LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

## Module Code & Module Title

## CS4059NI Fundamentals of Computing

## Assessment Weightage & Type

## 60% Individual Coursework

## Year and Semester

## 2021-22 Summer

**Student Name: Sharams Kunwar**

**Group: N3**

**London Met ID: 21049701**

**College ID: NP01NT4A210112**

**Assignment Due Date: 26th August 2022**

**Assignment Submission Date: 26th August 2022**

*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.*

# ACKNOWLEDGEMENT

I would like to express my deepest gratitude to the Islington College and the module teacher Mr. Hrishav Tandukar for assisting me in accomplishment of the coursework in the given time frame. Likewise, I would also like to convey my sincere appreciation to all the involved helping hands.

The guidance from lecturer Mr. Tandukar led to accomplishment of the coursework. I would like to greatly thank him for assisting me day and night to solve my errors and helping me to conduct researches on the topic.

At last, I would like to thank my friends and family and also seniors who helped me clear my confusions and solve problems I faced throughout the completion of the project.

Sincerely Yours,

Sharams Kunwar.

## Table of Contents

## List of Figures

## List of Tables:

# 1. INTRODUCTION

An application for a Costume Rental system had to be created using Python Programming language, as per the assigned coursework. The task was quite challenging and difficult as it was a greater leap from learning the fundamentals of the language to creating a fully functional application using it. The coursework had to be submitted by the end of week 12 of the module and also had to be well-explained using flowcharts, algorithm and properly written pseudocode.

*Figure 1 Python (Python, 2022)*

Many research and studies had to be done taking in regards the complexity of the coursework. Since, we had only begun with the fundamentals, completing the application was quite challenging as it replicated a real-time software which could be used in many stores. Studies were done to overcome the challenges of the coursework. To accomplish the final goal of the coursework within the deadline, the provide lecture slides were revised, and the module teachers were also consulted for further assistance.

## 1.1 Brief overview of Report:

An application with a costume rental system was built using Python programming language which would be applicable to use in real-time stores. The program would run in the shell. The user would be welcomed with a display message and the users of the program would be prompted if they desired to whether rent or return the costume. If the user selected an option to rent, a list of the costumes available for rent along with their respective prices and quantities would be displayed. The text file that serves as a source code is updated and in the process of updating it, a track of the details of the rented costume is kept and inserted in a new text file by the program. Similarly, on returning the costume, the text file used on the renting of the costume is taken in use and is simultaneously updated. Thus, a program which operates in accordance with the user’s preference is created which is the need of the coursework.

## 1.2 Goals and Objective:

The goal of the coursework is to create an application software which would represent a fully functional Costume Rental System that can be used in real-time in numerous stores around the globe. Similarly, final goal was achieved meeting several objectives which are as follows:

- To create a costume rental system using Python Programming language.
- To write the program with the right data structure and logic.
- To represent the running of the application effectively using flowcharts, pseudocode and algorithm.
- To test the program to discover faults in the program.

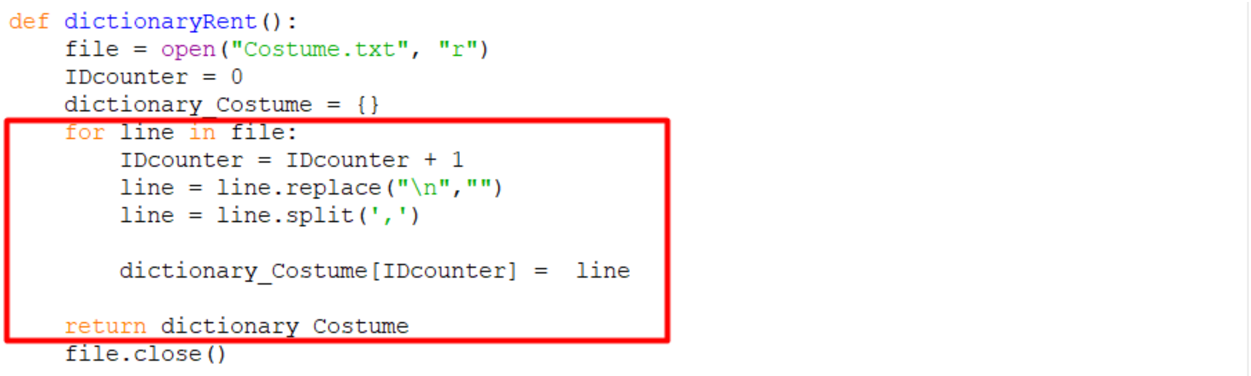
A text file was created which would enact as a source code for data and similarly, a python file to access the text file was also created. The data from text file was transferred to Python file using number of complex operations like creation of several loops and data structures. The obstacles were solved with assistance of module teacher and with properly researching the solutions. Finally, a program was created which functioned as per requirement.

## 2. Loop

Loop is a set of instructions used to execute a block of statements repeatedly until a given requirement is met or until satisfaction of a given condition (GeeksforGeeks, 2022). Throughout the coursework, different kinds of loops are taken in use. Few of them are:

### 2.1 FOR Loop

A for loop is taken in use for iterating over a sequence. With the help of it, a set of statements can be executed once for each item in a list, tuple, set, etc (w3schools, 2022).

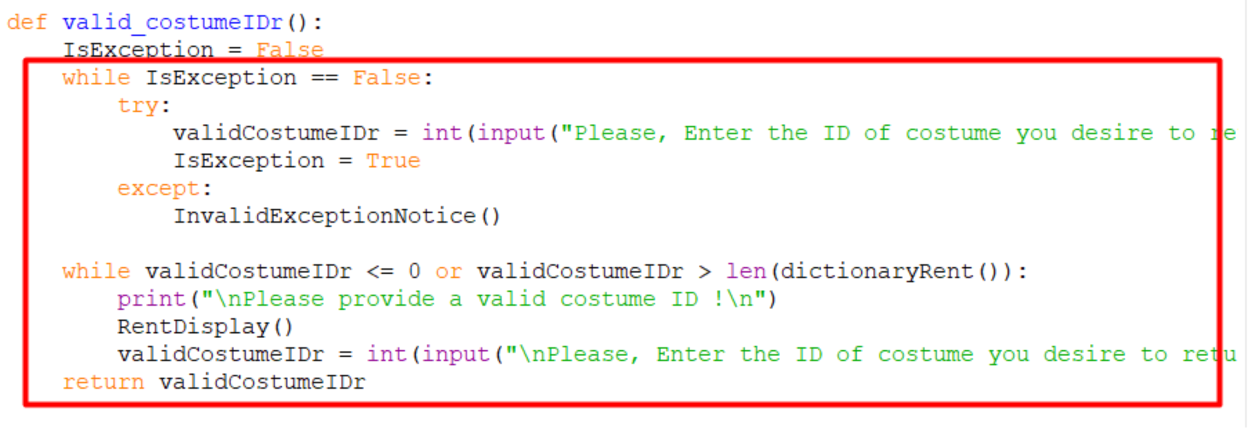


```
def dictionaryRent():
    file = open("Costume.txt", "r")
    IDcounter = 0
    dictionary_Costume = {}
    for line in file:
        IDcounter = IDcounter + 1
        line = line.replace("\n","")
        line = line.split(',')

        dictionary_Costume[IDcounter] =  line

    return dictionary_Costume
    file.close()
```

*Figure 2 Use of For loop*

### 2.2 WHILE Loop

A while loop is taken in use for iterating over a block of code as long as the condition is true. It is taken is use when the number of times of iteration is not known beforehand (Programiz, 2022).

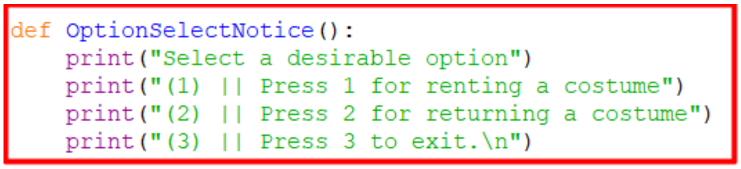


```
def valid_costumeIDr():
    IsException = False
    while IsException == False:
        try:
            validCostumeIDr = int(input("Please, Enter the ID of costume you desire to re
            IsException = True
        except:
            InvalidExceptionNotice()

    while validCostumeIDr <= 0 or validCostumeIDr > len(dictionaryRent()):
        print("\nPlease provide a valid costume ID !\n")
        RentDisplay()
        validCostumeIDr = int(input("\nPlease, Enter the ID of costume you desire to retu
    return validCostumeIDr
```

*Figure 3 Use of While loop*

## 2.3 Function

A function is a block of code which is destined to perform a single, related action. Python itself gives a built-in function but one can create their own functions as well (TutorialsPoint, 2022).



```
def OptionSelectNotice():
    print("Select a desirable option")
    print("(1) || Press 1 for renting a costume")
    print("(2) || Press 2 for returning a costume")
    print("(3) || Press 3 to exit.\n")
```

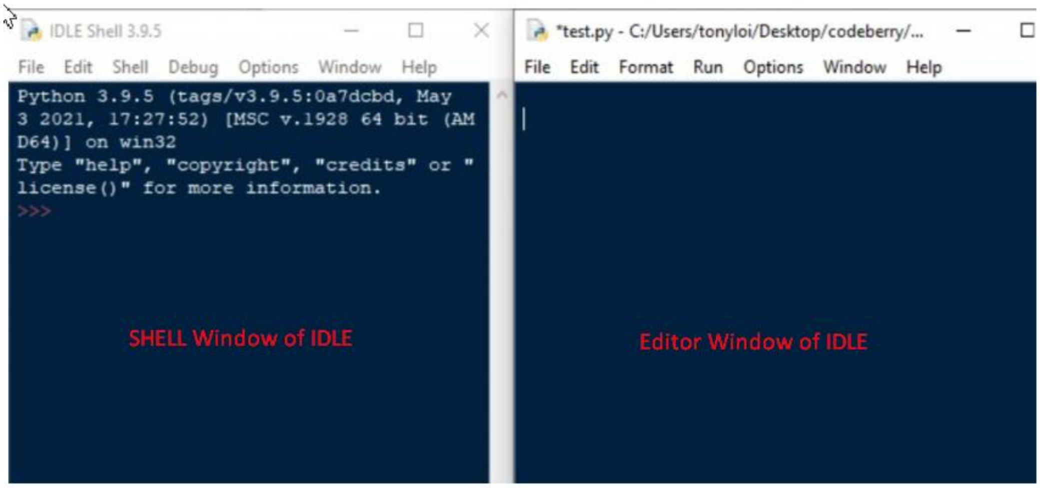
*Figure 4 Use of Function*

# 3. Discussion and Analysis

As per the coursework, a Costume Rental System had to be built and also representing algorithm, flowchart, pseudocode and also the program with the help of which system had been built had to be tested. The program had been developed using Python Programming Language in IDLE. Similarly, application like Draw.io was used to create flowchart that would represent the flow of the program. Likewise, notepad was used to create .txt file and generate bill details as well.

- IDLE Python

IDLE is Integrated Development and Learning Environment of Python Programming Language (python, 2022). In this coursework, IDLE was used to write the program in its editor window and run the program in its Shell window where the user would be able to interact with the program.



```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May
3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "
license()" for more information.
>>>
```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
*test.py - C:/Users/tonyloi/Desktop/codeberry/...
File Edit Format Run Options Window Help
SHELL Window of IDLE
Editor Window of IDLE

*Figure 5 IDLE and its Windows (CodeBerry, 2022)*

- Draw.io

Draw.io is a software used for making diagrams and charts. It was designed by Seibert Media. In this coursework, it is used for creating flowchart of the program.

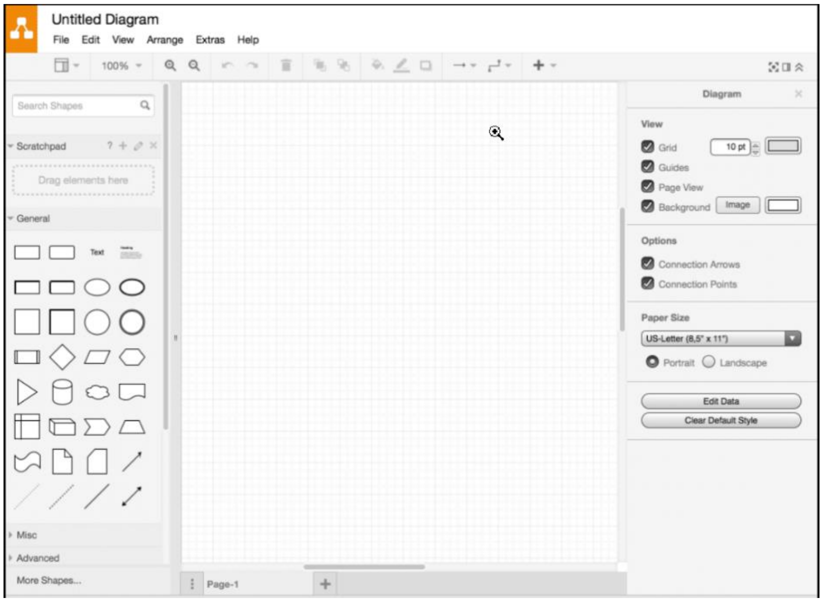
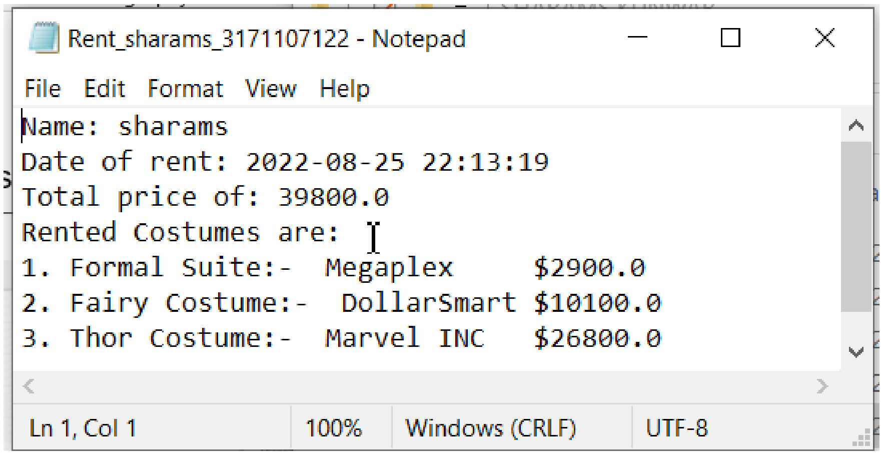


*Figure 6 Interface of Draw.io*

- Notepad

Notepad is a text editor used in editing plain text. In this coursework, it is used to create stock file and details of rent and return bill (ComputerHope, 2022).



```
Name: sharams
Date of rent: 2022-08-25 22:13:19
Total price of: 39800.0
Rented Costumes are:
1. Formal Suite:-  Megaplex     $2900.0
2. Fairy Costume:-  DollarSmart $10100.0
3. Thor Costume:-  Marvel INC   $26800.0
```

*Figure 7 Rent Bill Creation in Notepad*

Similarly, Algorithm, Flowchart and PseudoCode were created to represent the flow of the program.

## 3.1 Algorithm

Algorithms are a block of instructions which are executed to discover a solution to the problem. They are not language-specific and can be implemented in various programming languages as well. There are no specific guidelines to write algorithm (Sharma, 2022).

In the coursework, algorithm has been written to represent the program used in creation of Costume Rental System, as per the requirement of the coursework.

```
STEP 1: Start
STEP 2: Display WelcomeNotice
STEP 3: Display OptionSelectNotice
STEP 4: Select a desirable option
STEP 5: If n is 1 go to STEP 7 else if n is 2 go to next STEP 18 else if n is 3 go to STEP 27 else, go to STEP 28
STEP 6: Read Costume.txt file
STEP 7: Display all the costume available for renting in a table
STEP 8: Ask user to Enter the CostumeID and store in valid CostumeID
STEP 9: IF the inputted valid CostumeID by the user is valid then move to STEP 10 else return to STEP 8
STEP 10: IF quantity is less than 0, display UnavailableCostume message and then return to STEP 3 else move to STEP 11
STEP 11: Display AvailableCostume message
STEP 12: Ask user to Enter the quantity and store in quantityCostume
STEP 13: IF quantityCostume less than or equals to 0 or quantityCostume is greater than stockquantity then return to STEP 12 else move to STEP 14.
STEP 14: Update quantity of costume in the dictionary
STEP 15: Ask user if they want to rent another costume
STEP 16: IF user’s input in contrent is y then Move to STEP 7 and if the user input is anything else move to STEP 17.
```

STEP 17: Generate a rent bill in console and write a bill in txt file with a unique name and display OptionSelectNotice

STEP 18: Read Costume.txt file

STEP 19: Display all the costume available for returning in a table

STEP 20: Ask user to Enter the CostumeID and store in valid CostumeIDr

STEP 21: IF the inputted valid CostumeIDr by the user is valid then move to STEP 22 else return to STEP 20

STEP 22: Ask user to Enter the quantity and store in quantityCostume

STEP 23: Update quantity of costume in the dictionary

STEP 24: Ask user if they want to return another costume

STEP 25: IF user input in contrent is ‘y’ then Move to STEP 19 and if the user input is anything else move to STEP 26.

STEP 26: Generate a return bill in console and write a bill in txt file with a unique name and display OptionSelectNotice

