



islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2021-22 Summer

Student Name: Shishir Ghimire

Group: L1C15

London Met ID: 22015777

College ID: NP01CP4S220018

Assignment Due Date: 2022-08-26

Assignment Submission Date: 2022-08-26

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

| | |
|--|----|
| 1. INTRODUCTION:..... | 1 |
| 2. Discussion and Analysis | 2 |
| 2.1. Algorithm:..... | 2 |
| 2.2. Flow-Chart:..... | 4 |
| 2.3. Pseudocode | 5 |
| 2.3.1. Pseudocode for main.py..... | 5 |
| 2.3.2. Pseudocode for View.py | 6 |
| 2.3.3. Pseudocode for Rent.py | 7 |
| 2.3.4. Pseudocode for Return.py | 15 |
| 3. DATA STRUCTURE: | 20 |
| 3.1. Screenshots of data structure implementation: | 21 |
| 4. PROGRAM: | 26 |
| 4.1. Implementation of the program: | 26 |
| 4.2. Renting and returning the costume | 28 |
| 4.2.1. Renting | 28 |
| 4.2.2. Renting | 33 |
| 4.3. Creation of text file: | 36 |
| 4.4. Opening the text file and showing bill..... | 38 |
| 4.5. Terminating the program:..... | 40 |
| 5. TESTING | 41 |
| 5.1. Test One: | 41 |
| 5.2. Test Two: | 42 |
| 5.3. Test Three:..... | 43 |
| 5.4. Test four:..... | 45 |
| 5.5. Test Five: | 47 |
| 6. Conclusion: | 54 |
| Bibliography | 55 |
| APPENDIX: | 56 |

List of Figures

| | |
|---|----|
| Figure 1: Flow-chart | 4 |
| Figure 2: Screenshot of Data structure implementation | 21 |
| Figure 3: Screenshot of Data structure implementation | 22 |
| Figure 4:Screenshot of Data structure implementation | 22 |
| Figure 5: Screenshot of Data structure implementation | 23 |
| Figure 6: Screenshot of Data structure implementation | 23 |
| Figure 7: Screenshot of Data structure implementation | 24 |
| Figure 8: Screenshot of Data structure implementation | 24 |
| Figure 9: Screenshot of Data structure implementation | 25 |
| Figure 10: Screenshot of Data structure implementation | 25 |
| Figure 11: Implementation of program main menu..... | 26 |
| Figure 12: Implementation of program when view option is selected | 27 |
| Figure 13: Implementation of program during wrong input..... | 28 |
| Figure 14: Entering the customer name | 29 |
| Figure 15: Entering the costume Id | 30 |
| Figure 16: Entering the quantity of costume..... | 30 |
| Figure 17: Asking for conformation to continue or not | 31 |
| Figure 18: When user continues renting..... | 31 |
| Figure 19: When invalid Id is entered | 32 |
| Figure 20: When an invalid quantity is entered | 32 |
| Figure 21: Bill is generated..... | 33 |
| Figure 22: Entering the customer Id | 34 |
| Figure 23: Asking which costume customer wants to return | 34 |
| Figure 24: Asking the customer to check out | 35 |
| Figure 25: Bill print | 35 |
| Figure 26: Creation of the text file | 36 |
| Figure 27: Creation of the text file | 37 |
| Figure 28: Creation of the text file | 37 |
| Figure 29: Folder with the auto generated text files and python files | 38 |
| Figure 30: Auto generated text file with the data | 38 |

| | |
|---|----|
| Figure 31: Auto-generated text file which stores the bill | 39 |
| Figure 32: Bill printed in the terminal after renting..... | 39 |
| Figure 33: Program is terminated..... | 40 |
| Figure 34 Entered negative value of ID | 42 |
| Figure 35: Entered non-existing value as ID | 43 |
| Figure 36: Data are entered | 44 |
| Figure 37: Bill is generated..... | 44 |
| Figure 38: Text file is generated..... | 45 |
| Figure 39: Return Process | 46 |
| Figure 40: Return Process | 46 |
| Figure 41: Return Bill..... | 47 |
| Figure 42: Text file before renting | 48 |
| Figure 43: Renting process | 48 |
| Figure 44: Renting process | 49 |
| Figure 45: Renting process | 49 |
| Figure 46: Renting process | 50 |
| Figure 47: Text file after renting | 50 |
| Figure 48: Return Process | 51 |
| Figure 49: Return Process | 51 |
| Figure 50: Return Process | 52 |
| Figure 51: Return Process | 52 |
| Figure 52: Text File after returning | 53 |

List of Tables

| | |
|---------------------------------|----|
| Table 1: Test One..... | 41 |
| Table 2: Exception handled..... | 41 |
| Table 3: Test two..... | 42 |
| Table 4: Test three | 43 |
| Table 5: Test Four..... | 45 |
| Table 6: Test Five..... | 47 |

1. INTRODUCTION:

This is the first coursework in Fundamental of Computing. This coursework was assigned to us with the goal of developing a system for costume rental. We were required to develop this system with the help of python programming language and the Integrated Development and Learning Environment (IDLE). We were required to develop this program with the help of an algorithm and flowchart. In this system, we were required to develop a properly functioning system to view, rent, and return the costume. This system is very useful to the owners of Rental shops where people borrow and return items. This program can help those owners by keeping records of borrowing, returns, and transactions. Instead of keeping records on paper, this system can be very helpful to rental shop owners.

The goals and objectives of this project are very clear. This is the system for the Rental shop management system. People can view the available costumes, select their favorite costumes as per their desire, and can rent them for a certain time and return the costume in proper time as people do in real life in the rental shops. This system also helps in keeping the record of every transaction. This system eradicates the problem of keeping every record on physical paper and keeping it safe as its loss can disturb the entire business. User can select their desirable costume from the list and enter their name in order to rent the costume. Then a bill is generated for that costume with the customer name, customer Id, Name of the costume, Brand of the costume, Price of the costume, the total quantity of the costume, and the total price of the costume. Meanwhile returning the costume, the system automatically finds the bill which was generated while renting the costume with the user name.

While doing this coursework, I implemented every knowledge I got from the lecture, tutorial, and workshop classes of Islington college. I developed this program with the help of many concepts of loops, files, lists, and many more. This system is easy to use and user-friendly. I am very pleased to get this opportunity to do this work and got a chance to implement the things I learned in classes.

2. Discussion and Analysis

2.1. Algorithm:

Step 1: START

Step 2: Input a number from a range of 1 to 4.

Step 3: If an input is 1,

 Display the list of costumes available for rent.

 Return to Step 2

Step 4: If an input is 2,

 Go to Step 5

 If No, then Go to Step

Step 5: Enter the Name of the Customer

Step 6: Enter the Costume Id to rent

Step 7: Check if a Costume is available,

 If No Print Costume is not available

 If Yes Go to step 8

Step 8: Append the data of the costume to the dictionary

Step 9: Update the Stock/ quantity of costumes available

Step 10: Check if the customer wants to continue,

 If Yes, Return to step 7

 If No, Go to Step 11

Step 11: Print the bill of costume rented

 Return to step 2

Step 12: If an input is 3

Go to Step 13

Step 13: Enter the Customer Id

Step 14: Check whether the customer Id matches or not

If No, Print Invalid costume Id

If Yes, Go to Step 15

Step 15: Update the quantity of the costume

Step 16: Check whether the costumer wants to continue

If Yes, Return to Step 13

If No, Go to Step 17

Step 17: Print the bill of returned item

Return to Step 2

Step 18: If an input is 4

Go to Step 19

Step 19: Print Thank you, Visit Again.

Step 20: STOP

2.2. Flow-Chart:

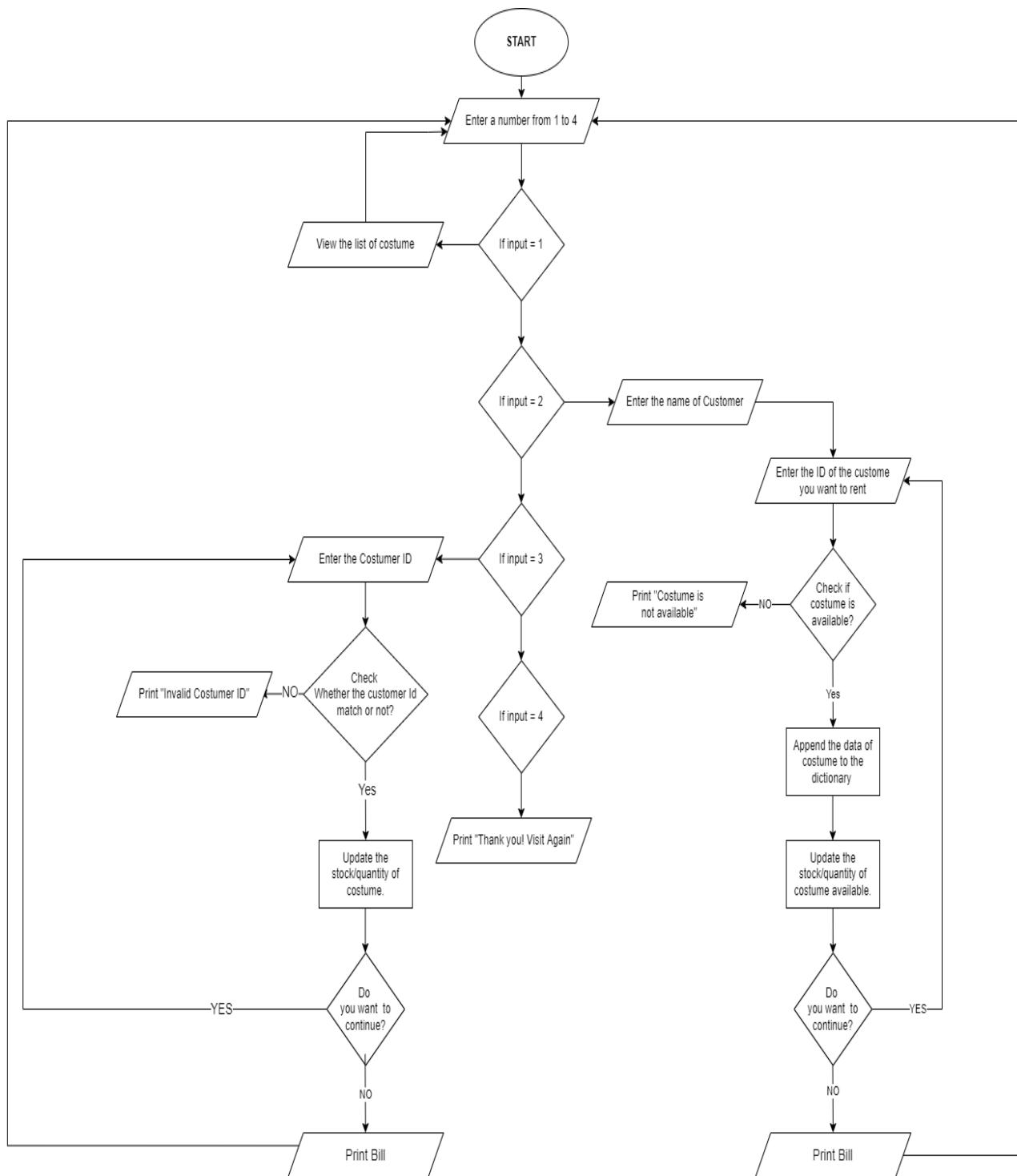


Figure 1: Flow-chart

2.3. Pseudocode

2.3.1. Pseudocode for main.py

IMPORT sys

CREATE a function named main

WHILE True

PRINT the text * 60 times

PRINT the text SHISHIR CLOTHING

PRINT the text * 60 times

PRINT the text Select an option

PRINT the text * 60 times

PRINT the text Press [1] to VIEW \n Press [2] to RENT \n Press [3] to
RETURN \n Press [4] to EXIST

INPUT the number to press any option in int data type as variable a

PRINT the text * 60 times

IF the value of a is 1

IMPORT View

ELIF the value of a is 2

IMPORT Rent

ELIF the value of a is 3

IMPORT Return

ELIF the value of a is 4

EXIST the code with sys.exist function. **PRINT** the text Thank you
for trusting us

ELSE

PRINT the text Invalid Input

END IF

END WHILE

CALL the function main

2.3.2. Pseudocode for View.py

CREATE the dictionary named dic_costume

CREATE the function display

ASSIGN the value of the counter as c = 0

PRINT the text * 50 times

PRINT the text ID, Costume, Brand, Price, Quantity

PRINT the text * 50 times

DECLARE the variable file. **OPEN** the text file Shishir.txt in read mode

DECLARE the variable costume_book. **READ** the text file.

SPLIT the data of text file by \n and **ASSIGN** its value in costume_book

WHILE empty text in costume_book

REMOVE the empty text of costume_book

END WHILE

FOR i in range of length costume_book

ASSIGN the value of c as c plus 1

APPEND the value of c in dic_costume dictionary and **ASSIGN** its value with split of text by comma in costume_book

END FOR

FOR key, value of dic_costume in items function

PRINT the text \t in variable key and end

FOR j in value

PRINT the text \t in variable j and end

END FOR

PRINT the text \n

END FOR

CALL the function display

2.3.3. Pseudocode for Rent.py

IMPORT datetime

IMPORT random

IMPORT sys

CREATE function display

ASSIGN the value of c to 0

PRINT * 50 times in row

PRINT ID, Costume, Brand, Price, and Quantity

PRINT * 50 times in row

OPEN Shishir.txt File in Read mode

ASSIGN the variable name for File as Costume_book and Split it bt \n

WHILE there is empty text in costume_book

REMOVE empty text

END WHILE

FOR i of Length of costume_book

 C += 1

ASSIGN the value of C in dic_costume Dictionary Which splits
 costume_book variable with i index

END FOR

FOR key value of dic_cosstume dictionary **ASSIGN** items function

PRINT the value of key and end with \t

FOR j in value

PRINT the value of j and end with \t

PRINT \n

END FOR

END FOR

CALL the function Valid with the value c

CALL the function update_stock

INPUT the Customer name by the variable Customer_Name

CREATE the function Valid with value c

ASSIGN the value of c as c

INITIALIZE the variable RentID. **INPUT** the ID by variable RentID

IF RentID greater than 0 and RentID less than equal to cThen

PRINT Costume is available

INITIALIZE the variable qty. **INPUT** the quantity the user want by the variable

qty

INITIALIZE the variable costume_select. **ASSIGN** costume_select as dic_costume with the index RentID

IF qty<= index 3 of variable costume_select

PRINT the text Quantity is Available. Please Continue.

INITIALIZE the variable update_qty. **ASSIGN** the value of update_qty as the difference of Index 3 of costume_select with qty

ASSIGN Index 3 of costume_select with the variable update_qty

CALL update_stock function

INITIALIZE the variable abcd. **ASSIGN** the value of abcd with index 0,1,tiday,qty of costume_select as String

APPEND the billing detail as abcd

REPLACE the value ‘ with \t by abcd variable

REPLACE the value , with empty text by abcd variable

INITIALIZE the variable custome_renting. **APPEND** the value abcd in custome_rent variable

INITIALIZE the variable xyz. **ASSIGN** the value of xyz with variable ID, Customer_Name, Costume_select with index 0, qty, and RentID by String

REPLACE the value ‘ by empty text in xyz

APPEND the variable xyz in list00

CALL function write()

INITIALIZE the variable total. **ASSIGN** the total with multiple of qty
with costume_select of index 2

APPEND the variable total in Total

DECLARE variable mnop and type casting int and **APPEND** in the
file.

REPLACE the value ‘ with empty text in mnop

REPLACE the value , with empty text in mnop

APPEND the value of mnop in variable renting

CALL the function write_two

DECLARE the variable Nam. **APPEND** the variable
Customer_Name and ID in String.

REPLACE the text ‘ with empty text in Nam variable

REPLACE the text , with \t in the Nam variable

APPEND the Nam variable in list01 list

CALL the function write

DECLARE the variable billing and **INPUT** the text Do you want to
continue? Type yes to continue or No to exist

IF billing == no

PRINT the text \n

PRINT the text \t and # 50 times

PRINT the text Shishir Clothing

PRINT the text – 60 times

PRINT the text Billing

PRINT the text – 60 times

PRINT the text Customer Name with the variable
Customer_Name

PRINT the text Customer ID with the variable ID

PRINT the text Costume, Brand, Booked Date, Quantity

PRINT the text # 90 times

FOR item in variable costume_renting

PRINT the text \t with variable item, end

PRINT the text \n

END FOR

PRINT the text Total with Variable sum **APPENDING** Total

PRINT # 90 times

CALL the function exit with sys

ELIF billing == yes

PRINT the text Thank you. Feel free to rent more

END IF

CALL the function upadate_stock

CALL the function display

ELSE

PRINT the text – 50 times

PRINT the text Sorry, Costume is out of stock.

