

Islington College



Information System CC4002NA Coursework 2

Submitted By:

Kushal Bhattarai

NP01CP4A170221

Group: L1C6

Date: 2018/1/12

Submitted To:

Mr. Sukrit Shakya

Module Leader

Information System

WORD COUNT: 4236

SEMESTER: Autumn

PROPOSAL

Purpose/Introduction

The project is to make a proper billing system. The application needs to be developed which will read the text file and display the products available. Then with each purchase, an invoice should be generated for the particular customer and should be written into a file. The inventory should also be updated with each purchase. The project should have better updating stock calculation system. For the electronic store, the bought products, amount, discount amount total price are to be displayed in the invoice with each customer having different invoice name.

PROBLEMS

The store owner may have faced problem while generating bill of the sold item. The inventory may not have been updated. Only one product's bill may have been printed while the customer had bought 2 or more products. The calculation done in the invoice may not be accurate and the store may have gone in loss. The other problem may be if wrong data is entered in the bill whole bill should be re generated.

OBJECTIVE:

- i. The main objective of the project is to generate an invoice bill where the customer can have data of the all the products he/she bought with proper calculation of each product.
- ii. The objective is to update the inventory when the products are bought by customer.

AIM:

- i. The aim is to give idea about the billing system used.
- ii. The aim of the project is to help us understand and implement the study materials read in lecture like, modules, package, and file manipulation.

Proposed approach:

The plan is to use different modules and function to simplify the billing system with inventory managing. The plan is to use loop in the program displaying the available products and waiting for the administrator to enter purchase details where any products bill can be generated in invoice. The calculation and the problem can be solved by using exception handling feature of python. In this way to solve the problem different plans were implemented.

Target audience

The project is mainly targeted to small stores who are in need of proper billing and inventory management. The people who are interested in python language are also targeted as different features of python are displayed in the project.

Hardware and software requirements

Software:

IDLE (Python 32bit) (python programming)

Microsoft Visio (flow chart and Gantt chart)

Microsoft Word (report)

Hardware:

Laptop

4 GB ram

I5 processor

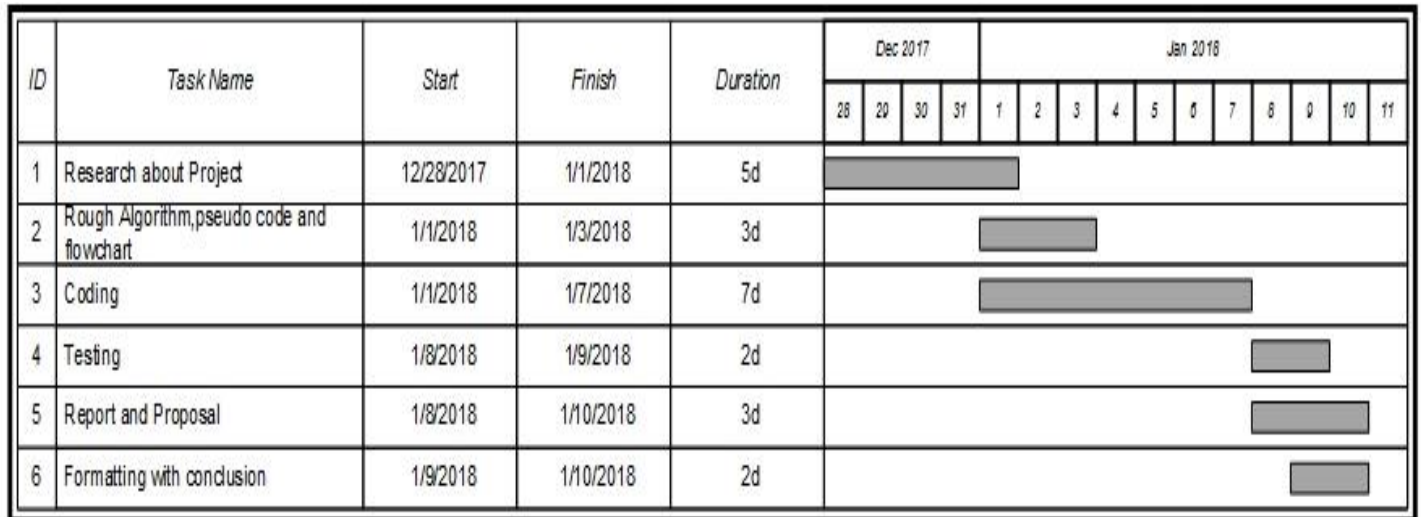
Gantt chart*Figure 1: Gantt chart*

Table of Contents

1. Introduction.....	1
2. Discussion and analysis	2
3. Algorithm	3
3.1 Algorithm.....	3
3.2 Flowchart.....	4
3.3 Pseudo code	5
4. Data structures	11
5. Program.....	13
6. Testing.....	22
Test no.1.....	22
Test no.2.....	23
Test no.3.....	24
Test no.4.....	25
Test no.5.....	26
7. Research	27
7.1 Websites	27
7.2 Books	28
7.3 Journals	31
8. Conclusion.....	34
References.....	35

Table of figures

Section 1

Figure 1: Gantt chart	iv
-----------------------------	----

Section 2

Figure 2: flowchart.....	4
Figure 3: data.txt	20
Figure 4: billing another bill	20
Figure 5: filling bill.....	20
Figure 6: displaying products	20
Figure 7: invoice 1	21
Figure 8: invoice 2	21
Figure 9: updated inventory.....	21
Figure 10: function to store data in 2d list.....	22
Figure 11: data was stored in 2 d list.....	22
Figure 12:code to check product in inventory	23
Figure 13: error handling tested result.....	23
Figure 14: tested code.....	24
Figure 15: tested code output.....	24
Figure 16: code for testing.....	25
Figure 17: invalid data input with output of each	25
Figure 18: function to display the available products.....	26
Figure 19: displayed available products	26

Table of tables

Table 1: Test1	22
Table 2: Test 2	23
Table 3: Test 3	24
Table 4:Test4	25
Table 5: Test 5	26

1. Introduction

First of all the project is to display the inventory. After that an invoice bill is to be generated for each customer. Since the customer can buy more than 1 item the loop should be used to help the customer buy more than 1 item. Since there are many customers each customer's bill should have unique file name and bill name. After the products are bought the inventory must be updated. The loop should be used to ask for the new customer details also loop should be used.

The features of programs are as follows: the inventory is updated after each purchase by customer, the program is in loop where the customer can buy as many items until the quantity left in inventory is finished, the program doesn't end after an error or wrong input is given but asks to input proper values. The need to developing the project is that the store did not have proper billing and updating software.