STEP 27: ThankyouNotice

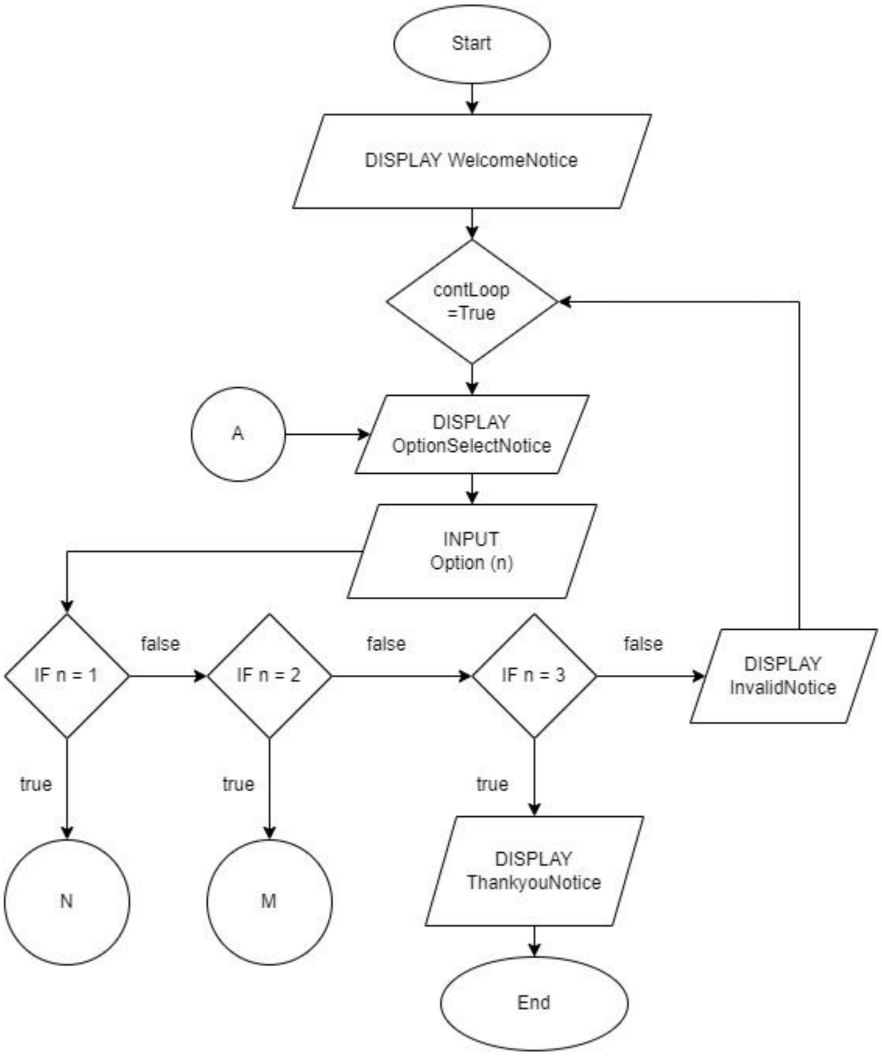
STEP 28: InvalidNotice

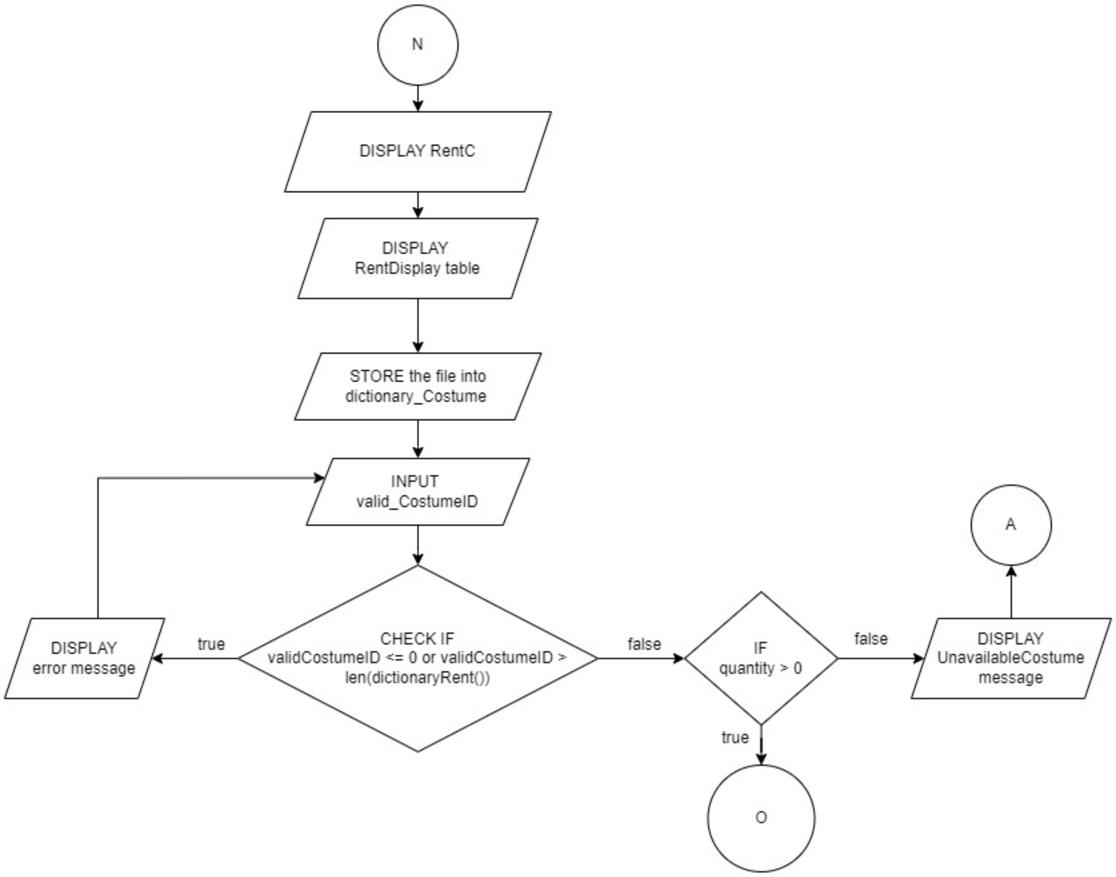
STEP 29: End

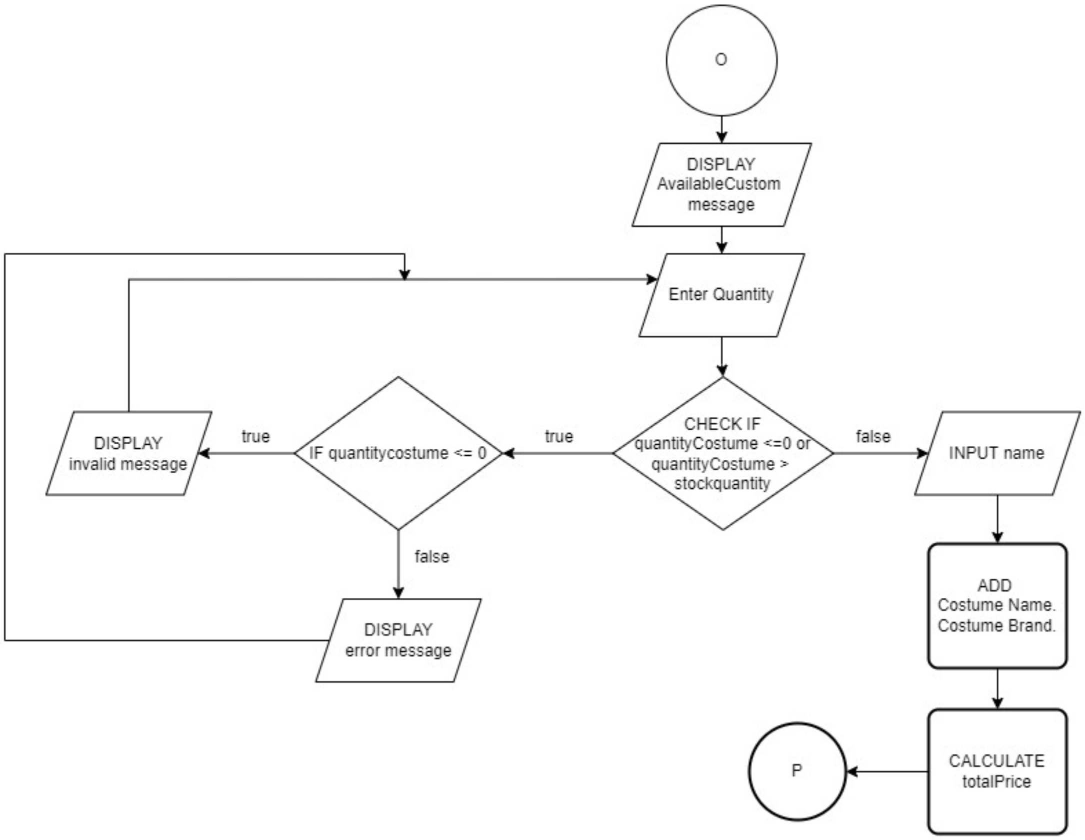
## 3.2 Flowchart

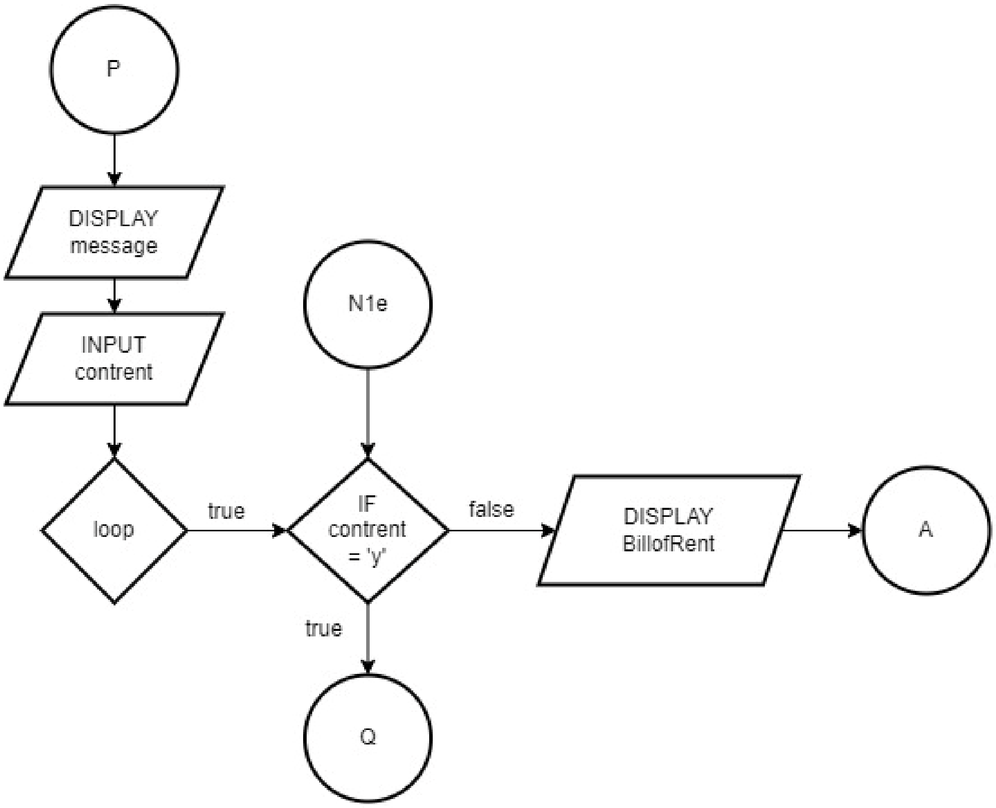
A flowchart is a diagrammatical representation of steps in a sequential order to represent the flow of a program (asq, 2022).

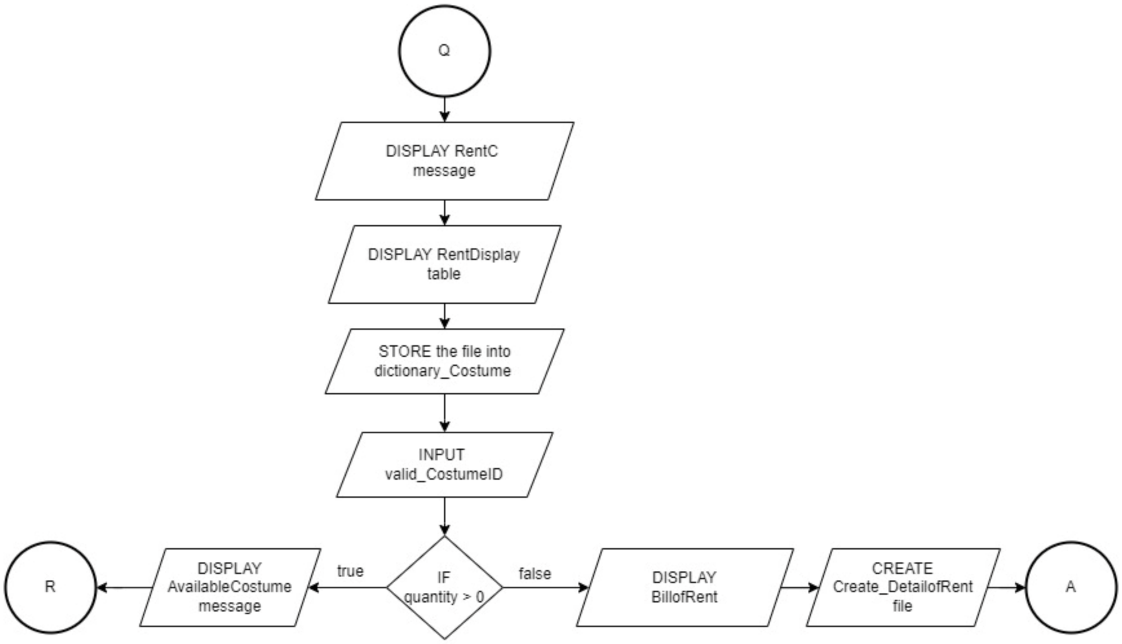
In the coursework too, flowchart has been developed to represent the flow of program used in Costume Rental System.

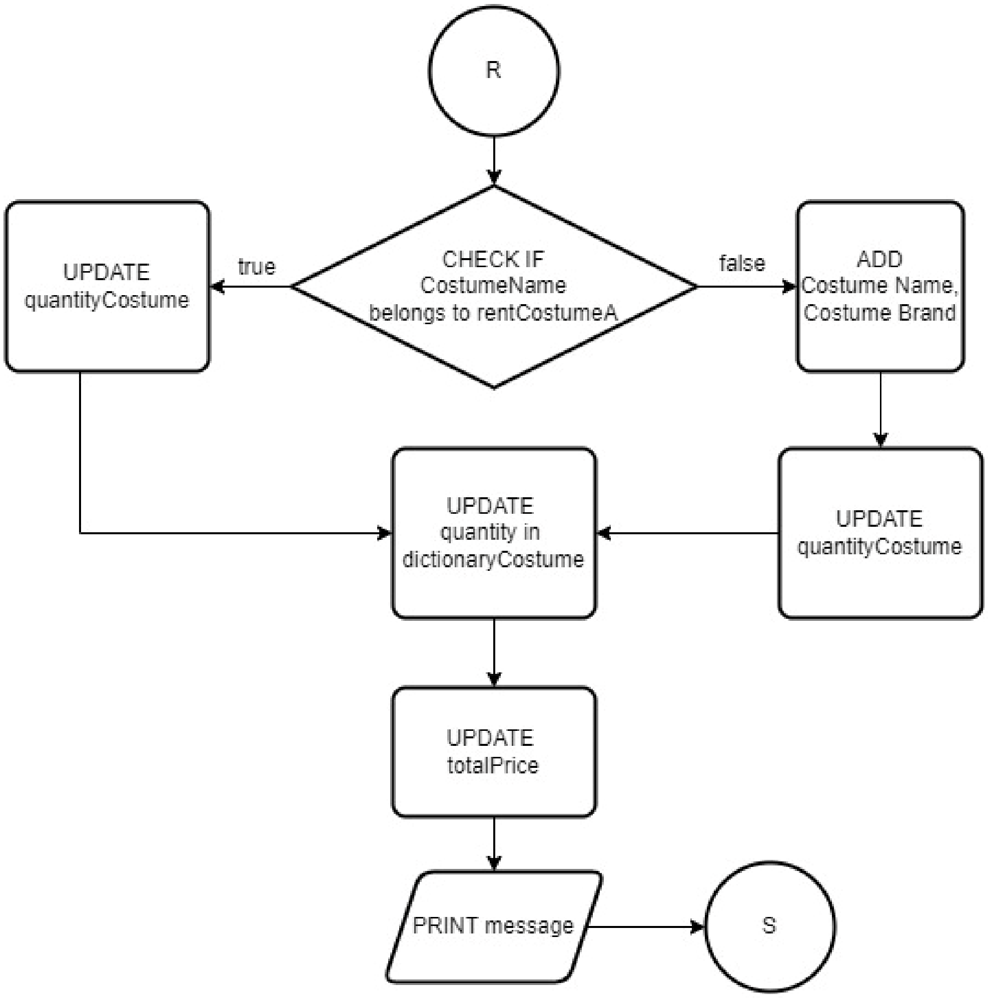


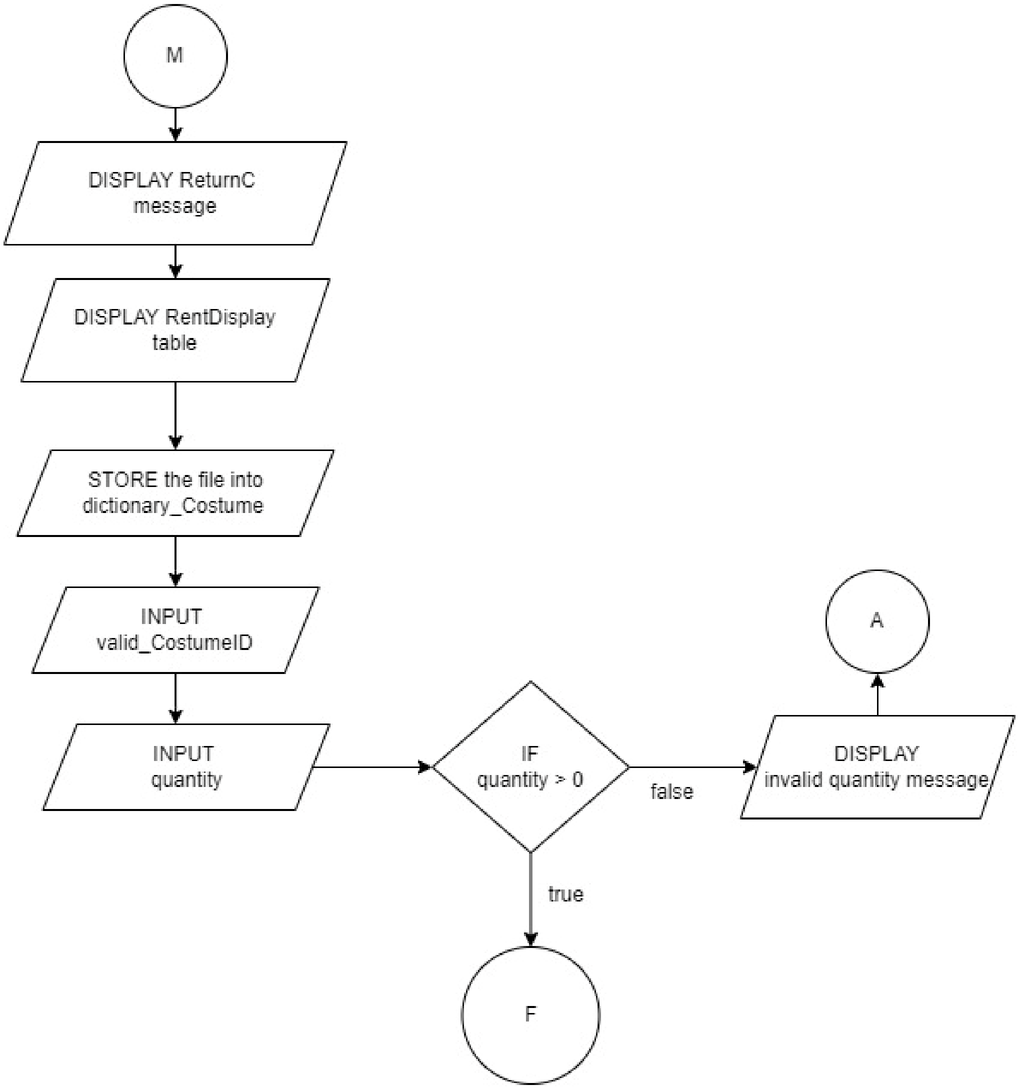

N
DISPLAY RentC
DISPLAY
RentDisplay table
STORE the file into
dictionary_Costume
INPUT
valid_CostumeID
CHECK IF
validCostumeID <= 0 or validCostumeID >
len(dictionaryRent())
true
DISPLAY
error message
false
IF
quantity > 0
false
DISPLAY
UnavailableCostume
message
A
true
O

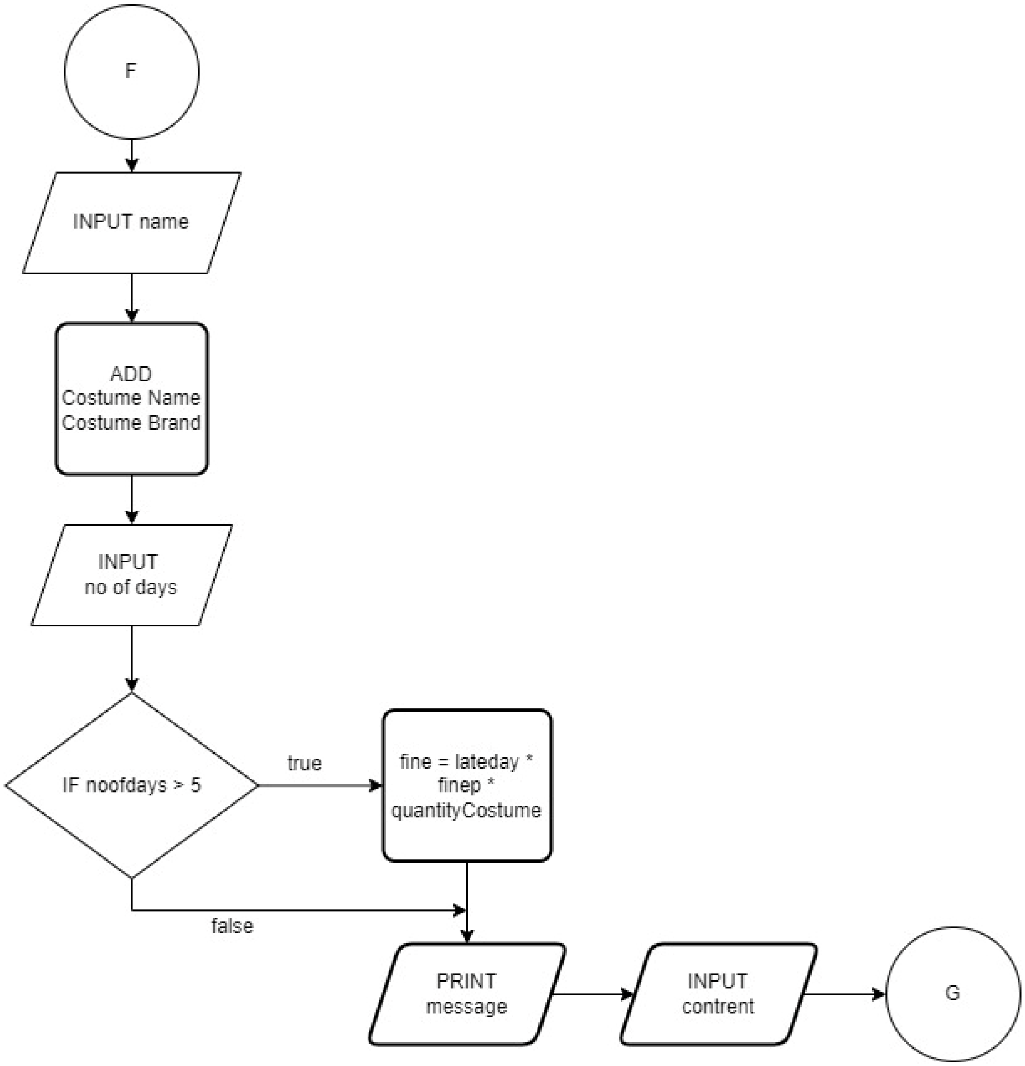

O
DISPLAY AvailableCustom message
Enter Quantity
CHECK IF quantityCostume <=0 or quantityCostume > stockquantity
true
IF quantitycostume <= 0
true
DISPLAY invalid message
false
DISPLAY error message
false
INPUT name
ADD Costume Name. Costume Brand.
CALCULATE totalPrice
P