PRINT the text – 50 times

CALL the method display

END IF

ELSE

PRINT the text – 50 times

PRINT the text INVALID! Try again.

PRINT the text – 50 times

END IF

END IF

CREATE a function Write

DECLARE the variable ghi. **OPEN** the text file ccrtxt.txt to append

FOR item in list00

APPEND item in text file as String

END FOR

CREATE the function write_two

CREATE a text file and **OPEN** it with the name user and **APPEND** user ID and in
write mode

FOR aaaa variable in renting variable

APPEND the text \n in text file

APPEND aaaa in text file in String

APPEND the text \n in text file

END FOR

CLOSE the text file

CREATE the function write

DECLARE the variable cc

CREATE a text file name COS_Bill. **APPEND** file name with ID in write mode.

WRITE the text – 60 times

WRITE the text \n

WRITE the text Shishir Clothing

WRITE the text \n

WRITE the text – 60 times

WRITE the text Bill Details

WRITE the text \n

WRITE the text Name:

APPEND the variable Customer_Name in text file

WRITE the text \n

WRITE the text Costomer_ID

APPEND the variable ID in string in text file

WRITE the text \n

WRITE the text – 60 times

WRITE the text \n

WRITE the text Costume

WRITE the text Brand

WRITE the text Booked Date

WRITE the text Quantity

WRITE the text \n

WRITE the text – 60 times

WRITE the text \n

FOR dd in list00

APPEND the variable dd in the text file

WRITE the text \n

END FOR

WRITE the text \n

WRITE the text – 60 times

WRITE the text – 60 times

WRITE the text total:

APPEND the list Total in sum variable

CLOSE the text file

CREATE the function update_stock

OPEN the text file Shishir.txt in write mode by oo variable

FOR key, value in dic_costume dictionary in items function

APPEND the , and join the value in text file

APPEND the text \n in the text file

CLOSE the text file

CALL the method display

2.3.4. Pseudocode for Return.py

```
FROM datetime IMPORT date  
  
DECLARE Variable for today date function as return_date  
  
CREATE the dictionary dic_return  
  
CREATE the dictionary dic_costume  
  
CREATE the list billing  
  
PRINT the text – 50 times  
  
DECLARE the variable costumer_id to INPUT costumer Id  
  
PRINT the text – 50 times  
  
CREATE a function return_dic  
  
    ASSIGN the value of counter c as 0  
  
    DECLARE a variable file. OPEN the text file Shishir.txt in read mode  
  
    DECLARE a variable return_data to READ the text file.  
  
    SPILT the text file data by \n  
  
    WHILE empty text in return_data  
  
        REMOVE the empty text  
  
    END WHILE  
  
    FOR i in range of length return_data  
  
        ASSIGN the value of c as c plus 1  
  
        APPEND the value of c in dic_costume which SPILT the return_data of  
        index i by comma  
  
    END FOR  
  
    ASSIGN the value of counter_two as zero
```

CREATE and **OPEN** the text file named COSTUMER added with costumer_id in read mode

DECLARE the variable returning_data to read the file

SPLIT the data in text file by \n

WHILE empty text in returning_data

REMOVE empty text

END WHILE

FOR i in range of length returning_data

ASSIGN the value of counter_two as counter plus 1

APPEND counter_two in dic_return and **SPLIT** the data by comma with returning_data variable of index i

END FOR

APPEND counter_two in valid_data function

CALL update_stock function

CREATE valid_data function and with counter_two in parenthesis

IF counter_two ==1 **THEN**

ASSIGN the value of identity_1 as 1

ELSE

FOR key, value of dic_return in items function

PRINT the text : in variable key and end

FOR j of value of index :1

PRINT the text \t with variable j and end

END FOR

PRINT the text \t

PRINT the text * 50 times

END FOR

INPUT which costume costumer wants as identity_1

PRINT * 50 times

END IF

DECLARE the variable selc_return. **ASSIGN** the variable with **APPENDED** identity_1 as int datatype in dic_return dictionary

DECLARE the variable qty. **APPEND** index 3 of selc_return variable as int datatype.

DECLARE the variable list0002. **ASSIGN** its value with index 0,1,3 of selc_return variable

APPEND list0002 in String on index 1: -1

REPLACE the text ‘ with empty text in list0002

REPLACE the text , with \t in list0002

APPEND the value of list0002 in billing list

APPEND the value of selc_return of index 4 in dic_costume in return_slec variable

DECLARE the variable UQDTY. **APPEND** the value of return_slec of index 3 and **ADD** with the variable qty

APPEND the value of UQDTY as String and **ASSIGN** its value in return_slec variable on index 3

CALL update_stock function

PRINT the text * 50 times

INPUT if the customer wants to continue or print bill in Boolean value

PRINT the text * 50 times

IF the value of sure is no

CALL update_stock function

PRINT the text * 60 times

PRINT the text Thank you. Visit again

PRINT the text * 60 times

PRINT the text Shishir Clothing

PRINT the text Bill Details

PRINT the text * 60 times

PRINT the text Costumer ID with variable Costumer_id

PRINT the text Booked Date with the variable selc_return of index 2

PRINT the text Return Date with the variable return_date

PRINT the text * 60 times

PRINT the text Costume, Brand, and Quantity

PRINT the text * 60 times

FOR item of billing list

PRINT the text \t\t with variable item and end

PRINT the text \n

END FOR

PRINT the text * 60 times

DECLARE the variable day. **ASSIGN** its value with selc_return variable of index 5

DECLARE the variable LATE. **ASSIGN** its value with sum of day plus 5 and minus return_day

IF return_day < 5 day plus 5

ASSIGN the value of total as 0

ELSE

ASSIGN the value LATE multiple of 5

PRINT the text Total Penalty with variable total and again text \$

PRINT the text * 60 times

END IF

ELIF the value of Sure is yes

PRINT the text Please continue

CALL the function return_dic

ELSE

PRINT the text Invalid Input

CALL the method return_dic

END IF

CREATE the function update_stock

DECLARE the variable abc and **OPEN** the text file Shishir.txt in write mode

FOR key, value of dic_costume in items method

WRITE the text comma and **JOIN** its value with value

WRITE the text \n

CLOSE the text file

CALL return_dic function

3. DATA STRUCTURE:

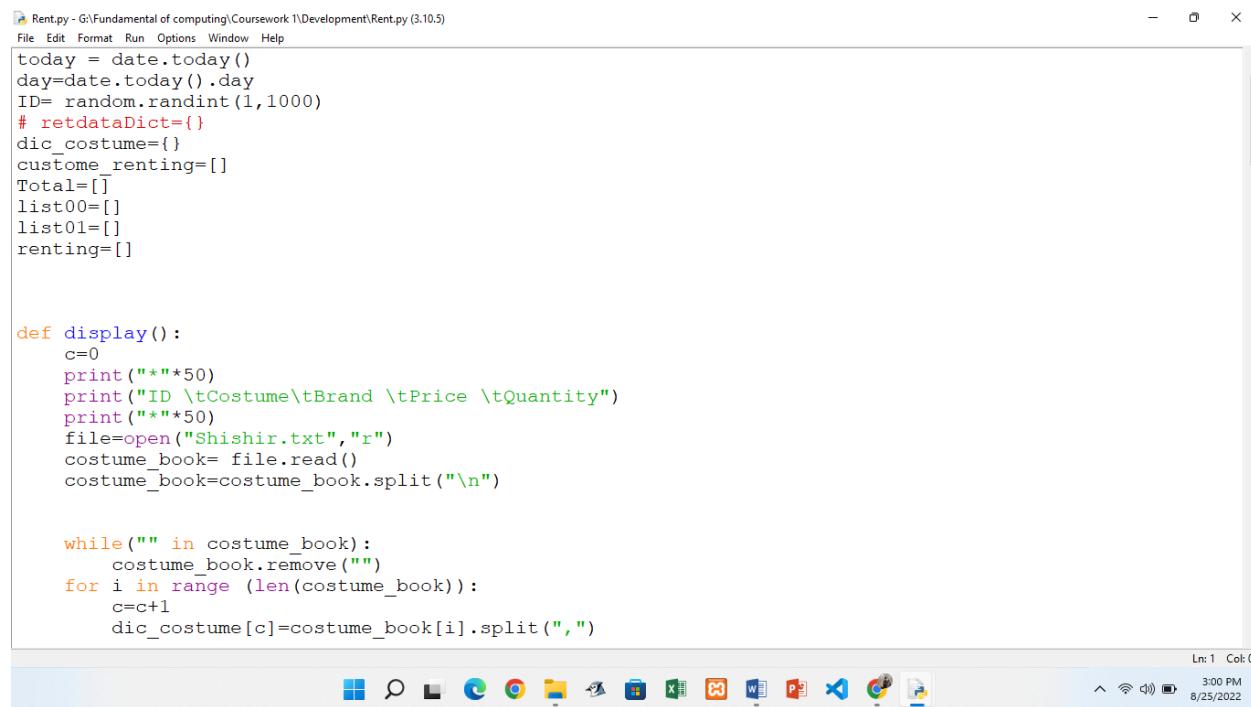
The data structure can be defined as the platform or format of saving, manipulating, managing, and organizing data specially used in programming languages. Programs are built around data structures as they are considered as the fundamental unit of any programming language. As python is considered an easier programming language than other programming languages, the fundamentals of data structure can be learned in an easier way. In python, there are mainly two types of data structures and they are, primitive data structures and non-primitive data structures. (Geeks for Geeks, 2022)

Primitive data structures are also known as basic data structures. The most common primitive data structures are Integer, float, double, String, and Boolean. Among these data structures, integer represents the numeric data ranging from negative infinity to positive infinity. In this coursework, integer data types are used to perform mathematical expressions like adding and subtracting quantities/stocks or the total amounts of money. Float data types are used to represent rational numbers like 8.11, 9.69, 1.11, etc. String data types represent the text data. I have used String data type in this coursework to print texts, and also used it in the process of converting integer data into String format contained in the text file. In programming languages, mostly string data are written inside the “” or “”. The Boolean data type always accepts two values either true or false. Boolean data types are used in conditional and loops in the programming languages. I have used Boolean data type in the while loop by assigning the value of while as either true or false. (web3school, n.d.)

Basically, Non-primitive data structures are known as reference types of data types. Usually, non-primitive data structures are used in refereeing the data in different operations. In python, some of the basic non-primitive data types are arrays, tuples,

lists, sets, dictionaries, and files. First, the array is the collection of different data types elements each as identified by its index number. In this coursework, I haven't used array data structure. Tuples are those data structures that cannot be updated after being created thus, we haven't also used tuples in this coursework. Lists in the data structure are used to store heterogeneous elements of data. Unlike tuples, we can use a list to update the data. In coursework, lists are used multiple times to store different heterogeneous data. Dictionary is a collection of data that stores data value in keys and does not allow the duplication of data. Data in a dictionary are stored inside the curly braces {}. File handling is one of the most important features of a programming language. Python provides several functions to create, manipulate, delete and edit data in the file. (datacamp, 2017)

3.1. Screenshots of data structure implementation:



```
Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
today = date.today()
day=date.today().day
ID= random.randint(1,1000)
# retdataDict={}
dic_costume={}
costume_renting=[]
Total=[]
list00=[]
list01=[]
renting=[]

def display():
    c=0
    print("*"*50)
    print("ID \tCostume \tBrand \tPrice \tQuantity")
    print("*"*50)
    file=open("Shishir.txt","r")
    costume_book= file.read()
    costume_book=costume_book.split("\n")

    while("") in costume_book:
        costume_book.remove("")
    for i in range (len(costume_book)):
        c=c+1
        dic_costume[c]=costume_book[i].split(",")

Ln: 1 Col: 0
3:00 PM
8/25/2022
```

Figure 2: Screenshot of Data structure implementation

```

Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
costume_book=costume_book.split("\n")

while("") in costume_book:
    costume_book.remove("")
for i in range (len(costume_book)):
    c=c+1
    dic_costume[c]=costume_book[i].split(",")

for key,value in dic_costume.items():
    print(key,end="\t")
    for j in value:
        print(j,end="\t")
    print("\n")

update_stock()
Valid(c)

Customer_Name=input("Please Enter Your Name: ")

def Valid(c):
    c=c

    RentID=int(input("Enter the ID: "))
    if (RentID>0 and RentID<=c):
        print("Costume is available")
    else:
        print("Costume is not available")
    update_stock()

Ln: 33 Col: 49
3:00 PM
8/25/2022

```

Figure 3: Screenshot of Data structure implementation

```

Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
RentID=int(input("Enter the ID: "))
if (RentID>0 and RentID<=c):
    print("Costume is available")
    qty=int(input("Enter the quantity you want: "))
    costume_select=dic_costume[RentID]

    if qty<=int(costume_select[3]):
        print("Quantity is Available please continue")
        update_qty=int(costume_select[3])-qty
        costume_select[3]=str(update_qty)
        update_stock()
        abcd=[str(costume_select[0]),str(costume_select[1]),str(today),str(qty)]
        abcd = str(abcd)[1:-1]
        abcd=abcd.replace("'", "\t")
        abcd=abcd.replace(",","")
        abcd=abcd.replace("$","")

        custome_renting.append(abcd)

        total=int(costume_select[2].rstrip("$"))*qty
        Total.append(total)

        xyz=[str(costume_select[0]),str(costume_select[1]),str(today),str(qty)]
        xyz = str(xyz)[1:-1]
        xyz=xyz.replace("'", "\t")
        xyz=xyz.replace(",","")
        xyz=xyz.replace("$","")

Ln: 33 Col: 49
3:01 PM
8/25/2022

```

Figure 4: Screenshot of Data structure implementation

```
Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
#
#     ghi.write(str(item))
#     ghi.write("\n")

def write_two():
    ghi=open('COSTUMER'+str(ID)+".txt", 'w')

    for aaaa in renting:
        ghi.write(str(aaaa))
        ghi.write("\n")
    ghi.close()

def write():
    cc=open('COS_Bill'+str(ID)+".txt",'w')
    cc.write("-"*60)
    cc.write("\n")
    cc.write("\t Shishir Clothing")
    cc.write("\n")
    cc.write("-"*60)
    cc.write("\n")
    cc.write("\n")
    cc.write("\t Bill Details")
    cc.write("\n")
    cc.write("Name: ")
    cc.write(Customer_Name)
    cc.write("\n")
    cc.write("Customer ID: ")
    cc.write(str(ID))
    cc.write("\n\n")

Ln: 33 Col: 49
3:01 PM
8/25/2022
```

Figure 5: Screenshot of Data structure implementation

```
Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
cc.write("\n")
cc.write("Costume")
cc.write("\t Brand")
cc.write("\t Book Date")
cc.write("\t Quantity")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
for dd in list00:
    cc.write(str(dd))
    cc.write("\n")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
cc.write("total: ")
cc.write(str(sum(Total)))
cc.close()

def update_stock():
    oo=open("Shishir.txt","w")
    for key,value in dic_costume.items():
        oo.write(",".join(value))
        oo.write("\n")
    oo.close()
display()

Ln: 33 Col: 49
3:01 PM
8/25/2022
```

Figure 6: Screenshot of Data structure implementation

```

Return.py - G:\Fundamental of computing\Coursework 1\Development\Return.py (3.10.5)
File Edit Format Run Options Window Help
from datetime import date
return_date=date.today()
return_day=date.today().day
dic_return={}
dic_costume={}
billing=[]

print("-"*50)
Costumer_id=input('Enter Costumer ID: ')
print("-"*50)

def return_dic():
    c=0
    file=open("Shishir.txt","r")
    return_data= file.read()
    return_data=return_data.split("\n")

    while("") in return_data:
        return_data.remove("")
    for i in range (len(return_data)):
        c=c+1
        dic_costume[c]=return_data[i].split(",")

    counter_two=0
    file=open('COSTUMER'+str(Costumer_id)+".txt","r")