Python is the main term used in the project. It is a programming language suitable to learn for new programmers, as it has powerful tools to implement the code with user's idea. (A et al., 2013) Testing without any knowledge of internal operation and black box testing only examines the important aspects of the system. When black box testing is performed the tester should have knowledge of the system design and should not have access to the source code. (Khan, 2012). Functions were used in the program. "The def statement is used to create a function and to invoke a function, the name of the function followed by its arguments enclosed in parentheses, such as `result = remainder(37, 15)` is written. Anything can be returned from the function." (Beazley, 2009) There are 4 functions in the program returning values and connected with each other. Modules and packages are the functions in the python program. "The random module has functions that generate random numbers which returns a random float between 0.0 and 1.0 (including 0.0 but not 1.0). Each time a random is imported and called different number is generated." (Allen, 2015) The package imported was `randint` from `random`. Reading a file and writing to file is also used in the program where invoice is generated by making a random file to a file.

2. Discussion and analysis

The algorithm and flowchart were written first. The main idea of the program was written in pseudo code. And the pseudo code was converted into python program. Python tutor was used to help understand the code and how the code was implemented. The program was developed in Python IDLE (32bit). The flowchart was made in Microsoft Visio 2016 and after testing and research work the proposal was written from Microsoft word.

Python is an excellent language for scientific codes written in other languages. However, with additional basic tools, Python transforms into a high-level language suited for scientific purpose which is often fast enough to be useful to the user. (Oliphant, 2007) Python programming was used to implement the logic and many books were referenced as well as websites about how to write the logics in the code form.

Python tutor is a site which helps to understand what happens as the computer runs each line of source code. (<http://www.pythontutor.com/>, n.d.)

In IDLE (Python 32bit) IDLE we write code line by line. One line will handle one thing. You type whatever you want in that line and press enter to execute it. IDLE works more like a terminal or command prompt - You write one line, press enter, it executes. (<http://www.studytonight.com>, n.d.) It is an IDE which helps us to implement our python code and run a program. The IDLE was an integral part to develop the application as the codes were written using this IDE. Testing was also done with the help of IDLE. Flowchart is graphical representation of the algorithm .it was made in the project to help to know the certain process in program. Microsoft Visio is an application designed Microsoft Corporation which helps the user to implement graphical diagram. Flow chart of the code was made with the help of Microsoft Visio. Microsoft Word is an application which helped to write report of the project.

3. Algorithm

3.1 Algorithm

- Step 1. Start
- Step 2. Read the file which has products stored
- Step 3. Displaying the data of the file
- Step 4. Name of the customer is entered
- Step 5. Product, bought amount is entered.
- Step 6. Discount for the product is calculated with price.
- Step 7. Write the details name, price, total discount to a file
- Step 8. The inventory is updated
- Step 9. If new product is needed go to line 7 else go to line 11
- Step 10. Final price is calculated of all the products bought
- Step 11. Invoice is generated and stored in file
- Step 12. The updated inventory is written to a file.
- Step 13. If new file is needed go to step 6 else go to step 15
- Step 14. Stop

3.2 Flowchart

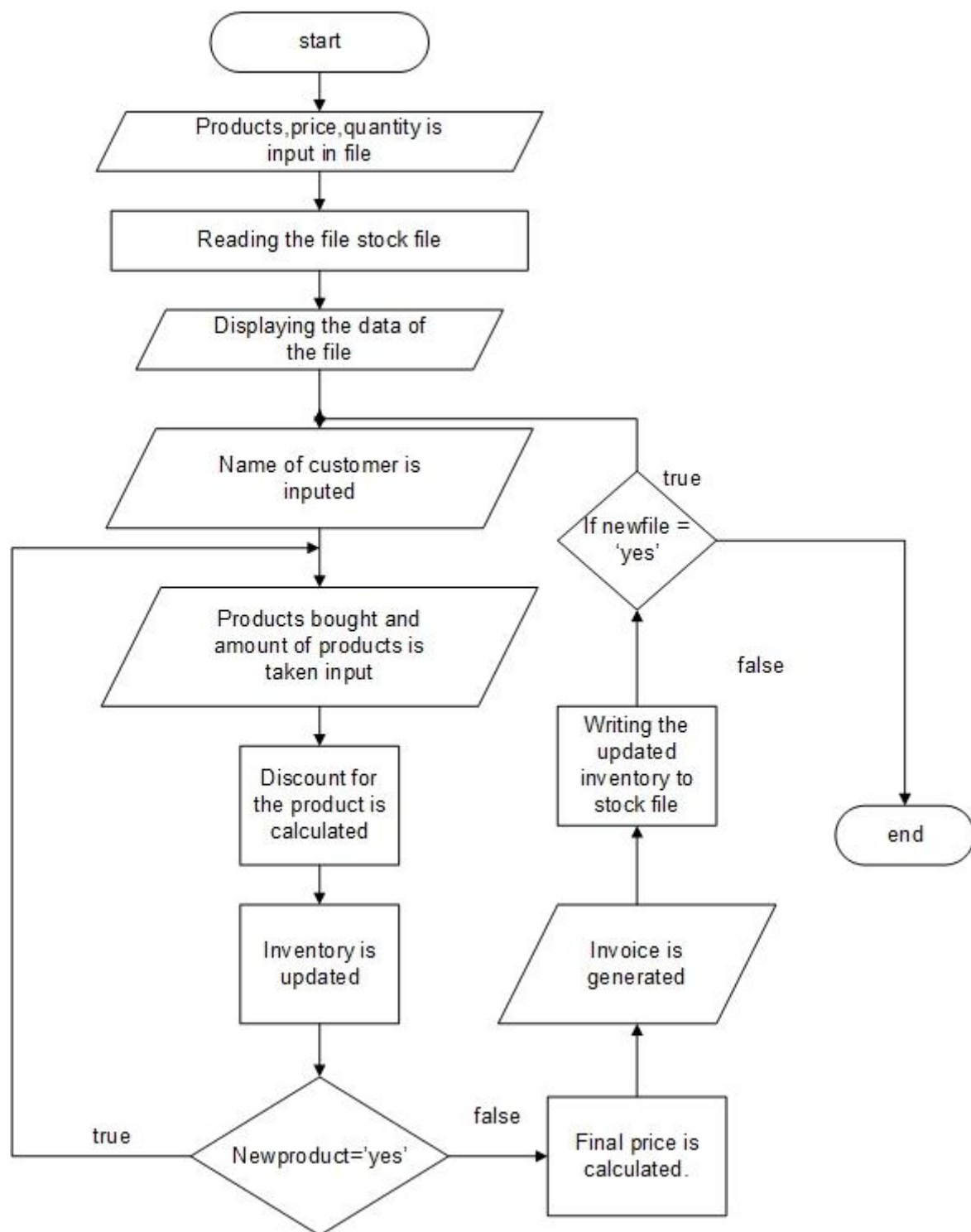


Figure 2: flowchart

3.3 Pseudo code

Module1 data output:

Algorithm storing data from file to 2dlist

```

file = open("data.txt","r")
l=[]
lines = file.readlines()
for line in lines:
    x=line.replace("\n","").split(',')
    l.append(x)
end for
file.close ()
for i in range(len(l)):
    l[i][0]=l[i][0].lower()
    l[i][1]=int(l[i][1])
    l[i][2]=int(l[i][2])
end for
return l
End 2dlist

```

Algorithm to display 2d list:

```

Display ("The available products are")
Display ('Product \t price \t quantity')
for each in inventory():
    if (Len(each[0])<7):
        Display (each [0],'\t\t', each[1],'\t',each[2])
    Else
        Display (each [0],'\t', each [1],'\t',each[2])
    End if
End for
End displaying 2d list.

