Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Pratik Shrestha

Group: C3

London Met ID: 20048957

College ID: NP01CP4S210063

Assignment Due Date: 10th September 2021

Assignment Submission Date: 10th September 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.

Table of Contents

Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

This is the final report for the coursework that was designed to create a library management system or application. The software was created with the help of precise algorithms and pseudocodes, as well as flowcharts.

As the tasks assigned were not so easy. It requires a lot of strength to express our skill to understand and respond to questions, as well as to illustrate them in an informative method as required. As a result, understanding the difficulty in performing assignments within the given time and to ensure that they are helpful to the targeted users, all efforts are put out until the final report outlook is created.

This coursework, which is a library management system, is like an inventory system in which it helps users in borrowing and returning books from the library. The library is a place where a user can borrow and borrow books as he or she desires. The library enables clients to borrow any book from its library. You may ask why we need to create this system when we don't need it. Because today's libraries are ready to manage effectively with their old techniques, such as keeping a record of user borrowed and returned items on paper rather than in software. Even without the help of today's modern software, libraries can perform and are not facing any difficulty. Yes, that is correct, but we must always keep an eye on the future. Because in today's world, advanced technologies can be found in every sector.

With the technological innovation and platforms, old methods and techniques should tend to change with the passage of time. This system should be built since our libraries are rapidly becoming much more modern, prompting the usage of recently developed software. This developed software will support libraries in keeping appropriate track of books borrowed and returned by customers, as well as assuring that no books are lost. This technique is user-friendly and time-saving, and it may help to minimize unnecessary data duplication while also making it understandable.

The main feature of this project is that it will teach the student how to utilize the IDLE (Integrated Development and Learning Environment) in Python to its full ability. Even individuals who understand little about programming will be able to use this software to its maximum potential and without causing any trouble. This software allows programmers to test with multiple functions and statements while also helps in the depth study of the programming language.

CHAPTER 2: DISCUSSION AND ANALYSIS:

A lot of research were done in order to tackle the problem in the discussion portion of this assignment, but there was also a brief description of python in different websites and books. The books and websites that I used for researching are listed in the Bibliography section with full descriptions. I could complete this course work after a lot of hard effort and constant discussion with the module leaders and colleagues. There are a few essential components without which this work would've been impossible to finish.

Figure 1: MS Word 2016

This coursework was created with the use of several tools included in the researching section. Microsoft Word 2016 was used to create the documentation portion of this coursework, that was then converted to PDF format.

Figure 2: IDLE PYTHON

Similarly, IDLE (Integrated Development and Learning Environment) of Python 3.7.2 was used to develop the developmental portion of the coursework. In the python shell window, all the programs were written and executed.

Furthermore, a free online diagram software called draw.io was used to create the flowcharts, which allows us to create different shapes and organized flowcharts with the required diagram. Additionally, using a web browser, research was done on a variety of coursework-related subjects.

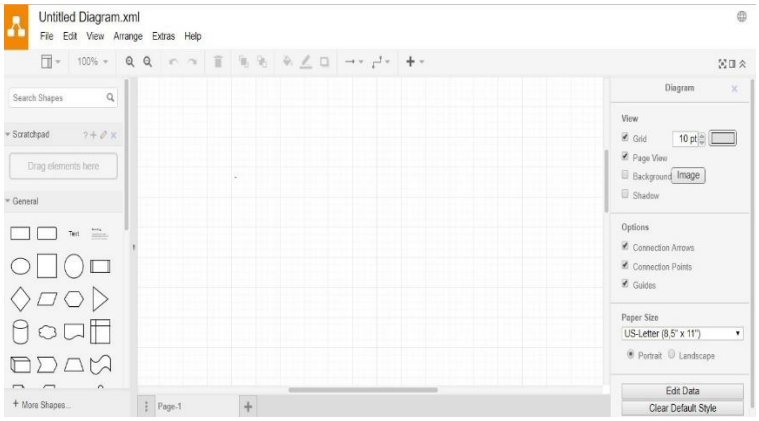


Figure 3: Draw.IO(flowchart)

A flowchart illustrates the distinct stages of a process in a logical sequence. It's extremely beneficial for collecting and interpreting business processes, IT systems, and computer algorithms. Draw.io is a free online drawing program. It makes it simple to create attractive flowcharts and process diagrams. Draw.io also simply connects with other suites, such as Google Drive, One Drive, and so on.