```

Figure 7: Screenshot of Data structure implementation

```

Return.py - G:\Fundamental of computing\Coursework 1\Development\Return.py (3.10.5)
File Edit Format Run Options Window Help
counter_two=0
file=open('COSTUMER'+str(Costumer_id)+".txt","r")
returing_data= file.read()
returing_data=returing_data.split("\n")
while("") in returing_data:
    returing_data.remove("")
for i in range (len(returing_data)):
    counter_two=counter_two+1
    dic_return[counter_two]=returing_data[i].split(",")
Valid_data(counter_two)
update_stock()

def Valid_data(counter_two):
    if counter_two==1:
        identity_1=1
    else:
        for key,value in dic_return.items():
            print(key,end=": ")
            for j in value[:1]:
                print(j,end="\t")
            print("\n")
            print("-"*50)
        identity_1= input("Which costume you want to return?: ")
        print("-"*50)

    selc_return=dic_return[int(identity_1)]

```

Figure 8: Screenshot of Data structure implementation

```

*Return.py - G:\Fundamental of computing\Coursework 1\Development\Return.py (3.10.5)
File Edit Format Run Options Window Help
for item in billing:
    print("\t \t",item,end="\t\t")
    print("\n")
print("*"*60)
day=int(selc_return[5])
LATE=(day+5)-return_day
if return_day<day+5:
    total=0
else:
    LATE*5
    print("\t Total Penalty",total,"$","\n")
    print("*"*60)
elif Sure=="yes" :
    print("Please continue")
    return_dic()
else:
    print("Invalid Input")
    return_dic()

def update_stock():
    abc=open("Shishir.txt","w")
    for key,value in dic_costume.items():
        abc.write(",".join(value))
        abc.write("\n")
    abc.close()

return_dic()

```

Ln: 102 Col: 20
3:02 PM 8/25/2022

Figure 9: Screenshot of Data structure implementation

```

*View.py - G:\Fundamental of computing\Coursework 1\Development\View.py (3.10.5)
File Edit Format Run Options Window Help
dic_costume={}

def display():
    c=0
    print("*"*50)
    print("ID \tCostume \tBrand \tPrice \tQuantity")
    print("*"*50)
    file=open("Shishir.txt","r")
    costume_book= file.read()
    costume_book=costume_book.split("\n")
    # removes empty string

    while("") in costume_book:
        costume_book.remove("")
    for i in range (len(costume_book)):
        c=c+1
        dic_costume[c]=costume_book[i].split(",")

    for key,value in dic_costume.items():
        print(key,end="\t")
        for j in value:
            print(j,end="\t")
        print("\n")

display()

```

Ln: 1 Col: 0
3:03 PM 8/25/2022

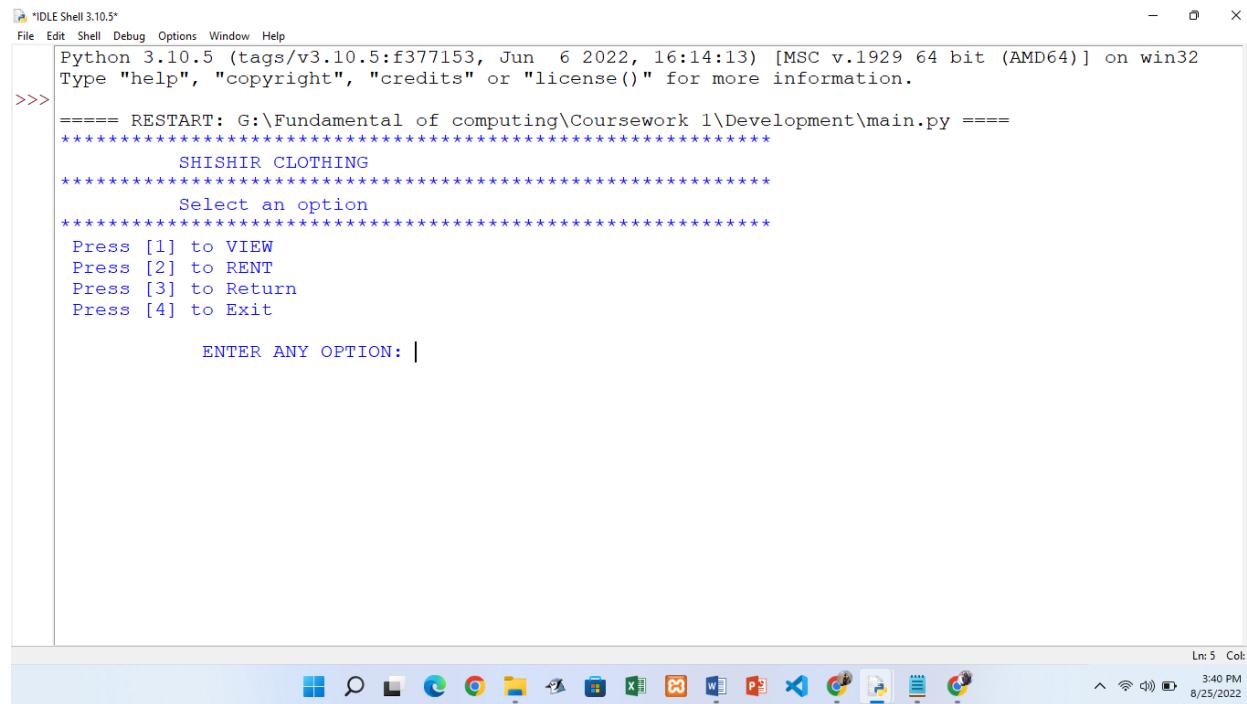
Figure 10: Screenshot of Data structure implementation

4. PROGRAM:

4.1. Implementation of the program:

The entire program is divided into some modules. Altogether there are four python files and one text file. The python files are main.py, rent.py, return.py, and view.py and the text file is Shishir.txt. The python file main.py is the main functioning file that is used to call every python file and run all of them from one single file. The main function of the Rent.py file is to rent the costume by selecting appropriate options and following the proper procedure. Another important python file Return.py is used to write the program for returning the costume which was rented earlier. The View.py file contains the costume name, brand, quantity, and the price of the costume which are required to rent the costume. The Shishir.txt text file is created to contain the available Costume name, brand, quantity, and price.

The following is an illustration of how the program works. In each step, I will show the implementation of my code. First, I will run the main.py python file and follow the instructions.



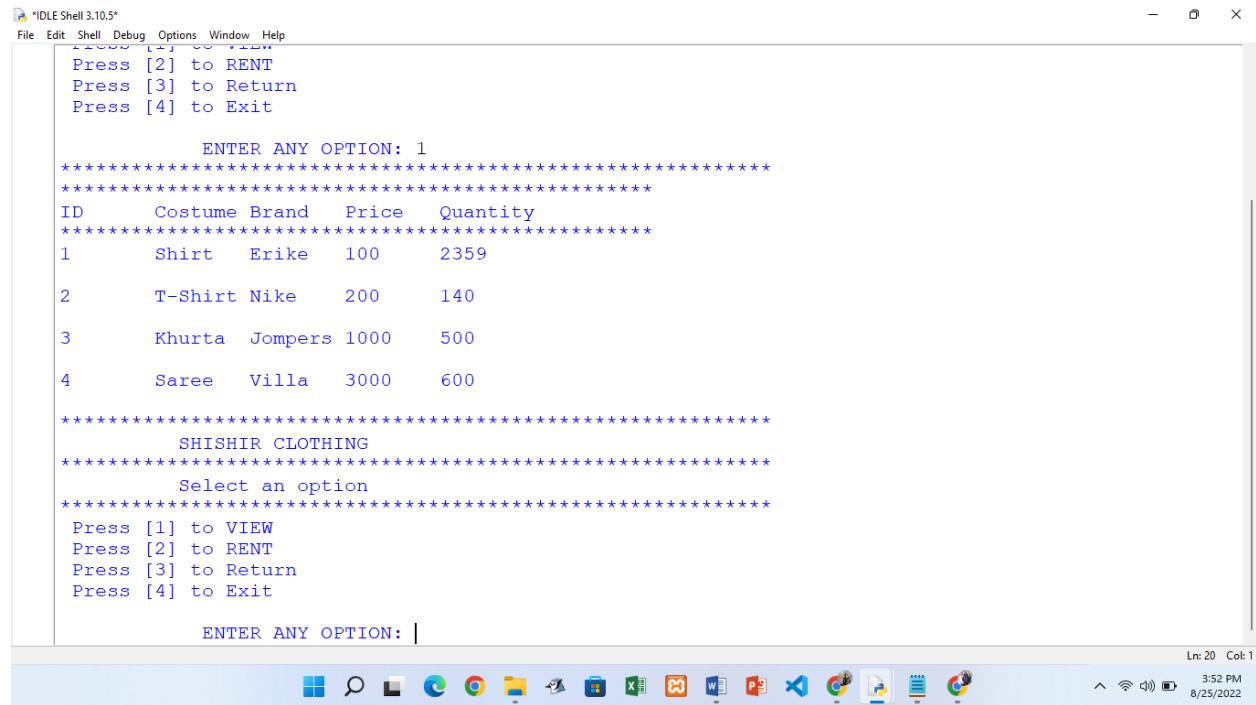
```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: G:\Fundamental of computing\Coursework 1\Development\main.py =====
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: |
```

Figure 11: Implementation of program main menu

Whenever we run the main.py file we get this interface where a system asks us to input any number from 1 to 4. 1 is for viewing the available costume, 2 is for renting the costume, 3 is for returning the rented costume, and 4 is to exit the program. When we enter the 1 program show us the available costume from the Shishir.txt file.



```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 1
*****
ID      Costume Brand   Price   Quantity
*****
1       Shirt    Erike    100     2359
2       T-Shirt  Nike     200     140
3       Khurta   Jompers  1000    500
4       Saree    Villa    3000    600
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: |

```

The screenshot shows a Windows desktop environment with a Python IDLE window open. The window title is "*IDLE Shell 3.10.5*". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main area displays a menu with options: Press [1] to VIEW, Press [2] to RENT, Press [3] to Return, and Press [4] to Exit. Below this, it prompts "ENTER ANY OPTION: 1". A table follows, listing four items with columns: ID, Costume, Brand, Price, and Quantity. The table is formatted with asterisks as separators. After the table, it says "SHISHIR CLOTHING" and "Select an option". It then repeats the menu options. At the bottom, it says "ENTER ANY OPTION: |". The taskbar at the bottom shows various application icons like File Explorer, Edge, and Microsoft Office. The status bar at the bottom right indicates "Ln: 20 Col: 1" and the date/time "3:52 PM 8/25/2022".

Figure 12: Implementation of program when view option is selected

In the above screenshot, the View option is selected and the available costume is shown to us. And if we enter the invalid option like entering the greater number than the available option then it shows the text invalid option.

```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
1      Shirt  Erike   100    2359
2      T-Shirt Nike    200    140
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 5
*****
Invalid input
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION:

Ln: 36 Col: 0
3:41 PM
8/25/2022

```

Figure 13: Implementation of program during wrong input

4.2. Renting and returning the costume

4.2.1. Renting

When option two is selected from the main menu, it directs the user to the renting interface where the user can rent any available costume of the available quantity. First, it asks the user their name to continue the future process of renting a costume. Entered customer name is used later in the bill with the randomly generated customer id to uniquely identify the customer.

```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
4      Saree    Villa    3000    600
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 2
*****
Please Enter Your Name: Shishir Ghimire
*****
ID      Costume Brand   Price   Quantity
*****
1      Shirt     Erike    100    2359
2      T-Shirt   Nike     200    140
3      Khurta   Jompers  1000   500
4      Saree    Villa    3000   600
Enter the ID:

```

The screenshot shows a Python script running in the IDLE shell. It displays a menu for a costume rental system. After selecting option 2 (RENT), it prompts the user to enter their name, which is "Shishir Ghimire". The program then lists all available costumes with their details (ID, name, brand, price, quantity). Finally, it asks the user to enter the ID of the costume they want to rent.

Figure 14: Entering the customer name

After entering the customer name program displays all the available costumes. After displaying the available costume details with name, brand, price, and quantity the program asks the user to enter the ID of the costume that the user wants to rent. With that Id in a row, there is entire detail of that costume with quantity and price. When we enter the option of our desirable costume, the program checks if the costume is available or not. If the costume is available, then a text is generated where the costume is available is written. Then We have to enter the quantity of the costume that we need then the program checks if the quantity is available or not. If the quantity is available, then the program asks the user if he/she still wants to continue renting. If yes, then again this available costume list is shown. Also if the user enters the invalid option then a text is generated where the Invalid option, Try again is shown. If the costume is not available in enough quantity, then the program informs the user that the costume is not available at the moment.

```

*IDLE Shell 3.10.5*
File Edit Shell Options Window Help
Enter the ID:
=====
RESTART: G:\Fundamental of computing\Coursework 1\Development\main.py =====
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 2
*****
Please Enter Your Name: Shishir Ghimire
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2359
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID: 1
Costume is available

```

Ln: 113 Col: 0
4:21 PM
8/25/2022

Figure 15: Entering the costume Id

```

*IDLE Shell 3.10.5*
File Edit Shell Options Window Help
=====
RESTART: G:\Fundamental of computing\Coursework 1\Development\main.py =====
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 2
*****
Please Enter Your Name: Shishir Ghimire
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2359
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID: 1
Costume is available
Enter the quantity you want: 100

```

Ln: 114 Col: 20
4:21 PM
8/25/2022

Figure 16: Entering the quantity of costume

```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 2
*****
Please Enter Your Name: Shishir Ghimire
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2359
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID: 1
Costume is available
Enter the quantity you want: 100
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to existyes

```

Ln: 118 Col: 0
4:21 PM
8/25/2022

Figure 17: Asking for conformation to continue or not

```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
*****
1 Shirt Erike 100 2359
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

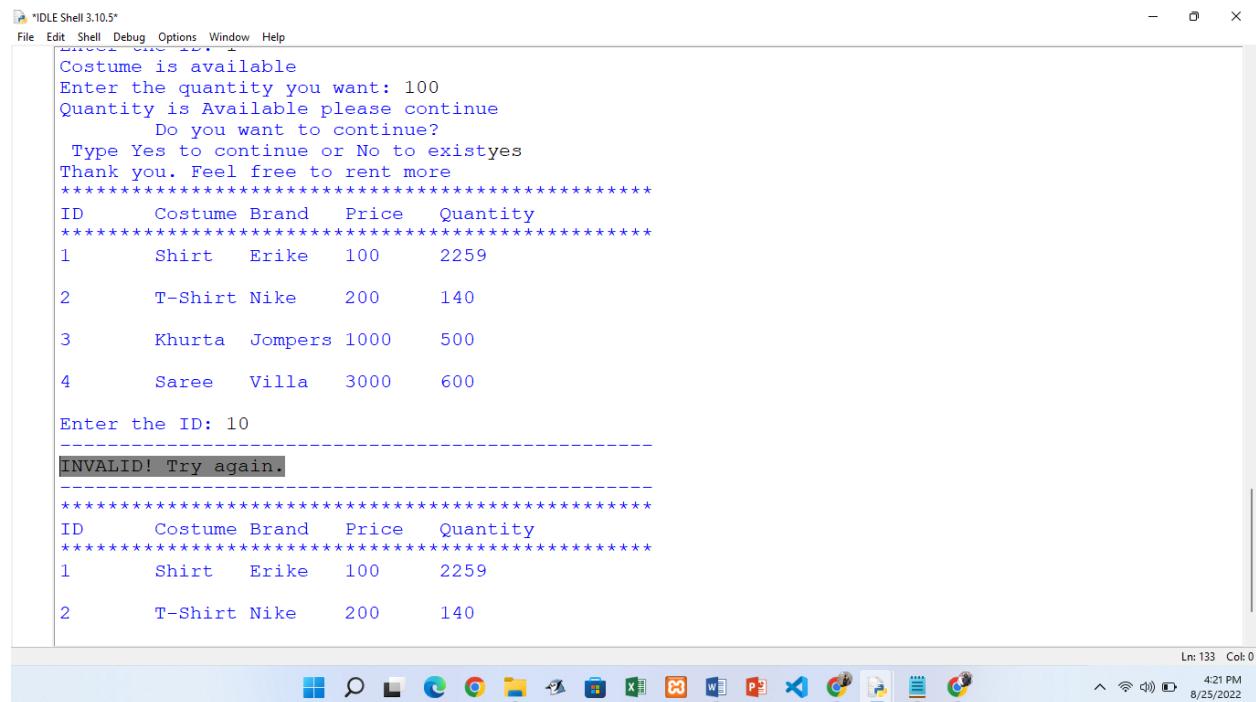
Enter the ID: 1
Costume is available
Enter the quantity you want: 100
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to existyes
Thank you. Feel free to rent more
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID: |