```

Module 2 maininvoice:

```
import datetime
```

```
from random import randint
```

```
from dataoutput import inventory FUNCTION
```

Algorithm to enter data in invoice and update inventory

```
inven=inventory()
```

```
display("FILLING BILL")
```

```
name = str(randint(0, 999999999))
```

```
file=open(name+".txt",'w')
```

```
file.write("\t****UNIQUE ELECTRONIC STORE ***\n")
```

```
file.write("\tINVOICE BILL \n\t-----\n")
```

```
file.write("\tINVOICE Name : UEC")
```

```
file.write(str(randint(1, 500)))
```

```
file.write('\t\t\t\t\tDATE AND TIME:  ')
```

```
file.write(str(datetime.datetime.now()))
```

```
file.write('\n')
```

```
file.write(str("\tNAME: "))
```

```
name=input('ENTER CUSTOMER NAME: ')
```

```
file.write(str(name.upper()))
```

```
file.write('\n')
```

```
nextproduct="y"
```

```
total=0
```

```
file.write('\n')
```

```
file.write(str("\tNAME OF PRODUCT: "))
```

```
file.write(str("\tBOUGHT QUANTITY: "))
```

```
file.write(str("\tAMOUNT per PRODUCT: "))
```

```
file.write(str('\tDISCOUNT AMOUNT: '))
```

```
file.write(str("\tFINAL PRICE: "))
```

```
file.write('\n')
```

```

while nextproduct=="y":
    a=""
    price=0
    success = False
    while success == False:
        product=str(input('ENTER PRODUCT: '))
        x=0
        for x in range(len(inventory())):
            if product.lower()==inventory()[x][0]and
(inventory()[x][2]!=0) then
                a=product.lower()
                success =True
                break
            end if
        end for
        if a==" " then
            display("PRODUCT NOT AVAILABLE. ")
        end if
        success2 = False
        while success2 == False:
            try:
                bought_quantity=int(input("ENTER BOUGHT
QUANTITY: "))
                if bought_quantity>0 and
(((inventory()[x][2])-bought_quantity)>=0)then
                    success = True
                else:
                    display("INVALID VALUE FOR QUANTITY.
ENTER VAID VALUE")
                end if
            except:
                display("INVALID VALUE FOR QUANTITY.
ENTER VAID VALUE")
        end while
    end while

```

```
inven[x][2]=inven[x][2]-bought_quantity
success3 = False
while success3 == False:
    try:
        discount=float(input("ENTER DISCOUNT
PERCENTAGE"))
        if discount<100 and discount>=0 then
            succes3=True
            break
        else
            DISPLAY("enter less than 100 for
discount or greater than or equal to zero")
    except:
        display("invalid value")
end while
for i in range(len(inventory()))
    if a==inventory()[i][0]then
        price=inventory()[i][1]
end for
file.write(str("\t") )
file.write(a)
file.write(str("\t\t\t") )
file.write(str(bought_quantity))
file.write(str("\t\t\t"))
file.write(str(price))
file.write(str("\t\t\t"))
discountamount=(discount/100)*(price)
file.write (str(discountamount))
final price=(price*bought_quantity)-discount
file.write (str("\t\t\t"))
b=str(finalprice)
file.write(b)
file.write("\n")
total=total float(b)
```

```

        nextproduct=input("ENTER y FOR NEW PRODUCT")
    file.write('\n\n\n')
    file.write('\n')
    file.write('\t-----')
    file.write('\n')
    file.write("\tTOTAL PRICE = ")
    file.write(str(total))
    file.write("\n\t*****")
    file.write("\n \t THANK YOU FOR YOUR TIME HERE.\n\t
HOPE      TO SEE YOU AGAIN")
    file.write("\n\t*****")
    file.close()
    return inven

end invoice generation

```

Module 3:

```

from maininvoice import invoice
from dataoutput import inventory

```

Algorithm to write updated products in file

```

    b=invoice()#calling invoice
    file=open("data.txt",'w')#opening file to write
    for i in range(len(b)):#loop
        a=b[i][0]+", "+str(b[i][1])+", "+str(b[i][2])#index
        file.write((str(a)))
        file.write('\n')
    end for
    file.close()
end algorithm

```


Module 4: Algorithm to run the billing system.

```
from dataoutput import display function
from edit import update function
newfile="yes"
while newfile=='yes':
    DISPLAY(display())
    DISPLAY(update())
    newfile=input("for next customer. enter yes to fill
bill.")
End While
```

4. Data structures

The primitive data structures used in the program are:

- int integer/numbers

Whole numbers are represented in integer data type. Values of type int can be positive or negative whole numbers. A numeric literal that does not contain a decimal point is an int value. (Zelle, 2004) From the data file when the price and quantity are read and stored in list, they are stored in list with int data type. While taking input for bought quantity and price integer were used.

- float real numbers /decimal point numbers

Numbers that can have fractional parts are represented as floating point (or float) values. While a literal that has a decimal point is represented by a float. (Zelle, 2004) While calculation of discount amount, discount percentage and total float was used in the program.

- bool either true or false

Boolean data type is used for setting conditions used to control the flow of the program. (Fuhrer et al., 2016) The datatypes were used to initialize loop where first the variable was False at first. Using the variable to control loop boolean data type was used.

- str string values character

A string is an order of characters surrounded in single or double quote. " & "" represent the empty string. Although it contains no characters, the empty string is a string nonetheless. Where a=12 has int data type b=str (a) then b="12" is a (Lambert, 2009) .The str data type was used in customer name , while reading file.

The collection data type used in the program is list .

- list

A list is a mutable, ordered sequence of elements, where strings, integers, floats, even another list can be stored. Lists stores values as well as list inside between square brackets [] starting with first list having index 0 which is also called 2 D list. (Fuhrer et al., 2016) For example: l=[[1,2,3],[3,3,4]]. Here [1,2,3] has 0 index and [3,3,4] has index 1. As well as in [1,2,3] 1 has index 0 and in [3,3,4] has index 3.

