LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: ADARSHA PANDEY

Group: C12

London Met ID: 20048813

College ID: NP01CP4S210301

Assignment Due Date: 10th SEP 2021

Assignment Submission Date: 9th SEP 2021

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.

Contents

List of Figures

List of Tables

INTRODUCTION:

This is the first coursework assigned to us for Fundamental (Computing facility). This coursework was given to us in week 6 which carries 60% of total marks in final. In this coursework we developed a library management system where we have to store books. If user wants to borrow books then we have to create a borrowed file. Moreover, if the user wants to return a book then we have to charge fine if the user was delayed to return the book.

Python is one of the open source and high level language which is used for developing desktop GUI applications, websites, games and many more. **(medium.com, n.d.**)

Python has one of the simplex syntax languages which can be understand by anyone. It was first released in 1991at Centrum Wiskunde and Informatica(CWI). The latest version of python is 3.7.6 which was released in 18 December 2019. It can be used in any platforms like Windows, Linux, Mac etc.

IDLE (Integrated Development and Learning Environment) is an integrated development environment for python. It helps to create, edit, and execute the files. It is one of the best text editors which also have features like syntax highlighting, code completion and auto-indentation. Furthermore, it also provides some tools which helps to debug the program**. (tutorialsteacher, n.d.)**

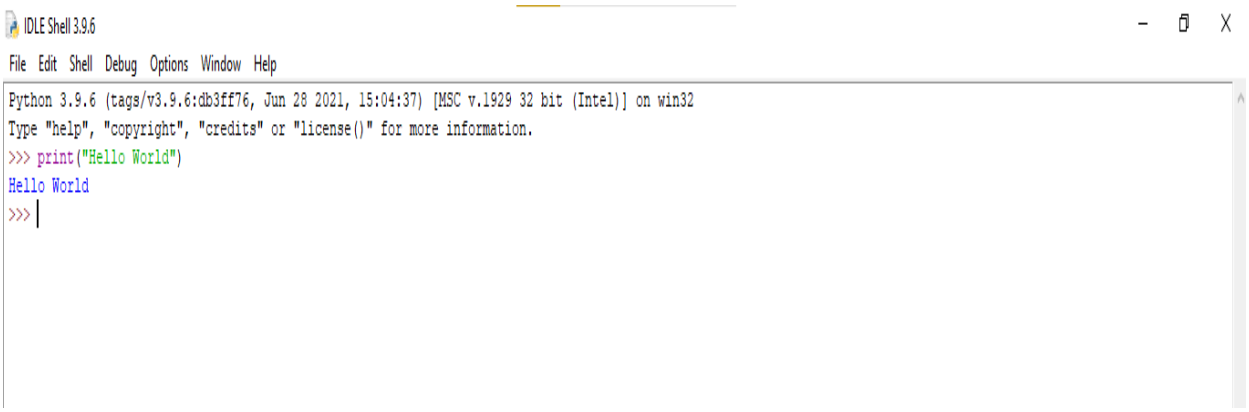

Figure 1: IDLE Shell

GOALS AND OBJECTIVE:

The main aim and goal of this coursework is:

- To make a systematic and user friendly library management system where users can borrow and return the books.
- To store the automatic borrow and return users personal details in a note.
- To decrease in stock after borrowing the books and increase after returning the book
- To charge some amount of fine if the user is delayed to return some books.

ALGORITHM:

It is a set steps defining how to find the solution of the program.

STEP 1: Start

STEP 2: Display “Welcome to the Library Management System”

STEP 3: ask the user to enter 1 to display books, enter 2 to borrow books, enter 3 to return books and enter 4 to exit the program.

STEP 4: If 1 is entered, then go to the STEP 6

Else if, 2 is entered then go to STEP 7

Else if,3 is entered then go to STEP 15

Else if, 4 is entered then go to STEP 24

STEP 5: If the entered number is not 1,2,3,4 then display the message and then go the Step 3.

STEP 6: If the user enters No. 1 it will display all the books from books.txt then go the step 3.

STEP 7: When the user enters No 2 if it will be true it will go to the borrow process else it will show the exception message.

STEP 8: Assign the first and last name of the borrower.

STEP 9: if the name is invalid it will go the Step 8 otherwise it will go the Step 10.

STEP 10: Creates a .txt file which will store the details of the borrower.

STEP 11: Displays all the books from books.txt to chooser the books for the user.

STEP 12: If the user entered book number is valid it will go to the Step 13 otherwise it will go to the Step 11 to display all the books again.

STEP 13: Display Do you want to borrow more books? Press y for yes and n for nope.

STEP 14: If user entered y then go to Step 11 else go to the Step 3.

STEP 15: When the user enters no 3 Return process will initiate else it will show the exception message.

STEP 16: Assign the name of the borrower.

STEP 17: if the name is invalid it will go the Step 16 otherwise it will go the Step 18.

STEP 18: Creates a .txt file which will store the details of the return user.

STEP 19: Displays Is the book return date has already been expired? Press Y for Yes and N for Nope

YES: Go to the Step 20

No: Go the the Step 3.

STEP 20: Displays By how many days was the book returned late?

STEP 21: Input how many days the book was delayed.

STEP 22: Then, it will calculate the fine by [fine = 2 * days expired].

STEP 23: Displays the total fee and go to the Step 3.

STEP 24: When the user enter No 4 the whole program will terminate.

STEP 25: Display “Thanks for visiting our library”

STEP 26: Stop

FLOWCHART

It is the graphical representation of the algorithm.