```

Ln: 121 Col: 0
4:21 PM
8/25/2022

Figure 18: When user continues renting



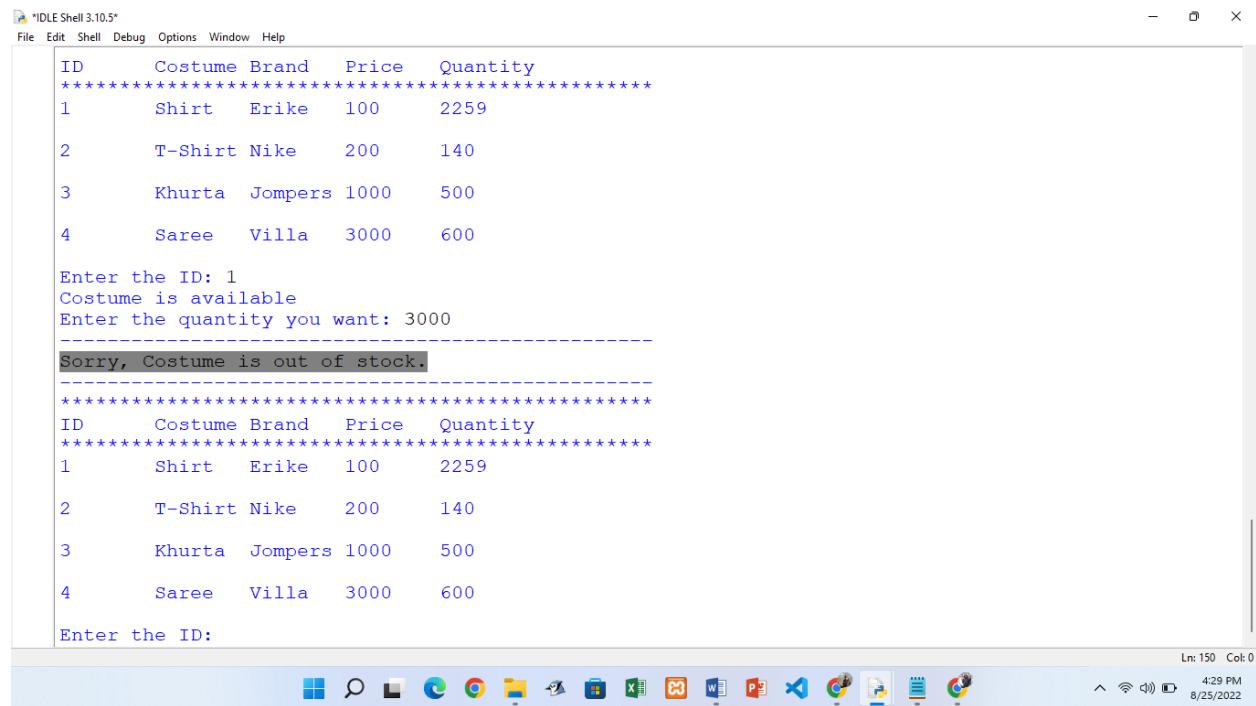
```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Costume is available
Enter the quantity you want: 100
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to existyes
Thank you. Feel free to rent more
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID: 10
-----
INVALID! Try again.
-----
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 140

```

The screenshot shows a Python script running in the IDLE shell. It displays a menu for renting costumes. When the user enters an invalid ID (10), the program prints an error message and asks for another input. The system then lists the available costumes again.

Figure 19: When invalid Id is entered


```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID: 1
Costume is available
Enter the quantity you want: 3000
-----
Sorry, Costume is out of stock.
-----
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 140
3 Khurta Jompers 1000 500
4 Saree Villa 3000 600

Enter the ID:

```

The screenshot shows a Python script running in the IDLE shell. It displays a menu for renting costumes. When the user enters an invalid quantity (3000 for costume ID 1), the program prints an error message and asks for another input. The system then lists the available costumes again.

Figure 20: When an invalid quantity is entered

When all the details of renting the costume are filled like in the above screenshots then a bill is generated showing all the details of costume name, brand, price, and quantity with the total sum of money to be paid.

```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Enter the ID: 1
Costume is available
Enter the quantity you want: 100
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to existno

#####
# Shishir Clothing
#####

-----
Billing
-----

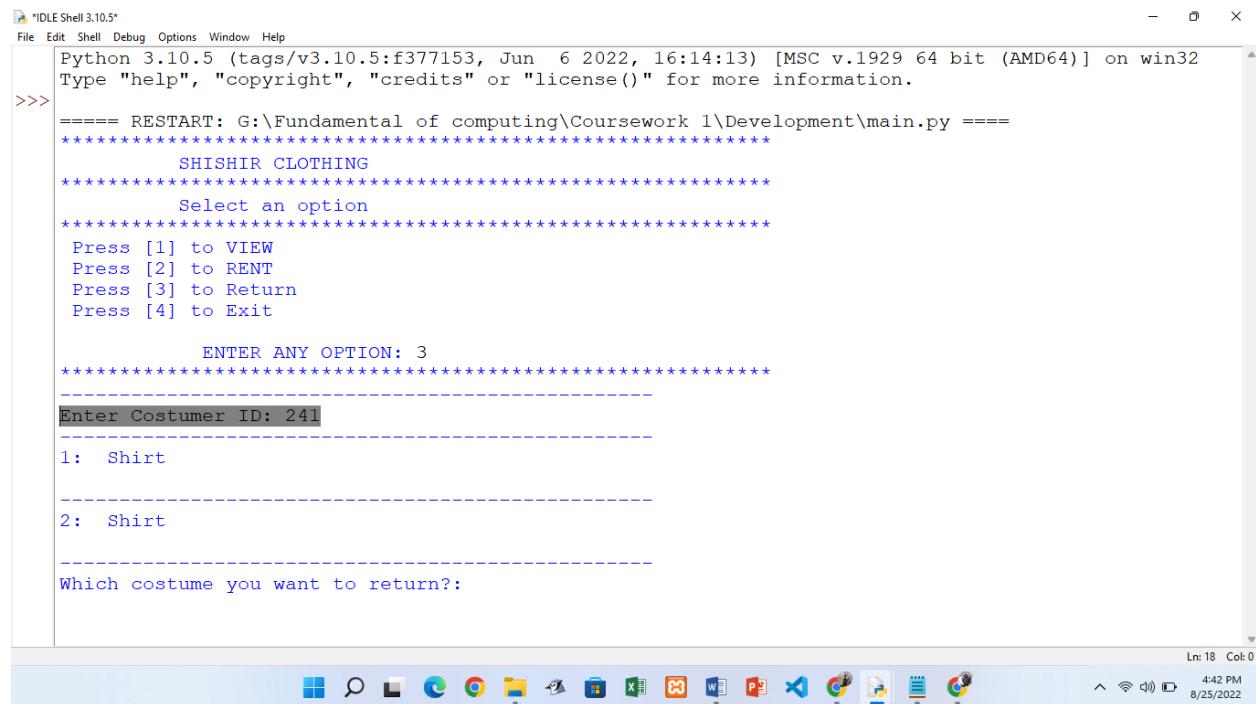
Customer Name: Shishir Ghimire
Customer ID: 241
Costume      Brand      Booked Date    Quantity
#####
Shirt        Erike     2022-08-25      100
Shirt        Erike     2022-08-25      100
Total= 40000
#####
*****
```

Ln: 171 Col: 0
4:37 PM 8/25/2022

Figure 21: Bill is generated

4.2.2. Renting

When we press option three from the menu the program directs us to the return interface then the program asks the user to enter the customer Id. From this Id, the program finds the data of that exact user from the text file which was generated when the customer had rented the costume. When we enter the customer id the program asks the user which costume the user wants to return. If the customer has rented only a single costume, then the program directly asks the customer if they want to print out the bill. After filling in all the information bill is printed. If the user had delayed returning the costume for more than five days, then the penalty is added to the bill.



```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

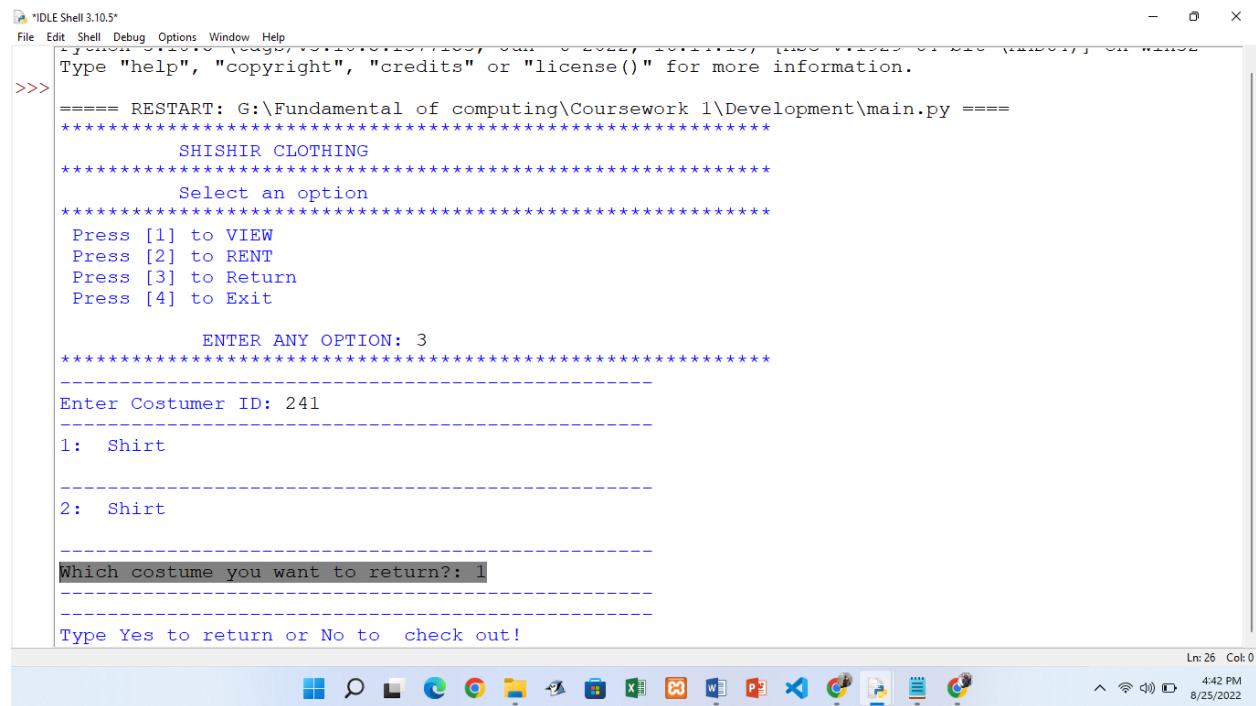
>>>
===== RESTART: G:\Fundamental of computing\Coursework 1\Development\main.py =====
***** SHISHIR CLOTHING *****
***** Select an option *****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 3
*****
----- Enter Costumer ID: 241
-----
1: Shirt

-----
2: Shirt

-----
Which costume you want to return?:
```

Ln: 18 Col: 0
4:42 PM
8/25/2022

Figure 22: Entering the customer Id


```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: G:\Fundamental of computing\Coursework 1\Development\main.py =====
***** SHISHIR CLOTHING *****
***** Select an option *****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 3
*****
----- Enter Costumer ID: 241
-----
1: Shirt

-----
2: Shirt

-----
Which costume you want to return?: 1
-----
Type Yes to return or No to check out!
```

Ln: 26 Col: 0
4:42 PM
8/25/2022

Figure 23: Asking which costume customer wants to return



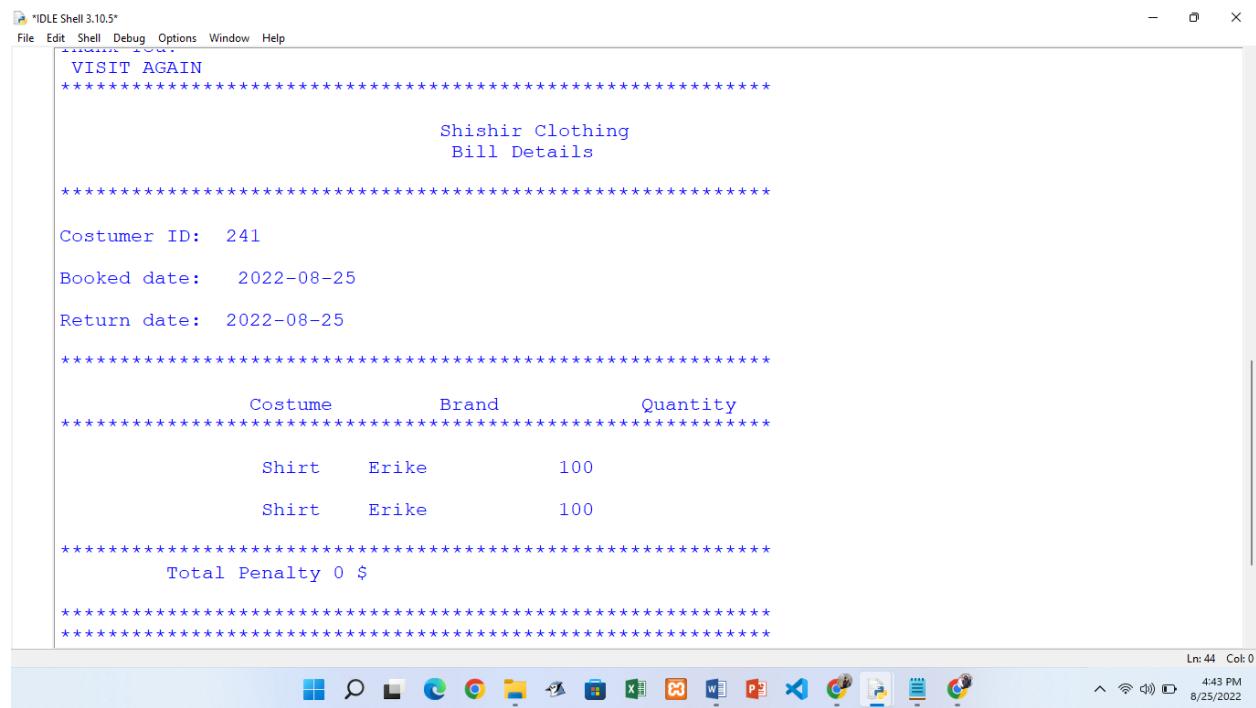
```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 3
*****
Enter Costumer ID: 241
-----
1: Shirt
-----
2: Shirt
-----
Which costume you want to return?: 1
-----
Type Yes to return or No to check out! yes
-----
Please continue
1: Shirt
-----
2: Shirt
-----
Which costume you want to return?:

```

Ln: 29 Col: 0
4:43 PM
8/25/2022

Figure 24: Asking the customer to check out


```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
VISIT AGAIN
*****
Shishir Clothing
Bill Details
*****
Costumer ID: 241
Booked date: 2022-08-25
Return date: 2022-08-25
*****
Costume       Brand       Quantity
*****
Shirt        Erike       100
Shirt        Erike       100
*****
Total Penalty 0 $
*****

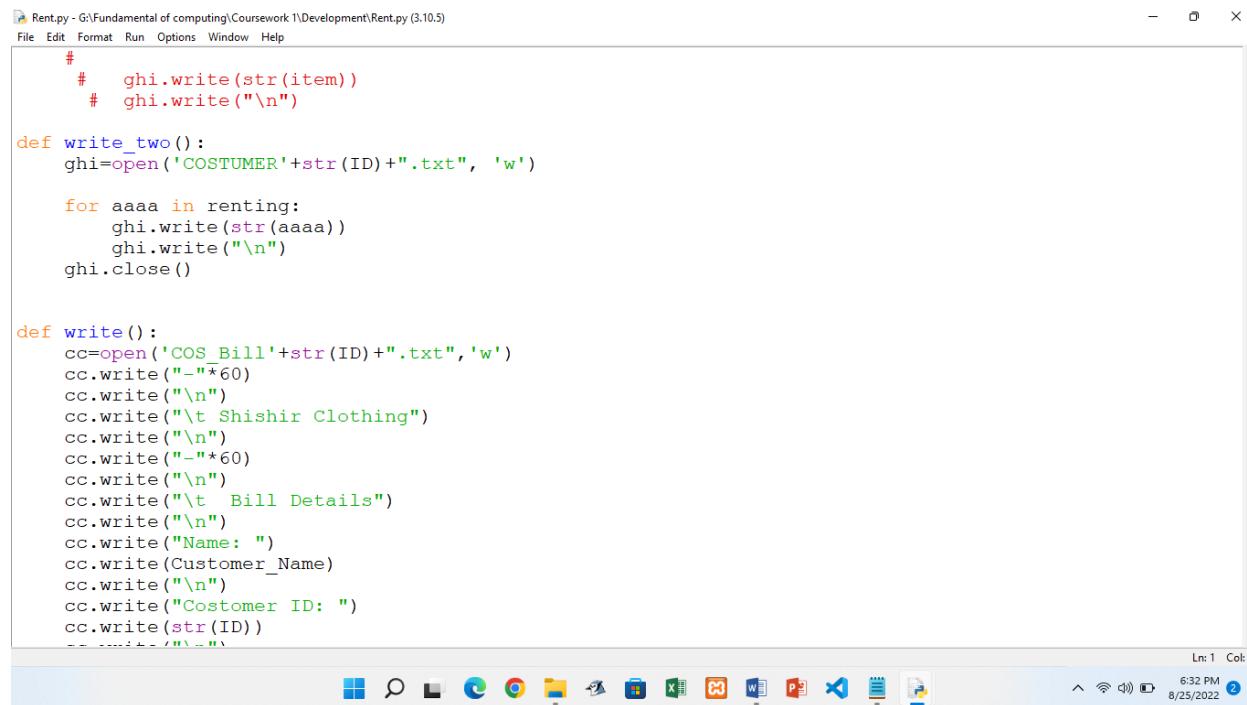
```