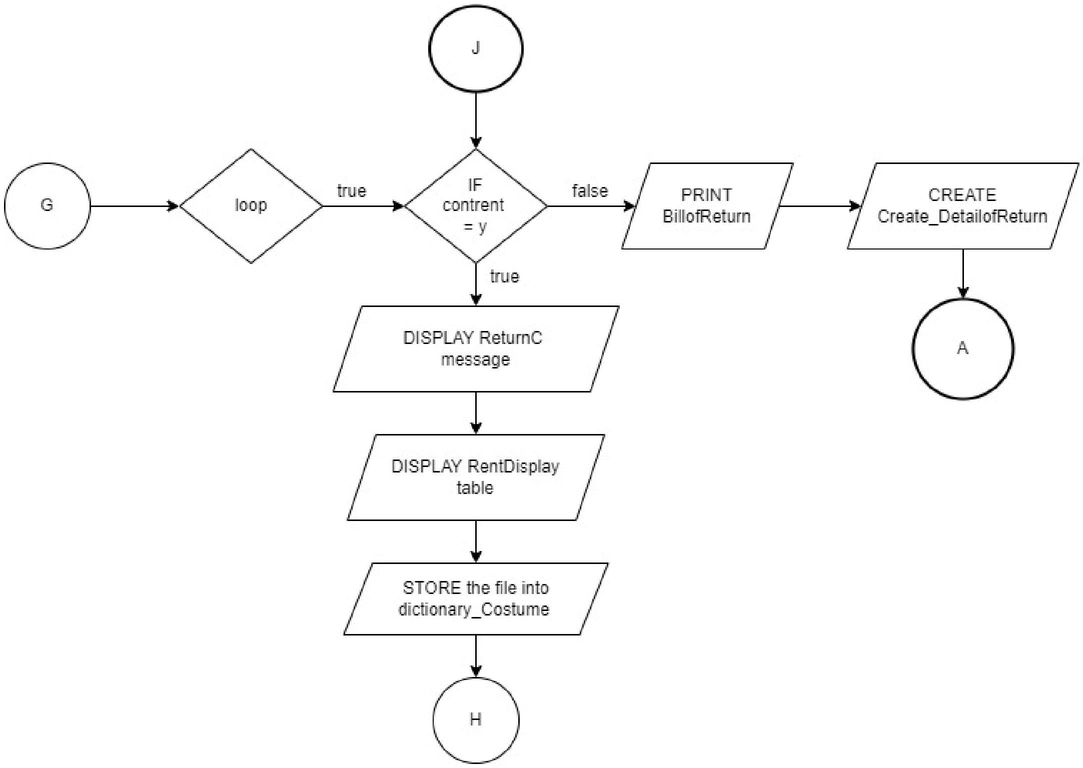

P
DISPLAY message
INPUT contrent
loop
true
N1e
IF contrent = 'y'
false
DISPLAY BillofRent
A
true
Q


Q
DISPLAY RentC message
DISPLAY RentDisplay table
STORE the file into dictionary_Costume
INPUT valid_CostumeID
IF quantity > 0
true
DISPLAY AvailableCostume message
R
false
DISPLAY BillofRent
CREATE Create_DetailofRent file
A


R
CHECK IF CostumeName belongs to rentCostumeA
true
false
UPDATE quantityCostume
ADD Costume Name, Costume Brand
UPDATE quantityCostume
UPDATE quantity in dictionaryCostume
UPDATE totalPrice
PRINT message
S


M
DISPLAY ReturnC message
DISPLAY RentDisplay table
STORE the file into dictionary_Costume
INPUT valid_CostumeID
INPUT quantity
IF quantity > 0
false
DISPLAY invalid quantity message
A
true
F


F
INPUT name
ADD
Costume Name
Costume Brand
INPUT
no of days
IF noofdays > 5
true
fine = lateday *
finep *
quantityCostume
false
PRINT
message
INPUT
contrent
G


J
G
loop
true
IF
contrent
= y
false
PRINT
BillofReturn
CREATE
Create_DetailofReturn
true
DISPLAY ReturnC
message
A
DISPLAY RentDisplay
table
STORE the file into
dictionary_Costume
H

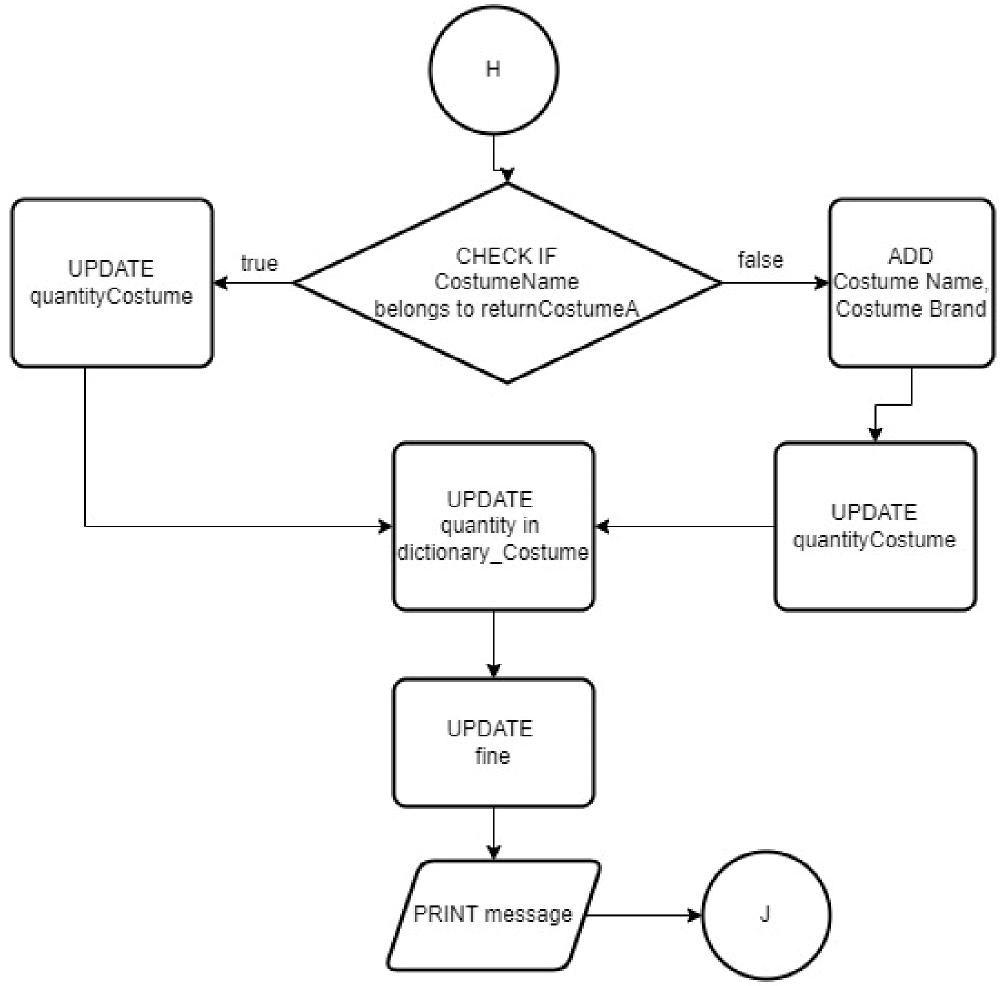


*Figure 8 Flowchart*

## 3.3 Pseudocode

Pseudocode is an informal language which assists programmers to develop algorithms. The rules of Pseudocode are reasonably straightforward i.e., all the statements which show 'dependency' shall be indented (UNF, 2022). Pseudocode has been written to develop algorithm for the program used in creation of Costume Rental System.

### 3.3.1 function.py pseudocode

```
        import datetime
CREATE function WelcomeNotice
   DISPLAY welcome message

CREATE function OptionSelectNotice
   OUTPUT desirable option
   OUTPUT the options to rent, return and exit

CREATE function ThankyouNotice
   OUTPUT message Thank You for using our application.

CREATE function InvalidNotice
   OUTPUT message Please enter a valid input
   OUTPUT message The value shall be selected as per the provided options

CREATE function RentC
   OUTPUT Rent

CREATE function AvailableCostume
  OUTPUT Costume is in Stock.



CREATE function UnavailableCostume
   OUTPUT Insufficient stock for the selected Costume.
```

```
CREATE function InvalidExceptionNotice
   OUTPUT "Please provide a valid option"



CREATE function RentDisplay
   OPEN file Costume.txt in read file
   OUTPUT ID Customer Name   Costume Brand  Price   Quantity
   CREATE IDcounter and set its value 0
        FOR line in file:
                UPDATE IDcounter by adding 1
                OUTPUT adding IDcounter and line replace
        CLOSE file


CREATE function dictionaryRent
   OPEN file Costume.txt in read file
   CREATE IDcounter and set its value 0
   CREATE a dictionary dictionary_Costume
        FOR line in file:
                UPDATE IDcounter by adding 1
                REPLACE ("\n","") in line
                SPLIT (',') in line

        CREATE a dictionary dictionary_Costume

   RETURN dictionary_Costume
```

```
    CLOSE file


CREATE function ReturnC
    OUTPUT Return

CREATE function valid_CostumeIDr
    CREATE a variable IsException and set its value to False
    WHILE IsException is False:
        TRY:
            DECLARE a variable ‘validCostumeIDr’ as input for CostumeID to return
            CREATE a variable IsException and sets its value to True
        EXCEPT:
                    CALL InvalidExceptionNotice
          WHILE validCostumeIDr less than equal to 0 or validCostumeIDr is greater than
length of ‘dictionaryRent’
        OUTPUT Please provide a valid costume ID !
        CALL RentDisplay
        CREATE a variable validCostumeIDr as input for costumeID
        RETURN validCostumeID


CREATE function valid_costumeID
    CREATE a variable IsException and sets its value to False
    WHILE IsException is False:
        TRY:
            CREATE a variable validCostumeID as input for CostumeID to rent
            CREATE a variable IsException and sets its value to true
        EXCEPT:
            CALL InvalidExceptionNotice
```

```
 WHILE validCostumeID less than equal to 0 or validCostumeID is greater than length of ‘dictionaryRent’
    OUTPUT Please provide a valid costume ID !
    CALL RentDisplay
    CREATE a variable validCostumeID as input for costumeID fro rent
 RETURN validCostumeID

CREATE function RentQuantity for stockquantity as parameter
 CREATE a variable IsException and sets its value to False
 WHILE IsException is False:
    TRY:
       CREATE a variable quantityCostume as input Costume number
       CREATE a variable IsException is True
    EXCEPT:
       CALL InvalidExceptionNotice

 WHILE quantityCostume less than equal to 0 or quantityCostume is greater than stockquantity:
    IF quantityCostume less than equal to 0:
       OUTPUT Please provide a valid quantity.
    ELSE:
       OUTPUT Provided quantity is greater than the stock quantity.



    CREATE a variable IsException and set its value to False
    WHILE IsException is False:
       TRY:
          CREATE a variable quantityCostume as input Costume number
          CREATE a variable IsException is True
       EXCEPT:
          CALL InvalidExceptionNotice
 RETURN quantityCostume

CREATE function ReturnQuantity(c):
    CREATE a variable IsException and set its value to False
    WHILE IsException is False:
        TRY:
            CREATE a variable quantityCostume as input Costume Number
            CREATE a variable IsException is True
        EXCEPT:
            CALL InvalidExceptionNotice

    WHILE quantityCostume less than equal to 0:
        IF quantityCostume less than equal to 0:

            OUTPUT Please provide valid quantity.

        CREATE a variable IsException and set its value to False
        WHILE IsException is False:
            TRY:
                CREATE a variable quantityCostume and store user's input
                CREATE a variable IsException is True
            EXCEPT:
                CALL InvalidExceptionNotice
    RETURN quantityCostume

CREATE function StockCostume with dictionary parameter
    OPEN Costume text file in write and store in file
    FOR i in dictionary values:
        line = str(i[0] + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]))
        WRITE line in file
        CLOSE file
```

```
CREATE function CalculatePrice with dictionary, quantitydetails, costumeID as parameter
    price = float(dictionary[costumeID][2].replace("$",""))
    OUTPUT The price of the is costume is price
    CREATE a variable pricePItem and store the value of multiplication of price and quantityDetails
    RETURN pricePItem

CREATE function BillofRent with name, todaysdate, totalprice, rentCostumename, rentCostumebrand as parameter

    OUTPUT  Bill Details

    OUTPUT Name:
    OUTPUT Date and Time of borrow:
    OUTPUT Total price:
    OUTPUT  Rented Costumes are:
    FOR x in range as length of rentCostumename
        print(str(x+1) + ". " + RentCostumeName[x] + ":- " + RentCostumeBrand[x])


CREATE function Create_DetailofRent with cname, date, total, costumename, costumebrand as parameter
    CREATE a variable fileName and store "Rent_" + cname + "_" + str(datetime.datetime.now().second) + str(datetime.datetime.now().microsecond) + str(datetime.datetime.now().hour) + ".txt"

    OPEN filename and store in file write
    WRITE costume Name in file
    WRITE Date in file
    WRITE Total price in file
    WRITE rented in file
    FOR i in range of costumename length
        WRITE CostumeName and CostumeBrand
    CLOSE file

CREATE function BillofReturn with parameters as name, todaysdate, days, fine, costumename, costumebrand

    OUTPUT  Bill Details

    OUTPUT Name:
    OUTPUT Time of return
    OUTPUT Total no of days:
    OUTPUT Applicable Fine:
    OUTPUT  Rented Costumes are:
    FOR i in range of costumename as length
        WRITE CostumeName and CostumeBrand


CREATE function create_DetailofReturn with parameters as cname, date, days, fine, costumeName, costumeBrand
    fileName = "Return_" + cname + "_" + str(datetime.datetime.now().second) + str(datetime.datetime.now().microsecond) + str(datetime.datetime.now().hour) + ".txt"

    OPEN filename in write and store in file
    WRITE Customer Name in file
    WRITE Date of returned in file
    WRITE Total No of days in file
    WRITE Fine Amount in file
    WRITE rented in file
```

```
FOR x in range of CostumeName length
    WRITE CostumeName and CostumeBrand
CLOSE file
```

### 3.3.2 main.py pseudocode

```
          import function
import datetime
CALL function WelcomeNotice()
CREATE a contLoop variable and set its value True

WHILE contLoop is True:

    CALL function OptionSelectNotice()

   CREATE IsException is variable and set its value to False
    WHILE IsException is False:
        TRY:
            CREATE n variable and take users' input
            IsException True
        EXCEPT:
            CALL function InvalidExceptionNotice
            CALL function OptionSelectNotice

    IF n is 1
        CALL function RentC
        CALL function RentDisplay
        CALL function dictionaryRent

        CREATE a variable dictionary_Costume and store function dictionaryRent

        CREATE a variable rentCostumeID and store function valid_costumeID

        CREATE an array rentCostumeA
        CREATE an array rentCostumeC
        IF int(dictionaryCostume[rentCostumeID][3]) > 0:
            CALL function AvailableCostume

            CREATE a variable quantityCostume and store function RentQuantity with parameter int(dictionaryCostume[rentCostumeID][3])
            UPDATE dictionary_Costume[rentCostumeID][3] with int(dictionary_Costume[rentCostumeID][3]) after sub quantityCostume

            CREATE a name variable and store user’s input
            CREATE a todaysdate and store current date and time

            ADD CostumeName in rentCostumeA
            ADD Costumebrand in rentCostumeC

            CALL function StockCostume(dictionary_Costume)

            CREATE a variable totalPrice and store function CalculatePrice with parameter dictionary_Costume, quantityCostume, rentCostumeID

            OUTPUT "#######################################################"
            OUTPUT "Do you desire to rent another costume as well?"
            CREATE variable contrent and store user’s input

            CREATE loop and set its value to True
            WHILE loop is True:
                IF contrent is "y":
                    CALL function RentC
                    CALL function RentDisplay
                    CREATE a variable dictionary_Costume and store function dictionaryRent
                    CREATE a variable rentCostumeID and store function valid_costumeID
                IF int(dictionaryCostume[rentCostumeID][3]) > 0:
                    CALL function AvailableCostume



                    IF dictionaryCostume[rentCostumeID][0] in rentCostumeA:
                        CREATE a variable quantityCostume and store after function RentQuantity with parameter int(dictionary_Costume[rentCostumeID][3]) is add quantityCostume

                    ELSE:
                        ADD costumeName in rentCostumeA
                        ADD costumeBrand in rentCostumeN
                        CREATE a variable quantityCostume and store after function RentQuantity with parameter int(dictionary_Costume[rentCostumeID][3]) is add quantityCostume



                    UPDATE dictionary_Costume[rentCostumeID][3] with int(dictionary_Costume[rentCostumeID][3]) after sub quantityCostume
                    CALL function StockCostume with parameter dictionary_Costume

                    CREATE a variable totalPrice and store after function CalculatePrice with parameter dictionary_Costume, quantityCostume, rentCostumeID and add totalPrice
                    OUTPUT "**********************************************"
                    OUTPUT "Do you desire to rent another costume as well?"
                    CREATE a variable contrent and store user input
                ELSE:
                CALL function UnavailableCostume
```

```
                    CALL function BillofRent with parameter name, todaysdate, totalPrice, rentCostumeA, rentCostumeC
                    CALL function Create_DetailofRent with parameter name, todaysdate, totalPrice, RentCostumeA, RentCostumeC
                    SET loop to False

        ELSE:
            CALL function UnavailableCostume

    ELSE IF n is 2
        CALL function ReturnC
        CALL function RentDisplay

        CREATE a variable dictionary_Costume and store function dictionaryRent

        CREATE a variable ReturnCostumeID and store function valid_costumeIDr

        CREATE an array ReturnCostumeA
        CREATE an array ReturnCostumeC


            CREATE a variable quantityCostume and store function ReturnQuantity with parameter int(dictionary_Costume[ReturnCostumeID][3])
            UPDATE dictionary_Costume[ReturnCostumeID][3] with int(dictionary_Costume[ReturnCostumeID][3]) after sub quantityCostume

            CREATE a name variable and store user's input
            CREATE a todaysdate and store current date and time

            ADD costumeName in ReturnCostumeA
            ADD costumeBrand in ReturnCostumeC
```

```
CALL function StockCostume with parameter dictionary_Costume
        CREATE a IsException variable and set its value to false
        WHILE IsException is false
                TRY
                        CREATE a variable days and store user input
                        SET IsException to true
                EXCEPT
                        CALL function InvalidExceptionNotice
        DELCARE a variable fine and set its value 0
        DECLARE a variable lateday and sets its value 0
        DECLARE a variable finep and sets its value 10
        IF days greater than 5
                UPDATE lateday with days after sub 5
                UPDATE fine after multiplication of lateDay fineNo quantityOfCostume
                OUTPUT "You have been fined: ", fine, "for returning ", lateday, "days late."

OUTPUT "*********************************************"
OUTPUT "Have you rented another costume as well?"
CREATE variable contrent and store user's input

CREATE loop and set its value to True
WHILE loop is True:
   IF contRent is "y":
      CALL function ReturnC
      CALL function RentDisplay
      CREATE a variable dictionary_Costume and store function dictionaryRent
      CREATE a variable ReturnCostumeID and store function valid_costumeIDr
```

```
        IF dictionary_Costume[ReturnCostumeID][0] in ReturnCostumeA:
            CREATE a variable quantityCostume and store after function ReturnQuantity with parameter int(dictionary_Costume[ReturnCostumeID][3]) is add quantityCostume

        ELSE:
            ADD costumeName in ReturnCostumeA
            ADD costumeBrand in ReturnCostumeC
            CREATE a variable quantityCostume and store after function ReturnQuantity with parameter int(dictionary_Costume[ReturnCostumeID][3]) is add quantityCostume


        UPDATE dictionary_Costume[ReturnCostumeID][3] with int(dictionary_Costume[ReturnCostumeID][3]) after sub quantityCostume
        CALL function StockCostume with parameter dictionary_Costume

        UPDATE fine after multiplication of lateDay fineNo quantityOfCostume and add fine
      OUTPUT "You have been fined: ", fine, “for returning”, lateday, “days late.”
        OUTPUT "********************************************"
        OUTPUT "Have you rented another costume as well?"
        CREATE a variable contrent and store user’s input
    ELSE:
      CALL function BillofReturn with parameter name, todaysdate, days, ReturnCostumeA, ReturnCostumeC
```

```
            CALL function Create_DetailofReturn with parameter name, todaysdate, days, fine, ReturnCostumeA, ReturnCostumeC
            SET loop to False

  ELIF num is 3
    CALL function ThankyouNotice
    SET contLoop to False

  ELSE:
    CALL function InvalidNotice
```