The collection data type used most is list where 2d list is used to store data from .txt file. Empty is created and the data from .txt file is stored in 2d list which was returned. A new variable was made in 2nd module and the returned 2d list was stored in the variable. The calculation part was done in the stored variable which was returned again. The returned list was now back being converted into txt format.

5. Program

4 different modules were made to run the billing system and inventory update in the electronic store. First the file (data.txt) of inventory is manually made by the store. In first module dataoutput.py the function inventory is made where the data of the file is converted into 2Dlist. The 2D list is returned in the function. The next function display in the module is to display the products.

```
def inventory():  
    file = open("data.txt","r")  
  
    l=[]  
  
    lines = file.readlines()  
  
    for line in lines:  
        x=line.replace("\n","").split(',')  
        l.append(x)  
  
    file.close()  
  
    for i in range(len(l)):  
        l[i][0]=l[i][0].lower()  
        l[i][1]=int(l[i][1])  
        l[i][2]=int(l[i][2])  
  
    return l  
  
def display():  
    print("The available products are")  
    print ('Product \t price \t quantity')  
  
    for each in inventory():  
        if (len(each[0])<7):  
            print (each[0],'\t\t',each[1],'\t',each[2])  
        else:  
            print (each[0],'\t',each[1],'\t',each[2])  
  
    return ""
```

In the 2nd module the function is made to generate the invoice and storing it in a text file. Datetime module is imported to write date and time. Unique bill name and unique filename is generated for the invoice importing package from python. Here name of the customer is inputted. Loop is used in the program to ask products, bought quantity, and discount percentage. the input are entered and checked through exception handling as well as comparing data with Boolean operators like (and ,or) and conditional operators (>,<=,<).The price is imported from list is 1st module and calculated and written in invoice. The loop is used to let the user buy more than 1 product. Discount for each product is also calculated inside loop and written in file. Total is price is first initialized before loop and calculated inside loop and output is written in file outside from the loop.

```
import datetime

from random import randint

from dataoutput import inventory

def invoice():

    inven=inventory()

    print("FILLING BILL")

    name = str(randint(0,999999999))

    file=open(name+".txt",'w')

    file.write("\t***** UNIQUE
ELECTRONIC STORE *****\n")

    file.write("\tINVOICE BILL \n\t-----
-----\n")

    file.write("\tINVOICE Name : UEC")

    file.write(str(randint(1,500)))

    file.write('\t\t\t\t\tDATE AND TIME:  ')

    file.write(str(datetime.datetime.now()))

    file.write('\n')

    file.write(str("\tNAME: "))
```

```
name=input('ENTER CUSTOMER NAME\t')

file.write(str(name.upper()))

file.write('\n')

nextproduct="y"

total=0

file.write('\n')

file.write(str("\tNAME OF PRODUCT\t"))

file.write(str("\tBOUGHT QUANTITY\t"))

file.write(str("\tAMOUNT per PRODUCT"))

file.write(str('\tDISCOUNT AMOUNT\t'))

file.write(str("\tFINAL PRICE"))

file.write('\n')


while nextproduct=="y":

    a=""

    price=0

    success = False

    while success == False:

        product=str(input('ENTER PRODUCT: '))

        x=0

        for x in range(len(inventory())):

            if product.lower()==inventory()[x][0]and

(inventory()[x][2]!=0):

                a=product.lower()

                success =True

                break

        if a=="":

            print("PRODUCT NOT AVAILABLE. RE ENTER THE PRODUCT:

")
```

```
    success2 = False

    while success2 == False:

        try:

            bought_quantity=int(input("ENTER BOUGHT QUANTITY:
"))

            if bought_quantity>0 and (((inventory()[x][2])-
bought_quantity)>=0):

                success2 = True

            else:

                print("INVALID VALUE FOR QUANTITY. ENTER VAID
VALUE")

        except:

            print("INVALID VALUE FOR QUANTITY. ENTER VAID
VALUE")

    inven[x][2]=inven[x][2]-bought_quantity#editing inventory

    success3 = False

    while success3 == False:

        try:

            discount=float(input("ENTER DISCOUNT PERCENTAGE"))

            if discount<100 and discount>=0:

                succes3=True

                break

            else:

                print("enter less than 100 for discount or
greater than or equal to zero")

        except:

            print("invalid value")
```

```
for i in range(len(inventory())):
    if a==inventory()[i][0]:
        price=inventory()[i][1]

file.write(str("\t") )
file.write(a)

file.write(str("\t\t\t") )
file.write(str(bought_quantity))

file.write(str("\t\t\t"))

file.write(str(price))

file.write(str("\t\t\t"))
discountamount=(discount/100)*(price)
file.write(str(discountamount))

finalprice=(price*bought_quantity)-discount

file.write(str("\t\t\t"))
b=str(finalprice)
file.write(b)
file.write("\n")
total=total+float(b)
nextproduct=input("ENTER y FOR NEW PRODUCT")

file.write('\n\n\n')
```



```

file.write('\n')

file.write('\t-----')
-----')

file.write('\n')

file.write("\tTOTAL PRICE = ")

file.write(str(total))

file.write("\n\t*****")

file.write("\n\tTHANK YOU FOR YOUR TIME HERE.\n\tHOPE TO SEE YOU
AGAIN")

file.write("\n\t*****")

file.close()

return invent

```

In the 3rd module I have imported invoice function from 2nd module and inventory function from 1st module. The returned value of invoice function is a 2dlist where the products bought with amount is stored. Now a file is written to store the edited list.

```

from maininvoice import invoice

from dataoutput import inventory

def update():

    b=invoice()#calling invoice

    file=open("data.txt",'w')#opening file to write

    for i in range(len(b)):#loop

        a=b[i][0]+", "+str(b[i][1])+", "+str(b[i][2])#index

        file.write((str(a)))

        file.write('\n')

    file.close()

    return "INVENTORY IS UPDATED"

```

In the 4th module loop was used ask for a new file for customer and called the function display function and update function where the question is asked if they need new bill for new customer.

```
from dataoutput import display

from edit import update

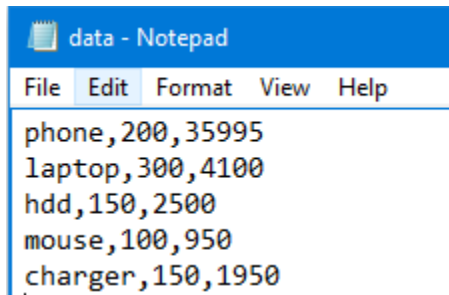
newfile="yes"

while newfile=='yes':

    print(display())

    print(update())

    newfile=input("for next customer. enter yes to fill
bill.")
```