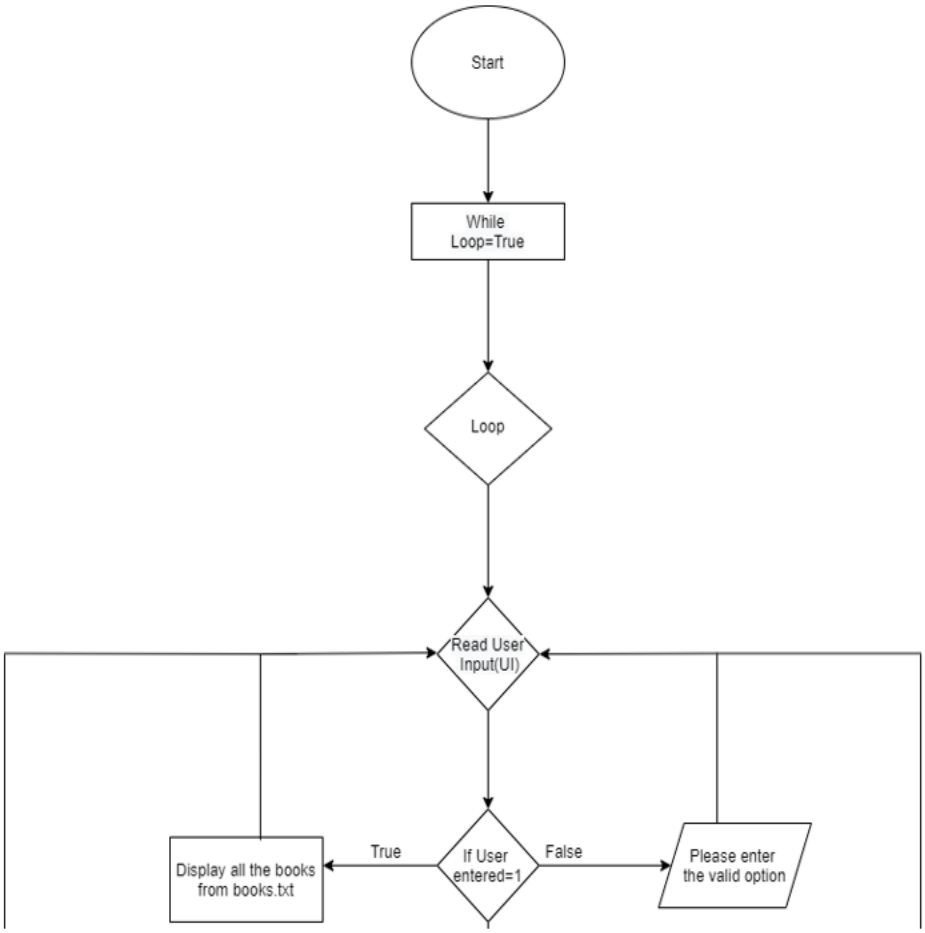

Figure 2: Flow Chart 1

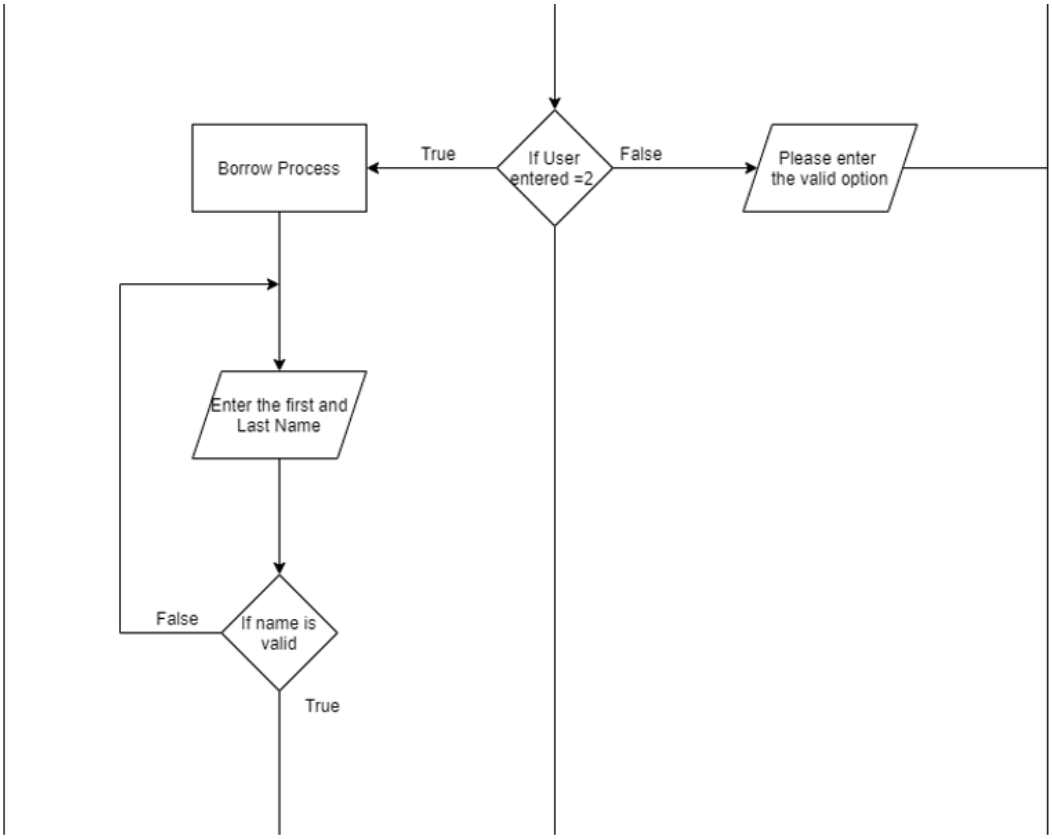

Figure 3: Flow Chart 2

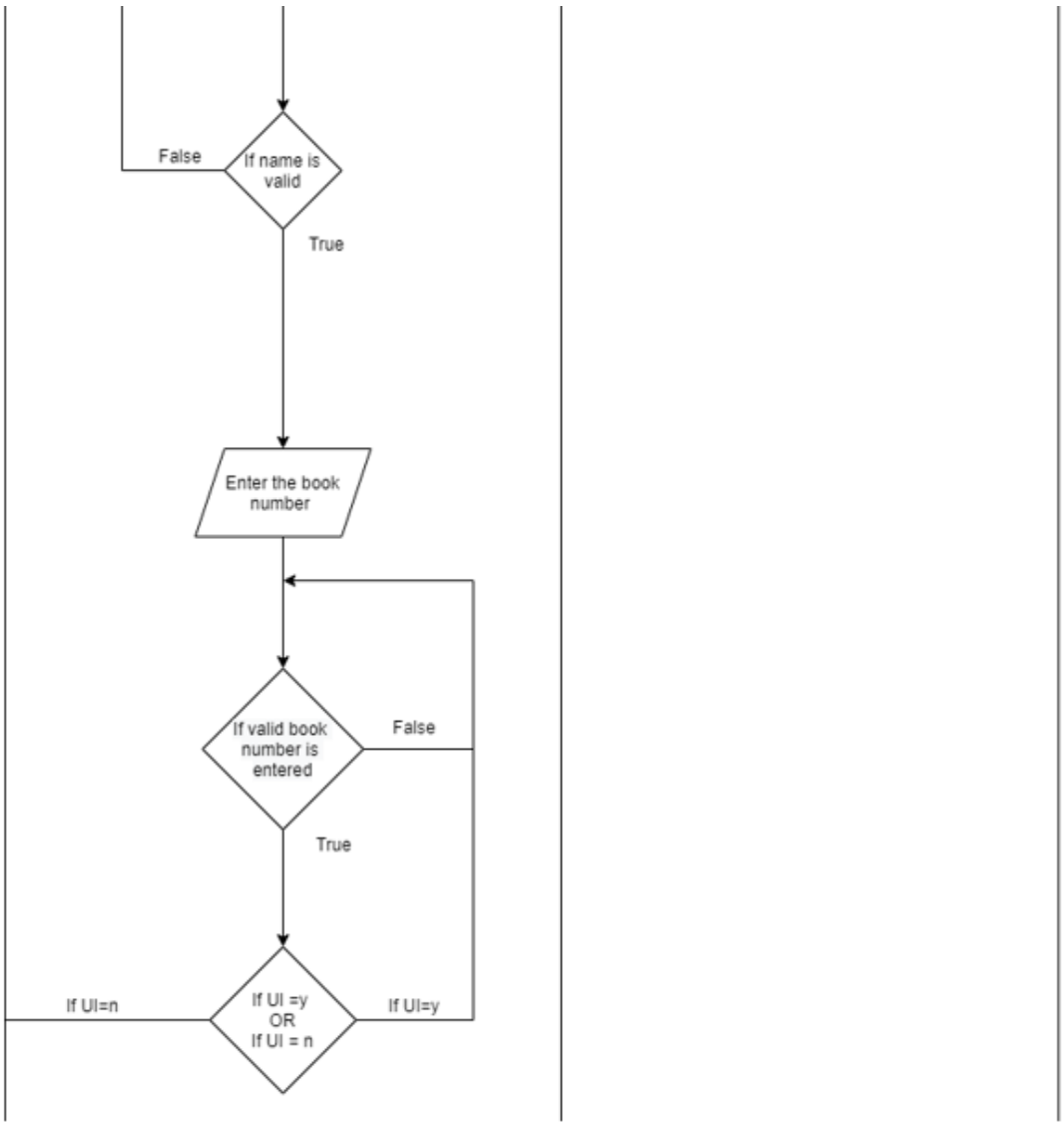

Figure 4: Flow Chart 3

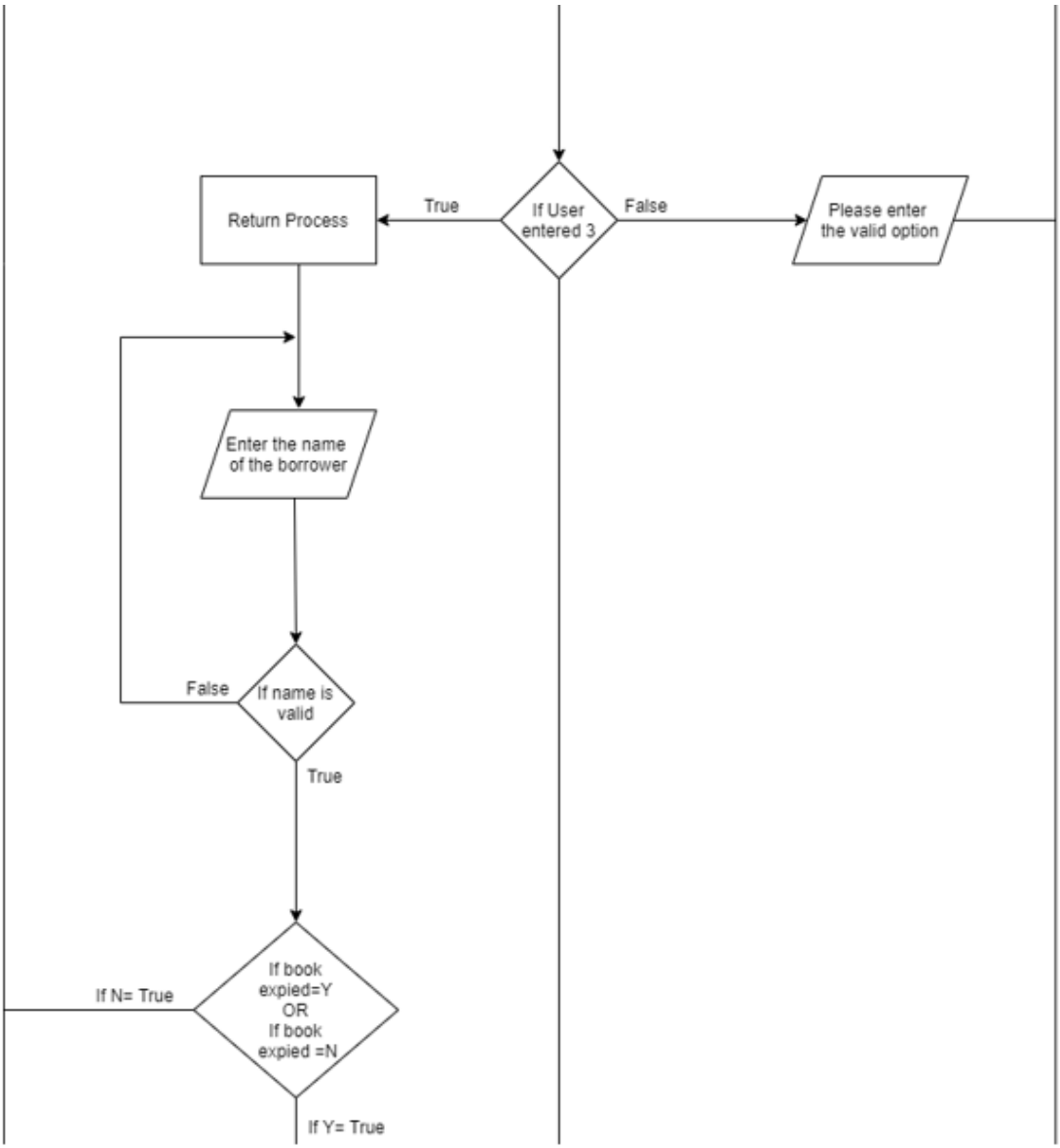

Figure 5: Flow Chart 4

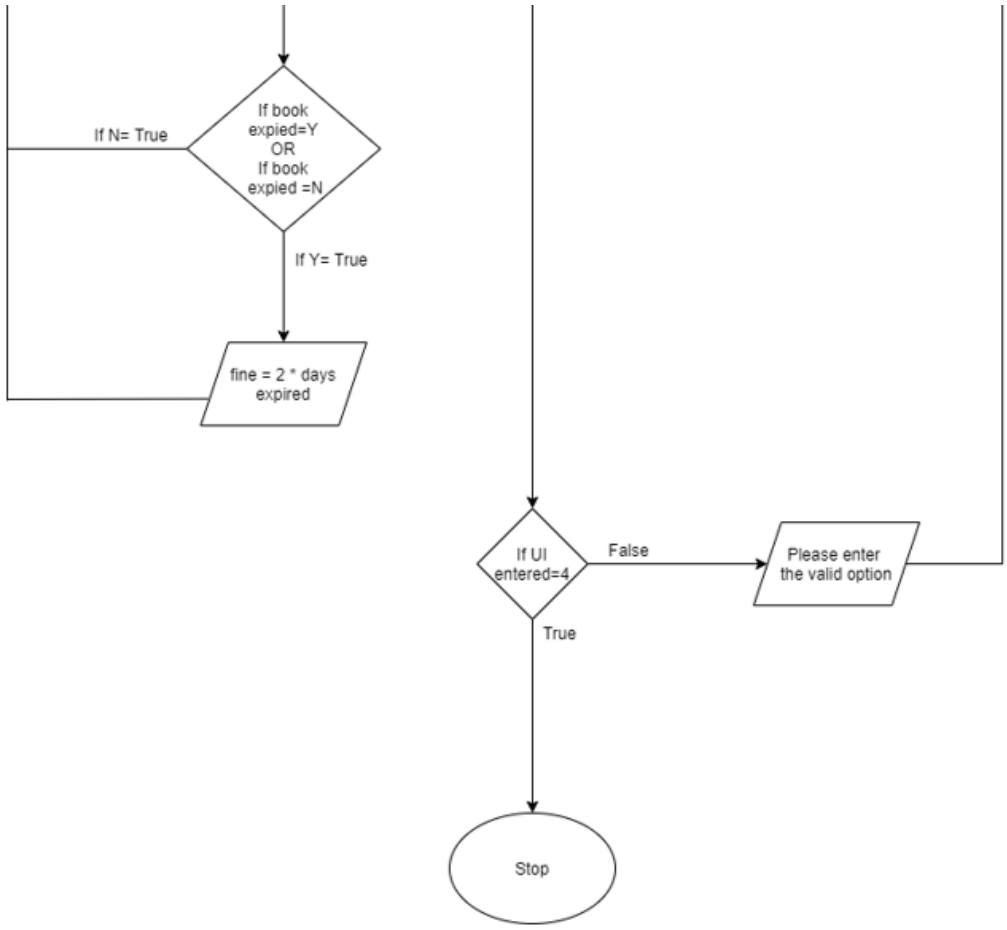

Figure 6: Flow Chart 5

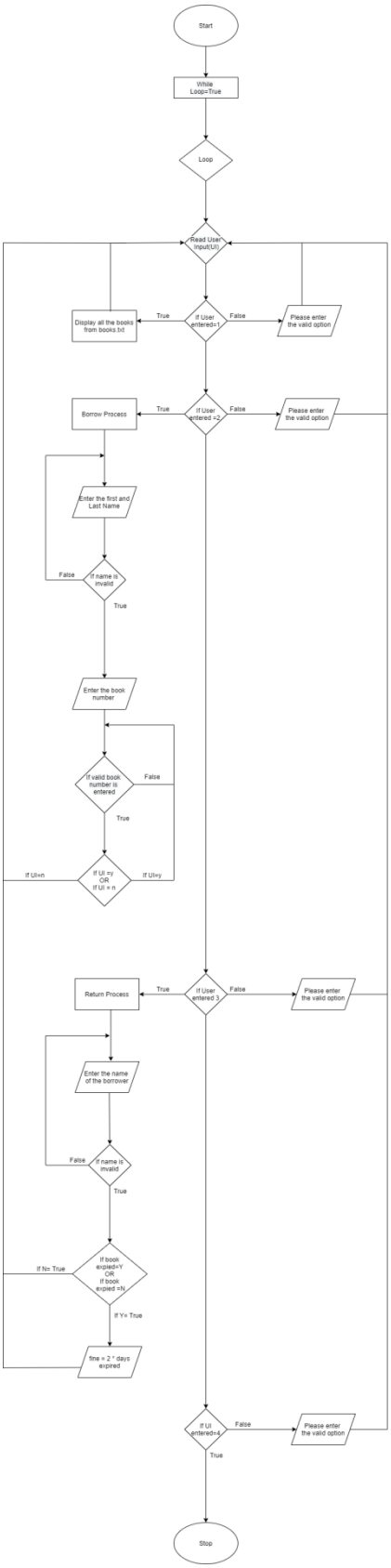

Figure 7: main Flow Chart

PSEUDOCODE:

It is an artificial language that helps the programmers to develop algorithms.

PSEUDOCODE OF MAIN:

```
START
import Return
import List
import DateAndTime
import Borrow

ALGORITHM start():
   WHILE (True):
      OUTPUT ("       Welcome to our library              ")
      OUTPUT ("-----------------------------------------------")
      OUTPUT ("Enter 1. Display available books in library")
      OUTPUT ("Enter 2. Borrow a Book from our library")
      OUTPUT ("Enter 3. Return a Book to out library")
      OUTPUT ("Enter 4. Exit")
    ENDWHILE
      TRY:
         a = int( INPUT ("Select a choice from 1-4: "))
         OUTPUT ()
               IF (a==1):
                     with open("books.txt","r") as f:
                     lines=f.read()
                     OUTPUT (lines)
                      OUTPUT ()
               ENDIF
               ELSE IF (a==2):
                     CALL list algorithm from List module
```

```
            CALL borrowBooks algorithm from Borrow module
        ENDELSE
        ELSE IF (a==3):
            CALL list algorithm from List module
            CALL returnBooks algorithm from Return module
        ENDELSE
        ELSE IF (a==4):
             OUTPUT  ("Thank you for visiting our Library. Pls visit us again")
            break
        ENDELSE
         ELSE:
            OUTPUT ("Please enter a valid option from 1-4")
        ENDELSE
    EXCEPT ValueError:
        OUTPUT ("Please input as suggested.")
END ALGORITHM
START
    CALL ALGORITHM start
END
```

PSEUDOCODE OF DATE ANDTIME

```
START
    ALGORITHM getDate():
        import datetime
         now = datetime.datetime.now
        RETURN str(now().date())

    ALGORITHM getTime():
         import datetime
         now = datetime.datetime.now
        RETURN str(now().time())
```

END ALGORITHM

PSUEDOCODE OF LIST

START

ALGORITHM List():

DECLARE bookName

DECLARE authorName

DECLARE quantity

DECLARE price

INITIALIZE bookName=[]

INITIALIZE authorName=[]

INITIALIZE quantity=[]

INITIALIZE price=[]

with open("books.txt","r") as f:

lines=f.readlines()

lines=[x.strip('\n') for x in lines]

FOR i in range(len(lines)):

ind=0

FOR a in lines[i].split(','):

IF (**INITIALIZE** ind==0):

bookName.append(a)

ENDIF

ELSE IF (ind==1):

authorName.append(a)

ENDELSE

ELSE IF (ind==2):

quantity.append(a)

ENDELSE

ELSE IF (ind==3):

```
                                price.append(a.strip("$"))
                            ENDELSE
                             ind+=1
END ALGORITHM
START
    CALL ALGORITHM List
END
```

PSEUDOCODE OF BORROW

```
START
import DateAndTime
import List

ALGORITHM borrowBooks():
    borrowCode=False
    WHILE (True):
        firstName=input("Enter the First Name: ")
        IF firstName.isalpha():
            break
        OUTPUT ("please enter a valid first name")
        ENDIF
    ENDWHILE
     WHILE (True):