## 3.4 Data Structures

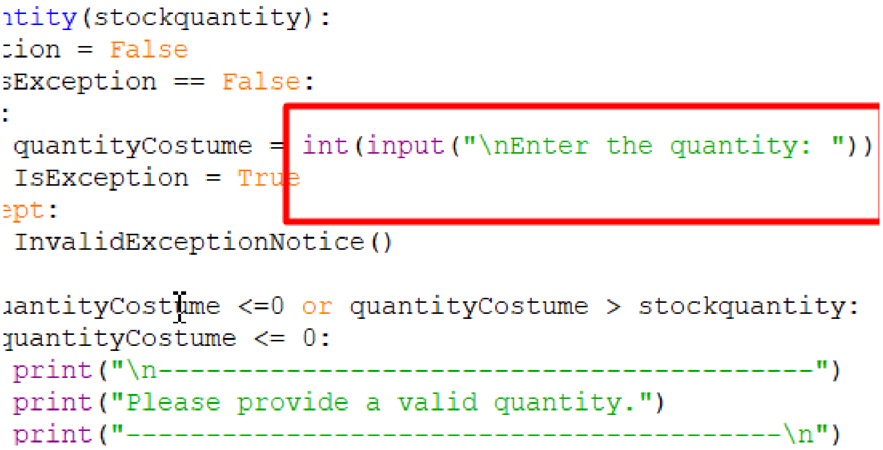
Data Structures are a way of data organization in order to access data in a much efficient manner based on the circumstances. They are the fundamentals of any programming languages including Python Programming Language around which it is built (GeeksforGeeks, 2022). The data types are of two types:

### 3.4.1 Primitive Data Type

They are the most basic data types containing pure and simple values of data. In Python, there are 4 primitive or fundamental data types.

#### 3.4.1.1 Integer:

Integer is used to represent numeric data from negative infinity to infinity.

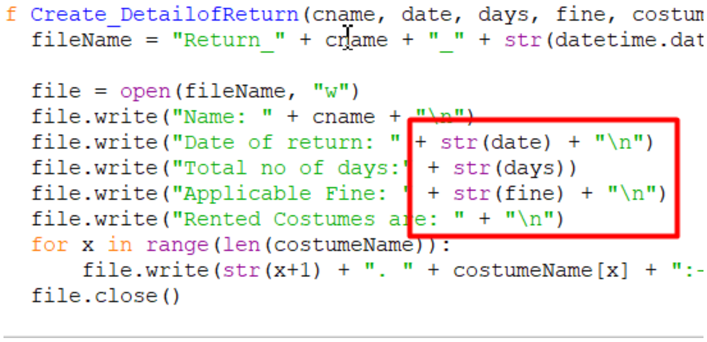


```
ntity(stockquantity):
tion = False
sException == False:
:
 quantityCostume = int(input("\nEnter the quantity: "))
 IsException = True
ept:
 InvalidExceptionNotice()

antityCostume <=0 or quantityCostume > stockquantity:
quantityCostume <= 0:
 print("\n-----------------------------------------")
 print("Please provide a valid quantity.")
 print("-----------------------------------------\n")
```

*Figure 9 Use of Integer Data Type in Program*

#### 3.4.1.2 String

String is collection of alphabets, words or other characters. For example: ‘cake’, ‘cookie’, etc.

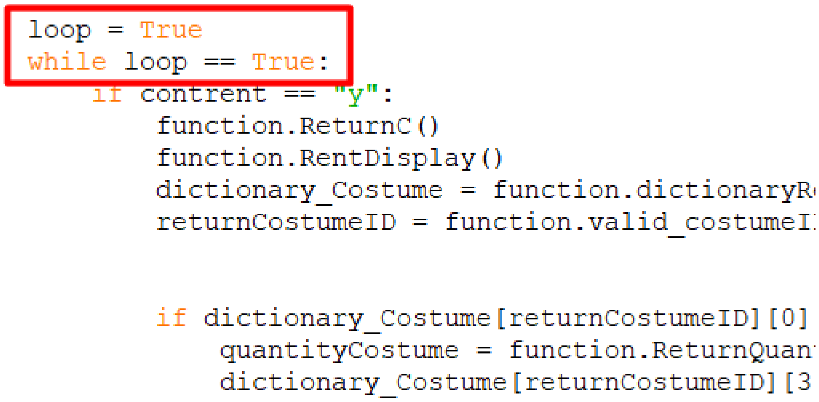


```
f Create_DetailofReturn(cname, date, days, fine, costum
  fileName = "Return_" + cname + "_" + str(datetime.dat

  file = open(fileName, "w")
  file.write("Name: " + cname + "\n")
  file.write("Date of return: " + str(date) + "\n")
  file.write("Total no of days:" + str(days))
  file.write("Applicable Fine: " + str(fine) + "\n")
  file.write("Rented Costumes are: " + "\n")
  for x in range(len(costumeName)):
      file.write(str(x+1) + ". " + costumeName[x] + ":-
  file.close()
```

*Figure 10 Use of String Data Type in Program*

#### 3.4.1.3 Boolean

Boolean can take up only two values: True and False, which is often interchangeable with integers 0 and 1.

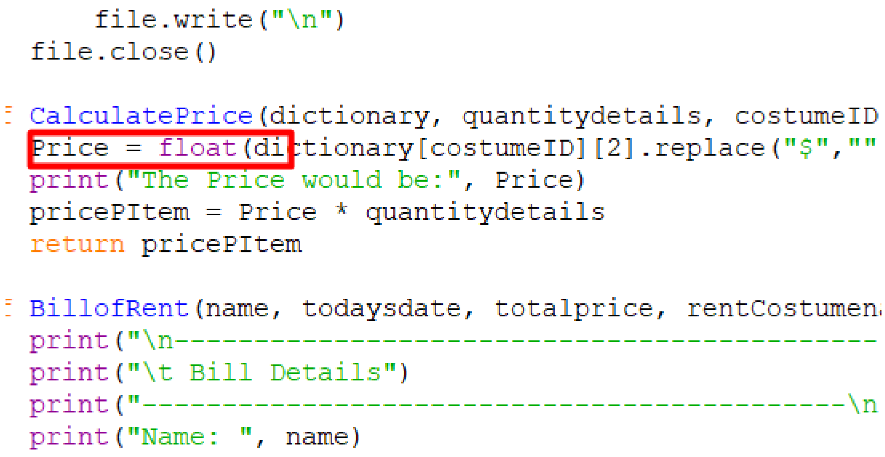


```
loop = True
while loop == True:
    if contrent == "y":
        function.ReturnC()
        function.RentDisplay()
        dictionary_Costume = function.dictionaryR
        returnCostumeID = function.valid_costumeI


        if dictionary_Costume[returnCostumeID][0]
            quantityCostume = function.ReturnQuan
            dictionary_Costume[returnCostumeID][3
```

*Figure 11 Use of Boolean Data Type in Program*

#### 3.4.1.4 Float

Float stands for ‘floating point number’. It is used to represent rational numbers, i.e., numbers ending with decimal like 1.1, 14.0, etc.



```
    file.write("\n")
file.close()

CalculatePrice(dictionary, quantitydetails, costumeID
Price = float(dictionary[costumeID][2].replace("$",""
print("The Price would be:", Price)
pricePItem = Price * quantitydetails
return pricePItem

BillofRent(name, todaysdate, totalprice, rentCostumen
print("\n-------------------------------------------
print("\t Bill Details")
print("--------------------------------------------\n
print("Name: ", name)
```

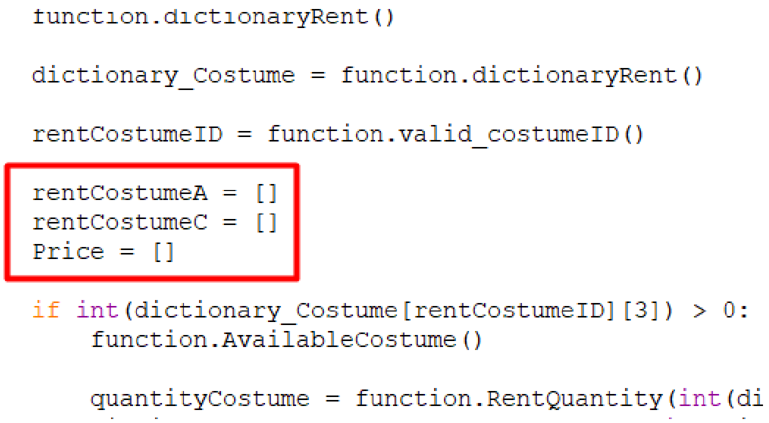
*Figure 12 Use of Float Data Type in Program*

### 3.4.2 Non-Primitive Data Type:

Non-Primitive Data Type can be often recalled as sophisticated members of data structure. They not only store a value but a collection of values and often in various formats. Few of the Non-Primitive Data Types are as follows:

#### 3.4.2.1 List:

Lists in Python stores collection of heterogeneous items and they are mutable in nature which means that their content can be changed without bringing a change in their identity. They can be recognized by square brackets [ ] which hold elements separated by commas (DataCamp, 2022). For example: list A = [ 123,456,7,78,9,0], list 2 = [“hello”, “hi”], etc.

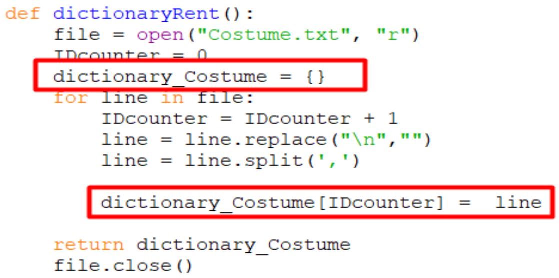


```
function.dictionaryRent()

dictionary_Costume = function.dictionaryRent()

rentCostumeID = function.valid_costumeID()

rentCostumeA = []
rentCostumeC = []
Price = []

if int(dictionary_Costume[rentCostumeID][3]) > 0:
    function.AvailableCostume()

    quantityCostume = function.RentQuantity(int(di
```

*Figure 13 Use of List Data Type in Program*

#### 3.4.2.2 Dictionary:

Dictionary is made up of key-value pairs. ‘key’ in dictionary is used to identify the item and ‘value’ holds the value of the item. One can apply many in-built functionalities on dictionaries in Python. It is mutable in nature and can be recognized by small brackets ( ) whose elements are separated by commas (Sharma, 2022).



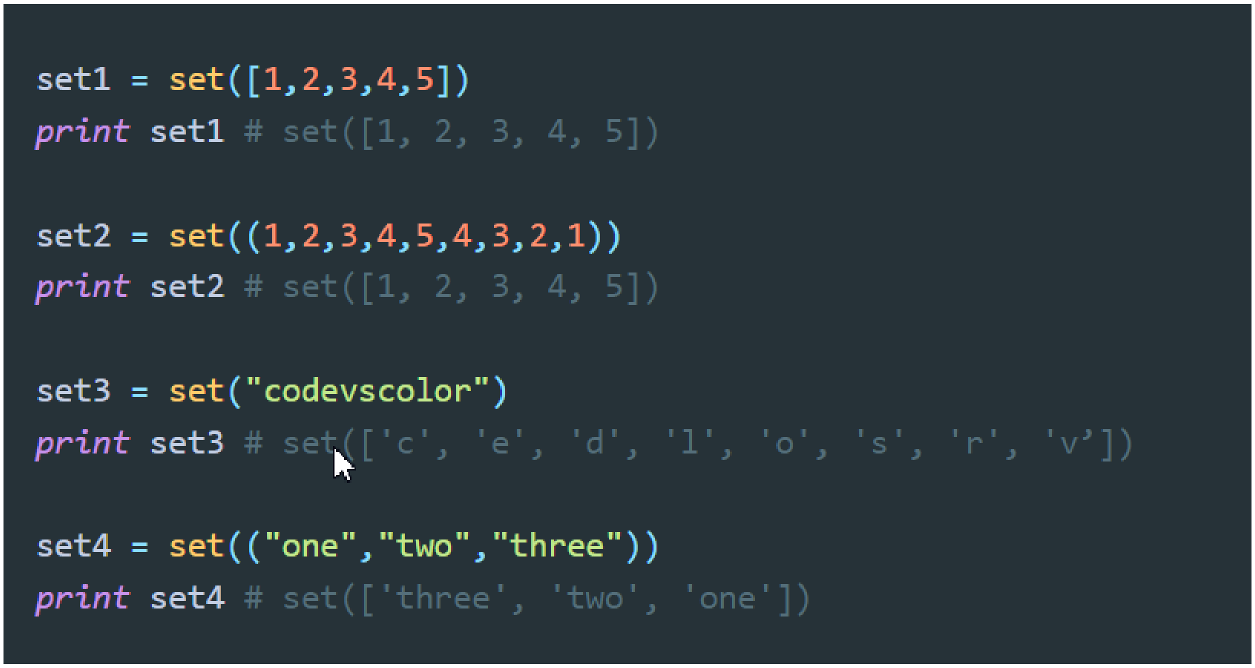
```
def dictionaryRent():
    file = open("Costume.txt", "r")
    IDcounter = 0
    dictionary_Costume = {}
    for line in file:
        IDcounter = IDcounter + 1
        line = line.replace("\n","")
        line = line.split(',')

        dictionary_Costume[IDcounter] =  line

    return dictionary_Costume
    file.close()
```

*Figure 14 Use of Dictionary in Program*

### 3.4.2.3 Sets:

Sets is the collection of unique objects. They are used in creating lists which holds unique values in the dataset. It is mutable in nature but is an unordered collection (DataCamp, 2022). For example: set = (“a”,” b”,” c”).

It is not taken in use in this coursework.



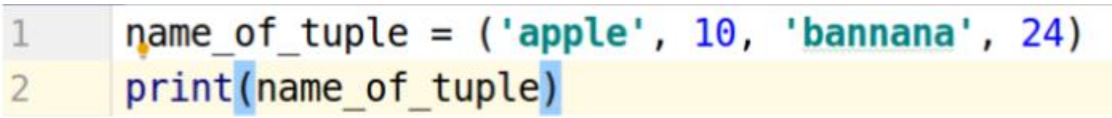
```
set1 = set([1,2,3,4,5])
print set1 # set([1, 2, 3, 4, 5])

set2 = set((1,2,3,4,5,4,3,2,1))
print set2 # set([1, 2, 3, 4, 5])

set3 = set("codevscolor")
print set3 # set(['c', 'e', 'd', 'l', 'o', 's', 'r', 'v’])

set4 = set(("one","two","three"))
print set4 # set(['three', 'two', 'one'])
```

*Figure 15 use of set() (codevscolor, 2022)*

### 3.4.2.4 Tuples:

Tuple is a standard sequence data type which is immutable. It means that once defined the values inside it can’t be deleted, added or edited which might come handy when passing control to someone else (DataCamp, 2022). For example: tuple1 = (1,2,3,4,5), tuple2 = (“Banana”, Orange”).

It is not taken in use un this coursework.



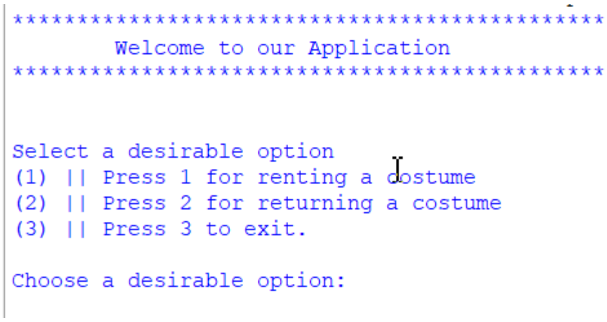
```
name_of_tuple = ('apple', 10, 'bannana', 24)
print(name_of_tuple)
```

*Figure 16 Declaration of tuple (H2K infosys, 2022)*

.

# 4 Program

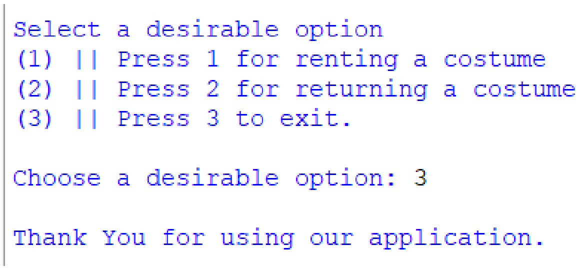
The project is done to develop a Costume Rental System which simulates a real time solution to inventory management and bill creation. The program is user-interactive and it would keep track of the costumes rented and returned by the user. The Program is divided into two processes: Rental process and Return process. At first, the user would be greeted with WelcomeNotice and an OptionSelectionNotice woud be displayed and the user would be asked to choose among options.



```
**********************************************
        Welcome to our Application
**********************************************


Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option:
```

*Figure 17 Start of the Program*

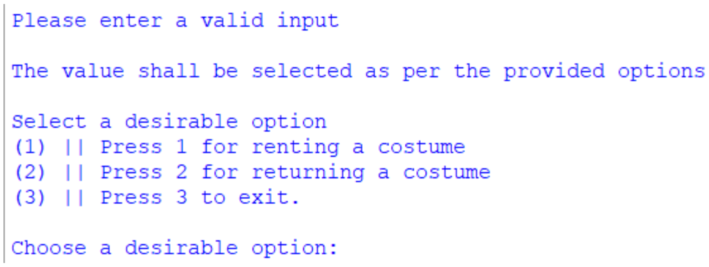
The program would then progress according to the option the user chooses. If the user inputs 3, ThankyouNotice would be displayed and program would end.



```
Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option: 3

Thank You for using our application.
```

*Figure 18 On pressing option 3*

If anything, else is pressed, InvalidNotice would be displayed followed by OptionSelectionNotice.