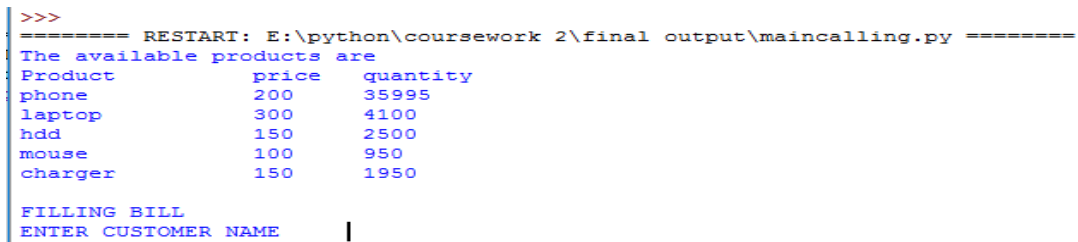


```

data - Notepad
File Edit Format View Help
phone,200,35995
laptop,300,4100
hdd,150,2500
mouse,100,950
charger,150,1950

```

Figure 3: data.txt



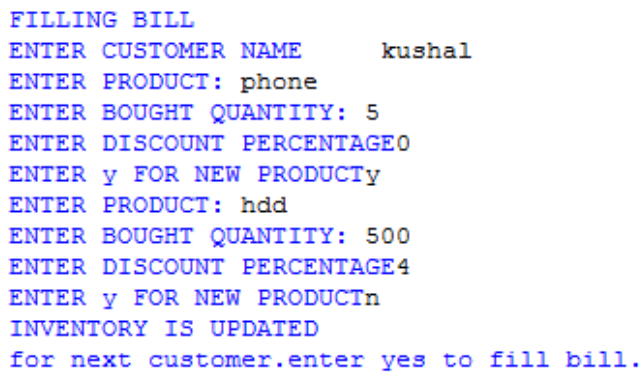
```

>>>
===== RESTART: E:\python\coursework 2\final output\maincalling.py =====
The available products are
Product      price    quantity
phone        200      35995
laptop       300      4100
hdd          150      2500
mouse        100      950
charger      150      1950

FILLING BILL
ENTER CUSTOMER NAME

```

Figure 4: displaying products

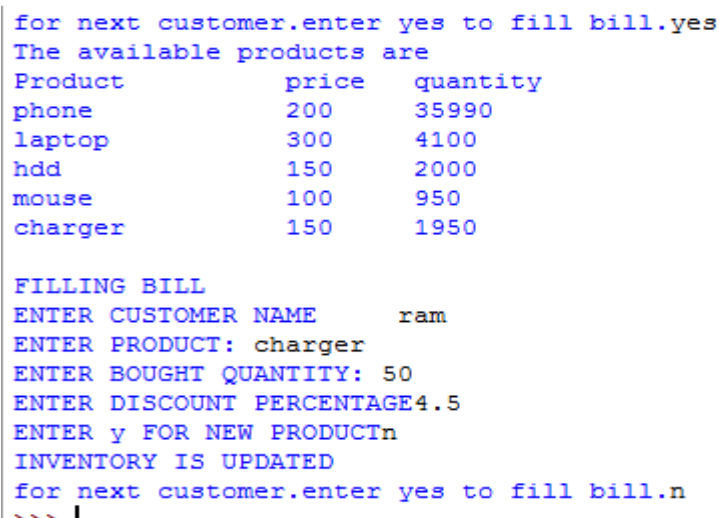


```

FILLING BILL
ENTER CUSTOMER NAME      kushal
ENTER PRODUCT: phone
ENTER BOUGHT QUANTITY: 5
ENTER DISCOUNT PERCENTAGE0
ENTER y FOR NEW PRODUCTy
ENTER PRODUCT: hdd
ENTER BOUGHT QUANTITY: 500
ENTER DISCOUNT PERCENTAGE4
ENTER y FOR NEW PRODUCTn
INVENTORY IS UPDATED
for next customer.enter yes to fill bill.

```

Figure 4: filling bill



```

for next customer.enter yes to fill bill.yes
The available products are
Product      price    quantity
phone        200      35990
laptop       300      4100
hdd          150      2000
mouse        100      950
charger      150      1950

FILLING BILL
ENTER CUSTOMER NAME      ram
ENTER PRODUCT: charger
ENTER BOUGHT QUANTITY: 50
ENTER DISCOUNT PERCENTAGE4.5
ENTER y FOR NEW PRODUCTn
INVENTORY IS UPDATED
for next customer.enter yes to fill bill.n
>>>

```

Figure 6: billing another bill

```

623236188 - Notepad
File Edit Format View Help
***** UNIQUE ELECTRONIC STORE *****
INVOICE BILL
-----
INVOICE Name : UEC117                      DATE AND TIME: 2018-01-10 02:35:43.174632
NAME: RAM

NAME OF PRODUCT      BOUGHT QUANTITY    AMOUNT per PRODUCT    DISCOUNT AMOUNT      FINAL PRICE
charger              50                  150                   6.75                   7495.5

-----
TOTAL PRICE = 7495.5
*****
THANK YOU FOR YOUR TIME HERE.
HOPE TO SEE YOU AGAIN
*****

```

Figure 7: invoice 1

```

50204208 - Notepad
File Edit Format View Help
***** UNIQUE ELECTRONIC STORE *****
INVOICE BILL
-----
INVOICE Name : UEC185                      DATE AND TIME: 2018-01-10 02:33:55.763417
NAME: KUSHAL

NAME OF PRODUCT      BOUGHT QUANTITY    AMOUNT per PRODUCT    DISCOUNT AMOUNT      FINAL PRICE
phone                5                   200                   0.0                   1000.0
hdd                  500                 150                   6.0                   74996.0

-----
TOTAL PRICE = 75996.0
*****
THANK YOU FOR YOUR TIME HERE.
HOPE TO SEE YOU AGAIN
*****

```

Figure 8: invoice 2

```

data - Notepad
File Edit Format View Help
phone,200,35990
laptop,300,4100
hdd,150,2000
mouse,100,950
charger,150,1900

```

Figure 9: updated inventory

6. Testing

Test no.	1
Action	The inventory function is called to check the data from txt file is stored in list.
Expected output	The data in the file be displayed in 2d list
Actual output	The data in the file be displayed in 2d list
Test result	Pass

Table 1: Test1

```
def inventory():
    file = open("data.txt", "r")
    l=[]
    lines = file.readlines()
    for line in lines:
        x=line.replace("\n", "").split(',')
        l.append(x)
    file.close()
    for i in range(len(l)):
        l[i][0]=l[i][0].lower()
        l[i][1]=int(l[i][1])
        l[i][2]=int(l[i][2])
    return l
print(inventory())
```

Figure 10: function to store data in 2d list

```
>>>
===== RESTART: E:\python\coursework 2\final output\dataoutput.py =====
[['phone', 200, 425], ['laptop', 300, 450], ['hdd', 150, 90], ['mouse', 100, 600], ['charger', 150, 445]]
>>> |
```