        lastName=input("Enter the Last Name: ")
        IF lastName.isalpha():
            break
        OUTPUT ("please enter a valid last name")
        ENDIF
    t="Borrowed by-"+firstName+".txt"
    with open(t,"w+") as f:
        f.write("Library Management System  \n")
```

```
        f.write("Borrowed By: "+ firstName+" "+lastName+"\n")
        f.write("Date: " + DateAndTime.getDate()+"    Time:"+DateANDTime.
        .getTime()+"\n\n")
        f.write("S.N. \t\t Bookname \t      Authorname \t\t price \n" )
    ENDWHILE
    WHILE borrowCode==False:
        OUTPUT ("Please select a option below:")
        FOR i in range(len(List.bookName)):
                OUTPUT Enter", i, "to borrow book", List.bookName[i])
    TRY:
        a = int (INPUT())
        TRY:
            IF (int(List.quantity[a])>0):
            OUTPUT ("Book is available in our library")
            with open(t,"a") as f:
            f.write("1. \t\t"+ List.bookName[a]+"\t\t
            "+List.authorName[a]+"\t\t  "+"$"+List.price[a]+"\n")

            List.quantity[a]=int(List.quantity[a])-1
            with open("books.txt","w+") as f:
             FOR i in range(3):

            f.write(List.bookName[i]+","+List.authorName[i]+","+str(List.quantity[
            i])+","+"$"+List.price[i]+"\n")
             loop=True
            count=1
        ENDWHILE
            WHILE loop==True:
            option= str(INPUT("Do you want to borrow more books?Press y for
            yes and n for nope."))
             IF (option.upper()=="Y"):
```

```
                count= count+1
                OUTPUT ("Please select an option below:")
                 FOR i in range(len(List.bookName)):
                    OUTPUT ("Enter", i, "to borrow book", List.bookName[i])
                   a=int(input())
            ENDIF
                   IF (int(List.quantity[a])>0):
                        OUTPUT ("Book is available in our library")
                        with open(t,"a") as f:
                       f.write(str(count) +". \t\t"+ List.bookName[a]+"\t\t
                       "+List.authorName[a]+"\t\t  "+"$"+List.price[a]+"\n")

          List.quantity[a]=int(List.quantity[a])-1
           with open("books.txt","w+") as f:
               FOR i in range(3):

                   f.write(List.bookName[i]+","+List.authorName[i]+","+str(List.q
                   uantity[i])+","+"$"+List.price[i]+"\n")
                   borrowCode =False
       ENDIF
    ENDWHILE
        ELSE:
          loop=False
          break
         ENDELSE
   ELSE IF(option.upper()=="N"):
        OUTPUT ("Thank you for borrowing books from our library ")
       OUTPUT ("")
       loop=False
       borrowCode=True
    END ELSE
```

```
            ELSE IF:
                OUTPUT ("Please choose as instructed")
            END ELSE
        ELSE IF
            OUTPUT ("Book is not available in our library. Pls check other books")
            borrowBooks()
            borrowCode=False
        END ELSE
        EXCEPT IndexError:
          OUTPUT ("")
          OUTPUT ("Please choose book according to the given instruction. Thank You")
      EXCEPT ValueError:
        OUTPUT ("")
        OUTPUT ("Please choose as given suggested.")
END ALGORITHM
START
        CALL ALGORITHM borrowBooks
END
```