Ln: 44 Col: 0
4:43 PM
8/25/2022

Figure 25: Bill print

4.3. Creation of text file:

The text file is created when the user rents any costume. The text file is created by the name of “COSTUMER”+ “Costumer_id” “.txt”. In this text, the file program writes all the data required for further implementation of the program. It stores the data like costume name, costume brand, booked date, quantity, Id of the costume, and the day when the costume was rented. Also, another text file is generated after renting the costume and is created with the file name of “COS_bill” + “Costumer_id” “.txt”. This auto-generated text file program saves the bill which is given to the customer so that they can bring it with them at the time of returning the costume. The screenshot for the implementation of the creation of the text file is given below:



```
Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
#
#     ghi.write(str(item))
#     ghi.write("\n")

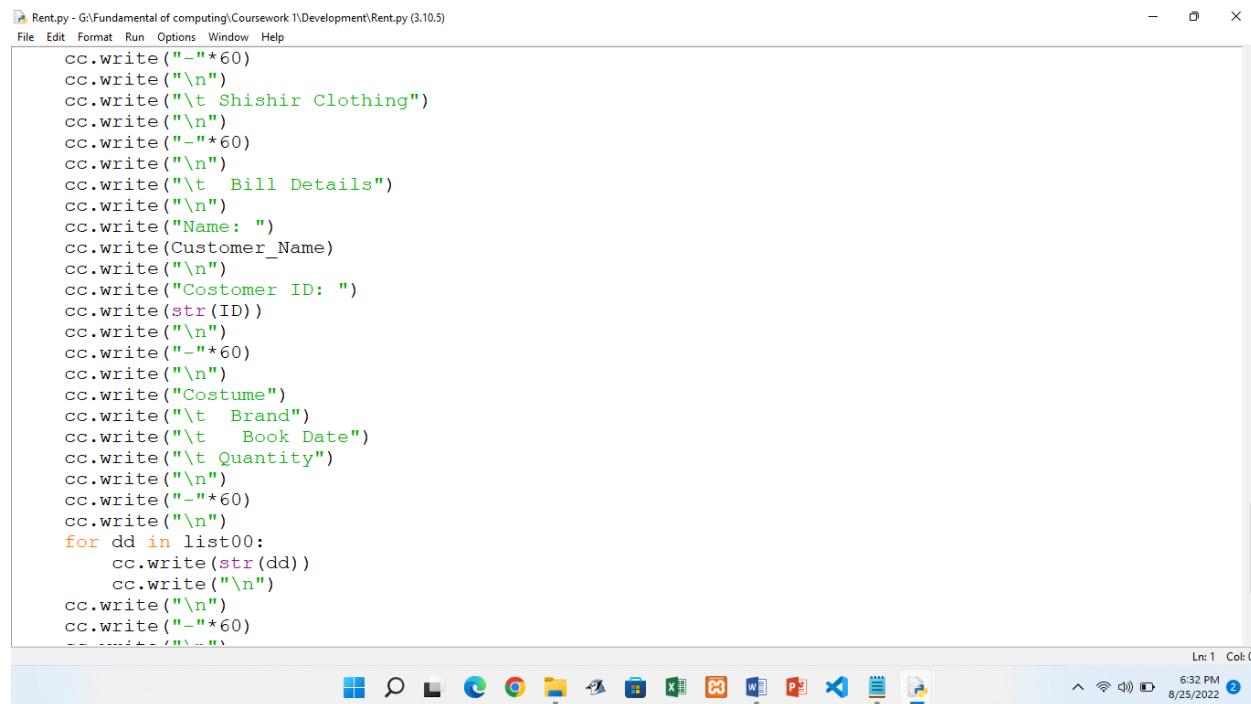
def write_two():
    ghi=open('COSTUMER'+str(ID)+".txt", 'w')

    for aaaa in renting:
        ghi.write(str(aaaa))
        ghi.write("\n")
    ghi.close()

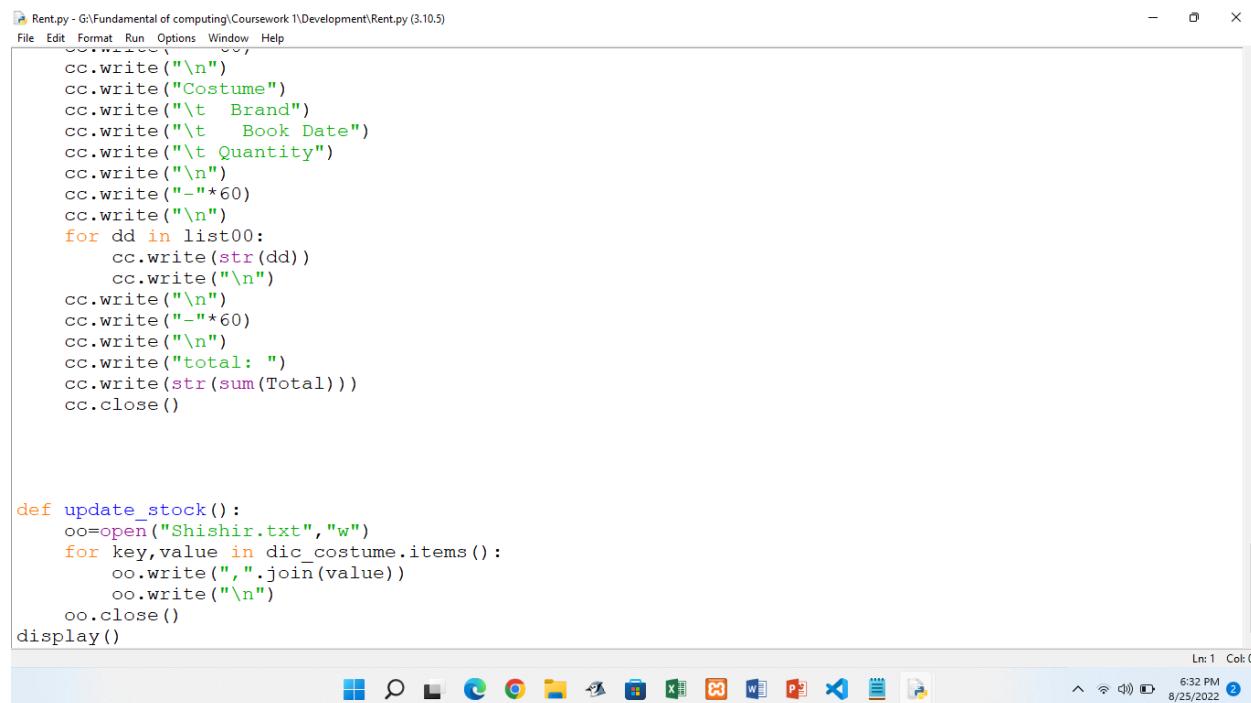
def write():
    cc=open('COS_Bill'+str(ID)+".txt", 'w')
    cc.write("-"*60)
    cc.write("\n")
    cc.write("\t Shishir Clothing")
    cc.write("\n")
    cc.write("-"*60)
    cc.write("\n")
    cc.write("\t Bill Details")
    cc.write("\n")
    cc.write("Name: ")
    cc.write(Customer_Name)
    cc.write("\n")
    cc.write("Customer ID: ")
    cc.write(str(ID))
    cc.write("\n")
    cc.write("-"*60)

Ln: 1 Col: 0
6:32 PM 8/25/2022
```

Figure 26: Creation of the text file



```
Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
cc.write("-"*60)
cc.write("\n")
cc.write("\t Shishir Clothing")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
cc.write("\t Bill Details")
cc.write("\n")
cc.write("Name: ")
cc.write(Customer_Name)
cc.write("\n")
cc.write("Costomer ID: ")
cc.write(str(ID))
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
cc.write("Costume")
cc.write("\t Brand")
cc.write("\t Book Date")
cc.write("\t Quantity")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
for dd in list00:
    cc.write(str(dd))
    cc.write("\n")
cc.write("\n")
cc.write("-"*60)
Ln: 1 Col: 0
6:32 PM 8/25/2022
```

Figure 27: Creation of the text file


```
Rent.py - G:\Fundamental of computing\Coursework 1\Development\Rent.py (3.10.5)
File Edit Format Run Options Window Help
cc.write("\n")
cc.write("Costume")
cc.write("\t Brand")
cc.write("\t Book Date")
cc.write("\t Quantity")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
for dd in list00:
    cc.write(str(dd))
    cc.write("\n")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
cc.write("total: ")
cc.write(str(sum(Total)))
cc.close()

def update_stock():
    oo=open("Shishir.txt","w")
    for key,value in dic_costume.items():
        oo.write(",".join(value))
        oo.write("\n")
    oo.close()
display()
Ln: 1 Col: 0
6:32 PM 8/25/2022
```

Figure 28: Creation of the text file

4.4. Opening the text file and showing bill

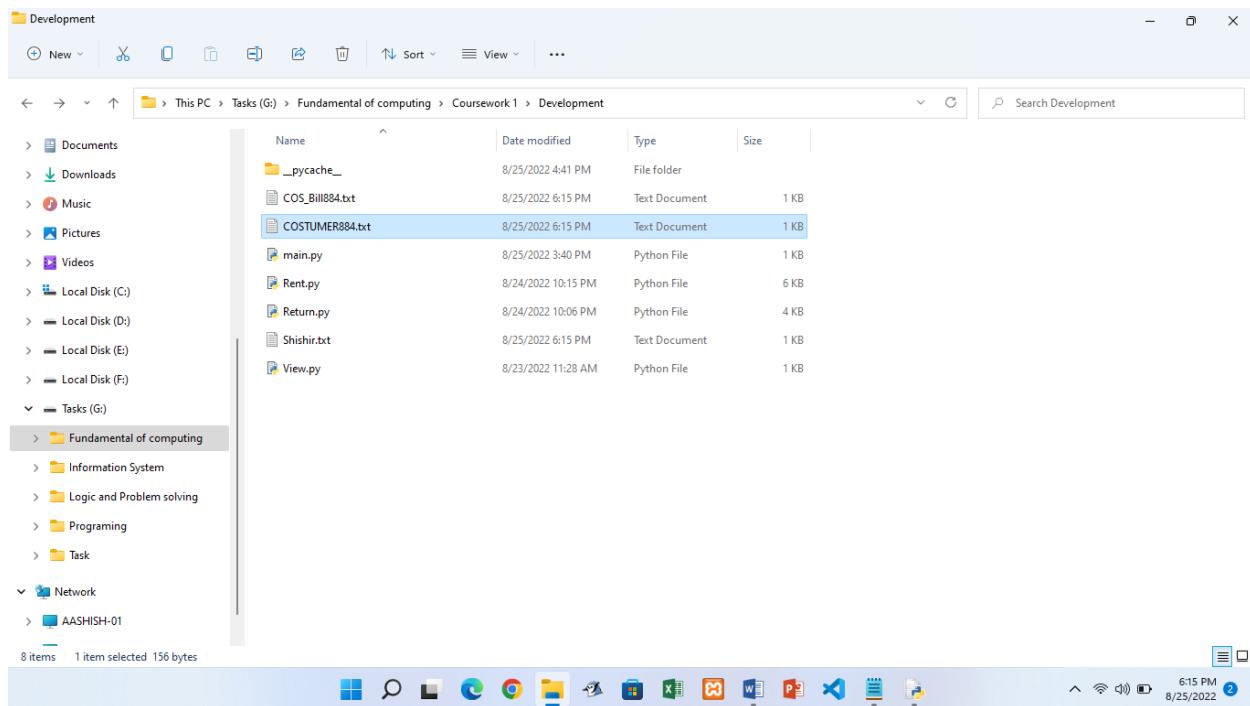


Figure 29: Folder with the auto generated text files and python files

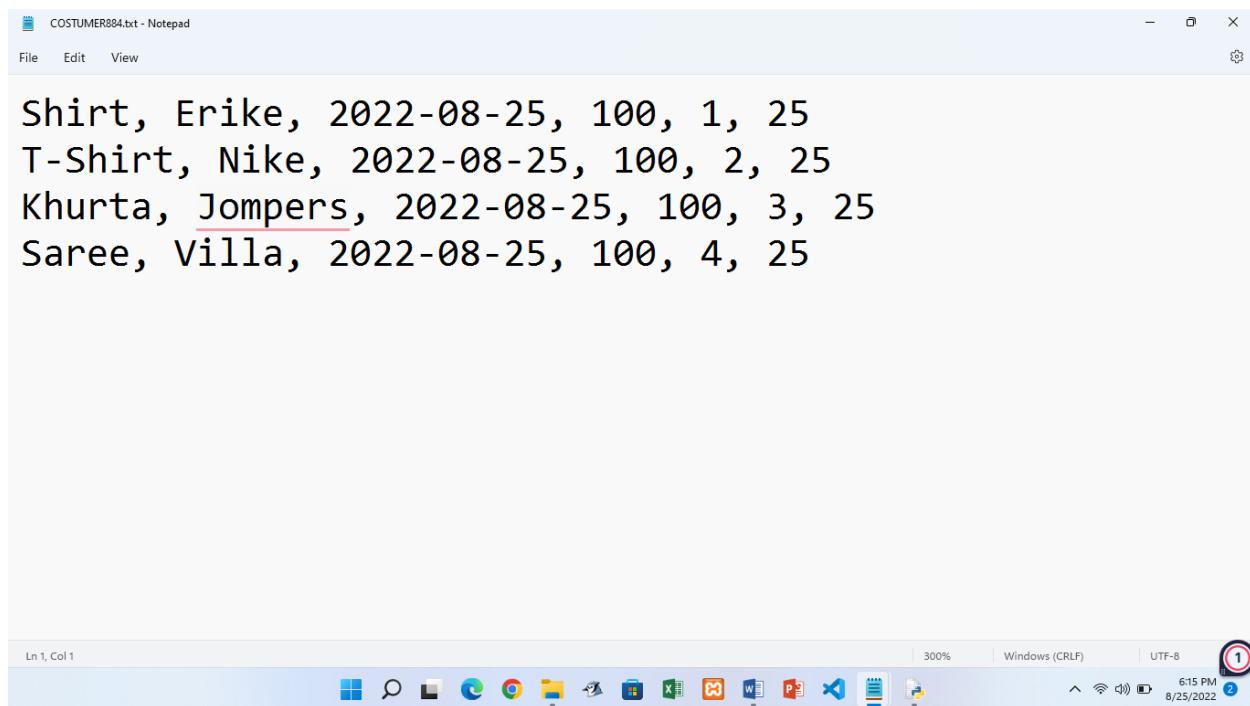
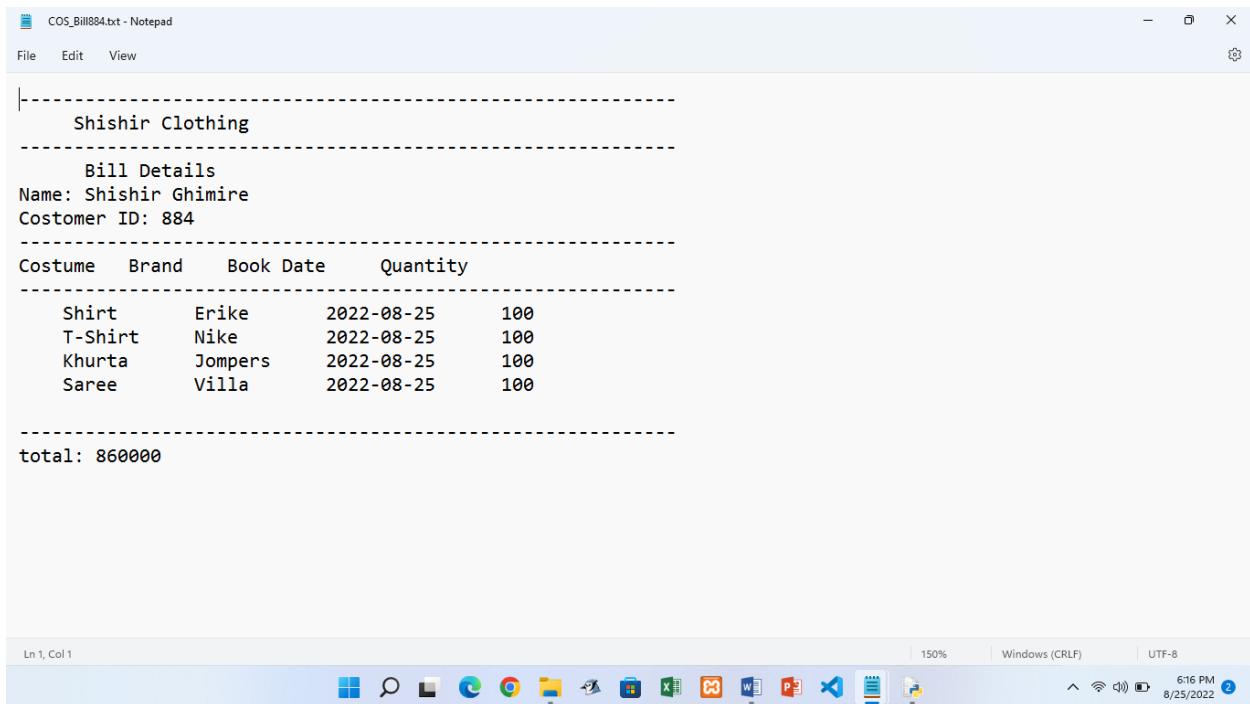


