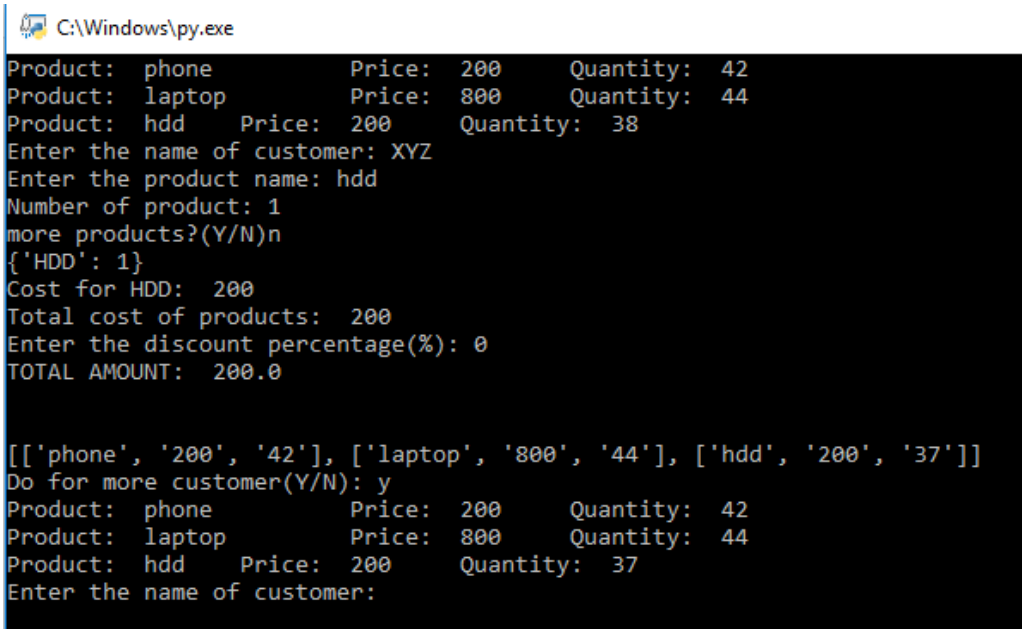

Figure 14: Test 5

Action	'y' was input in do for more customers?
Expected result	New name of customer will be asked
Actual result	New name of customer was asked
Test result	Test successful

Table 5: Test 5

7. Resources

- http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004867
- http://ieeexplore.ieee.org/document/4160250/
- https://docs.python.org/2/reference/index.html
- https://www.python.org/doc/essays/blurb/
- https://www.omicsonline.org/scholarly/python-for-bioinformatics--journals-articles-ppts-list.php
- https://docs.python.org/3/tutorial/datastructures.html
- Python Essential Reference
- Data Structures and Algorithms

8. Conclusion

The coursework was provided to us on the 7th week. The coursework was lengthier than the previous coursework. After I went through the questions, I had an outline about what are to be done and what tools will be required. I also found that I will have to conduct some researches as well. I started collecting ideas for the coursework. I went through lecture slides that were related to file handling. I gathered some idea through various valid sources on internet. I had also consulted our tutor regarding different ideas I had come up with. Preparation was not an easy task.

As per the requirement of the project, firstly algorithm was written. Then after, the pseudo codes were prepared and flowcharts were drawn. Then finally code development in python started. The code was tested time and again so that there are no bugs and errors in the program. And finally the program without bugs or errors were developed and after all the assigned tasks were completed, submission was done.

I had not thought that program would get complex to that extent. While in preparation stage, I had a general idea about how the program would flow. But once the development of code was started, numerous errors started to occur. Some of them were logical errors whereas some of them were syntax errors. The logics along with syntax were getting complex. But still I managed to complete the coursework in the given time period in the best way possible. Lengthy coding was a bit tough. I had not expected our coursework to be a complete billing system.

The coursework taught us a lot more things than just only developing python codes. While developing the code and documentation part, the coursework taught us about time management, error handling, planning and other very informative things. We did not just build a system but also learnt a lot about exception handling processes and debugging errors in a lengthy code. Attempting these kinds of project will not only fulfill our assessment but will also help a lot in our upcoming professional career. Practicing development of simple systems from now onwards will definitely have a lot of positive impact in our future.

References

Gilberg, R.F. & Forouzan, B.A. (2005) *Data Structures: A Pseudocode Approach with C*. 2nd ed. Boston: Course Technology.