PSEUDOCODE OF RETURN

```
START
import List
import DateAndTime
ALGORITHM returnBooks():
   name=input("Enter the first name of borrower: ")
   a="Borrowed by-"+name+".txt"
   TRY:
        with open(a,"r") as f:
         lines=f.readlines()
         lines=[a.strip("$") for a in lines]
```

```
        with open(a,"r") as f:
            data=f.read()
            OUTPUT (data)
    EXCEPT:
       OUTPUT ("The borrower name is incorrect. Pls check once")
        returnBooks()

    b="Returned by-"+name+".txt"
    with open(b,"w+")as f:
          f.write("Library Management System \n")
          f.write("Returned By: "+ name+"\n")
           f.write("Date: " + DateAndTime.getDate()+"   Time:"+
           DateAndTime.getTime()+"\n\n")
           f.write("S.N.\t\tBookname\t      Authorname \t\tprice\n")
    total=0.0
    FOR i in range(3):
        IF List.bookName[i] in data:
            with open(b,"a") as f:
            f.write(str(i+1)+"\t\t"+List.bookName[i]+"\t\t "+List.authorName[i]+"\t\t
           "+"$"+List.price[i]+"\n")
                List.quantity[i]=int(List.quantity[i])+1
               total+=float(List.price[i])
          ENDIF
     ENDFOR
    OUTPUT ("\t\t\t\t\t\t\t\t\t"+"$"+str(total))
   OUTPUT ("Is the book return date has already been expired?")
    OUTPUT ("Press Y for Yes and N for Nope")
    stat=input()
    IF (stat.upper()=="Y"):
        OUTPUT ("By how many days was the book returned late?")
        day=int(INPUT())
```

```
        fine=2*day
        with open(b,"a")as f:
            f.write("\t\t\t\t\t\t\t\t\tFine: $"+ str(fine)+"\n")
        total=total+fine
            OUTPUT ("Final Total: "+ "$"+str(total))
    ENDIF
          with open(b,"a")as f:
                  f.write("\t\t\t\t\t\t\t\t\tTotal: $"+ str(total))
           with open("books.txt","w+") as f:
           FOR i in range(3):

          f.write(List.bookName[i]+","+List.authorName[i]+","+str(List.quantity[i])+","+"$"+Li
          st.price[i]+"\n")
          ENDFOR
END ALGORITHM
START
          CALL ALGORITHM Returnbooks
END
```

DATA STRUCTURE

Data Structures are the way of organizing and sorting data which can be worked efficiently. In other words it is the relationship between the data and the operation which needed to be performed. **(datacamp, n.d.)** In general there are two types of data structure and they are:

- Primitive data type
- Non Primitive data type

PRIMITIVE DATA TYPE:

It is defined by as the predefined types of data which contain pure and simples values of a data. **(datacamp, n.d.)** There are four types of primitive data types and they are as follows:

1. Integers:

   Integers are the whole number which is ranged from -2,147,483,647 to 2,147,483,647. It contains negative, positive and zero values. For example 5,-8, 0, 2, 6 etc.

   In my application the integer data type is used to store integer numbers. In the below figure I have store integer value assigning the variable name “a”.

```
def start():
    while(True):
        print("        Welcome to our library                ")
        print("----------------------------------------------")
        print("Enter 1. Display available books in library")
        print("Enter 2. Borrow a Book from our library")
        print("Enter 3. Return a Book to out library")
        print("Enter 4. Exit")
        try:
            a=int(input("Select a choice from 1-4: "))
            print()
```

Figure 8: Integers data type

2. Float:

   Floats are the irrational numbers such which ends with the decimal figures such as 2.33, 6.5 etc. It is also known as floating number.

   In my application the float data type is used to store float numbers. In the below figure I have store float value in the variable name “total”. The fee can be in decimal therefore I have store the total fee in decimal numbers.

```
total=0.0
for i in range(3):
    if List.bookName[i] in data:
        with open(b,"a") as f:
            f.write(str(i+1)+"\t\t"+List.bookName[i]+"\t\t "+List.authorName[i]+"\t\t  "+"$"+List.price[i]+"\n")
            List.quantity[i]=int(List.quantity[i])+1
        total+=float(List.price[i])
```

Figure 9: Float data types

3. Strings:

   Strings are the collection of alphabets and words of characters. It is the non-numeric values. For example: “assignment”, “Richard kaka” etc. In my application I have used String data types to display the message in my program.

```
def start():
    while(True):
        print("        Welcome to our library                ")
        print("----------------------------------------------")
        print("Enter 1. Display available books in library")
        print("Enter 2. Borrow a Book from our library")
        print("Enter 3. Return a Book to out library")
        print("Enter 4. Exit")
        try:
            a=int(input("Select a choice from 1-4: "))
            print()
```

Figure 10: String data types

4. Boolean:

   Boolean are the built-in data type which can only take the values True and False. In my application I have used Boolean data type to continue the loop. Furthermore, I have also used in try except assigning the different values.

```
def borrowBooks():
    borrowCode=False
    while(True):
        firstName=input("Enter the First Name: ")
        if firstName.isalpha():
            break
        print("please enter a valid first name")
    while(True):
        lastName=input("Enter the Last Name: ")
        if lastName.isalpha():
            break
        print("please enter a valid last name")
```

Figure 11: Boolean data type

NON PRIMITIVE DATA TYPES:

Non Primitive data types are the non-predefined programming language. There are four types of non-primitive types and they are as follows:

1. List:

   Lists are the mutable items which can be changed. This type of data is used to store the data. It is represented by the square brackets [ ]. In this application I have used list in the bookName, authorName, quantity and price which stores the list values. Here, I have initialized as the empty list which can be used to store the details of the books later on

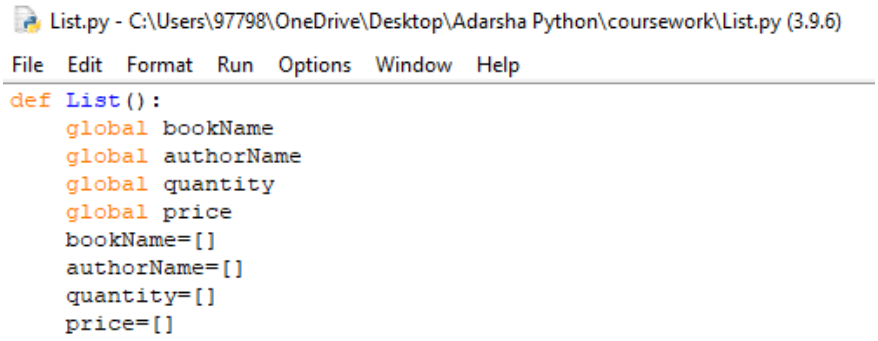

Figure 12: List data type

2. Tuples:

   Tuples are the immutable items which can be not changed. Since, it cannot be modified they are faster than list.

3. Dictionary:

   Dictionaries consist of key value pairs. This data type stores huge amount of data. It is also changeable and does not allow duplicate data.

   For Example: {

   “brand”: “Yamaha”

   “name”: “MT15”

}

4. Sets:

   Sets are the unique values. It is unordered, unchangeable, and do not allow any duplicate data.

   For Example: Set1 = {“Yamaha”, “Honda”, “BMW”}

PROGRAM

When the program is opened, the user will get four options to choose. The user can enter 1 to display all the books from the .txt file, 2 to borrow books from the library, 3 to return books to the library and 4 to terminate the whole program.

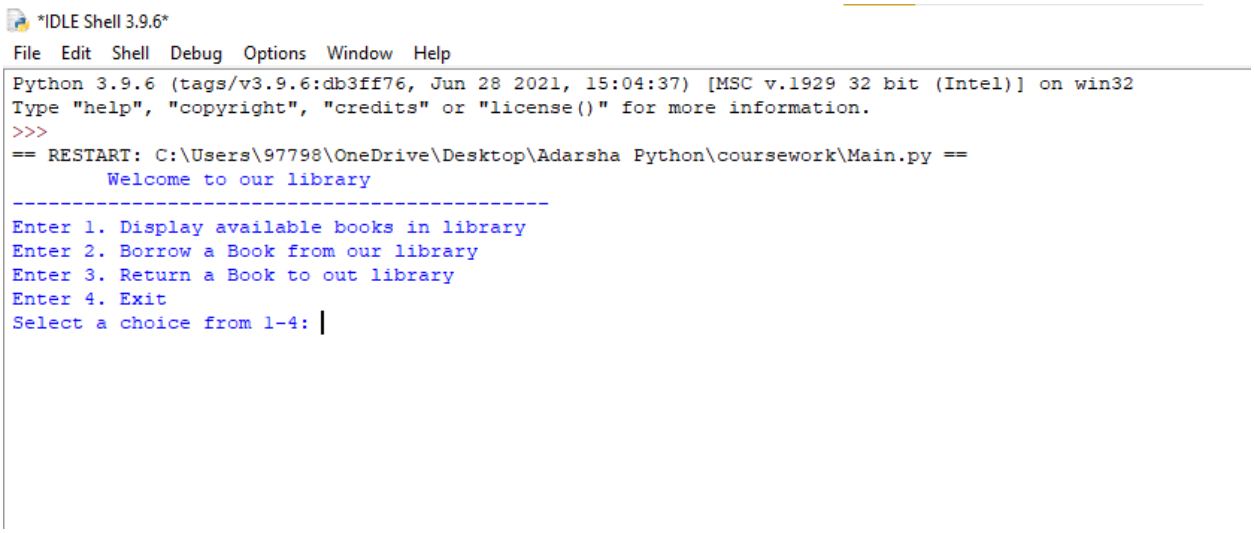

Figure 13: Choosing Option from 1- 4

When the user will enter number 1 it will display all the books from the store which user can borrow.

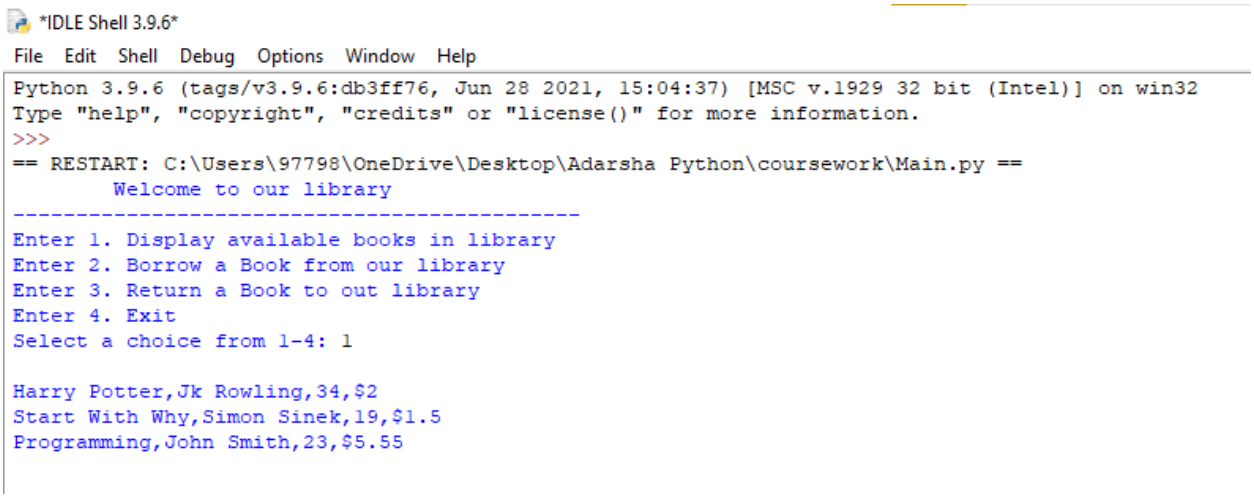

Figure 14: Displaying the books

Borrow Process

Figure 15: Full Borrow Process

When the user enters no 2 then borrow process will initiate. In the beginning the program will ask the user to enter first and last name. If it is not valid then again same process will repeated. Then the available books in the library will be displayed and user has to select the books according to the given number. In case, if the user wants to borrow more books then user has to follow the given instruction. When the user has completed his/her purchase of books a .txt file will be generated where all the information of the borrower will be kept as shown below.