Figure11: data was stored in 2 d list

Test no.	2
Action	Wrong product is inputed.
Expected output	PRODUCT NOT AVAILABLE. RE ENTER THE PRODUCT
Actual output	PRODUCT NOT AVAILABLE. RE ENTER THE PRODUCT
Test result	Pass

Table 2: Test 2

```

nextproduct="y"
total=0
while nextproduct=="y":
    a=""
    success = False
    while success == False:
        product=str(input('ENTER PRODUCT: '))
        x=0
        for x in range(len(inventory())):
            if product.lower()==inventory()[x][0]and (inventory()[x][2]!=0):
                a=product.lower()
                success =True
                break
    if a=="":
        print("PRODUCT NOT AVAILABLE. RE ENTER THE PRODUCT: ")

```

Figure 12:code to check product in inventory

```

The available products are
Product      price  quantity
phone        200    35990
laptop       300    4100
hdd          150    2000
mouse        100    950
charger      150    1900

FILLING BILL
ENTER CUSTOMER NAME    kushal
ENTER PRODUCT: abc
PRODUCT NOT AVAILABLE. RE ENTER THE PRODUCT:
ENTER PRODUCT: 56
PRODUCT NOT AVAILABLE. RE ENTER THE PRODUCT:
ENTER PRODUCT: phone
ENTER BOUGHT QUANTITY: |

```

Figure 13: error handling tested result

Test no.	3
Action	0, QUANTITY MORE THAN AVAILABLE AND STRING VALUE IN INTEGER DATATYPE IS INPUTED.
Expected output	INVALID VALUE FOR QUANTITY. ENTER VAID VALUE
Actual output	INVALID VALUE FOR QUANTITY. ENTER VAID VALUE
Test result	Pass

Table 3: Test 3

```

success2 = False
while success2 == False:
    try:
        bought_quantity=int(input("ENTER BOUGHT QUANTITY: "))
        if bought_quantity>0 and (((inventory()[x][2])-bought_quantity)>=0):
            success2 = True
        else:
            print("INVALID VALUE FOR QUANTITY. ENTER VAID VALUE")
    except:|
        print("INVALID VALUE FOR QUANTITY. ENTER VAID VALUE")

```

Figure 14: tested code

```

ENTER BOUGHT QUANTITY: 0
INVALID VALUE FOR QUANTITY. ENTER VAID VALUE
ENTER BOUGHT QUANTITY: ac
INVALID VALUE FOR QUANTITY. ENTER VAID VALUE
ENTER BOUGHT QUANTITY: 12a
INVALID VALUE FOR QUANTITY. ENTER VAID VALUE
ENTER BOUGHT QUANTITY: 359900
INVALID VALUE FOR QUANTITY. ENTER VAID VALUE
ENTER BOUGHT QUANTITY: 60
ENTER DISCOUNT PERCENTAGE|

```

Figure 15: tested code output

Test no.	4
Action	String value are inputted, and valid values are inputted
Expected output	Invalid value is displayed for invalid value and entering option is displayed. But the program continues if valid value is given.
Actual output	Invalid value is displayed for invalid value and entering option is displayed. But the program continues if valid value is given.
Test result	Pass

Table 4:Test4

```

success3 = False
while success3 == False:
    try:
        discount=float(input("ENTER DISCOUNT PERCENTAGE"))
        if discount<100 and discount>=0:
            succes3=True
            break
        else:
            print("enter less than 100 for discount or greater than or equal to zero")
    except:
        print("invalid value")

```

Figure 16: code for testing

```

ENTER DISCOUNT PERCENTAGE100
enter less than 100 for discount or greater than or equal to zero
ENTER DISCOUNT PERCENTAGE-9
enter less than 100 for discount or greater than or equal to zero
ENTER DISCOUNT PERCENTAGE222.8.8
invalid value
ENTER DISCOUNT PERCENTAGEabc
invalid value
ENTER DISCOUNT PERCENTAGE2.5
ENTER y FOR NEW PRODUCT

```

Figure 17: invalid data input with output of each

Test no.	5
Action	To check the available products in inventory.
Expected output	The available products, price and quantity are displayed in list.
Actual output	The products, price, quantity are displayed .
Test result	Pass

Table 5: Test 5

```
def display():
    print("The available products are")
    print ('Product \t price \t quantity')
    for each in inventory():
        if (len(each[0])<7):
            print (each[0], '\t\t', each[1], '\t\t', each[2])
        else:
            print (each[0], '\t', each[1], '\t', each[2])
    return ""
print(display())
```

Figure 18: function to display the available products

```
===== RESTART: E:\python\coursework 2\final output\dataoutput.py =====
The available products are
Product      price    quantity
phone        200      35990
laptop       300      4100
hdd          150      2000
mouse        100      950
charger      150      1900
```

Figure 19: displayed available products

7. Research

7.1 Websites

- i) <https://docs.python.org>

The site is an official site of Python Software Foundation. This site helped me to know different information about importing packages from python like random, datetime. The site gave me information about IDLE. The site gives information and guides to use different methods in function.

- ii) <https://www.microsoft.com>

This site helped me to know about Microsoft Visio and Microsoft word. gave me instruction about how to use the applications. The official site of Microsoft was in my research because it had 2 important applications needed for the developing my program. Visio and Word which help me to make flowchart and formatting of the report proposal

- iii) www.thesai.org

This site was researched to search to attain articles ,journals about testing. The site contains many articles related to computer technology where the researchers of the company publish their journals and articles. The site is a domain for technology research papers.

- iv) <http://www.studytonight.com>

This site was used as a reference to do coding of python . It is an online platform that allows user to learn the desired subject with the help of lessons giving proper ideas about many concepts of the subject.

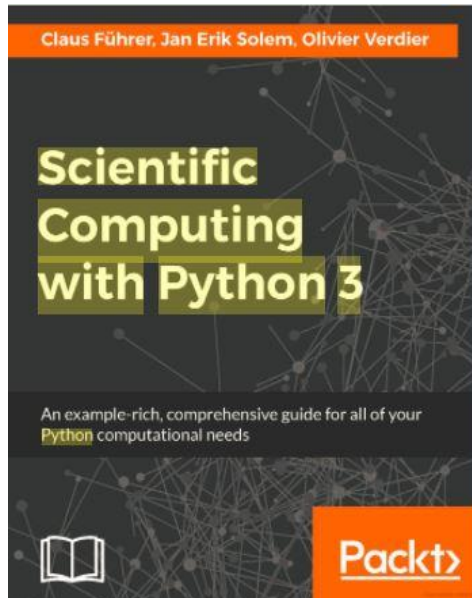
- v) <http://pythontutor.com/visualize.html#mode=edit>

This site is very useful to test our logic to implement in the. Many logics used in the program were first tested in this site. It is very useful tool giving step wise explanation of the program.