Figure 30: Auto generated text file with the data



COS_Bill884.txt - Notepad

File Edit View

```

-----
Shishir Clothing
-----
Bill Details
Name: Shishir Ghimire
Customer ID: 884
-----
Costume Brand Book Date Quantity
-----
Shirt Erike 2022-08-25 100
T-Shirt Nike 2022-08-25 100
Khurta Jompers 2022-08-25 100
Saree Villa 2022-08-25 100
-----
total: 860000

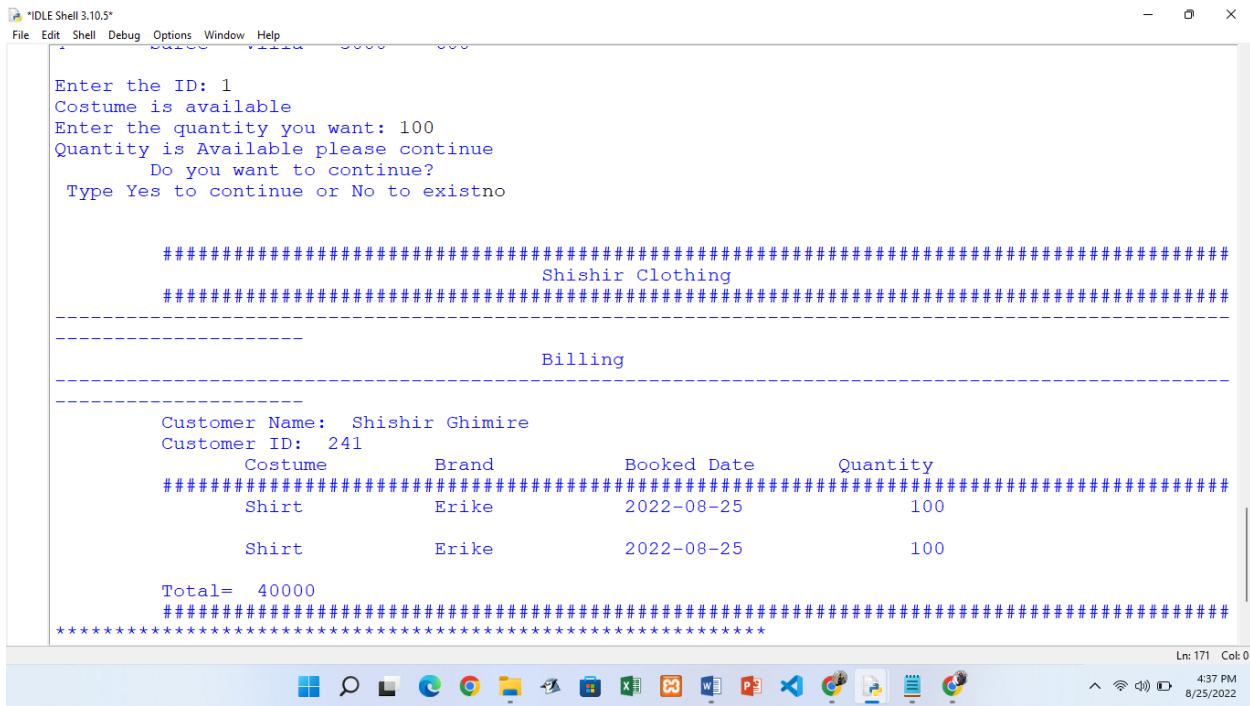
```

Ln 1, Col 1

150% Windows (CRLF) UTF-8

6:16 PM 8/25/2022

Figure 31: Auto-generated text file which stores the bill



IDLE Shell 3.10.5

File Edit Shell Debug Options Window Help

```

Enter the ID: 1
Costume is available
Enter the quantity you want: 100
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to existno

#####
# Shishir Clothing #
#####

-----
Billing
-----

Customer Name: Shishir Ghimire
Customer ID: 241
Costume Brand Booked Date Quantity
#####
# Shirt Erike 2022-08-25 100
# Shirt Erike 2022-08-25 100
#
Total= 40000
#####
*****
```

Ln: 171 Col: 0

4:37 PM 8/25/2022

Figure 32: Bill printed in the terminal after renting

4.5. Terminating the program:

When we press option four, the program is terminated after displaying a text message.

The screenshot shows the Python 3.10.5 IDLE Shell interface. The window title is "IDLE Shell 3.10.5". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the following output:

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: G:\Fundamental of computing\Coursework 1\Development\main.py =====
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 4
*****
SystemExit: Thank You For trusting us
>>>
```

The status bar at the bottom right shows "Ln: 18 Col: 0", the system tray icons, and the date and time "7:54 PM 8/25/2022".

Figure 33: Program is terminated

5. TESTING

5.1. Test One:

| | |
|-----------------|--|
| TEST No. | ONE |
| OBJECTIVE | To implement try catch |
| ACTION | = Pressed option two rent = Entered the costumer name = Entered the String value in ID |
| EXPECTED RESULT | The program will handle the exception and message will be displayed. |
| ACTUAL RESULT | The program has handled the exception and message was displayed. |
| CONCLUSION | Test was successful. |

Table 1: Test One

```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
*****ENTER THE OPTION*****
Please Enter Your Name: Shisir Ghimire
*****
... ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 980
2 T-Shirt Nike 200 4000
3 Khurta Jompers 1000 4500
4 Saree Villa 3000 5000

Enter the ID: ehfuh
Please, Enter numeric value only
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 980
2 T-Shirt Nike 200 4000
3 Khurta Jompers 1000 4500
4 Saree Villa 3000 5000

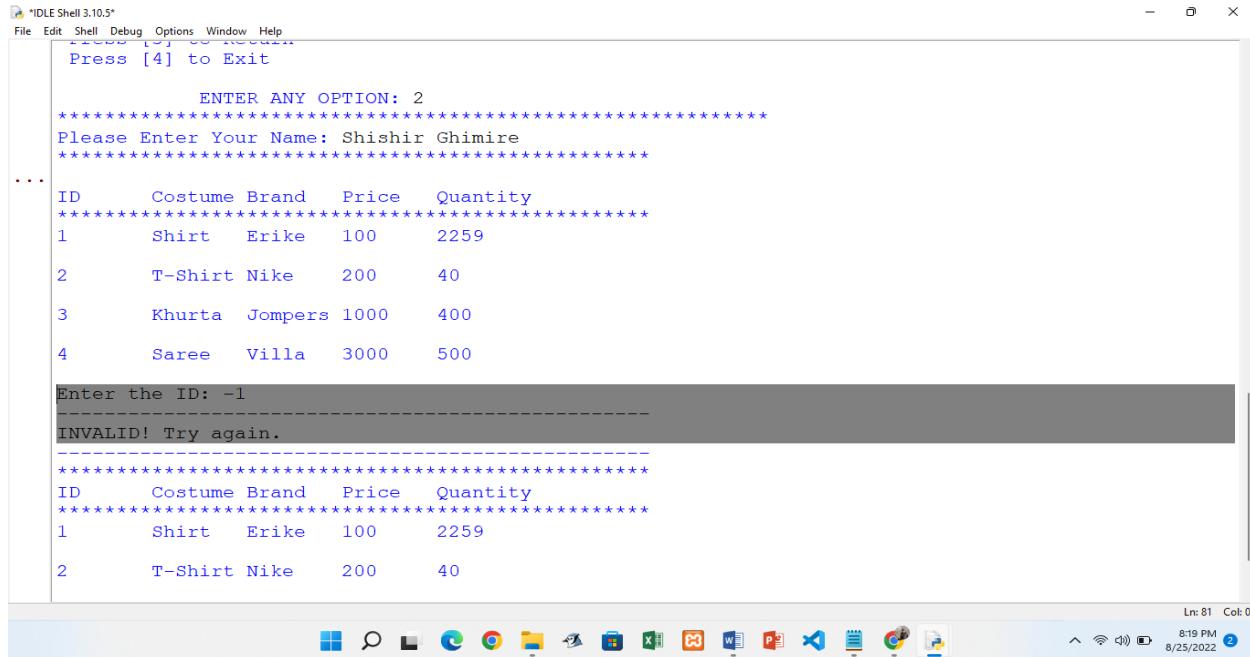
Enter the ID:
```

Table 2: Exception handled

5.2. Test Two:

| | |
|-----------------|--|
| TEST No. | TWO |
| OBJECTIVE | To rent and return custom with invalid inputs and showing the resulting message. |
| ACTION | = Pressed option two rent = Entered the costumer name = Inputted negative ID value as input = Inputted non-existing number of ID value as input |
| EXPECTED RESULT | The program will print a text informing about the invalid input. |
| ACTUAL RESULT | The program has printed a text informing about the invalid input. |
| CONCLUSION | Test was successful. |

Table 3: Test two



```

IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Press [4] to Exit

ENTER ANY OPTION: 2
*****
Please Enter Your Name: Shishir Ghimire
*****


ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 40
3 Khurta Jompers 1000 400
4 Saree Villa 3000 500

Enter the ID: -1
-----
INVALID! Try again.
-----
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 2259
2 T-Shirt Nike 200 40

```

Figure 34 Entered negative value of ID

```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
4      Saree    Villa   3000   500
Enter the ID: -1
-----
INVALID! Try again.
-----
*****ID      Costume Brand  Price  Quantity*****
*****ID      Costume Brand  Price  Quantity*****
1      Shirt     Erike    100    2259
2      T-Shirt   Nike     200    40
3      Khurta   Jompers  1000   400
4      Saree    Villa   3000   500
Enter the ID: 10
-----
INVALID! Try again.
-----
*****ID      Costume Brand  Price  Quantity*****
*****ID      Costume Brand  Price  Quantity*****
1      Shirt     Erike    100    2259
2      T-Shirt   Nike     200    40

```

Ln: 96 Col: 0
8:20 PM 8/25/2022

Figure 35: Entered non-existing value as ID**5.3. Test Three:**

| | |
|-----------------|--|
| TEST No. | THREE |
| OBJECTIVE | To generate the text file after rent |
| ACTION | = Pressed option two rent = Entered the costumer name = Entered 1 as Id of costume = Entered 200 as quantity = Typed No to check out = Bill and Text File are generated |
| EXPECTED RESULT | The text file will be generated. |
| ACTUAL RESULT | The text file has been generated. |
| CONCLUSION | Test was successful. |

Table 4: Test three

```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 2
*****
Please Enter Your Name: Shishir Ghimire
*****
ID      Costume Brand   Price    Quantity
*****
1       Shirt     Erike    100     2259
2       T-Shirt   Nike     200     40
3       Khurta   Jompers  1000    400
4       Saree     Villa    3000    500

Enter the ID: 1
Costume is available
Enter the quantity you want: 200
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to existNO
```

Figure 36: Data are entered

```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help

#####
# Shishir Clothing
#####
----- Billing -----
-----
Customer Name: Shishir Ghimire
Customer ID: 869
    Costume      Brand      Booked Date      Quantity
#####
# Shirt        Erike      2022-08-25          200
#
Total= 40000
#####
***** SHISHIR CLOTHING *****
*****
Select an option
*****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION:

Ln: 61 Col: 30
```

Figure 37: Bill is generated

The screenshot shows a Notepad window titled "COS_Bill869.txt - Notepad". The content of the file is as follows:

```

Shishir Clothing
-----
Bill Details
Name: Shishir Ghimire
Customer ID: 869
-----
Costume    Brand    Book Date   Quantity
-----
Shirt      Erike    2022-08-25  200
-----
total: 40000

```

The Notepad window includes standard menu options (File, Edit, View) and a toolbar at the bottom. The status bar at the bottom right shows "Windows (CRLF)", "UTF-8", "230%", "8:36 PM", and the date "8/25/2022".

Figure 38: Text file is generated

5.4. Test four:

| | |
|-----------------|--|
| TEST No. | FOUR |
| OBJECTIVE | To generate the text file after return |
| ACTION | = Pressed option two rent = Entered the costumer name = Entered 1 as Id of costume = Entered 200 as quantity = Typed No to check out = Bill and Text File are generated |
| EXPECTED RESULT | The text file will be generated. |
| ACTUAL RESULT | The text file has been generated. |
| CONCLUSION | Test was successful. |

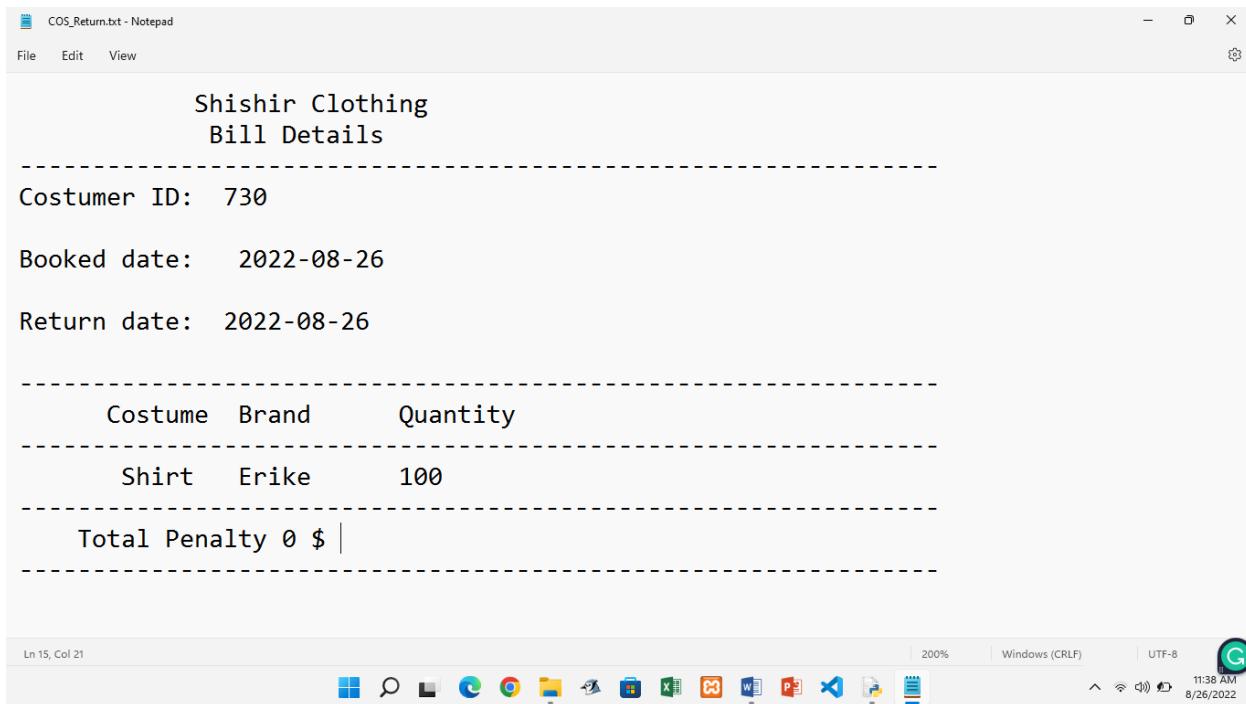
Table 5: Test Four

Figure 39: Return Process