Figure 16: txt file of borrowing

Return Process.

When the user will enter number three then return program will initiate. In the beginning the program will ask the user to enter first and last name. If it is not valid then again same program will run. After that, the books which the borrower has borrowed will be displayed. Then, the program will ask if the books were delayed or not. If it was delayed then it will ask how many days the books were delayed to return. Finally, the program will calculate the total fees of the user and the return process will terminate.

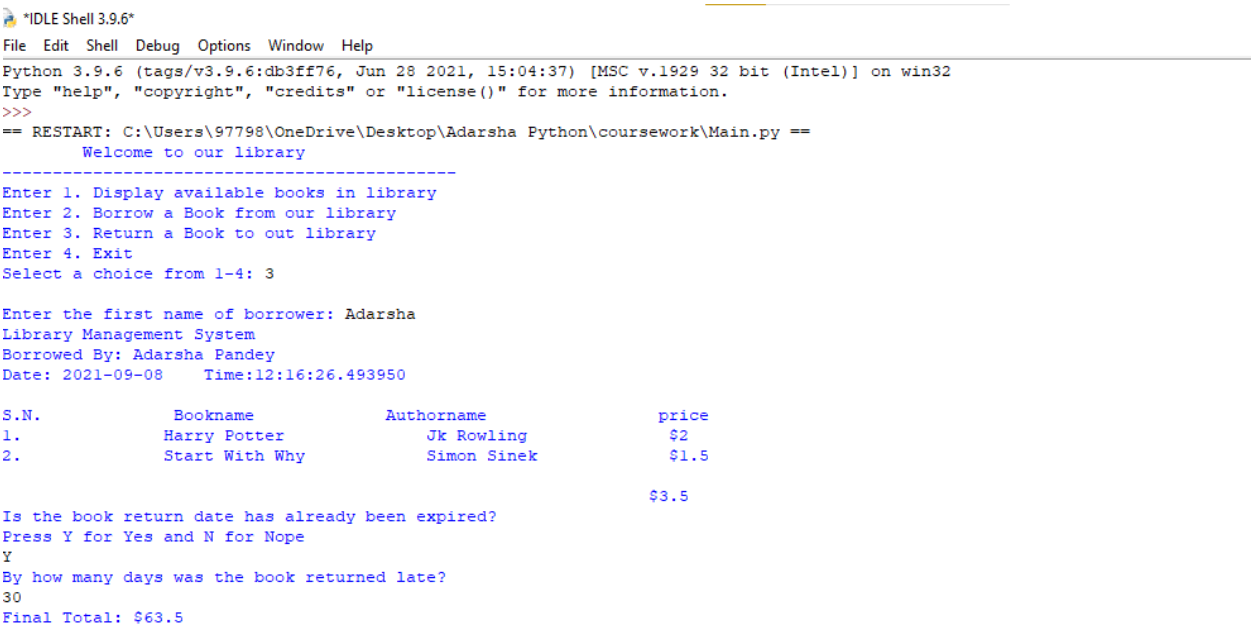

Figure 17: Return Process

Similarly, when the user has completed his/her return of books a .txt file will be generated where all the information of the person who has returned will be kept as shown below

.

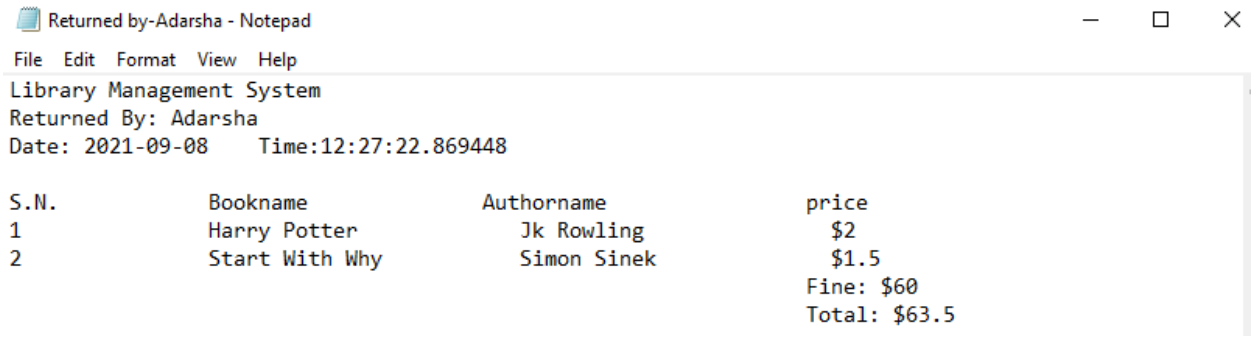

Figure 18: txt file of Return Process

Finally when the user will enter number 4 then whole program will be terminated with a message as shown below:

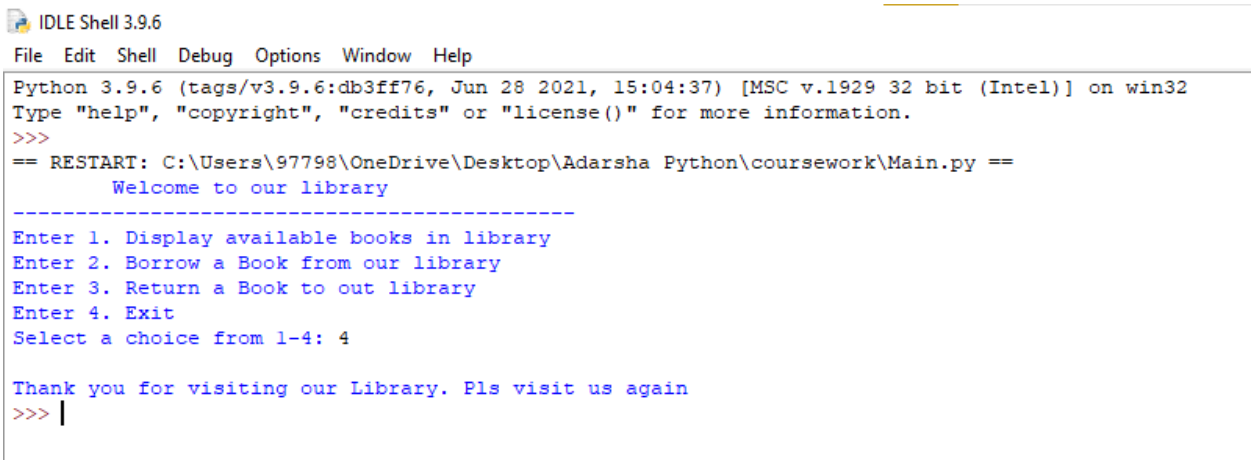

Figure 19: Terminating the program

TESTING:

TEST 1: TO SHOW IMPLEMENTATION OF TRY, EXCEPT

Objective:	To show the implementation of try, except
Action:	➢ Open the IDLE Shell ➢ Select an option except from 1- 4 ➢ Click Enter button
Expected Result:	Please enter a valid option from 1- 4 will be shown
Actual Result:	Please enter a valid option from 1- 4 is showed
Conclusion:	The test was successful.

Table 1: To test the implementation of try, except

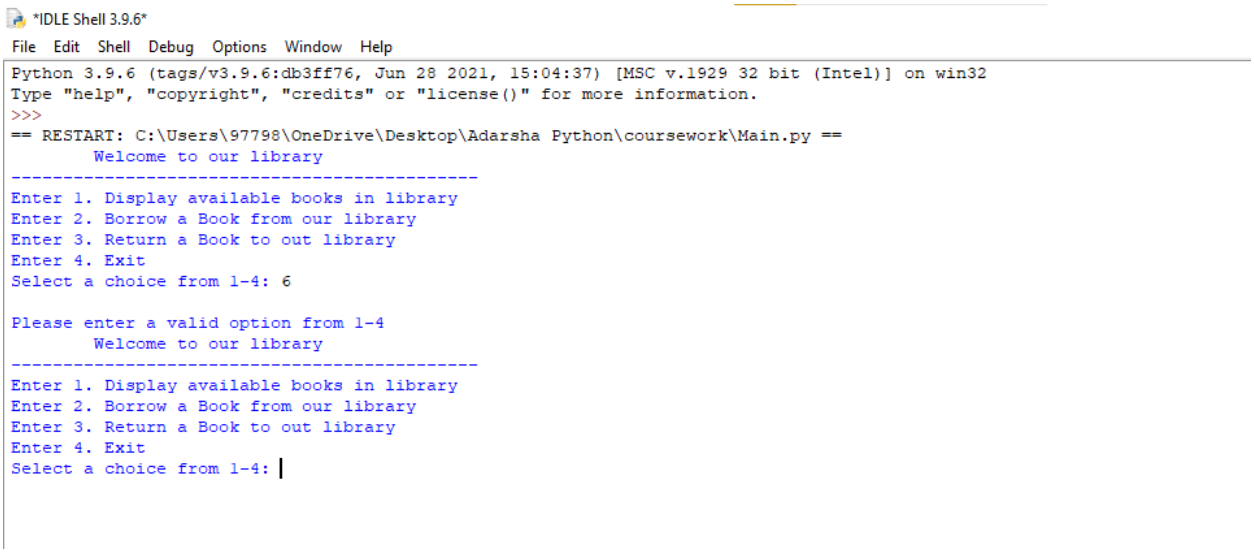

Figure 20: To show the implementation of try, except

TEST 2: TO SHOW THE SELECTION OF BORROW AND RETURN OPTION

TEST2.1: TO SHOW THE NEGATIVE AND NON EXISTED VAULE AS INPUT

Objective:	To show the negative and non-existed value as input in borrow
Action:	➢ Open the IDLE Shell ➢ Enter -2 and 6 number ➢ Click Enter button
Expected Result:	Please enter a valid option from 1- 4 will be shown
Actual Result:	Please enter a valid option from 1- 4 is showed
Conclusion:	The test was successful.