7.2 Books

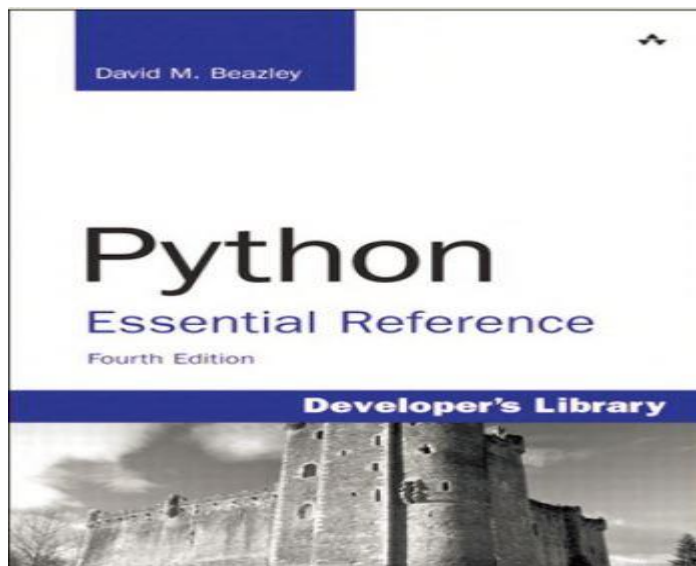
i) Scientific Computing with Python 3

The book gave information about the python. Primitive data types and collection data types were referenced from this book.



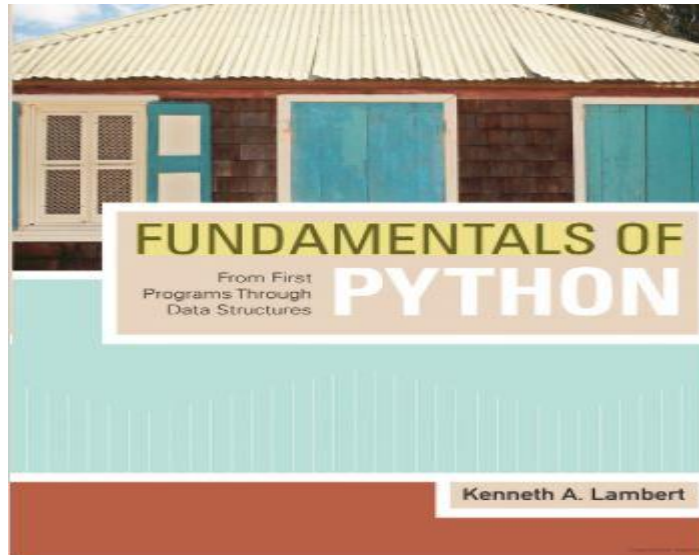
ii) Python Essential Reference

This book is a guide to Python programming. The book gave me reference about everything in python. I learned about functions, calling a function, returning values from function.



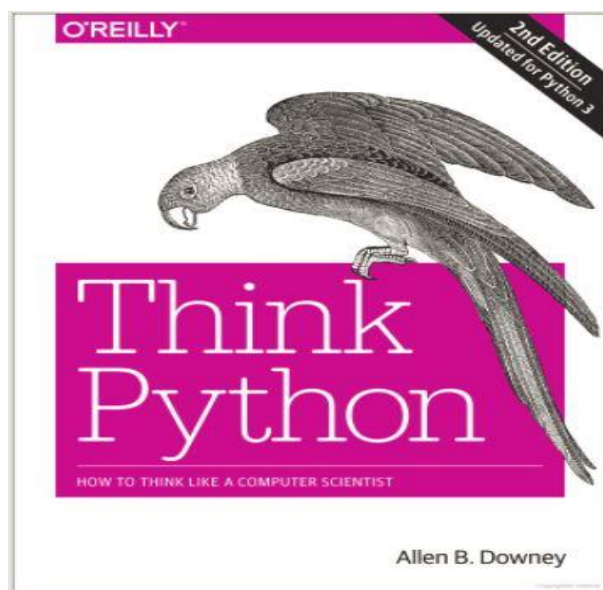
iii) Fundamentals of Python: From First Programs through Data Structures

This book was consulted to gain more information about Python datatypes, detecting syntax errors as well as testing of the program



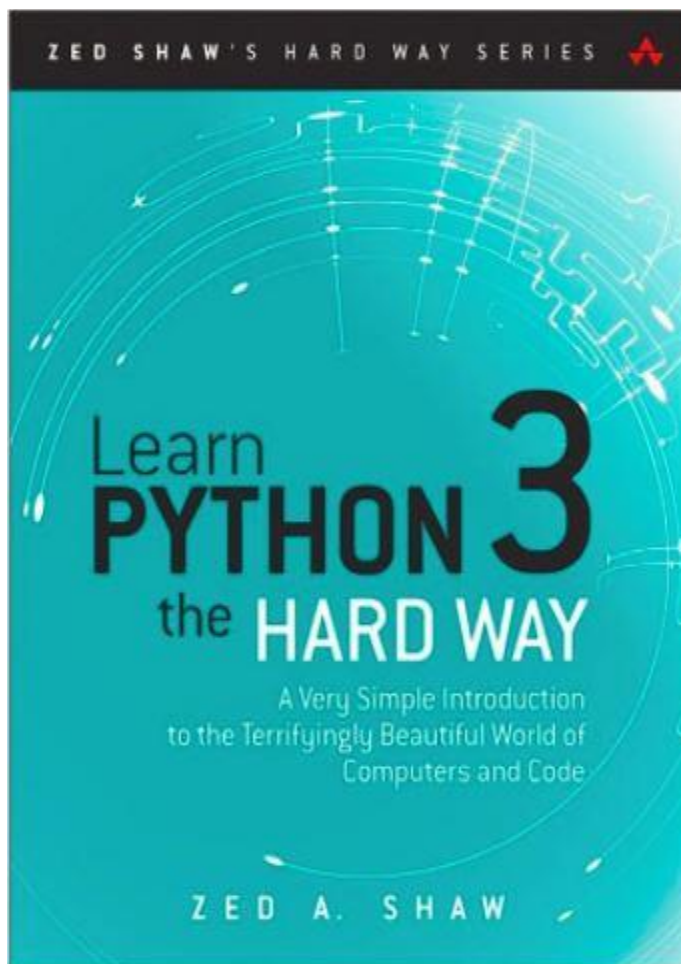
iv) Think Python: How to Think Like a Computer Scientist

This book gave information about importing modules and packages in python for example random.



v) Learn Python The Hard Way, 3rd Edition

This book is for beginners and it has simple language which helps to understand clearly about concepts of python.



7.3 Journals

- i. BLACK BOX AND WHITE BOX TESTING TECHNIQUES A LITERATURE REVIEW - International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June

From this journal the thing about black box testing and how is it done was learned.