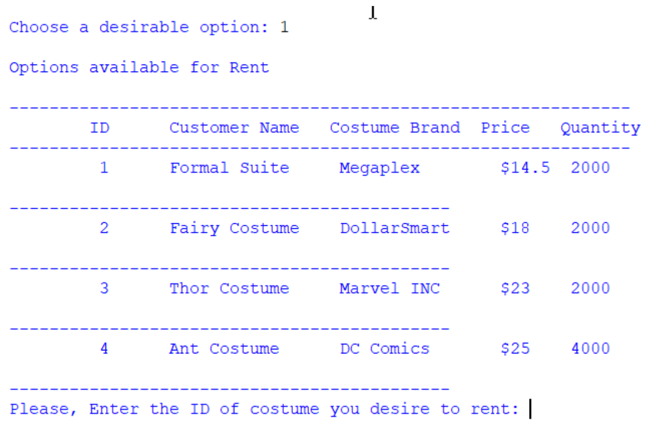


```
Please enter a valid input

The value shall be selected as per the provided options

Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option:
```

*Figure 19 On pressing Invalid Option*

## 4.1 Rental Process

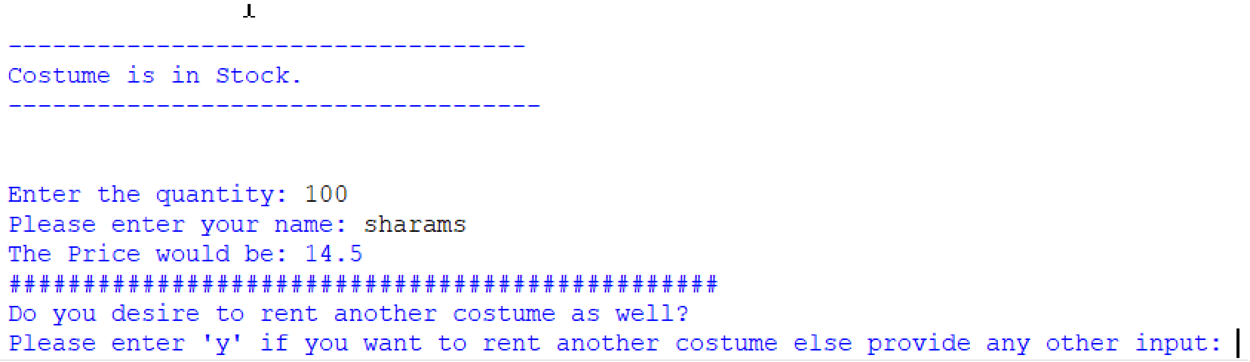
If the user chooses to input option 1, then the Rental Process of the program would begin. At first, all the options available for rent would be displayed asking user to input desirable valid ID of the Costume for the costume they wish to rent.



```
Choose a desirable option: 1

Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  2000

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    2000

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: 
```

*Figure 20 Rent Table and Valid CustomerID input in Rental Process*

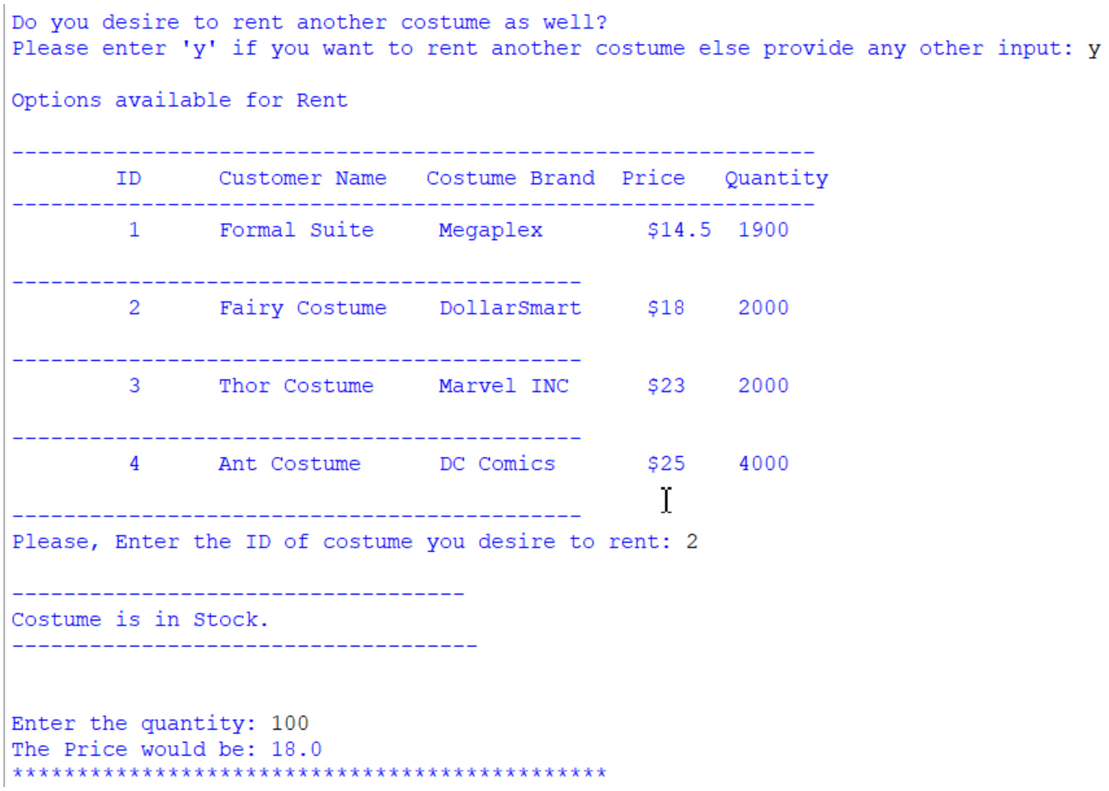
Then, after inputting a valid Costume ID, a message stating that a costume is available would be displayed in the SHELL. Simultaneously, user would be asked to input the quantity of costume they wish to rent. After entering the quantity, user would then be asked to input their name. The Price would be then displayed and a prompt to rent another costume would be displayed.



```
-----------------------------------
Costume is in Stock.
------------------------------------


Enter the quantity: 100
Please enter your name: sharams
The Price would be: 14.5
################################################
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: 
```

*Figure 21 User Input and Prompt to continue further in Rental Process*

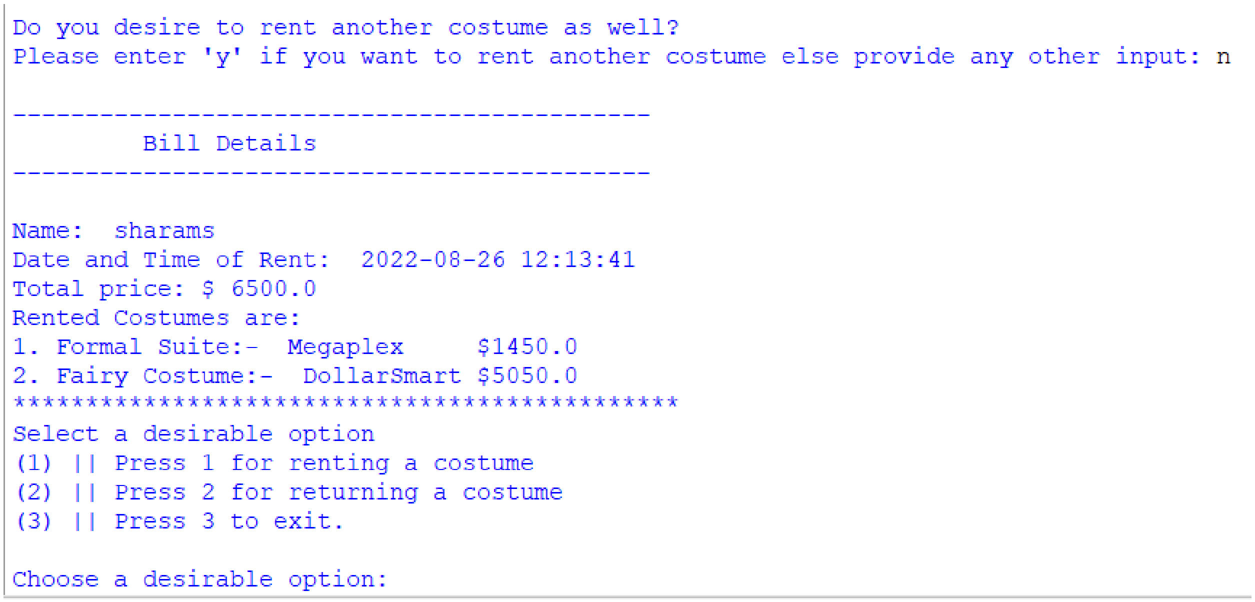
Suppose, if a user presses ‘y’, then the user would again be displayed the rent table and would be asked to input as above.



```
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: y

Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1900

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    2000

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: 2

-----------------------------------
Costume is in Stock.
------------------------------------


Enter the quantity: 100
The Price would be: 18.0
**********************************************
```

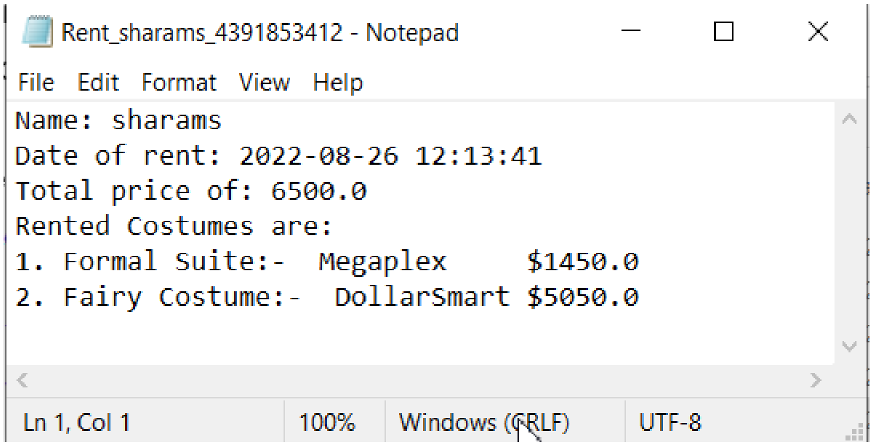
*Figure 22 Rental Process after user presses 'y'*

On, again displaying the prompt to continue further, if the user presses anything else other than ‘y’, Bill details of the costume rented would be then displayed and a unique text file containing details of bill of rent would be created separately. Also, OptionSelectionNotice would be displayed after bill details.



```
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: n

--------------------------------------------
         Bill Details
--------------------------------------------

Name:  sharams
Date and Time of Rent:  2022-08-26 12:13:41
Total price: $ 6500.0
Rented Costumes are:
1. Formal Suite:-  Megaplex     $1450.0
2. Fairy Costume:-  DollarSmart $5050.0
**********************************************
Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option:
```

*Figure 23 Bill details in Rental Process*



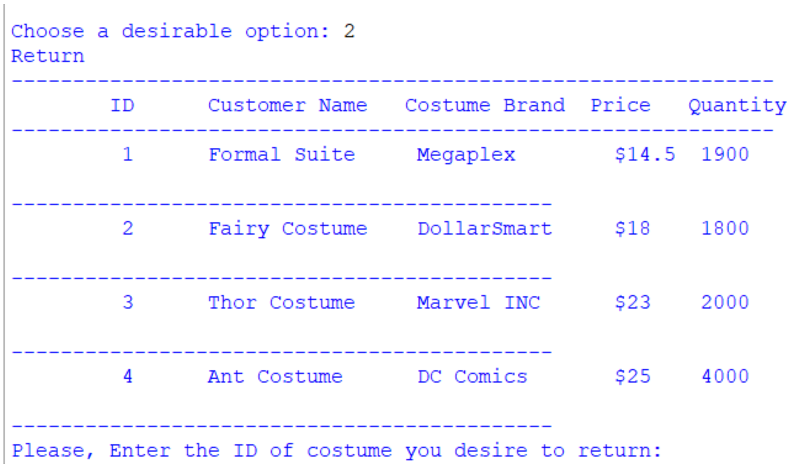
```
Name: sharams
Date of rent: 2022-08-26 12:13:41
Total price of: 6500.0
Rented Costumes are:
1. Formal Suite:-  Megaplex     $1450.0
2. Fairy Costume:-  DollarSmart $5050.0
```

*Figure 24 Details of Bill in text file*

This would conclude Rental Process of the Program.

## 4.2 Return Process

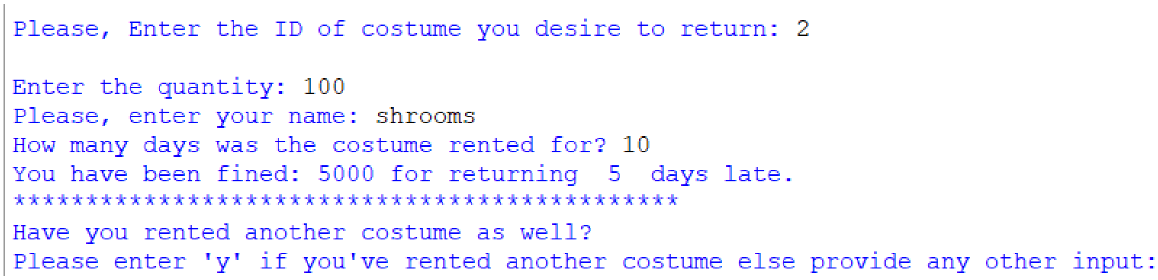
Similarly, if the user chooses option 2, the program would then progress into return process. Once, option is selected, user would be displayed a return table asking user to choose the costume they want to return using a valid costume ID.



```
Choose a desirable option: 2
Return
--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1900

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    1800

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to return:
```

*Figure 25 Return Table Display and Valid Costume ID input in Return Process*

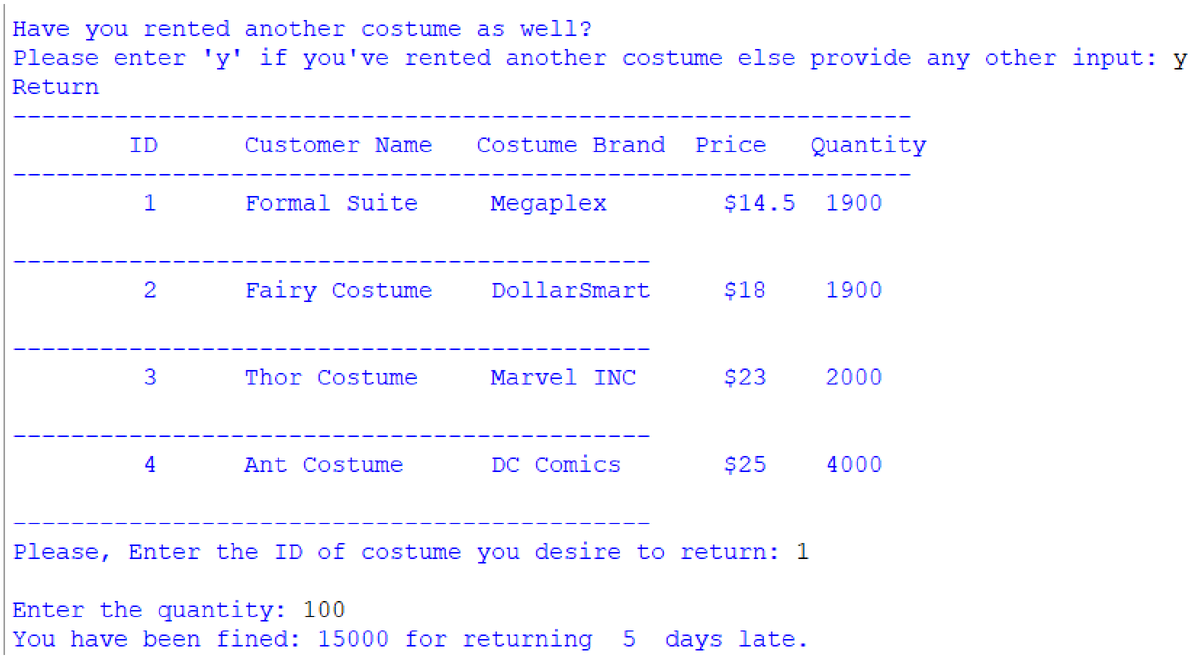
After, a valid Costume Id is entered, the user would be asked to input quantity of costume they wish to return along with their name and the number of days the costume had been rented for. Then, a message showing the total amount of fine would be displayed to the user along with prompt to continue further.



```
Please, Enter the ID of costume you desire to return: 2

Enter the quantity: 100
Please, enter your name: shrooms
How many days was the costume rented for? 10
You have been fined: 5000 for returning  5  days late.
**********************************************
Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input:
```

*Figure 26 User input and Prompt to continue further in Return Process*

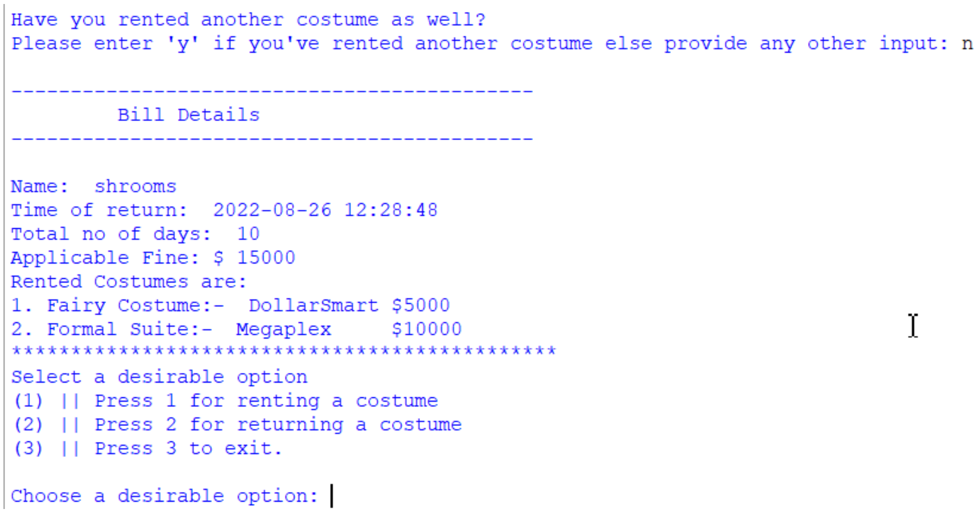
If the user presses ‘y’, again the return table would be displayed and the user would be asked to input as above.



```
Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input: y
Return
--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1900

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    1900

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to return: 1

Enter the quantity: 100
You have been fined: 15000 for returning  5  days late.
```

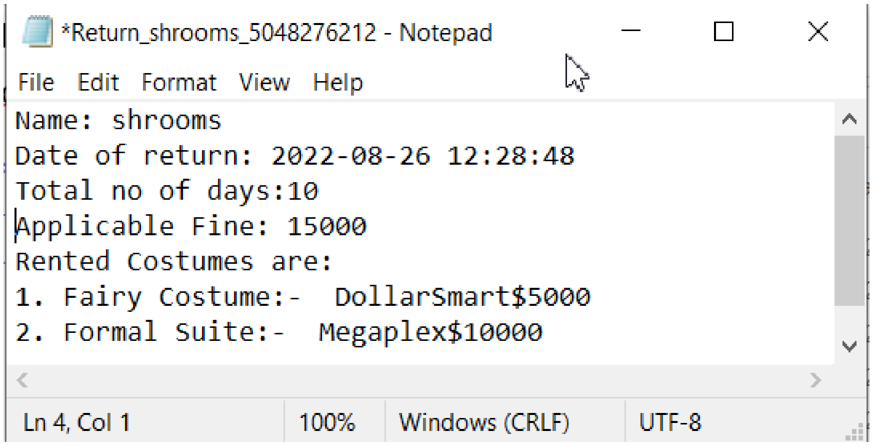
*Figure 27 Return Process after user presses 'y'*

On, again displaying the prompt to continue further, if the user presses anything else other than 'y', Bill details of the costume returned would be then displayed and a unique text file containing details of bill of return would be created separately. Also, OptionSelectionNotice would be displayed after bill details.



```
Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input: n

--------------------------------------------
         Bill Details
--------------------------------------------

Name:  shrooms
Time of return:  2022-08-26 12:28:48
Total no of days:  10
Applicable Fine: $ 15000
Rented Costumes are:
1. Fairy Costume:-  DollarSmart $5000
2. Formal Suite:-  Megaplex     $10000
**********************************************
Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option: 
```

*Figure 28 Bill Details in Return Process*



*Return_shrooms_5048276212 - Notepad
File Edit Format View Help
Ln 4, Col 1
100%
Windows (CRLF)
UTF-8

```
Name: shrooms
Date of return: 2022-08-26 12:28:48
Total no of days:10
Applicable Fine: 15000
Rented Costumes are:
1. Fairy Costume:-  DollarSmart$5000
2. Formal Suite:-  Megaplex$10000
```

*Figure 29 Details of return Bill in text file*

This would conclude Return Process.