Table 2: To test the negative and non-existed value as input in borrow

Figure 21: To show the negative and non-existed value as input in borrow

TEST2.2: TO SHOW THE NEGATIVE AND NON EXISTED VAULE AS INPUT

Objective:	To show the negative and non-existed value as input in return
Action:	➢ Open the IDLE Shell ➢ Enter -3 and 8 number ➢ Click Enter button
Expected Result:	Please enter a valid option from 1- 4 will be shown
Actual Result:	Please enter a valid option from 1- 4 is showed
Conclusion:	The test was successful.

Table 3: To test the negative and non-existed value as input in return

```
        Welcome to our library
-------------------------------------------
Enter 1. Display available books in library
Enter 2. Borrow a Book from our library
Enter 3. Return a Book to out library
Enter 4. Exit
Select a choice from 1-4: -3

Please enter a valid option from 1-4
        Welcome to our library
-------------------------------------------
Enter 1. Display available books in library
Enter 2. Borrow a Book from our library
Enter 3. Return a Book to out library
Enter 4. Exit
Select a choice from 1-4: 8

Please enter a valid option from 1-4
        Welcome to our library
-------------------------------------------
Enter 1. Display available books in library
Enter 2. Borrow a Book from our library
Enter 3. Return a Book to out library
Enter 4. Exit
Select a choice from 1-4:
```

Figure 22: To show the negative and non-existed value as input in return

TEST 3: TO SHOW THE GENERATION OF BORROW

Objective:	To show the generation of borrow file
Action:	➢ Run the main file ➢ Enter No.2 to borrow the books ➢ Enter all the personal details and books which you want to borrow
Expected Result:	The books will be borrowed.
Actual Result:	The book is borrowed.
Conclusion:	The test was successful.

Table 4: To test the generation of borrow file

Figure 23: To show the generation of borrow file

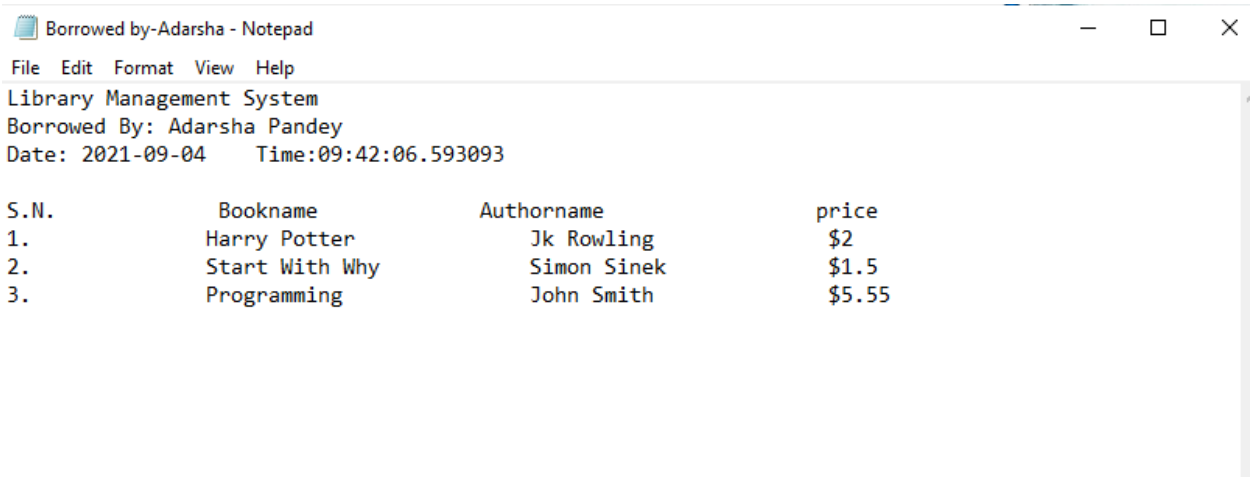

Figure 24: txt file of borrowing

TEST4: TO SHOW THE GENERATION OF RETURN FILE

Objective:	To show the generation of Return file
Action:	➢ Run the main file ➢ Enter No.3 to Return the books ➢ Enter all the personal details and books which you want to return
Expected Result:	The books will be returned
Actual Result:	The book is returned.
Conclusion:	The test was successful.

Table 5: To test the generation of Return file

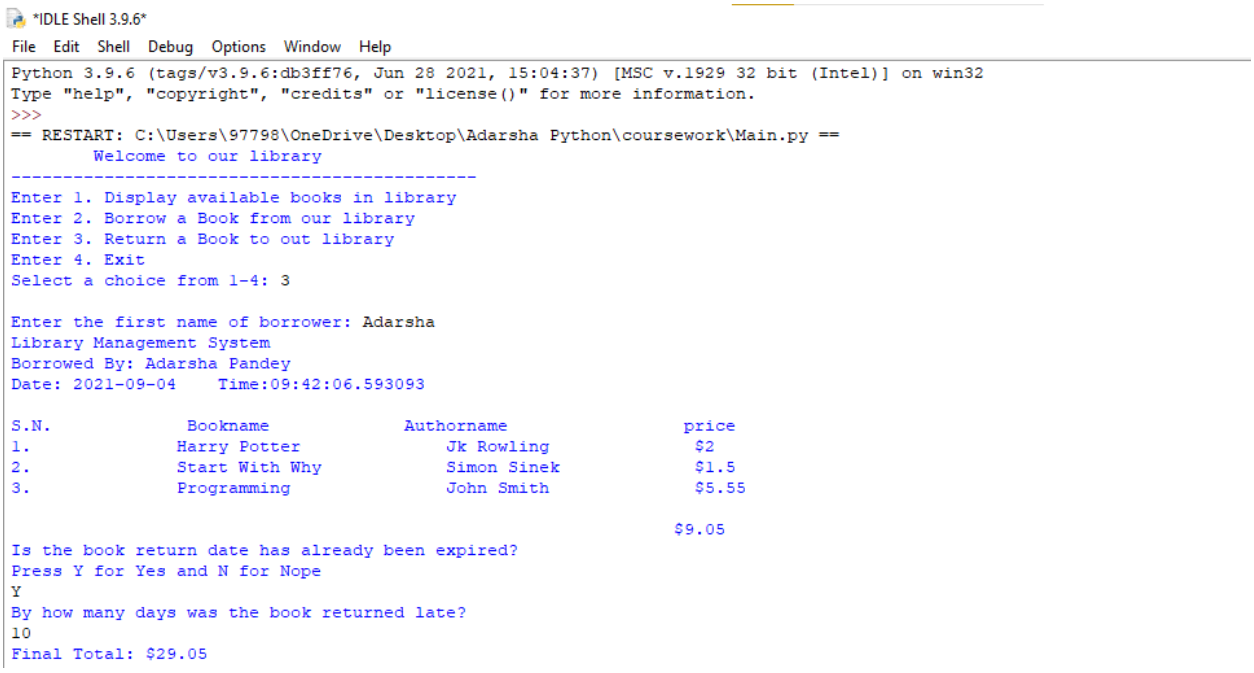

Figure 25: To show the generation of Return file

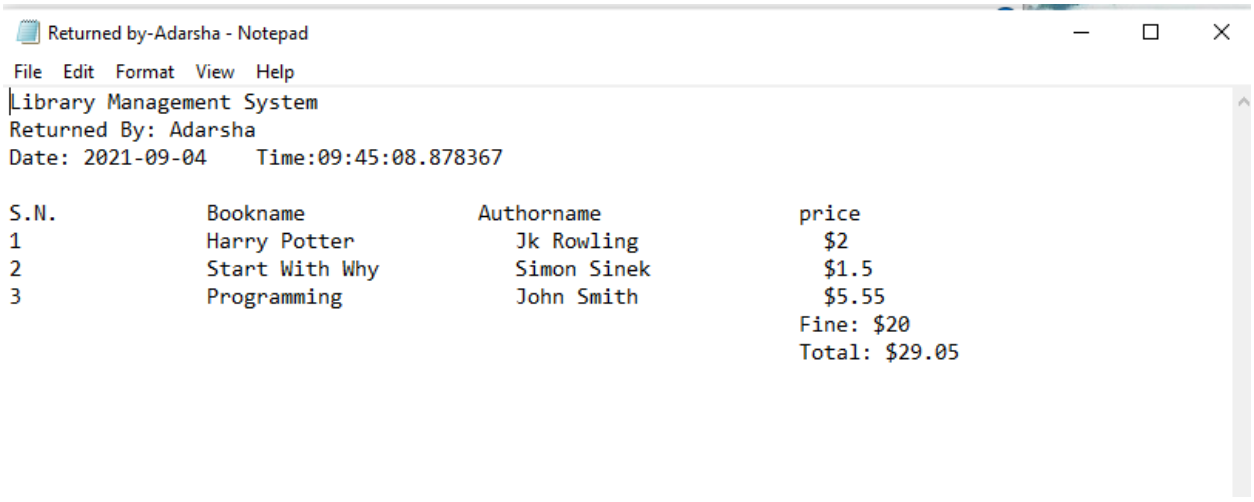

Figure 26: txt file of Return Process

TEST 5: TO SHOW THE UPDATE IN STOCK

TEST 5.1: TO SHOW THE QUANTITY BEING DEDUCTED WHILE BORROWING THE BOOK

Objective:	To show the quantity being deducted while borrowing the books.
Action:	➢ Run the main file ➢ Enter No.2 to borrow the books ➢ Enter all the personal details and books which you want to borrow ➢ Finally, see the number of books where it has been stored.
Expected Result:	The quantity of the books will be reduced
Actual Result:	The quantity of the books is decrease.
Conclusion:	The test was successful.

Table 6: To test the quantity being deducted while borrowing the books.