International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012

BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW

Srinivas Nidhra¹ and Jagruthi Dondeti²

¹School of Computing, Blekinge Institute of Technology, Karlskrona, Sweden
nidhra.srinivas@gmail.com

²Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India
jagruthi.sri@gmail.com

ABSTRACT

There are several methods for automatic test case generation has been proposed in the past. But most of these techniques are structural testing techniques that require the understanding of the internal working of the program. There is less practical coverage of all testing techniques together. In this paper we conducted a literature study on all testing techniques together that are related to both Black and White box testing techniques, moreover we assume a case situation of Insurance premium calculation for driver and we derive the test cases and test data for white box testing methods such as Branch testing, Statement testing, Condition Coverage testing, multiple condition coverage testing, in the similar way we derive the test cases and test data for the black box testing methods such as: Equivalence partitioning and Boundary value analysis.

The overall aim of this literature study is to clearly explain different testing techniques along with a case situation and their advantages.

KEYWORDS

Software testing, Functional testing, Structural testing, test cases, black box testing, white box testing, Testing techniques.

- ii. A Comparative Study of White Box, Black Box and Grey Box Testing Techniques (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012

This journal also helped to give certain idea about black box testing and its implementation.

(IJACSA) International Journal of Advanced Computer Science and Applications,
Vol. 3, No.6, 2012

A Comparative Study of White Box, Black Box and Grey Box Testing Techniques

Mohd. Ehmer Khan
Department of Computer Science,
Singhania University,
Jhunjhunu, Rajasthan, India

Farmeena Khan
Department of Computer Science,
EILM University,
Jorethang, Sikkim, India

Abstract—Software testing is the process to uncover requirement, design and coding errors in the program. It is used to identify the correctness, completeness, security and quality of software products against a specification. Software testing is the process used to measure the quality of developed computer software. It exhibits all mistakes, errors and flaws in the developed software. There are many approaches to software testing, but effective testing of complex product is essentially a process of investigation, not merely a matter of creating and following route procedure. It is not possible to find out all the errors in the program. This fundamental problem in testing thus throws an open question, as to what would be the strategy we should adopt for testing. In our paper, we have described and compared the three most prevalent and commonly used software testing techniques for detecting errors, they are: white box testing, black box testing and grey box testing.

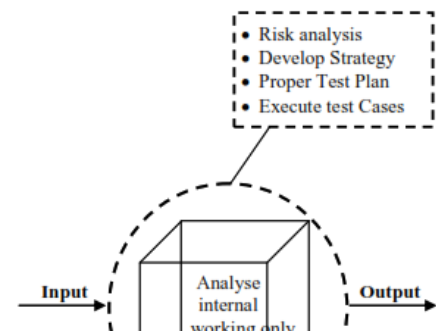
Keywords- Black Box; Grey Box; White Box.

I. INTRODUCTION

Software testing identifies defects, flaws or errors in the application code that must be fixed. We can also define software testing as a process of accessing the functionality and

3) *Grey Box Testing Technique*: White box + Black box = Grey box, it is a technique to test the application with limited knowledge of the internal working of an application and also has the knowledge of fundamental aspects of the system.

II. WHITE BOX TESTING TECHNIQUE



- iii. Gantt charts: A centenary appreciation -European Journal of Operational Research 149 (2003) 430–437.

This journal helped to know about Gantt chart and its history. As a reference to the chart in the journal, Gantt chart was made.



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

European Journal of Operational Research 149 (2003) 430–437

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCH

www.elsevier.com/locate/dsw

Gantt charts: A centenary appreciation

James M. Wilson *

Department of Business and Management, Glasgow University, University Avenue, Glasgow G12 8QQ, Scotland, UK

Abstract

With the proliferation of microcomputer based project management packages Gantt charts have enjoyed a revival in their use. Although Henry L. Gantt is recognized as their developer their origins and provenance are less well known. Gantt was a close associate of Frederick W. Taylor and an advocate of Scientific Management. His paper describing the use of “graphics” for general production planning appeared alongside Taylor’s *Shop Management* in 1903 and was an integral and critical component of Taylor’s system. Without Gantt’s charts to plan the workloads for men and machines both in departments and throughout the factory Taylor’s system would have been unworkable. The focus of this paper is to describe more fully their development and early history; and review their contemporary uses and future prospects.

- iv. Python to learn programming- Journal of Physics: Conference Series 423 (2013) 012027

This journal helped to know more about python program and how it is different from other programming language.

- v. Python for Scientific Computing- Computing in Science & Engineering (Volume: 9, Issue: 3, May-June 2007)

This journal was referenced to see what python programming is and its uses in fields.

8. Conclusion

The main thing of the project was to build an application which could manage billing system and make a unique bill of every customer with proper inventory update system. The whole project was done with exceptional handling, using control loop statements. I think I have met the expectations of the project. Python idle was used to write down code where python 3.6 language formats and syntax was used. I have developed a program which uses many datatypes but most used data type was list. The research section was the most important part in the project. I learned many things doing research which helped me write algorithm, flowchart and proper coding. The research was an integral part because of the research done and with the help from lecture notes I made the program.

I have done testing for the code in the process. The testing results were as expected and I didn't find any flaw, but The idiom "every part has two coins" is relatable in the scenario, where I have made a program it may contain some bugs or some errors, but in my understanding and my belief in the research I have done I can say that the program is flawless. The findings in the testing were as expected. According to the scenario, I can say that the requirements are met as there are 4 modules, invoice is generated, inventory is updated, loop is used to check if there are more than two products needed as well as loop is also used to enter data of more than one customer.

References

- A, B., Zhaparov, & Suliyeu, R. (2013) *J. Phys.: Conf. Ser.* 423 012027.
- Allen, D.B. (2015) *Think Python: How to Think Like a Computer Scientist*. 2nd ed. O'Reilly Media Inc.
- Beazley, D.M. (2009) *Python*. 4th ed. Addison-Wesley Professional.
- Fuhrer, C., Solem, J.E. & Verdier, O. (2016) Scientific Computing with Python 3. In *Scientific Computing with Python 3*. Birmingham: Packt Publishing Ltd. p.332.
- <http://www.pythontutor.com/>. (n.d.) *pythontutor* [Online]. Available from: <http://www.pythontutor.com/> [Accessed 11 January 2018].
- <http://www.studytonight.com>. (n.d.) *introduction-to-idle* [Online]. Available from: <http://www.studytonight.com/python/introduction-to-idle> [Accessed 10 January 2018].
- Khan, M.E. (2012) A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 3(6).
- Lambert, K.A. (2009) *Fundamentals of Python: From First Programs through Data Structures*. illustrated ed. Cengage Learning.
- Oliphant, T.E. (2007) Python for Scientific Computing. *Computing in Science & Engineering* , 9(3).
- Zelle, J.M. (2004) *Python Programming:An Introduction to Computer Science*. illustrated ed. Franklin, Beedle & Associates, Inc.