# 5 Testing of the project

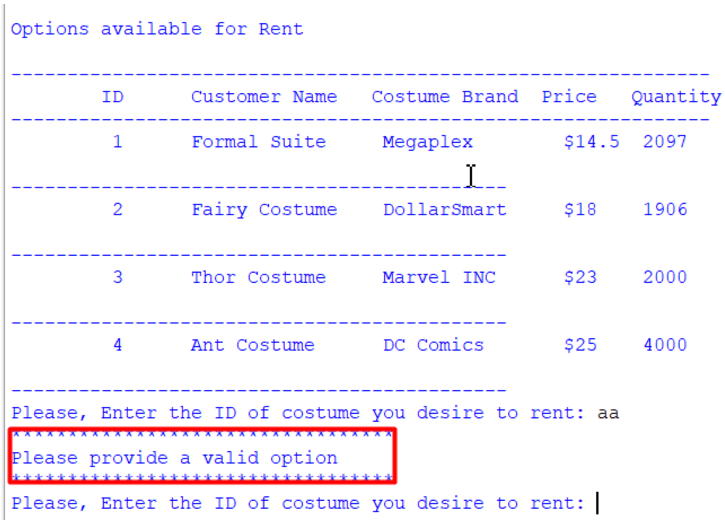
After the creation of the program, its testing is very crucial to find bugs and test the usability of the program or if the software is running as, it is originally supposed to. Testing is a part of development and shouldn't be neglected in any way.

In the coursework too after the creation of Costume Rental System, the program has been tested in several ways. Few of the conducted test to test the usability of the program are as follows:

## 5.1 Test 1**:**

| Test | 1 |
|---|---|
| Objective | To display appropriate message on entering invalid input. |
| Action | IDLE Shell was opened and the value of Customer ID was entered as 'aa'. |
| Expected Output | Error message stating "Please provide valid option" shall be shown. |
| Actual Output | Error message stating "Please provide valid option" was shown. |
| Result | The test was successful |

*Table 1 Test Table 1*



```
Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  2097

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    1906

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: aa
**********************************
Please provide a valid option
**********************************
Please, Enter the ID of costume you desire to rent: 
```

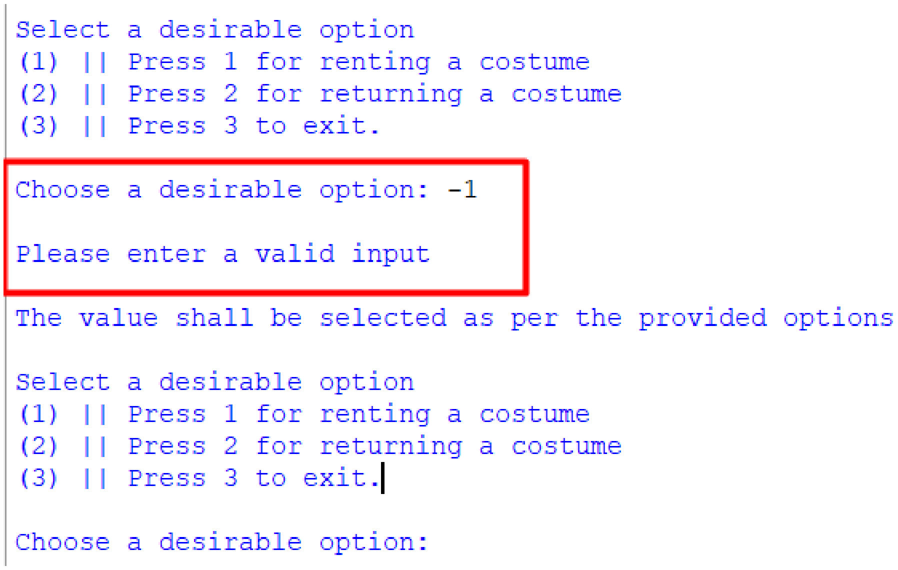
*Figure 30 Screenshot of output of Test 1*

## 5.2 Test 2:

### 5.2.1 Providing negative value as input:

| Test | 2.1 |
|---|---|
| Objective | To display appropriate message on providing negative value as input. |
| Action | IDLE shell was opened. '-1' was entered when asked to choose a desirable option. |
| Expected Output | Invalid message as "Please enter a valid input" shall be displayed. |
| Actual Output | Invalid message as "Please enter a valid input" was displayed. |
| Result | The test was successful |

*Table 2 Test table 2.1*



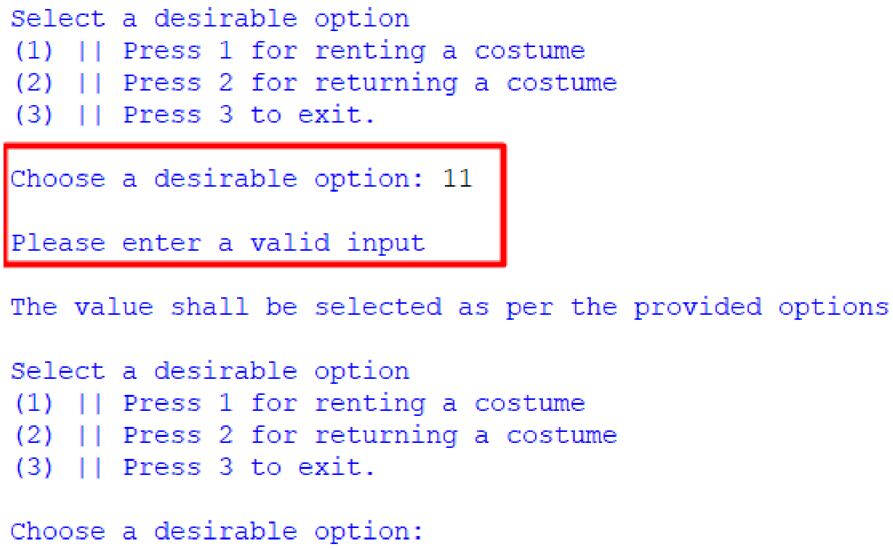
```
Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option: -1

Please enter a valid input

The value shall be selected as per the provided options

Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option:
```

*Figure 31 Screenshot of Output of Test 2.1*

### 5.2.2 Providing non-existing value as input

| Test | 2.2 |
|---|---|
| Objective | To display appropriate message on providing non-existent value as input. |
| Action | IDLE shell was opened. ‘11’ was entered when asked to choose a desirable option. |
| Expected Output | Invalid message as “Please enter a valid input” shall be displayed. |
| Actual Output | Invalid message as “Please enter a valid input” was displayed. |
| Result | The test was successful |

*Table 3 Test table 2.2*



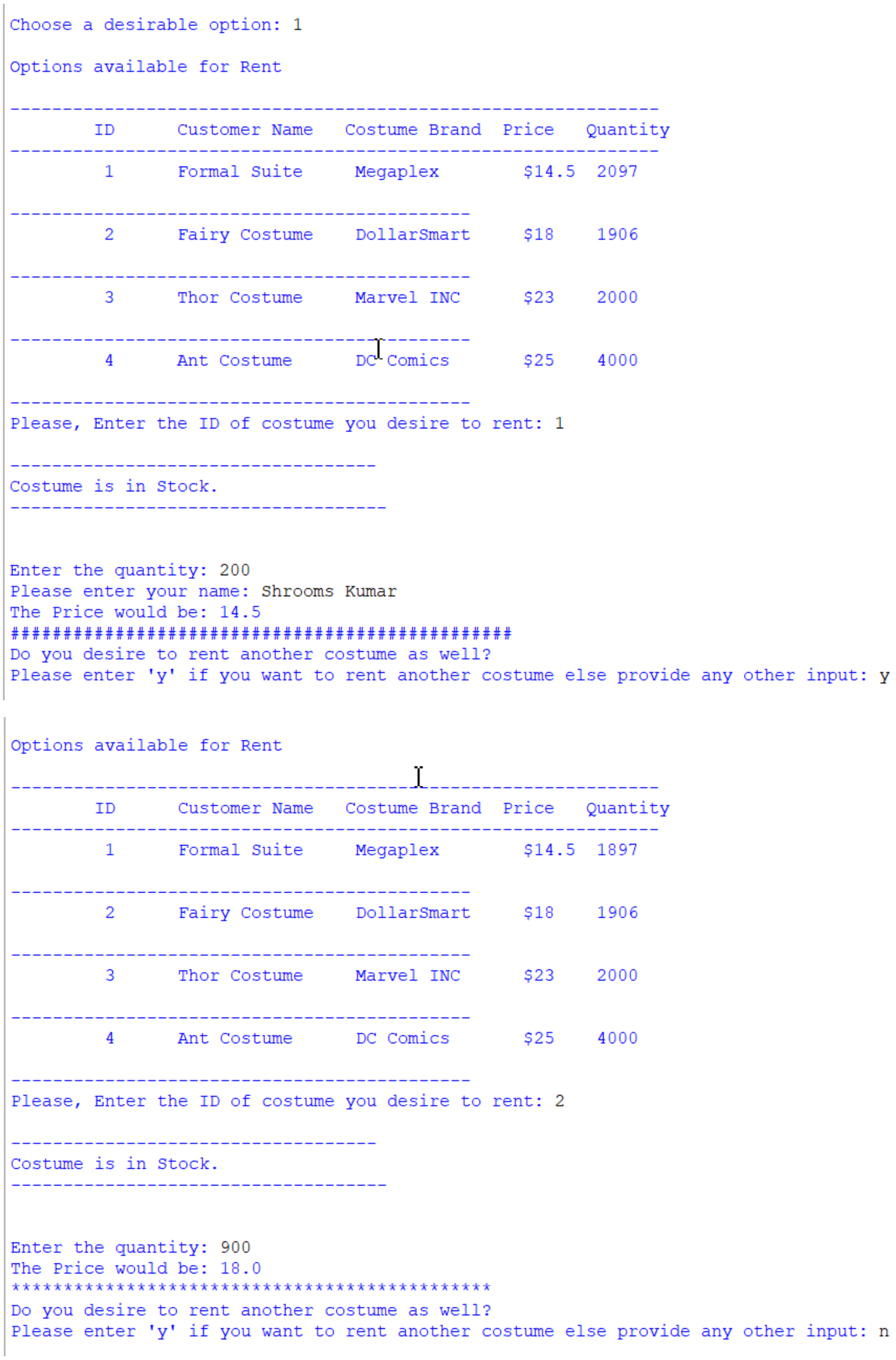
```
Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option: 11

Please enter a valid input

The value shall be selected as per the provided options

Select a desirable option
(1) || Press 1 for renting a costume
(2) || Press 2 for returning a costume
(3) || Press 3 to exit.

Choose a desirable option:
```

*Figure 32 Screenshot of Output of Test 2.2*

## 5.3 Test 3:

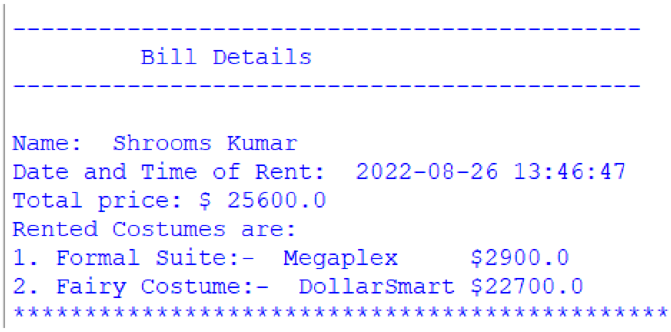
| Test | 3 |
|---|---|
| Objective | To show the file created of bill when the costume is rented. |
| Action | IDLE shell was opened and appropriate options to rent were pressed. After that, ID and quantity of costume were entered and the name of the user was also inputted. |
| Expected Output | The costumes shall be rented and the bill details shall be printed in the console and .txt file containing details of rent bill shall be created. |
| Actual Output | The costumes were rented and the bill details was printed in the console and .txt file containing details of rent bill was created. |
| Result | The test was successful |

*Table 4: Test Table 3*



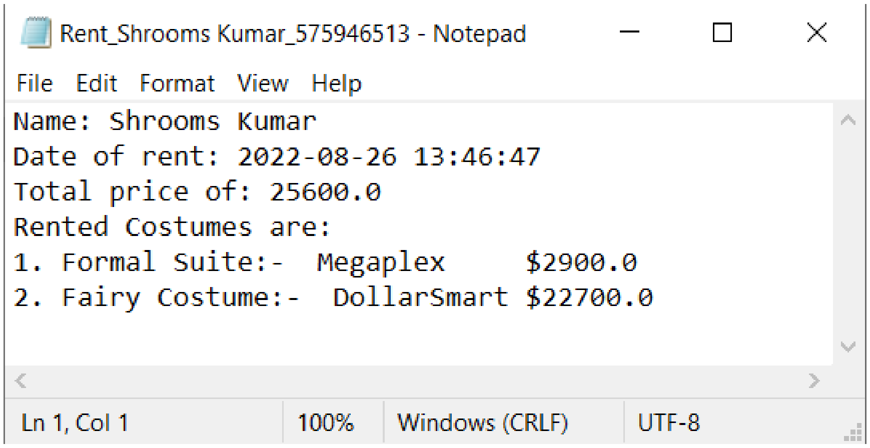
```
Choose a desirable option: 1

Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  2097

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    1906

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: 1

-----------------------------------
Costume is in Stock.
------------------------------------


Enter the quantity: 200
Please enter your name: Shrooms Kumar
The Price would be: 14.5
################################################
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: y

Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1897

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    1906

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: 2

-----------------------------------
Costume is in Stock.
------------------------------------


Enter the quantity: 900
The Price would be: 18.0
**********************************************
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: n
```

*Figure 33 Screenshot of Renting Costume*



```
---------------------------------------------
         Bill Details
---------------------------------------------

Name:  Shrooms Kumar
Date and Time of Rent:  2022-08-26 13:46:47
Total price: $ 25600.0
Rented Costumes are:
1. Formal Suite:-  Megaplex     $2900.0
2. Fairy Costume:-  DollarSmart $22700.0
***********************************************
```

*Figure 34 Screenshot of Bill Details*



Rent_Shrooms Kumar_575946513 - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

```
Name: Shrooms Kumar
Date of rent: 2022-08-26 13:46:47
Total price of: 25600.0
Rented Costumes are:
1. Formal Suite:-  Megaplex     $2900.0
2. Fairy Costume:-  DollarSmart $22700.0
```

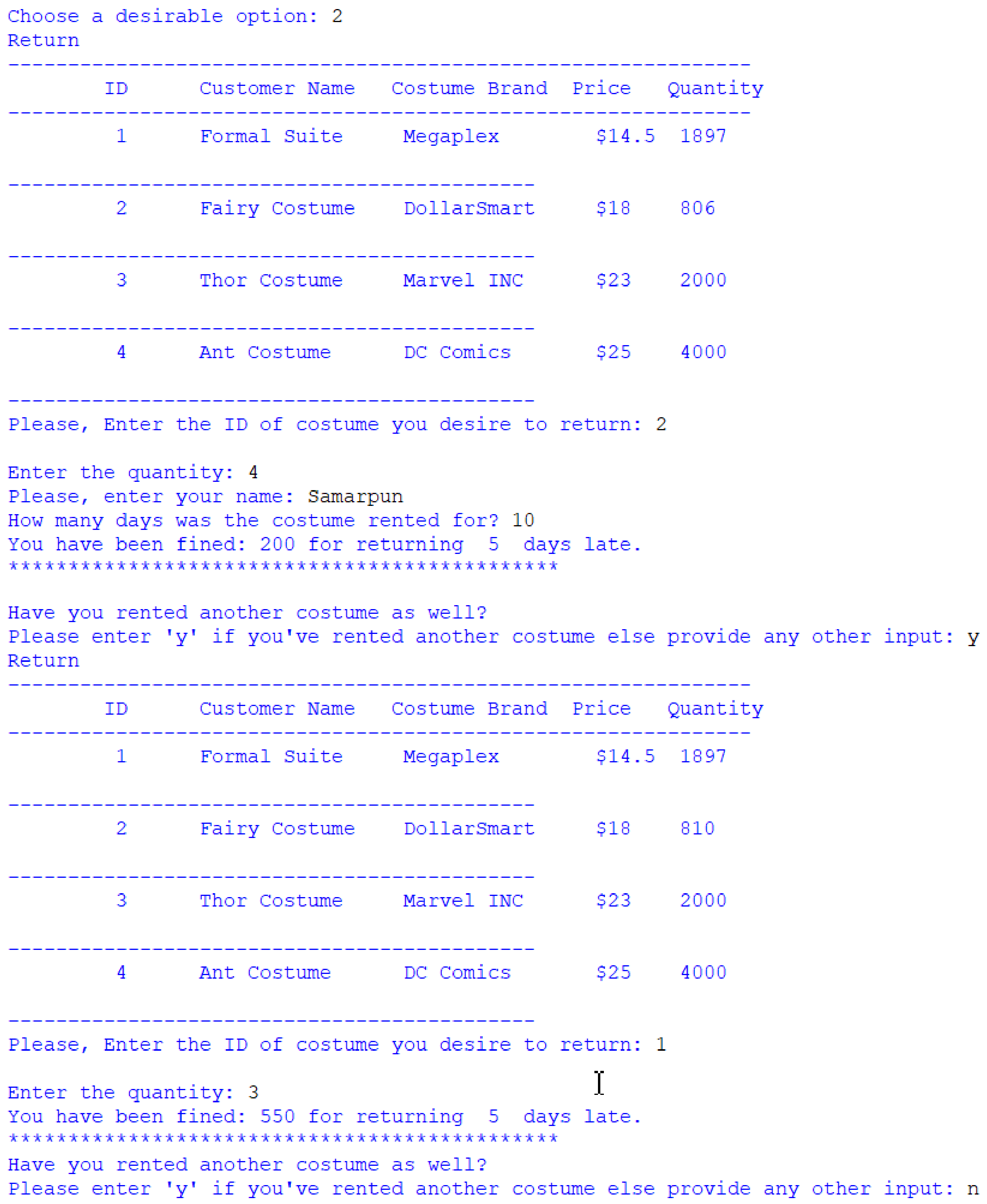
*Figure 35 Screenshot of Details of Bill in Text File*

*Figure 36 Screenshots of Output of Test 3*

## 5.4 Test 4:

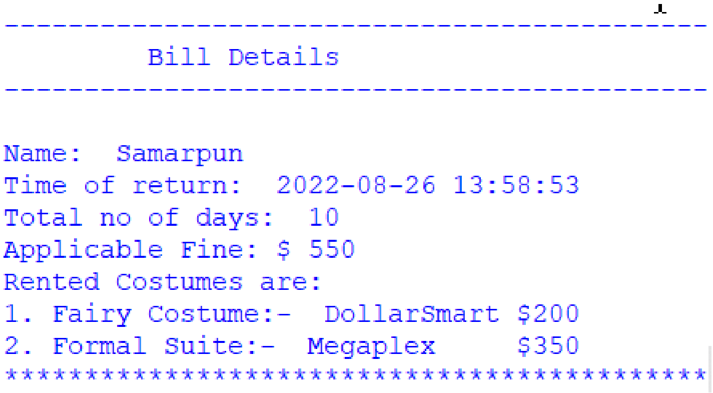
| Test | 4 |
|---|---|
| Objective | To show the file created of bill when the costume is returned. |
| Action | IDLE shell was opened and appropriate options to return were pressed. After that, ID and quantity of costume were entered and the name of the user was also inputted. |
| Expected Output | The costumes shall be rented and the bill details shall be printed in the console and .txt file containing details of rent bill shall be created. |
| Actual Output | The costumes were rented and the bill details was printed in the console and .txt file containing details of rent bill was created. |
| Result | The test was successful |