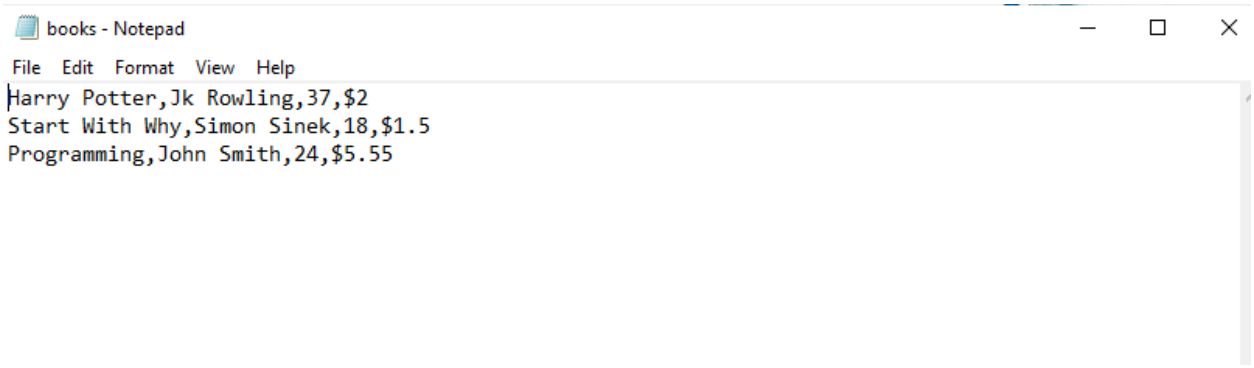

Figure 27:books.txt file before borrowing

Figure 28: Borrowing Process

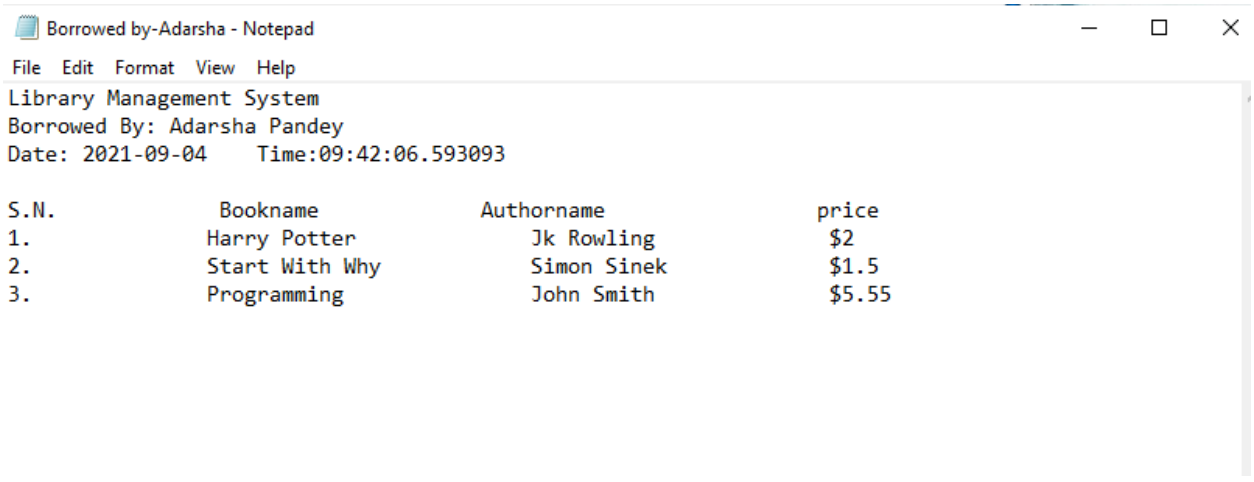

Figure 29: txt file of borrowed

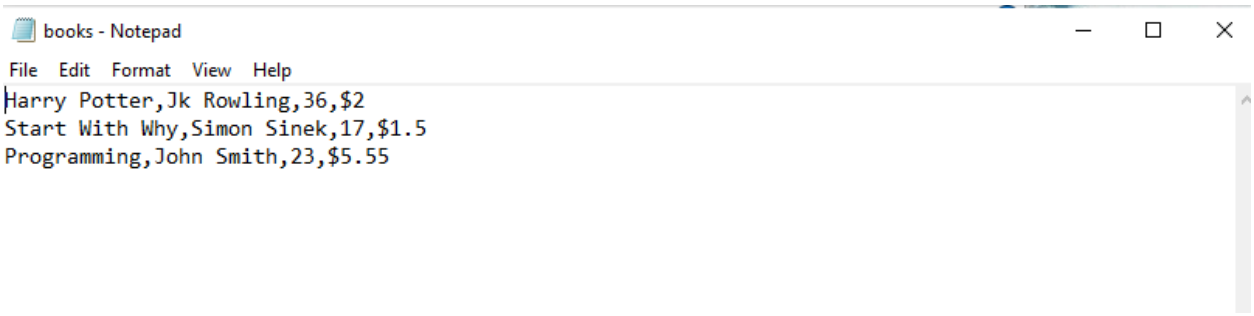

Figure 30: books.txt after borrowing

TEST 5.2: TO SHOW THE QUANTITY BEING ADDED WHILE RETURNING THE BOOK

Objective:	To show the quantity being added while returning the books.
Action:	➢ Run the main file ➢ Enter No.3 to return the books ➢ Enter all the personal details and books which you want to return ➢ Finally, see the number of books where it has been stored.
Expected Result:	The quantity of the books will be added after returning the books.
Actual Result:	The quantity of the books is increase after returning the books.
Conclusion:	The test was successful.

Table 7: To test the quantity being added while returning the books.

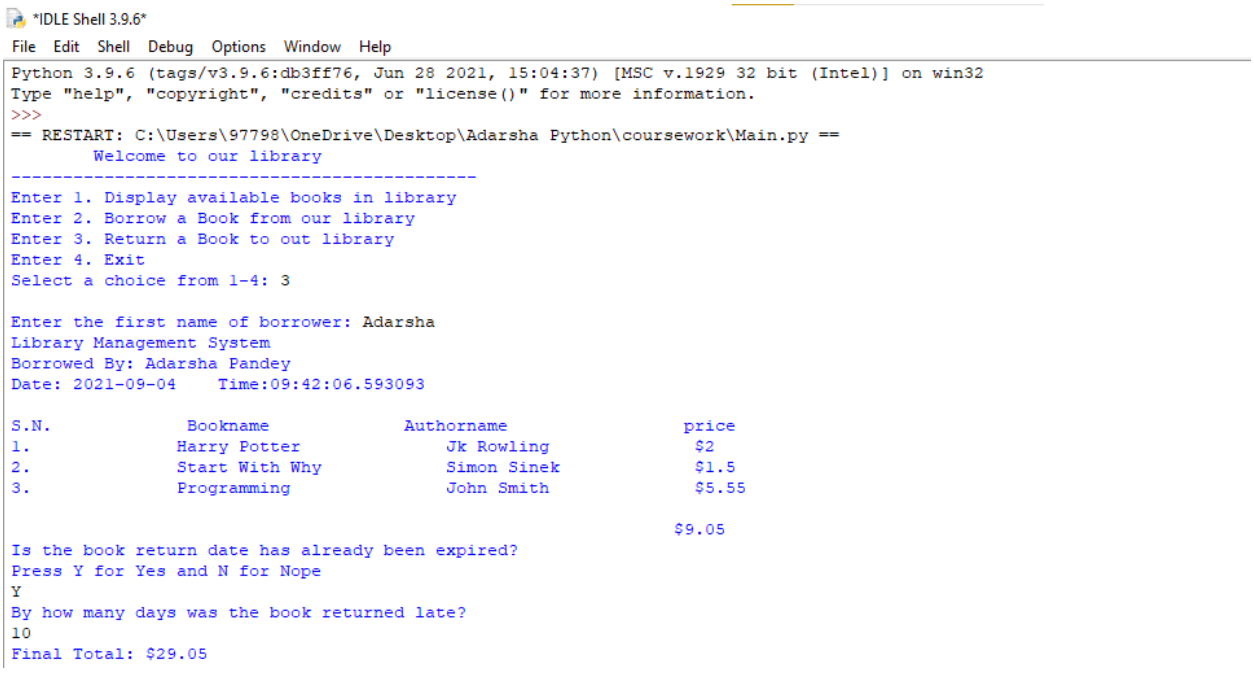

Figure 31: Return Process

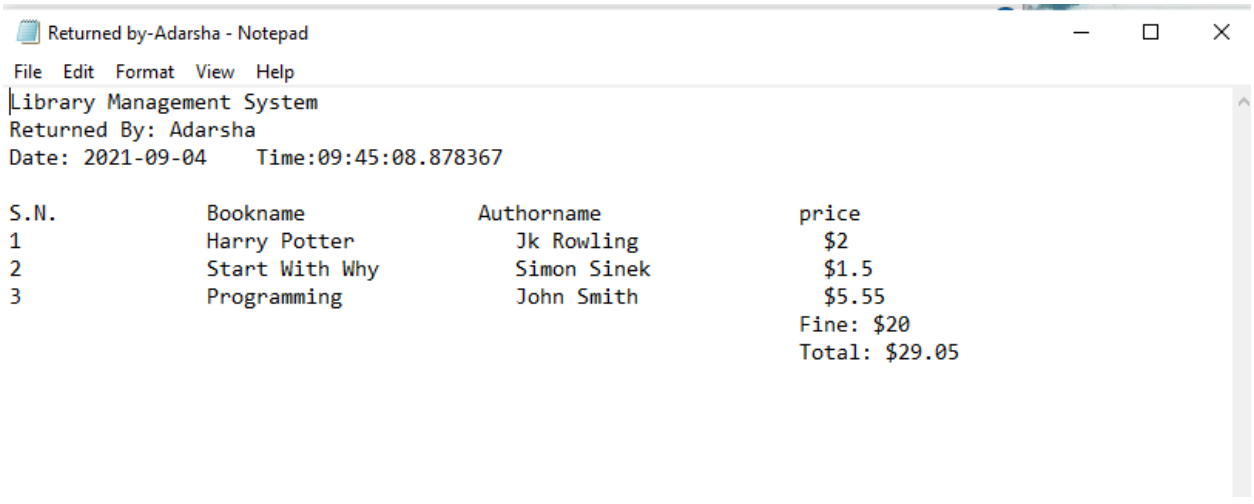

Figure 32: txt file of Return

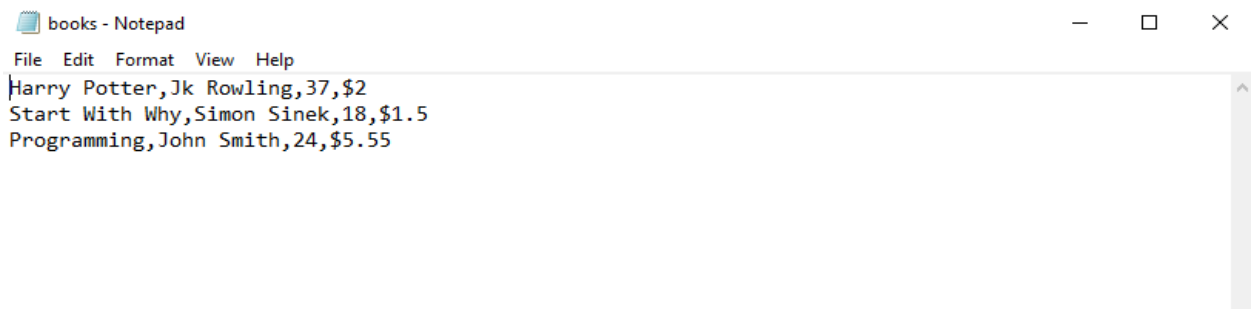

Figure 33: books.txt file after returning

CONCLUSION:

In Conclusion, there is no doubt that this coursework was completed with a lot of hardwork and dedication. After my completion of +2 examinations I was keen to learn to code. Therefore I took the class of CS50 which was organized by Harvard University. There I got the chance the learn some basic concept of python. So, I was quite familiar with the syntax of python. This project totally reflects of every week from 2nd Semester which was taught by our beloved lecturer and tutorial teachers. In this coursework we have made a library where we can store the books. Furthermore, if someone wants to borrow the books there information will be stored and so on. We also learned to make algorithms, flowchart and pseudocode. There were many difficulties while doing this course work but I encountered every problem which was ahead of me.

Research and findings are also the key part to accomplish every coursework. By doing research we can broaden our knowledge. I also did a lot of research due to which I was able to finish my coursework. It helped me lot to building my knowledge to do the coursework which were very limited in lecture and tutorial sessions. I also got chance to explore many different new things. Some of the websites which I used to learn are as follows:

- https://www.freecodecamp.org/
- https://www.w3schools.com/
- https://www.datacamp.com/

These websites really helped me in a great way to explore new things in python. At last but not the least I want to thank our teachers who helped us in every problem which was ahead of us and we successfully completed this project.

.

BIBILOGRAPHY

- datacamp, n.d. *www.datacamp.com.* [Online]
  Available at: https://www.datacamp.com/community/tutorials/data-structures-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&ut

  [Accessed 8 September 2021].
- datacamp, n.d. *www.datacamp.com.* [Online]
  Available at: https://www.datacamp.com/community/tutorials/data-structures-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&ut#primitive

  [Accessed 8 September 2021].
- medium.com, n.d. *https://medium.com/.* [Online]
  Available at: https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b
  [Accessed 8 September 2021].

- tutorialsteacher, n.d. *www.tutorialsteacher.com.* [Online]
  Available at: https://www.tutorialsteacher.com/python/python-idle
  [Accessed 8 September 2021].

APPENDIX

APPENDIX OF MAIN

```
import Return #importing Return file

import List #importing List file

import DateAndTime #importing Date and time

import Borrow #importing Borrow file


def start():

    while(True):

        print("        Welcome to our library                ")

        print("-----------------------------------------------")

        print("Enter 1. Display available books in library")

        print("Enter 2. Borrow a Book from our library")

        print("Enter 3. Return a Book to out library")

        print("Enter 4. Exit")

        try:

            a=int(input("Select a choice from 1-4: "))

            print()

            if(a==1):

                with open("books.txt","r") as f:

                    lines=f.read()

                    print(lines)

                    print ()


            elif(a==2):
```

```
            List.List()
            Borrow.borrowBooks()
        elif(a==3):
            List.List()
            Return.returnBooks()
        elif(a==4):
            print("Thank you for visiting our Library. Pls visit us again")
            break
        else:
            print("Please enter a valid option from 1-4")
    except ValueError:
        print("Please input as suggested.")
start()
```

APPENDIX OF DATEANDTIME

```
def getDate():
    import datetime #importing date and time
    now=datetime.datetime.now
    #print("Date: ",now().date())
    return str(now().date())

def getTime():
    import datetime
    now=datetime.datetime.now
```

```
    #print("Time: ",now().time())
    return str(now().time())
```

APPENDIX OF LIST

```
def List():
    global bookName
    global authorName
    global quantity
    global price
    bookName=[]
    authorName=[]
    quantity=[]
    price=[]

    with open("books.txt","r") as f:

        lines=f.readlines()
        lines=[x.strip('\n') for x in lines]
        for i in range(len(lines)):
            ind=0
            for a in lines[i].split(','):
                if(ind==0):
                    bookName.append(a)
                elif(ind==1):
                    authorName.append(a)
```

```
            elif(ind==2):
                quantity.append(a)
            elif(ind==3):
                price.append(a.strip("$"))
            ind+=1
```

APPENDIX OF BORROW

```
import DateAndTime
import List


def borrowBooks():
    borrowCode=False
    while(True):
        firstName=input("Enter the First Name: ")
        if firstName.isalpha():
            break
        print("please enter a valid first name") #if the first name is not valid it will print out
    while(True):
        lastName=input("Enter the Last Name: ")
        if lastName.isalpha():
            break
        print("please enter a valid last name") #if the last name is not valid it will print out.


    t="Borrowed by-"+firstName+".txt" #Generating a borrowd file
    with open(t,"w+") as f:
```

```
        f.write("Library Management System  \n")
        f.write("Borrowed By: "+ firstName+" "+lastName+"\n")
        f.write("Date: " + DateAndTime.getDate()+"    Time:"+
DateAndTime.getTime()+"\n\n")
        f.write("S.N. \t\t Bookname \t      Authorname \t\t price \n" )

    while borrowCode==False:
        print("Please select a option below:")
        for i in range(len(List.bookName)):
            print("Enter", i, "to borrow book", List.bookName[i]) #Displaying books to
boorrow

        try:
            a=int(input())
            try:
                if(int(List.quantity[a])>0): #if the qunatity is greater than 0 then message will
print out
                    print("Book is available in our library")
                    with open(t,"a") as f:
                        f.write("1. \t\t"+ List.bookName[a]+"\t\t  "+List.authorName[a]+"\t\t
"+"$"+List.price[a]+"\n") #Storing the data in a txt file

                    List.quantity[a]=int(List.quantity[a])-1 #reducing the qunatity
                    with open("books.txt","w+") as f:
                        for i in range(3):
```

```
f.write(List.bookName[i]+","+List.authorName[i]+","+str(List.quantity[i])+","+"$"+List.price
[i]+"\n")



            loop=True
            count=1
            while loop==True:
                option=str(input("Do you want to borrow more books?Press y for yes and
n for nope."))
                if(option.upper()=="Y"): #if user enters Y books will be display
                    count=count+1
                    print("Please select an option below:")
                    for i in range(len(List.bookName)):
                        print("Enter", i, "to borrow book", List.bookName[i]) #Displaying the
books
                    a=int(input())
                    if(int(List.quantity[a])>0): #if the qunatity is greater than 0 then
message will print out
                        print("Book is available in our library")
                        with open(t,"a") as f:
                            f.write(str(count) +". \t\t"+ List.bookName[a]+"\t\t
"+List.authorName[a]+"\t\t  "+"$"+List.price[a]+"\n")  #Storing the data in a txt file


                        List.quantity[a]=int(List.quantity[a])-1 #reducing the qunatity
                        with open("books.txt","w+") as f:
```

```
                    for i in range(3):

f.write(List.bookName[i]+","+List.authorName[i]+","+str(List.quantity[i])+","+"$"+List.price
[i]+"\n")
                        borrowCode=False
                else:
                    loop=False
                    break
            elif (option.upper()=="N"): #if the user enters N then it will terminate
                print ("Thank you for borrowing books from our library ")
                print("")
                loop=False
                borrowCode=True
            else:
                print("Please choose as instructed")


        else:
            print("Book is not available in our library. Pls check other books")
            borrowBooks()
            borrowCode=False
    except IndexError:
        print("")
        print("Please choose book according to the given instruction. Thank You")
except ValueError:
    print("")
    print("Please choose as given suggested.")
```

APPENDIX OF RETURN

```
import List

import DateAndTime

def returnBooks():

    name=input("Enter the first name of borrower: ")

    a="Borrowed by-"+name+".txt"

    try:

        with open(a,"r") as f:

            lines=f.readlines()

            lines=[a.strip("$") for a in lines]


        with open(a,"r") as f:

            data=f.read()

            print(data)

    except:

        print("The borrower name is incorrect. Pls check once")

        returnBooks()


    b="Returned by-"+name+".txt" #generating the Return file

    with open(b,"w+")as f:

        f.write("Library Management System \n")

        f.write("Returned By: "+ name+"\n")

        f.write("Date: " + DateAndTime.getDate()+"    Time:"+
DateAndTime.getTime()+"\n\n")

        f.write("S.N.\t\tBookname\t      Authorname \t\tprice\n")
```

```
    total=0.0
    for i in range(3):
        if List.bookName[i] in data:
            with open(b,"a") as f:
                f.write(str(i+1)+"\t\t"+List.bookName[i]+"\t\t "+List.authorName[i]+"\t\t
"+"$"+List.price[i]+"\n")
                List.quantity[i]=int(List.quantity[i])+1 #adding the stock
            total+=float(List.price[i])


    print("\t\t\t\t\t\t\t\t\t"+"$"+str(total)) #total
    print("Is the book return date has already been expired?")
    print("Press Y for Yes and N for Nope") #if yes then it will charge some money else it
will terminate
    stat=input()
    if(stat.upper()=="Y"):
        print("By how many days was the book returned late?")
        day=int(input())
        fine=2*day #calucalating the total charge
        with open(b,"a")as f:
            f.write("\t\t\t\t\t\t\t\tFine: $"+ str(fine)+"\n") #fine
        total=total+fine



    print("Final Total: "+ "$"+str(total))
```

```
    with open(b,"a")as f:
        f.write("\t\t\t\t\t\t\t\tTotal: $"+ str(total))


    with open("books.txt","w+") as f:
            for i in range(3):

f.write(List.bookName[i]+","+List.authorName[i]+","+str(List.quantity[i])+","+"$"+List.price
[i]+"\n")
```