```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help

Thank You!
VISIT AGAIN
*****
Shishir Clothing
Bill Details
*****
Costumer ID: 730
Booked date: 2022-08-26
Return date: 2022-08-26
*****
Costume       Brand       Quantity
*****
Shirt        Erike       100
*****
Total Penalty 0 $
```

Figure 40: Return Process



The screenshot shows a Notepad window titled "COS_Return.txt - Notepad". The content of the file is as follows:

```

Shishir Clothing
Bill Details

Costumer ID: 730

Booked date: 2022-08-26

Return date: 2022-08-26

-----
Costume Brand Quantity
-----
Shirt Erike 100
-----
Total Penalty 0 $ |
-----
```

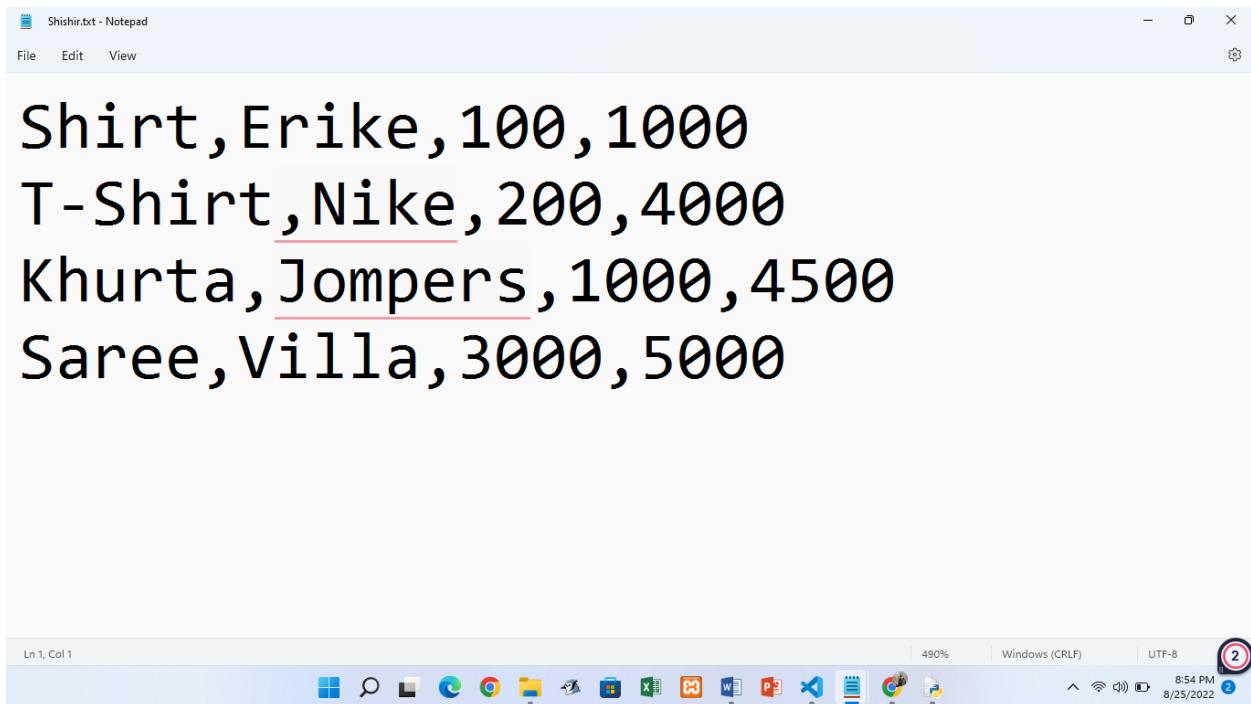
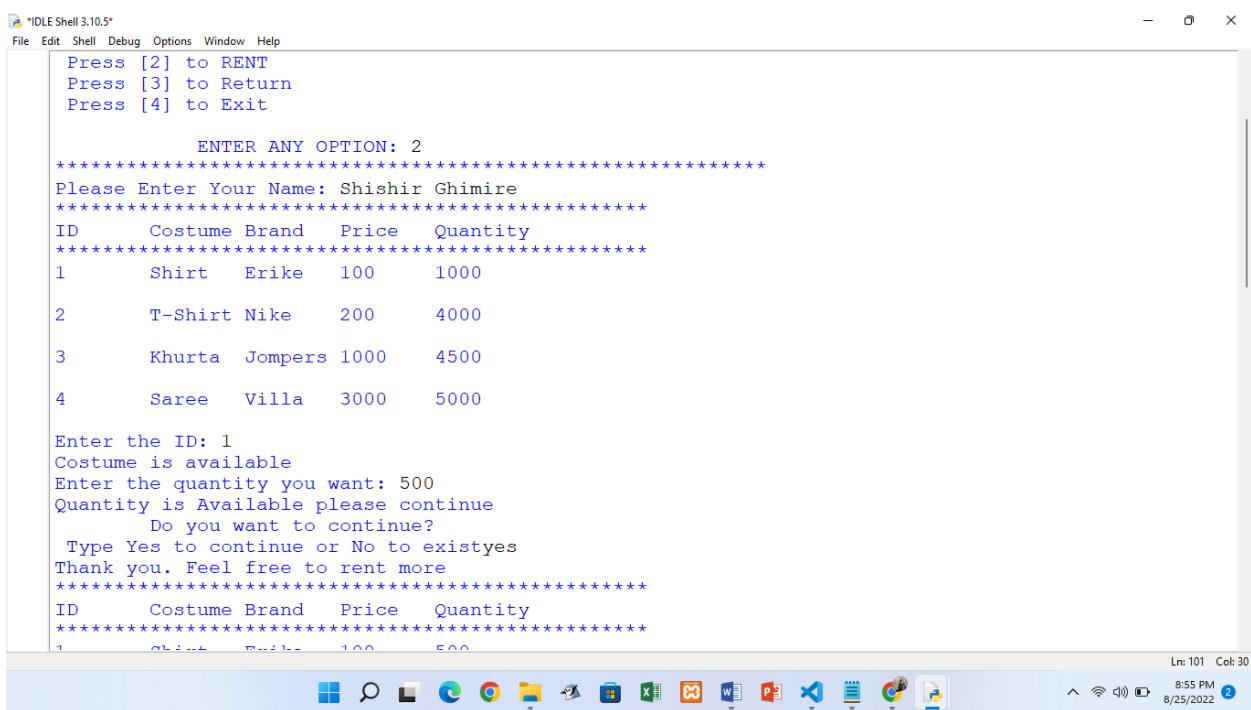
The Notepad window includes standard menu options (File, Edit, View) and a toolbar at the bottom with various icons. The status bar at the bottom right shows "Ln 15, Col 21", "Windows (CRLF)", "UTF-8", "11:38 AM", and "8/26/2022".

Figure 41: Return Bill

5.5. Test Five:

| | |
|-----------------|--|
| TEST No. | FIVE |
| OBJECTIVE | To be able to update the stock |
| ACTION | = Shishir.txt file is opened before renting = Some costume is rented = Shishir.txt file is opened again after renting = Costumes are returned = Shishir.txt file is opened again after returning |
| EXPECTED RESULT | The quantity will be updated. |
| ACTUAL RESULT | The quantity has been updated. |
| CONCLUSION | Test was successful. |

Table 6: Test Five

**Figure 42: Text file before renting****Figure 43: Renting process**

```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 500
2 T-Shirt Nike 200 4000
3 Khurta Jompers 1000 4500
4 Saree Villa 3000 5000

Enter the ID: 2
Costume is available
Enter the quantity you want: 2000
Quantity is Available please continue
Do you want to continue?
Type Yes to continue or No to exist yes
Thank you. Feel free to rent more
*****
ID Costume Brand Price Quantity
*****
1 Shirt Erike 100 500
2 T-Shirt Nike 200 2000
3 Khurta Jompers 1000 4500
4 Saree Villa 3000 5000

Ln: 101 Col: 30

8:55 PM
8/25/2022
```

Figure 44: Renting process

Figure 45: Renting process

```
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help

#####
# Shishir Clothing
#####
-----  

# Billing
-----
Customer Name: Shishir Ghimire
Customer ID: 63
    Costume      Brand      Booked Date      Quantity
#####
# Shirt        Erike       2022-08-25        500
# T-Shirt      Nike        2022-08-25        2000
# Saree        Villa       2022-08-25        1000
#
Total= 6900000
#####
***** SHISHIR CLOTHING *****
Select an option
*****
Press [1] to VIEW
-----
```

Figure 46: Renting process

Figure 47: Text file after renting



```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
***** Select an option *****
Press [1] to VIEW
Press [2] to RENT
Press [3] to Return
Press [4] to Exit

ENTER ANY OPTION: 3
*****
----- Enter Costumer ID: 63 -----
1: Shirt

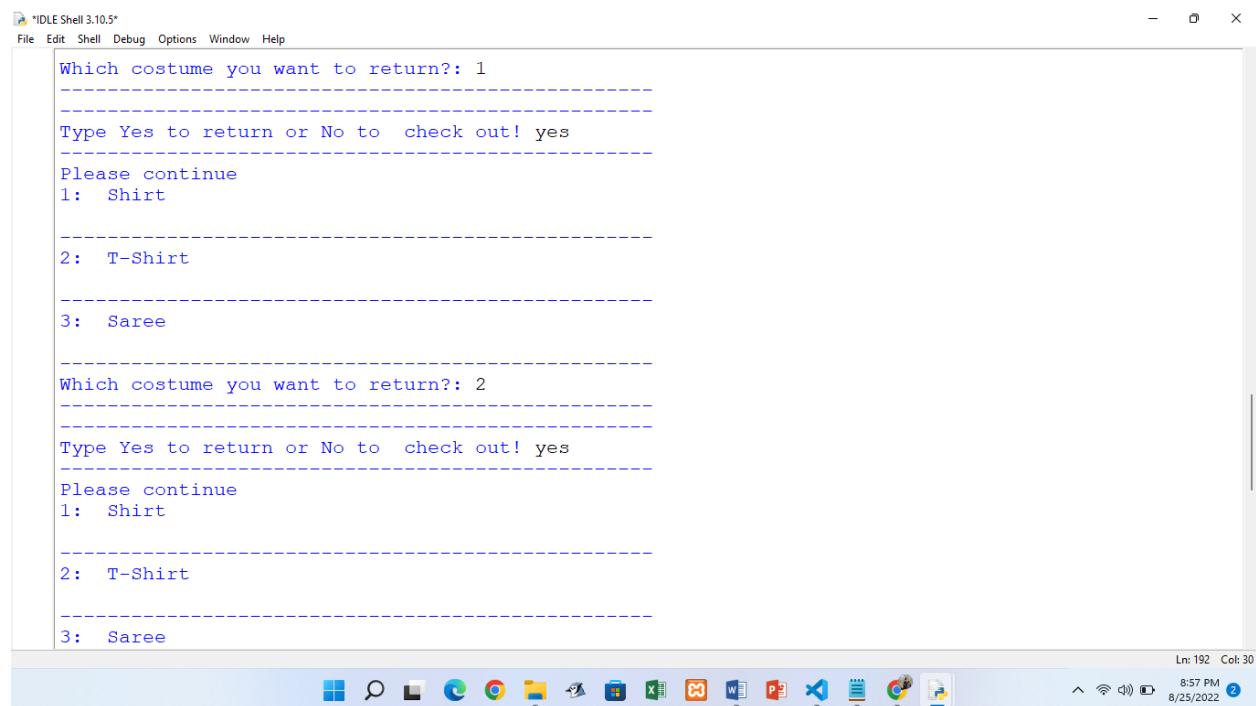
-----
2: T-Shirt

-----
3: Saree

-----
Which costume you want to return?: 1
-----
Type Yes to return or No to check out! yes
-----
Please continue
1: Shirt

```

Ln: 150 Col: 60
8:56 PM 8/25/2022

Figure 48: Return Process


```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
----- Which costume you want to return?: 1 -----
----- Type Yes to return or No to check out! yes -----
----- Please continue -----
1: Shirt

-----
2: T-Shirt

-----
3: Saree

-----
Which costume you want to return?: 2
-----
Type Yes to return or No to check out! yes
-----
Please continue
1: Shirt

-----
2: T-Shirt

-----
3: Saree

```

Ln: 192 Col: 30
8:57 PM 8/25/2022

Figure 49: Return Process



```

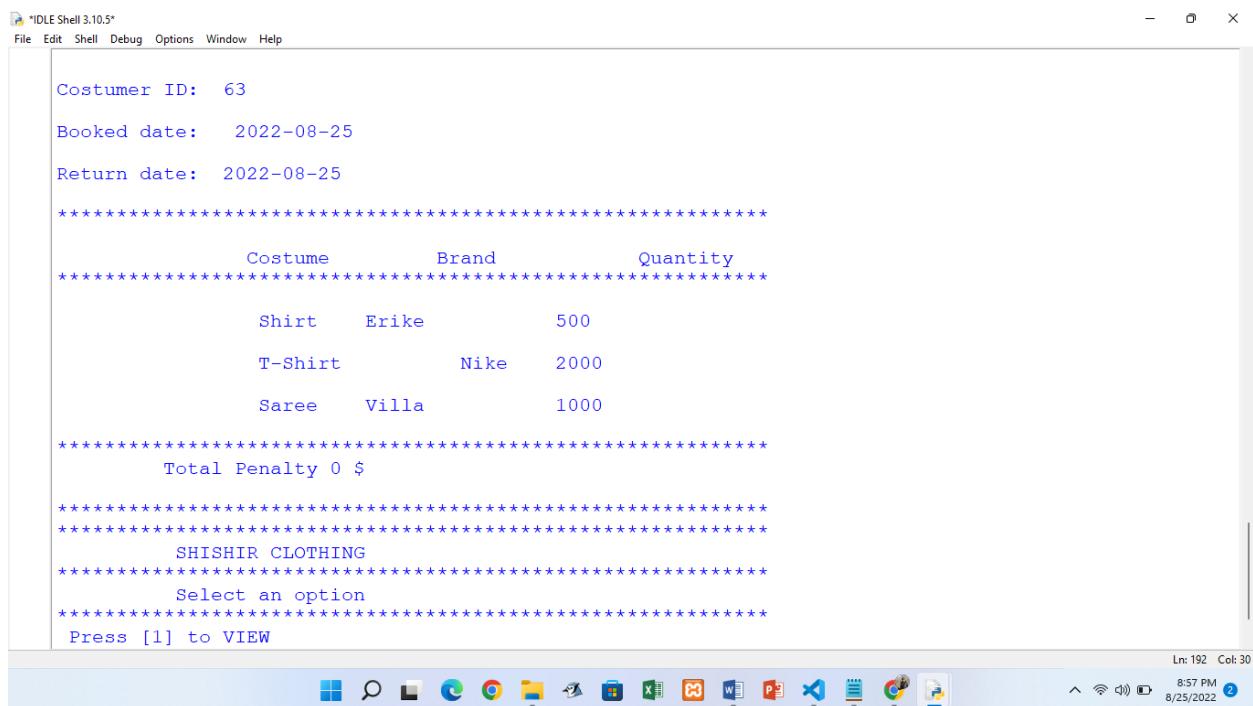
*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
E. Saree

-----
3: Saree

-----
Which costume you want to return?: 3
-----
Type Yes to return or No to check out! no
*****
-----
Thank You!
VISIT AGAIN
*****
Shishir Clothing
Bill Details
*****
Costumer ID: 63
Booked date: 2022-08-25
Return date: 2022-08-25
*****

```

Ln: 192 Col: 30
8:57 PM 8/25/2022

Figure 50: Return Process