*Table 5 Test Table 4*



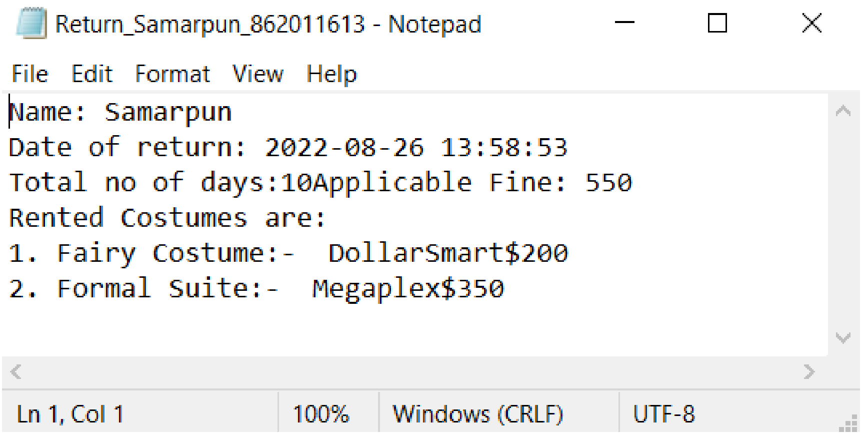
```
Choose a desirable option: 2
Return
--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1897

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    806

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to return: 2

Enter the quantity: 4
Please, enter your name: Samarpun
How many days was the costume rented for? 10
You have been fined: 200 for returning  5  days late.
**********************************************

Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input: y
Return
--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1897

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    810

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to return: 1

Enter the quantity: 3
You have been fined: 550 for returning  5  days late.
**********************************************
Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input: n
```

*Figure 37 Multiple returning processes*



```
--------------------------------------------
         Bill Details
--------------------------------------------

Name:  Samarpun
Time of return:  2022-08-26 13:58:53
Total no of days:  10
Applicable Fine: $ 550
Rented Costumes are:
1. Fairy Costume:-  DollarSmart $200
2. Formal Suite:-  Megaplex     $350
********************************************
```

*Figure 38 Screenshot of Bill Creation of Returning*



```
Name: Samarpun
Date of return: 2022-08-26 13:58:53
Total no of days:10Applicable Fine: 550
Rented Costumes are:
1. Fairy Costume:-  DollarSmart$200
2. Formal Suite:-  Megaplex$350
```

*Figure 39 Screenshot of detail of Return Bill Creation in text file*

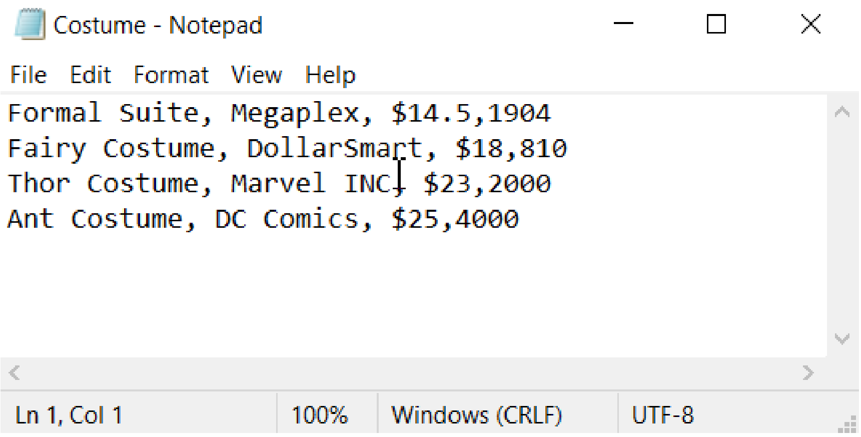
*Figure 40 Screenshots of Output of Test 4*

## 5.5 Test 5:

### 5.5.1 For renting the costume

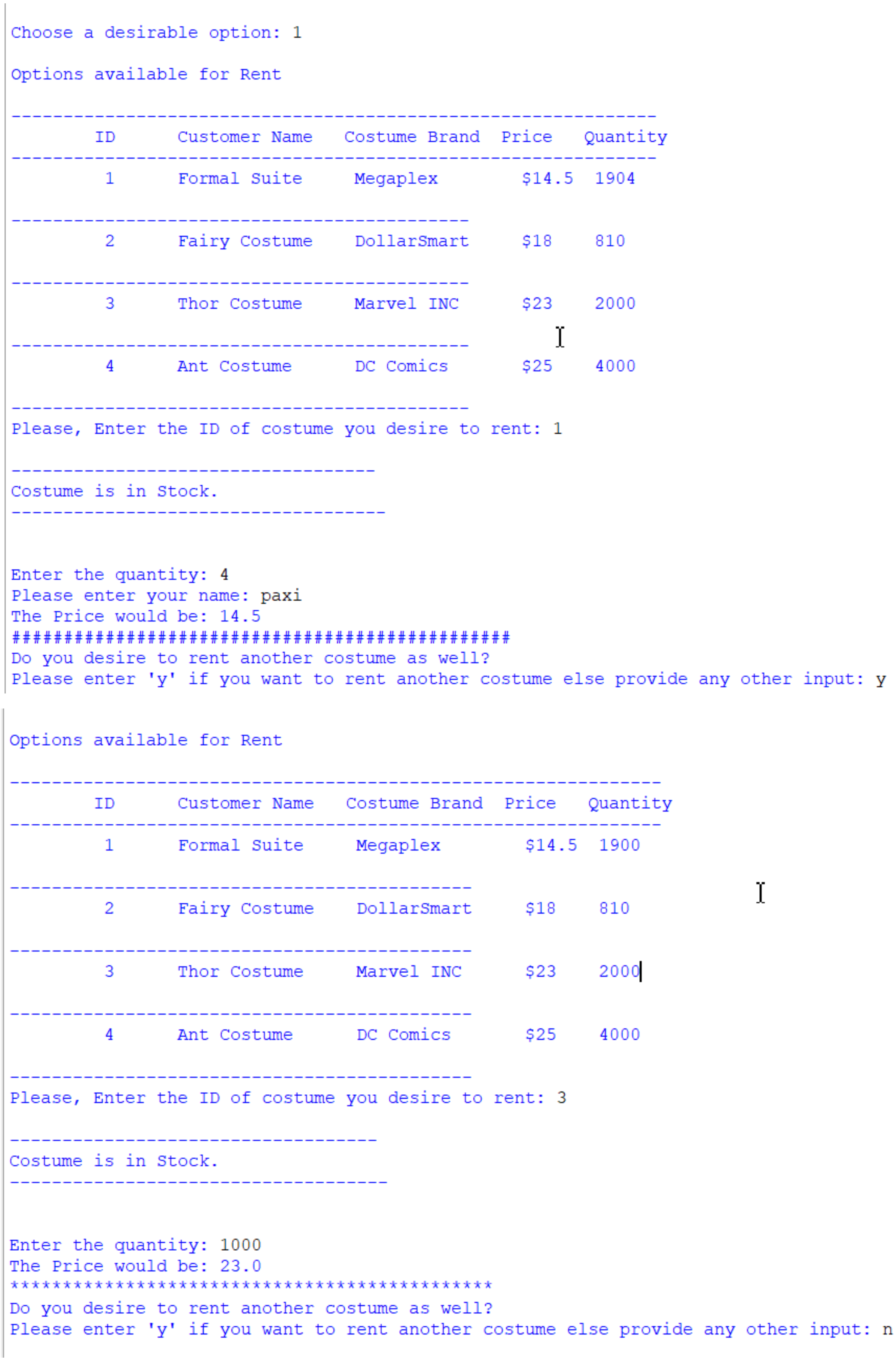
| Test | 5.5.1 |
|---|---|
| Objective | To show the update in stock of costume while renting. |
| Action | IDLE Shell was opened and multiple costumes were rented and change in stock was viewed. |
| Expected Output | The quantity of the stock after renting shall be reduced. |
| Actual Output | The quantity of the stock after renting was reduced. |
| Result | The test was successful |

*Table 6 Test table 5.5.1*



```
Formal Suite, Megaplex, $14.5,1904
Fairy Costume, DollarSmart, $18,810
Thor Costume, Marvel INC, $23,2000
Ant Costume, DC Comics, $25,4000
```

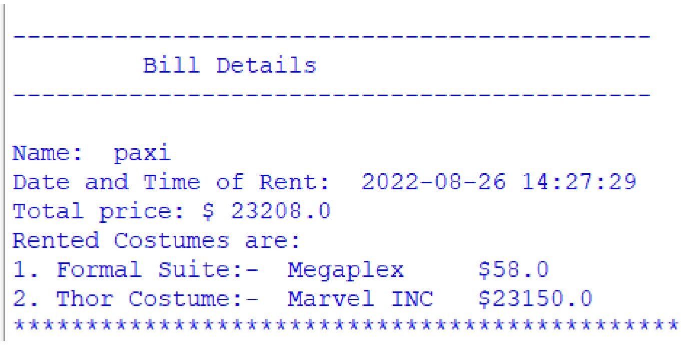
*Figure 41 Stock before Renting*



```
Choose a desirable option: 1

Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1904

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    810

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: 1

-----------------------------------
Costume is in Stock.
------------------------------------


Enter the quantity: 4
Please enter your name: paxi
The Price would be: 14.5
################################################
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: y
```

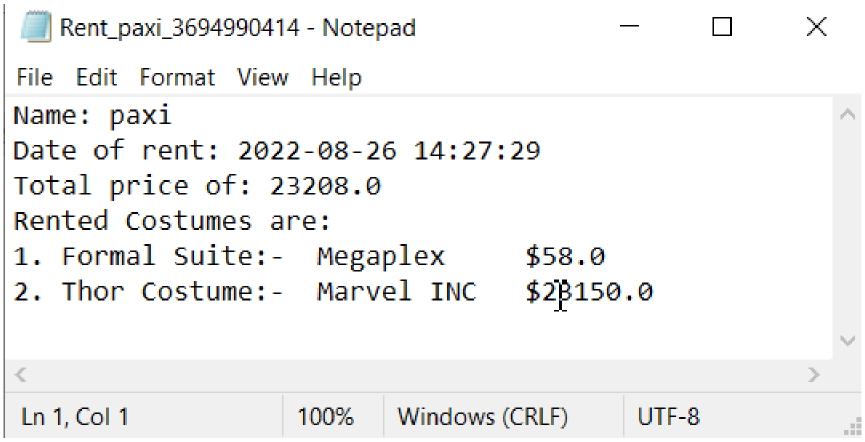
```
Options available for Rent

--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1900

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    810

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    2000

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to rent: 3

-----------------------------------
Costume is in Stock.
------------------------------------


Enter the quantity: 1000
The Price would be: 23.0
***********************************************
Do you desire to rent another costume as well?
Please enter 'y' if you want to rent another costume else provide any other input: n
```

*Figure 42 Screenshot of multiple renting*



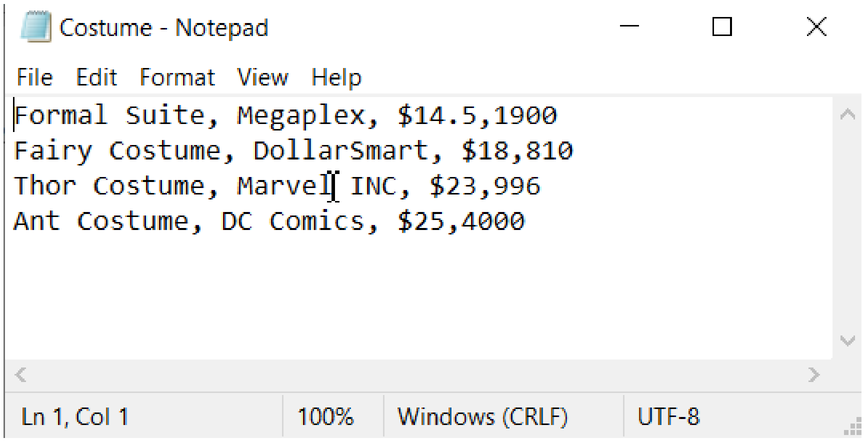
```
--------------------------------------------
         Bill Details
--------------------------------------------

Name:  paxi
Date and Time of Rent:  2022-08-26 14:27:29
Total price: $ 23208.0
Rented Costumes are:
1. Formal Suite:-  Megaplex     $58.0
2. Thor Costume:-  Marvel INC   $23150.0
**********************************************
```

*Figure 43 Screenshot of Bill*



```
Name: paxi
Date of rent: 2022-08-26 14:27:29
Total price of: 23208.0
Rented Costumes are:
1. Formal Suite:-  Megaplex    $58.0
2. Thor Costume:-  Marvel INC  $23150.0
```

*Figure 44 Screenshot of Detail of Bill in Text file*



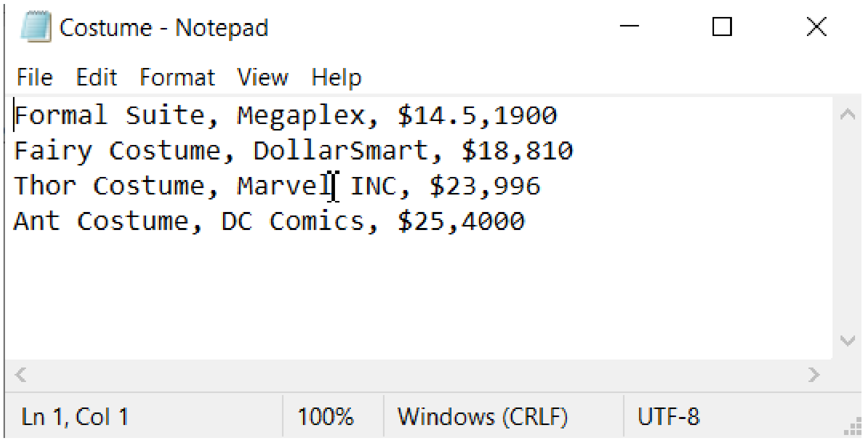
```
Formal Suite, Megaplex, $14.5,1900
Fairy Costume, DollarSmart, $18,810
Thor Costume, Marvel INC, $23,996
Ant Costume, DC Comics, $25,4000
```

*Figure 45 Screenshot of Stock after Renting*

### 5.5.2 For returning the costumes:

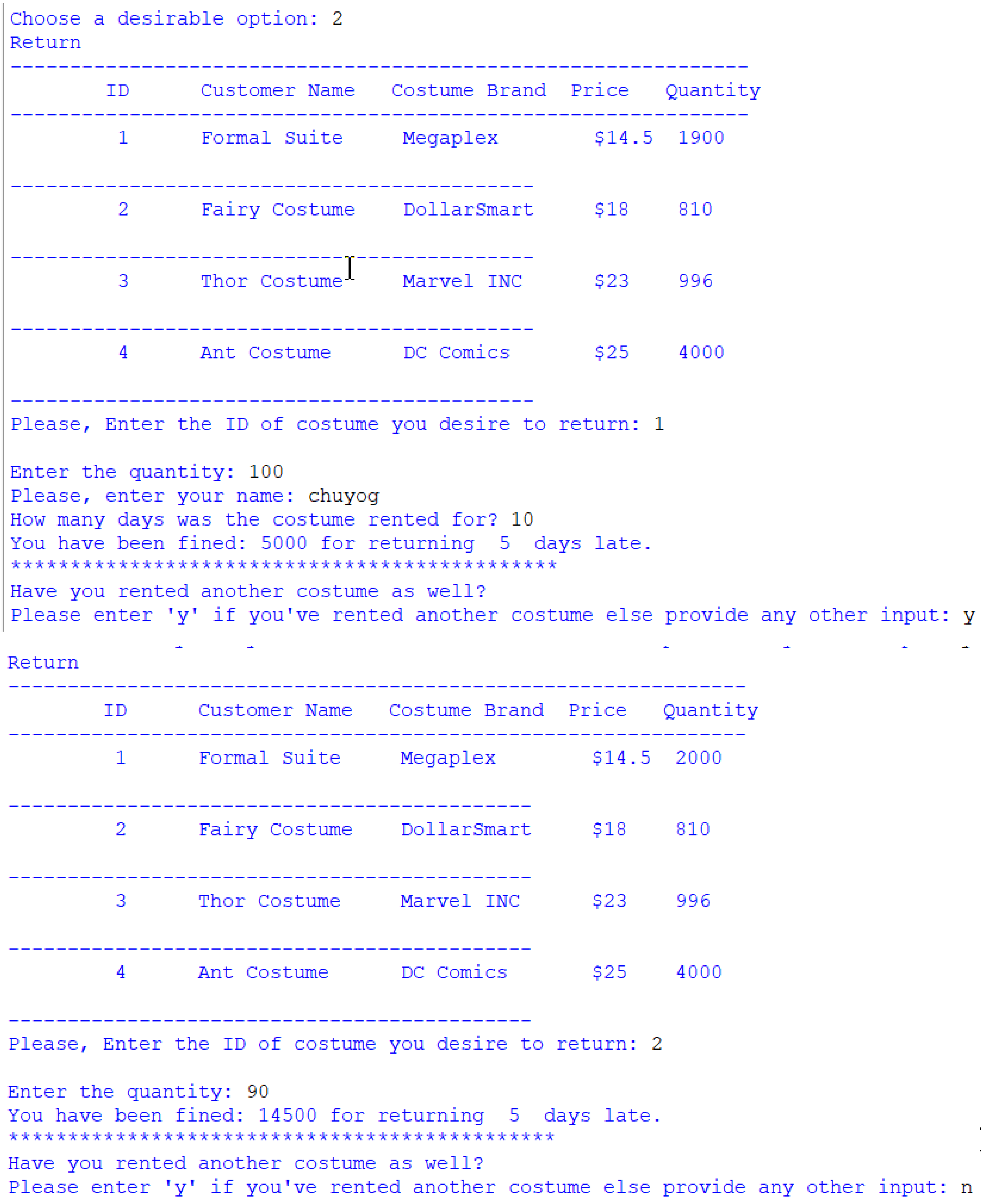
| | |
|---|---|
| Test | 5.2.2 |
| Objective | To show the update in stock of costume after returning. |
| Action | IDLE Shell was opened and multiple costumes were returned and change in stock was viewed. |
| Expected Output | The quantity of the stock after returning shall be increased. |
| Actual Output | The quantity of the stock after renting was increased. |
| Result | The test was successful |

*Table 7 Test Table 5.2.2*



```
Formal Suite, Megaplex, $14.5,1900
Fairy Costume, DollarSmart, $18,810
Thor Costume, Marvel INC, $23,996
Ant Costume, DC Comics, $25,4000
```

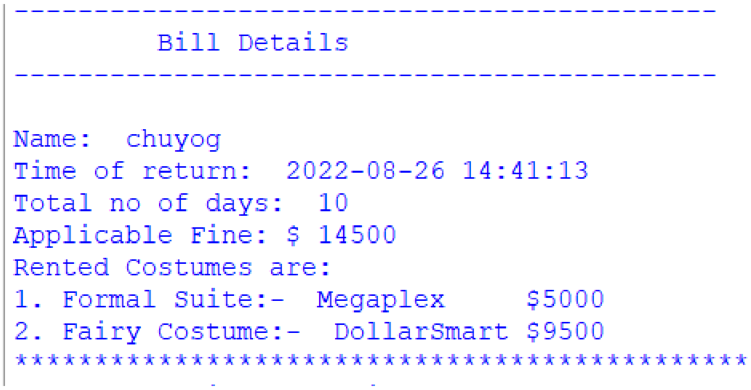
*Figure 46 Screenshot of Stock before returning*



```
Choose a desirable option: 2
Return
--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  1900

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    810

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    996

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to return: 1

Enter the quantity: 100
Please, enter your name: chuyog
How many days was the costume rented for? 10
You have been fined: 5000 for returning  5  days late.
**********************************************
Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input: y
```

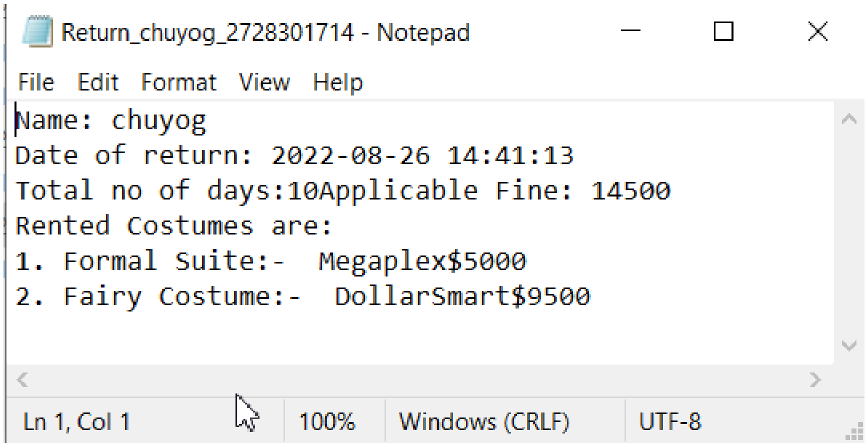
```
Return
--------------------------------------------------------------
        ID      Customer Name   Costume Brand  Price   Quantity
--------------------------------------------------------------
         1      Formal Suite     Megaplex        $14.5  2000

--------------------------------------------
         2      Fairy Costume    DollarSmart     $18    810

--------------------------------------------
         3      Thor Costume     Marvel INC      $23    996

--------------------------------------------
         4      Ant Costume      DC Comics       $25    4000

--------------------------------------------
Please, Enter the ID of costume you desire to return: 2

Enter the quantity: 90
You have been fined: 14500 for returning  5  days late.
**********************************************
Have you rented another costume as well?
Please enter 'y' if you've rented another costume else provide any other input: n
```

*Figure 47 Screenshot of multiples returning*



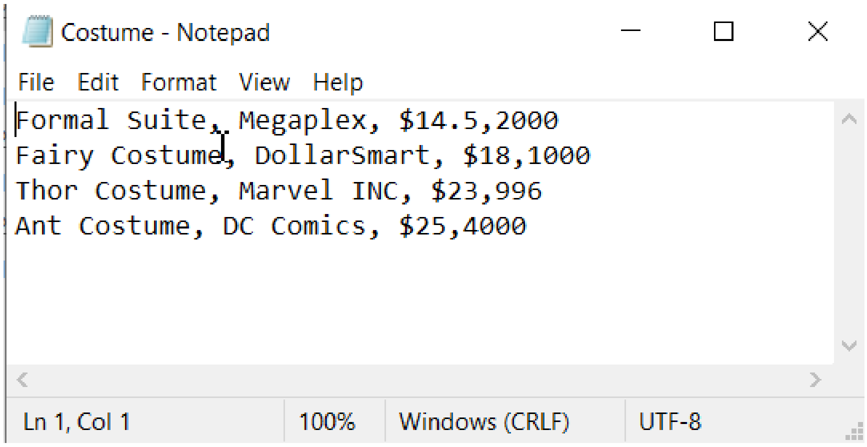
```
--------------------------------------------
         Bill Details
--------------------------------------------

Name:  chuyog
Time of return:  2022-08-26 14:41:13
Total no of days:  10
Applicable Fine: $ 14500
Rented Costumes are:
1. Formal Suite:-  Megaplex     $5000
2. Fairy Costume:-  DollarSmart $9500
**********************************************
```

*Figure 48 Screenshot of bill*


Return_chuyog_2728301714 - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

```
Name: chuyog
Date of return: 2022-08-26 14:41:13
Total no of days:10Applicable Fine: 14500
Rented Costumes are:
1. Formal Suite:-  Megaplex$5000
2. Fairy Costume:-  DollarSmart$9500
```

*Figure 49 Screenshot of Detail of return Bill text file creation*



```
Formal Suite, Megaplex, $14.5,2000
Fairy Costume, DollarSmart, $18,1000
Thor Costume, Marvel INC, $23,996
Ant Costume, DC Comics, $25,4000
```

*Figure 50 Screenshot of Stock after returning*

## References

asq, 2022. *WHAT IS A FLOWCHART?.* [Online]
Available at: https://asq.org/quality-resources/flowchart

CodeBerry, 2022. *How to run a program in Python IDLE – with examples.* [Online]
Available at: https://codeberryschool.com/blog/en/program-in-python-idle/

codevscolor, 2022. *Python set and built in methods : Python tutorial : Part 10.* [Online]
Available at: https://www.codevscolor.com/python-tutorial-part-10-python-set

ComputerHope, 2022. *Notepad.* [Online]
Available at: https://www.computerhope.com/jargon/n/notepad.htm

DataCamp, 2022. *DataTypes in Python.* [Online]
Available at: http://datacamp.com

GeeksforGeeks, 2022. [Online]
Available at: https://www.geeksforgeeks.org/python-while-loop/
[Accessed 22 August 2022].

GeeksforGeeks, 2022. *loops in python.* [Online]
Available at: https://www.geeksforgeeks.org/loops-in-python/

H2K infosys, 2022. *What is Python TUPLE ?.* [Online]
Available at: https://www.h2kinfosys.com/blog/what-is-python-tuple/

Programiz, 2022. *What is while loop in Python?.* [Online]
Available at: https://www.programiz.com/python-programming/while-loop

python, 2022. *IDLE.* [Online]
Available at: https://docs.python.org/3/library/idle.html

Python, 2022. *Python.* [Online]
Available at: https://www.python.org/

Sharma, R., 2022. *Data Structures & Algorithm in Python: Everything You Need to Know.* [Online]
Available at: https://www.upgrad.com/blog/data-structures-algorithm-in-python/#:~:text=Python%20algorithms%20are%20a%20set,guide%20the%20writing%20of%20algorithms.

TutorialsPoint, 2022. *Python - Functions.* [Online]
Available at: https://www.tutorialspoint.com/python/python_functions.htm#

UNF, 2022. *Pseudocode Examples.* [Online]

Available at: https://www.unf.edu/~broggio/cop2221/2221pseu.htm

w3schools, 2022. *Python For Loops.* [Online]

Available at: https://www.w3schools.com/python/python_for_loops.asp

# Appendix

## function.py

```
import datetime
def WelcomeNotice(): #message displayed to welcome user
    print("*********************************************")
    print("\tWelcome to our Application")
    print("*********************************************")
    print("\n")

def OptionSelectNotice(): #message displayed to select option
    print("Select a desirable option")
    print("(1) || Press 1 for renting a costume")
    print("(2) || Press 2 for returning a costume")
    print("(3) || Press 3 to exit.\n")

def ThankyouNotice(): # message displayed when exitted from program
    print("\nThank You for using our application.\n")

def InvalidNotice(): # message displayed in case of invalid input
    print("\nPlease enter a valid input")
    print("\nThe value shall be selected as per the provided options\n")

def RentC(): #message displayed containing options available to rent
    print("\nOptions available for Rent\n")

def AvailableCostume(): #message displayed when costume is available
    print("\n----------------------------------")
    print("Costume is in Stock.")
    print("-----------------------------------\n")
```

```
def UnavailableCostume(): #message displayed when costume is unavailable
    print("\n----------------------------------")
    print("Insufficient stock for the selected Costume.")
    print("-----------------------------------\n")


def InvalidExceptionNotice(): #message displayed when invalid input is given
    print("*********************************")
    print("Please provide a valid option")
    print("*********************************")


def RentDisplay(): # Displaying rent table
    file = open("Costume.txt","r")
    print("---------------------------------------------------------------")
    print("\tID \tCustomer Name   Costume Brand  Price   Quantity")
    print("---------------------------------------------------------------")
    IDcounter = 0
    for line in file:
        IDcounter = IDcounter + 1
        print("\t", IDcounter, "\t" + line.replace(",","\t"))
        print("---------------------------------------------")
    file.close()


def dictionaryRent(): #creating dictionary
    file = open("Costume.txt", "r") #opening text file
    IDcounter = 0
    dictionary_Costume = {}
    for line in file:
        IDcounter = IDcounter + 1
        line = line.replace("\n","")
        line = line.split(',')

        dictionary_Costume[IDcounter] =  line

    return dictionary_Costume
    file.close() #closing text file


def ReturnC(): #message displayed during return
    print("Return")

def valid_costumeIDr(): #valid costume ID during return
    IsException = False #exception handling
    while IsException == False:
        try:
            validCostumeIDr = int(input("Please, Enter the ID of costume you desire to return: "))
            IsException = True
        except:
            InvalidExceptionNotice()

    while validCostumeIDr <= 0 or validCostumeIDr > len(dictionaryRent()):
        print("\nPlease provide a valid costume ID !\n")
        RentDisplay()
        validCostumeIDr = int(input("\nPlease, Enter the ID of costume you desire to return: "))
    return validCostumeIDr


def valid_costumeID(): #valid costume ID during rent
```

```
    IsException = False
    while IsException == False:
        try:
            validCostumeID = int(input("Please, Enter the ID of costume you desire to rent: 
"))
            IsException = True
        except:
            InvalidExceptionNotice()

    while validCostumeID <= 0 or validCostumeID > len(dictionaryRent()):
        print("\nPlease provide a valid costume ID !\n")
        RentDisplay()
        validCostumeID = int(input("\nPlease, Enter the ID of costume you desire to rent: 
"))
    return validCostumeID

def RentQuantity(stockquantity): # checking the quantity of costume during rent
    IsException = False
    while IsException == False:
        try:
            quantityCostume = int(input("\nEnter the quantity: "))
            IsException = True
        except:
            InvalidExceptionNotice()

    while quantityCostume <=0 or quantityCostume > stockquantity:
        if quantityCostume <= 0:
            print("\n----------------------------------------")
            print("Please provide a valid quantity.") #message if invalid input
            print("----------------------------------------\n")
        else:
            print("\n----------------------------------------")
            print("Provided quantity is greater than the stock quantity.") #message if input is greater than stock
            print("----------------------------------------\n")

        IsException = False
        while IsException == False:
            try:
                quantityCostume = int(input("\nEnter the quantity: "))
                IsException = True
            except:
                InvalidExceptionNotice()
    return quantityCostume

def ReturnQuantity(c): # checking the quantity of costume during rent
    IsException = False
    while IsException == False:
        try:
            quantityCostume = int(input("\nEnter the quantity: "))
            IsException = True
        except:
            InvalidExceptionNotice()

    while quantityCostume <=0:
        if quantityCostume <= 0:
            print("\n----------------------------------------")
            print("Please provide valid quantity.")#message if invalid input
            print("----------------------------------------\n")
        IsException = False
        while IsException == False:
            try:
                quantityCostume = int(input("\nEnter the quantity: "))
                IsException = True
            except:
                InvalidExceptionNotice()
    return quantityCostume

def StockCostume(dictionary):
    file = open("Costume.txt","w")
    for i in dictionary.values():
        line = str(i[0] + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]))
        file.write(line)
        file.write("\n")
    file.close()

def CalculatePrice(dictionary, quantitydetails, costumeID): #display price
    Price = float(dictionary[costumeID][2].replace("$",""))
    print("The Price would be:", Price)
    pricePItem = Price * quantitydetails
    return pricePItem

def BillofRent(name, todaysdate, totalprice, rentCostumename, rentCostumebrand,
Price): #display bill details
    print("\n--------------------------------------------")
    print("\t Bill Details")
    print("--------------------------------------------\n")
    print("Name: ", name)
    print("Date and Time of Rent: ", todaysdate)
    print("Total price: " + "$", totalprice)
    print("Rented Costumes are: ")
    for x in range(len(rentCostumename)):
        print(str(x+1) + ". " + rentCostumename[x] + ":- " + rentCostumebrand[x] + "\t$"+
str(Price[x]))
    print("************************************************")

def Create_DetailofRent(cname, date, total, costumename, costumebrand, Price):
    fileName = "Rent_" + cname + "_" + str(datetime.datetime.now().second) +
str(datetime.datetime.now().microsecond) + str(datetime.datetime.now().hour) + ".txt"
#creating detail of rent bill in text file
    file = open(fileName, "w")
    file.write("Name: " + cname + "\n")
    file.write("Date of rent: " + str(date) + "\n")
    file.write("Total price of: " + str(total) + "\n")
    file.write("Rented Costumes are: " + "\n")
    for i in range(len(costumename)):
        file.write(str(i+1) + ". " + costumename[i] + ":- " + costumebrand[i] + "\t$"+
str(Price[i]) + "\n")
    file.close()

def BillofReturn(name, todaysdate, days, fine, costumename, costumebrand, listA):
#creating bill during return
    print("\n-----------------------------------------------")
    print("\t Bill Details")
    print("-----------------------------------------------\n")
    print("Name: ", name)
    print("Time of return: ", todaysdate)
    print("Total no of days: ", days)
    print("Applicable Fine: " + "$", fine)
    print("Rented Costumes are: ")
    for i in range(len(costumename)):
        print(str(i+1) + ". " + costumename[i] + ":- " + costumebrand[i] + "\t$" + str(listA[i]))
    print("************************************************")
```

```
def Create_DetailofReturn(cname, date, days, fine, costumeName, costumeBrand,
listA):
    fileName = "Return_" + cname + "_" + str(datetime.datetime.now().second) +
str(datetime.datetime.now().microsecond) + str(datetime.datetime.now().hour) + ".txt"
#creating detail of return bill in text file
    file = open(fileName, "w")
    file.write("Name: " + cname + "\n")
    file.write("Date of return: " + str(date) + "\n")
    file.write("Total no of days:" + str(days))
    file.write("Applicable Fine: " + str(fine) + "\n")
    file.write("Rented Costumes are: " + "\n")
    for x in range(len(costumeName)):
        file.write(str(x+1) + ". " + costumeName[x] + ":- " + costumeBrand[x] + "$" +
str(listA[x]) +"\n")
    file.close()
```

### main.py

```
import function #importing function.py
import datetime
function.WelcomeNotice() #calling Welcome message
contLoop = True
while contLoop == True:

    function.OptionSelectNotice() #calling Option selection message

    IsException = False
    while IsException == False:
        try:
            n = int(input("Choose a desirable option: ")) #Taking input from the user
            IsException = True
        except:
            function.InvalidExceptionNotice()
            function.OptionSelectNotice()



    if n == 1: # option to rent
        function.RentC() #call options available message
        function.RentDisplay() # call rent table
        function.dictionaryRent() #call dictionary

        dictionary_Costume = function.dictionaryRent() #checking

        rentCostumeID = function.valid_costumeID() #checking

        rentCostumeA = [] #list creation
        rentCostumeC = [] #list creation
        Price = [] #list creation
        if int(dictionary_Costume[rentCostumeID][3]) > 0:
            function.AvailableCostume()

            quantityCostume =
function.RentQuantity(int(dictionary_Costume[rentCostumeID][3]))
            dictionary_Costume[rentCostumeID][3] =
int(dictionary_Costume[rentCostumeID][3]) - quantityCostume

            name = input("Please enter your name: ") #taking user input
            todaysdate = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

            rentCostumeA.append(dictionary_Costume[rentCostumeID][0]) #adding to list
            rentCostumeC.append(dictionary_Costume[rentCostumeID][1])   #adding to list

            function.StockCostume(dictionary_Costume)

            totalPrice =
function.CalculatePrice(dictionary_Costume,quantityCostume,rentCostumeID)
            Price.append(totalPrice) #adding to list

            print("#######################################################")
            print("Do you desire to rent another costume as well?") #prompt
            contrent = input("Please enter 'y' if you want to rent another costume else
provide any other input: ").lower()

            loop = True
            while loop == True:
                if contrent == "y":
                    function.RentC()
                    function.RentDisplay()
```

```
          dictionary_Costume = function.dictionaryRent()
          rentCostumeID = function.valid_costumeID()

          if int(dictionary_Costume[rentCostumeID][3]) > 0:
            function.AvailableCostume()




            if dictionary_Costume[rentCostumeID][0] in rentCostumeA:
              quantityCostume =
function.RentQuantity(int(dictionary_Costume[rentCostumeID][3])) + quantityCostume

            else:
              rentCostumeA.append(dictionary_Costume[rentCostumeID][0])
              rentCostumeC.append(dictionary_Costume[rentCostumeID][1])
              quantityCostume =
function.RentQuantity(int(dictionary_Costume[rentCostumeID][3])) + quantityCostume

            dictionary_Costume[rentCostumeID][3] =
int(dictionary_Costume[rentCostumeID][3]) - quantityCostume
            function.StockCostume(dictionary_Costume)

            totalprice =
function.CalculatePrice(dictionary_Costume,quantityCostume,rentCostumeID) +
totalPrice
            Price.append(totalprice)
            totalPrice = totalprice + totalPrice
            print("****************************************************")
            print("Do you desire to rent another costume as well?")
```

```
                        contrent = input("Please enter 'y' if you want to rent another costume
else provide any other input: ").lower()
                    else:
                        function.UnavailableCostume()
                        function.BillofRent(name, todaysdate, totalPrice, rentCostumeA,
rentCostumeC, Price) #calling rent bill
                        function.Create_DetailofRent(name, todaysdate, totalPrice,
rentCostumeA, rentCostumeC, Price) #calling detail of rent bill
                        loop = False

                else:
                    function.BillofRent(name, todaysdate, totalPrice, rentCostumeA,
rentCostumeC, Price)
                    function.Create_DetailofRent(name, todaysdate, totalPrice, rentCostumeA,
rentCostumeC, Price)
                    loop = False

        else:
            function.UnavailableCostume() #calling unavailablecostume notice



    elif n == 2: #option to return
        function.ReturnC() #calling functions
        function.RentDisplay()
        dictionary_Costume = function.dictionaryRent()
        returnCostumeID = function.valid_costumeIDr()

        returnCostumeA = [] #creating lists
        returnCostumeC = []
        listA = []
        quantityCostume = function.ReturnQuantity(int(dictionary_Costume[returnCostumeID][3]))
        dictionary_Costume[returnCostumeID][3] = int(dictionary_Costume[returnCostumeID][3]) + quantityCostume

        name = input("Please, enter your name: ") #taking input
        todaysdate = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

        returnCostumeA.append(dictionary_Costume[returnCostumeID][0]) #adding to lists
        returnCostumeC.append(dictionary_Costume[returnCostumeID][1])

        function.StockCostume(dictionary_Costume)
        IsException = False
        while IsException == False:
            try:
                days = int(input("How many days was the costume rented for? ")) #Taking input from the user
                IsException = True
            except:
                function.InvalidExceptionNotice()



        fine = 0
        lateday = 0
        finep = 10
        if days > 5: #condition
            lateday = days - 5
            fine = lateday * finep * quantityCostume
            print("You have been fined:", fine, "for returning ", lateday, " days late.")
            listA.append(fine) #adding to listA
        print("*****************************************************")
        print("Have you rented another costume as well?")
        contrent = input("Please enter 'y' if you've rented another costume else provide any
    other input: ").lower()

        loop = True
        while loop == True:
            if contrent == "y":
                function.ReturnC()
                function.RentDisplay()
                dictionary_Costume = function.dictionaryRent()
                returnCostumeID = function.valid_costumeIDr()


                if dictionary_Costume[returnCostumeID][0] in returnCostumeA:
                    quantityCostume =
    function.ReturnQuantity(int(dictionary_Costume[returnCostumeID][3])) +
    quantityCostume
                    dictionary_Costume[returnCostumeID][3] =
    int(dictionary_Costume[returnCostumeID][3]) + quantityCostume

                else:
                    returnCostumeA.append(dictionary_Costume[returnCostumeID][0])
                    returnCostumeC.append(dictionary_Costume[returnCostumeID][1])
                    quantityCostume =
    function.ReturnQuantity(int(dictionary_Costume[returnCostumeID][3])) +
    quantityCostume
                    dictionary_Costume[returnCostumeID][3] =
    int(dictionary_Costume[returnCostumeID][3]) + quantityCostume

                function.StockCostume(dictionary_Costume)
```

```
                Fine = (lateday * finep * quantityCostume)
                fine = Fine + fine
                listA.append(Fine)
                print("You have been fined:", fine, "for returning ", lateday, " days late.")

                print("**************************************************")
                print("Have you rented another costume as well?")
                contrent = input("Please enter 'y' if you've rented another costume else
provide any other input: ").lower()

            else:
                function.BillofReturn(name, todaysdate, days, fine, returnCostumeA,
returnCostumeC, listA) #calling rentbill
                function.Create_DetailofReturn(name, todaysdate, days, fine,
returnCostumeA, returnCostumeC, listA) #calling detail of rent bill
                loop = False



    elif n == 3: #option to exit
        function.ThankyouNotice() #calling thank you message
        contLoop = False

    else:
        function.InvalidNotice() #invalid option selection
```