```

*IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Costumer ID: 63
Booked date: 2022-08-25
Return date: 2022-08-25
*****
Costume      Brand      Quantity
*****
Shirt        Erike      500
T-Shirt       Nike       2000
Saree        Villa      1000
*****
Total Penalty 0 $
*****
SHISHIR CLOTHING
*****
Select an option
*****
Press [1] to VIEW

```

Ln: 192 Col: 30
8:57 PM 8/25/2022

Figure 51: Return Process

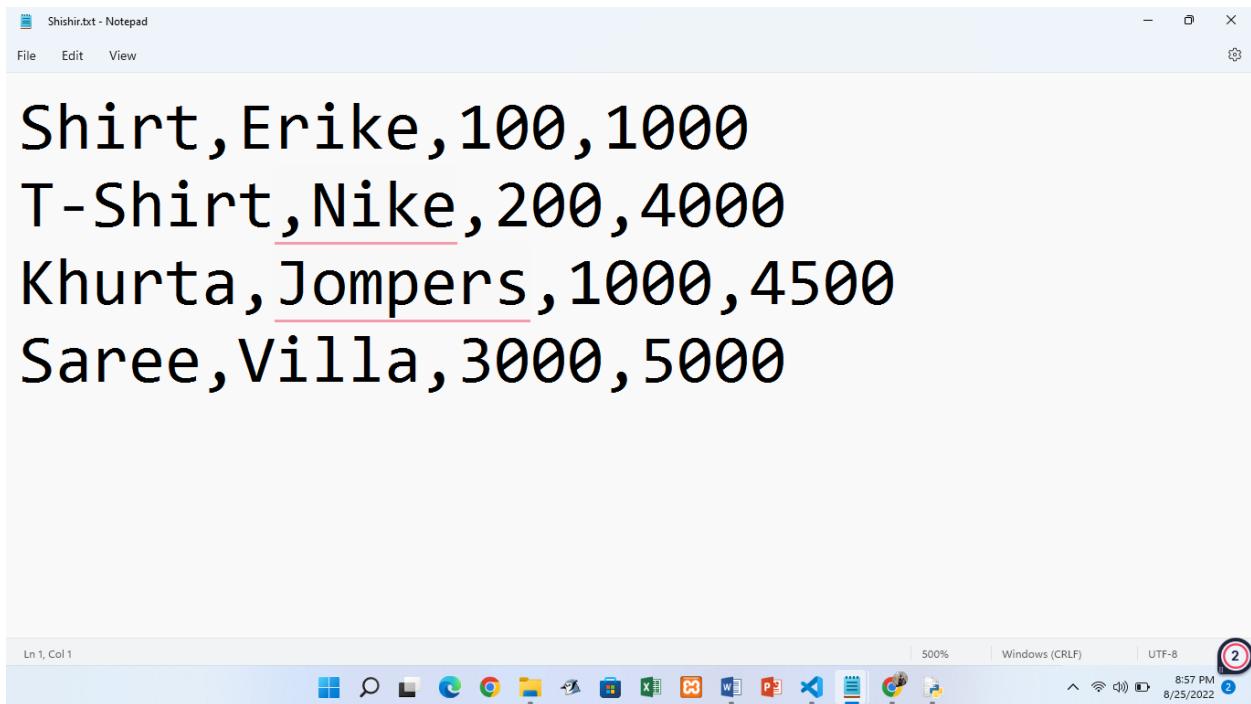


Figure 52: Text File after returning

6. Conclusion:

This was our first coursework for Fundamental of Computing and one of the most challenging coursework we got in this semester. In order to complete this coursework in time, I had to push myself much harder as I had to do lots of research and revision of the classes from lectures, tutorials, and workshops. I had to take help from my teachers and fellow friends in order to complete this task on time. One of the challenging parts of this coursework was to complete the development and documentation part of this coursework ahead of the given deadline. Time wasn't sufficient as I had to do three coursework from different modules simultaneously.

The question had so less detail to understand but I would like to thank my module teachers for briefing the question again and again which helped us to understand the given question in a broad way. I watched lots of YouTube videos and read many websites to have a better understanding of file handling topics. Not only coursework but writing a report was also challenging this time and took a lot of time to complete. To complete the documentation part, I had to implement my knowledge about the algorithm, flowcharts, and pseudocode. Developing an algorithm and flowchart helped me while writing the code for the system. I was basically new in this module as I had never done python programming before I had tons of challenges to complete this coursework. Unlike most of the students who had worked on python, I was one of those few who didn't do that before.

After completing this coursework, I had knowledge about loops and conditions, file handling, and so on. I am very optimistic that the knowledge I got from this coursework will help me to build a better future and I will be able to use my knowledge to be a better programmer. The experience I got from this coursework is a big deal for me and doing this coursework was a quite fun experience too.

Bibliography

(n.d.). Retrieved from web3school: <https://www.w3schools.com/>

datacamp. (2017, Dec 8). Retrieved from datacamp:

<https://www.datacamp.com/tutorial/data-structures-python>

Geeks for Geeks. (2022, August 01). Retrieved from

<https://www.geeksforgeeks.org/python-data-structures/>

APPENDIX:**FOR main.py:**

```
import sys
def main():
    while True:
        print("****60)
        print("      SHISHIR CLOTHING")
        print("****60)
        print("      Select an option")
        print("****60)
        print(" Press [1] to VIEW \n Press [2] to RENT \n Press [3] to Return \n Press [4]
to Exit")
        a=int(input(""
ENTER ANY OPTION: ""))
        print("****60)
        if a==1:
            import View
        elif a==2:
            import Rent
        elif a==3:
            import Return
        elif a==4:
            sys.exit("Thank You For trusting us")
        else:
            print("Invalid input")
    main()
```

FOR View.py

```
dic_costume={}

def display():

    c=0

    print("*****50)

    print("ID \tCostume\tBrand \tPrice \tQuantity")
    print("*****50)

    file=open("Shishir.txt", "r")

    costume_book= file.read()

    costume_book=costume_book.split("\n")



    while("") in costume_book):

        costume_book.remove("")

    for i in range (len(costume_book)):

        c=c+1

        dic_costume[c]=costume_book[i].split(",")





    for key,value in dic_costume.items():

        print(key,end="\t")

        for j in value:

            print(j,end="\t")

        print("\n")

display()
```

FOR Rent.py

```
from datetime import date
import random
import sys

today = date.today()
day=date.today().day
ID= random.randint(1,1000)
dic_costume={}
custome_renting=[]
Total=[]
list00=[]
list01=[]
renting=[]

def display():
    #count = 0
    c=0
    print("*****50)
    print("ID \tCostume\tBrand \tPrice \tQuantity")
    print("*****50)
    file=open("Shishir.txt", "r")
    costume_book= file.read()
    costume_book=costume_book.split("\n")
```

```
while("") in costume_book):
    costume_book.remove("")

for i in range (len(costume_book)):
    c=c+1
    dic_costume[c]=costume_book[i].split(",")

for key,value in dic_costume.items():
    print(key,end="\t")
    for j in value:
        print(j,end="\t")
    print("\n")

update_stock()
Valid(c)

Customer_Name=input("Please Enter Your Name: ")

def Valid(c):
    c=c

try:
    RentID=int(input("Enter the ID: "))
except ValueError:
```

```
print("Please, Enter numeric value only")
display()
if (RentID>0 and RentID<=c):
    print("Costume is available")
try:
    qty=int(input("Enter the quantity you want: "))
except ValueError:
    print("Invalid! Enter numeric value only")
    display()
costume_select=dic_costume[RentID]

if qty<=int(costume_select[3]):
    print("Quantity is Available please continue")
    #Updating the stocks
    update_qty=int(costume_select[3])-qty
    costume_select[3]=str(update_qty)
    update_stock()
    abcd=[str(costume_select[0]),str(costume_select[1]),str(today),str(qty)]
    abcd = str(abcd)[1:-1]
    abcd=abcd.replace("\n", " \t")
    abcd=abcd.replace(","," ")
    custome_renting.append(abcd)

total=int(costume_select[2].rstrip("$"))*qty
```

```
Total.append(total)
```

```
xyz=[str(costume_select[0]),str(costume_select[1]),str(today),str(qty)]
```

```
xyz = str(xyz)[1:-1]
```

```
xyz=xyz.replace("", "\t")
```

```
xyz=xyz.replace(", "")
```

```
list00.append(xyz)
```

```
Nam=[str(Customer_Name), str(ID)]
```

```
Nam= str(Nam)[1:-1]
```

```
Nam=Nam.replace("", "")
```

```
Nam=Nam.replace(", ", "\t ")
```

```
list01.append(Nam)
```

```
total=int(costume_select[2].rstrip("$"))*qty
```

```
Total.append(total)
```

```
mnop=[str(costume_select[0]),str(costume_select[1]),str(today),str(qty),str(RentID),str(day)]
```

```
mnop = str(mnop)[1:-1]
mnop=mnop.replace("", "")
renting.append(mnop)

write_two()

Nam=[str(Customer_Name), str(ID)]
Nam= str(Nam)[1:-1]
Nam=Nam.replace("", "")
Nam=Nam.replace(",","\t\t")

list01.append(Nam)

write()
```

```
billing = (input("      Do you want to continue? \n Type Yes to continue or No to
exist")).lower()
```

```
if billing=="no":
    print("\n")
    print("\t",("#"*90))
    print("\t\t\t\t\t", "Shishir Clothing")
    print("\t",("#"*90))
    print("-"*120)
```

```
print("\t\t\t\t", "Billing")
print("-"*120)
print("\t","Customer Name: ",Customer_Name)
print("\t","Customer ID: ",ID)
print("\t","\t""Costume","\t""Brand","\t"\t""Booked Date","\t""  " "Quantity")
print("\t","#"*90)
for item in custome_renting:
    print("\t",item,end="\t")
    print("\n")
    print('\t',"Total= ",sum(Total))
    print("\t","#"*90)

elif billing=="yes":
    print("Thank you. Feel free to rent more")

display()
else:
    print("-"*50)
    print("Sorry, Costume is out of stock.")
    print("-"*50)
    display()
else:
    print("-"*50)
    print("INVALID! Try again.")
```

```
print("-"*50)
display()

def write_two():
    #Creating the auto generating text file to store costumer data
    ghi=open('COSTUMER'+str(ID)+".txt", 'w')
    for aaaa in renting:
        ghi.write(str(aaaa))
        ghi.write("\n")
    ghi.close()

def write():
    cc=open('COS_Bill'+str(ID)+".txt",'w')
    cc.write("-"*60)
    cc.write("\n")
    cc.write("\t Shishir Clothing")
    cc.write("\n")
    cc.write("-"*60)
    cc.write("\n")
    cc.write("\t Bill Details")
    cc.write("\n")
    cc.write("Name: ")
    cc.write(Customer_Name)
    cc.write("\n")
    cc.write("Costomer ID: ")
    cc.write(str(ID))
    cc.write("\n")
```

```
cc.write("-"*60)
cc.write("\n")
cc.write("Costume")
cc.write("\t Brand")
cc.write("\t Book Date")
cc.write("\t Quantity")
cc.write("\n")
cc.write("-"*60)
cc.write("\n")
for dd in list00:
    cc.write(str(dd))
    cc.write("\n")
    cc.write("\n")
    cc.write("-"*60)
    cc.write("\n")
    cc.write("total: ")
    cc.write(str(sum(Total)))
    cc.close()
def update_stock():
    oo=open("Shishir.txt","w")
    for key,value in dic_costume.items():
        oo.write(",".join(value))
        oo.write("\n")
    oo.close()
display()
```

FOR Return.py

```
from datetime import date

return_date=date.today()

return_day=date.today().day

#Creating the dictionary

dic_return={}

dic_costume={}

#Creating the list

billing=[]

print("-"*50)
```

```
Costumer_id=input('Enter Costumer ID: ')
```

```
print("-"*50)
```

```
def return_dic():
```

c=0

```
file=open("Shishir.txt","r")
```

```
return_data= file.read()
```

```
return_data=return_data.split("\n")
```

```
while("") in return data):
```

```
return data.remove("")
```

```
for i in range (len(return_data)):
```

```
c=c+1  
dic_costume[c]=return_data[i].split(",")  
  
counter_two=0  
file=open('COSTUMER'+str(Costumer_id)+".txt","r")  
returing_data= file.read()  
returing_data=returing_data.split("\n")  
while("") in returing_data:  
    returing_data.remove("")  
for i in range (len(returing_data)):  
    counter_two=counter_two+1  
    dic_return[counter_two]=returing_data[i].split(",")  
Valid_data(counter_two)  
update_stock()  
#Validating the data  
  
def Valid_data(counter_two):  
    if counter_two==1:  
        identity_1=1  
    else:  
        for key,value in dic_return.items():  
            print(key,end=": ")  
            for j in value[:1]:  
                print(j,end="\t")  
            print("\n")
```

```
print("-"*50)

identity_1= input("Which costume you want to return?: ")

print("-"*50)

selc_return=dic_return[int(identity_1)]

qty=int(selc_return[3])


list0002=[selc_return[0],selc_return[1],selc_return[3]]

list0002=str(list0002)[1:-1]

list0002=list0002.replace("", "")

list0002=list0002.replace(", ", "\t")

billing.append(list0002)

return_slec=dic_costume[int(selc_return[4])]

UQDTY=int(return_slec[3])+qty

return_slec[3]=str(UQDTY)

update_stock()

print("-"*50)

Sure=input("Type Yes to return or No to check out! ").lower()

print("-"*50)

if(Sure=="no"):

    update_stock()

    print("*"*60, "\n")

    print("Thank You! \n VISIT AGAIN")

    print("*"*60, "\n")

    print("\t \t \t \t Shishir Clothing")

    print("\t \t \t \t Bill Details", "\n")
```

```
print("""*60,"\\n")
print("Costumer ID: " , Costumer_id,"\\n")
print("Booked date: " , selc_return[2],"\\n")
print("Return date: ",return_date,"\\n")
print("""*60,"\\n")
print("\\t","\\t""Costume","\\t""Brand","\\t","\\t" " "Quantity")
print("""*60,"\\n")
for item in billing:
    print("\\t \\t",item,end="\\\t\\\t")
    print("\\n")
print("""*60)
day=int(selc_return[5])
LATE=(day+5)-return_day
if return_day<day+5:
    total=0
else:
    LATE*5
print("\\t Total Penalty",total,"$","\\n")
print("""*60)
elif Sure=="yes" :
    #Printing text to appealing to continue
    print("Please continue")
    #Returning to start returning again
    return_dic()
else:
```

```
print("Invalid Input")  
return_dic()  
  
def update_stock():  
    abc=open("Shishir.txt","w")  
  
    for key,value in dic_costume.items():  
        abc.write(",".join(value))  
        abc.write("\n")  
  
    abc.close()  
  
    return_dic()
```

ORIGINALITY TEST:

8/26/22, 12:15 PM 22015777 Shishir Ghimire (1)

Originality report

COURSE NAME
CS4051NI - Fundamentals of Computing

STUDENT NAME
Shishir Ghimire Computing

FILE NAME
22015777 Shishir Ghimire (1)

REPORT CREATED
26 Aug 2022

Summary

| | | |
|-----------------------|---|------|
| Flagged passages | 1 | 0.8% |
| Cited/quoted passages | 0 | 0% |
| Web matches | | |
| coursehero.com | 1 | 0.8% |

My Drive - Google Drive 22015777 Shishir Ghimire (1) - G... FOC - Coursework (Originality C... 22015777 Shishir Ghimire (1) +

classroom.google.com/g/sr/NDk1ODE3NjgzNTM0/NDk3MDQwMTAxMzIz/1ngGJz125K7lMio_54h82-DjbUCXQRXsR3kAgMffjaVk

22015777 Shishir Ghimire (1)

Module Code & Module Title
CS4051NI Fundamentals of Computing

Assessment Weightage & Type
60% Individual Coursework

Year and Semester
2021-22 Summer

Student Name: Shishir Ghimire
Group: L1C15
London Met ID: 22015777
College ID: NP01CP45220018
Assignment Due Date: 2022-08-26
Assignment Submission Date: 2022-08-26

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted. I am fully aware that late submissions will be treated as non-submission and no marks will be awarded.

Count %

0.8% flagged content
0% cited or quoted content

Web matches (0.8%)
coursehero.com (0.8%)

Summary

Originality report expires on 10 Oct 2022

12:15 PM 8/26/2022