CHAPTER 3: DATA STRUCTURE:

Organizing, managing, and storing data is important as it enables easier access and efficient modifications. Data Structures allows you to organize your data in such a way that enables you to store collections of data, relate them and perform operations on them accordingly. (edureka, 2021)Data structure is highly essential in many algorithms since it helps the programmer to manage data effectively.

This project provides massive use of the collection data type list to handle multiple input/output and data storage functions in Python. Python also supports a variety of data types, including integers, strings, Booleans, floats, and so on. Some data types and data structures were used to store, modify, and execute various operations on the data when creating the software.

The below are the data types and structures utilized in the program:

- Integer
- Float
- String
- Boolean
- List

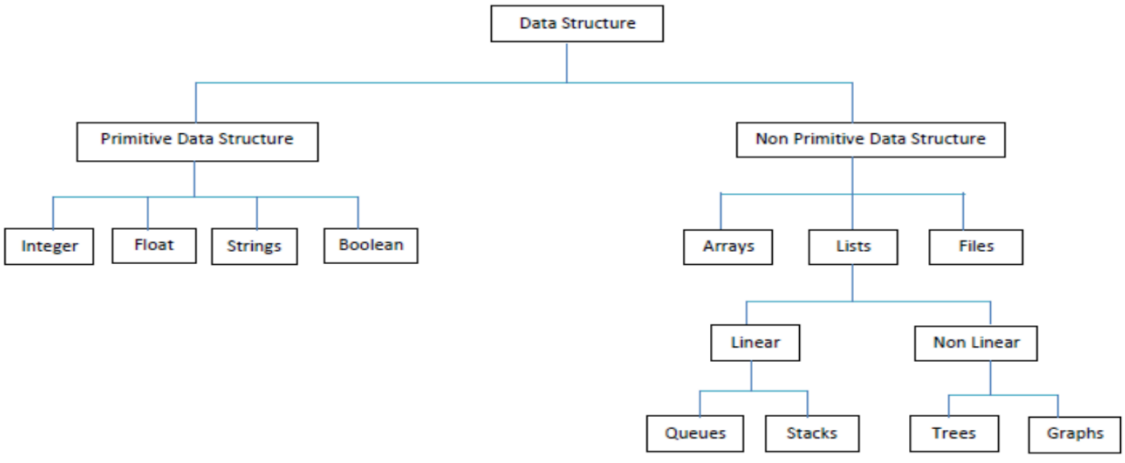


Figure 4: Data Structure in Python

- Integer

The integer data type was used in the software to store all of the numerical values input by the users. For example, It is used to check out how many number of a book a user needs to borrow

- Float

Similarly, the application uses the float data type to store decimal values. The variable total is used to keep a record of the total cost of borrowed books.

- String

String data types, on the other hand, are used to store text information that consists of one or more characters. Several operations, such as choice and fine amount, are done in string. In the different techniques of our.py file, string data types are used to store the user's decision.

- Boolean

Similarly, Boolean is used to store a single character (either true or false). It is used in the program to test conditions and is saved in variables such as loop and complete.

- List

In addition, a list is used in our.py file to keep track of information such as the name of the book, the author's name, the quantity, and the cost. The values on the list can be added as needed using the methods append() , which have been successfully included in the various modules of our.py folder.

- Tuples

A tuple is a collection of ordered and immutable objects. Tuples, like lists, are sequences. Tuples and lists differ within this tuples cannot be modified, but lists can, and tuples use parentheses whereas lists use square brackets.

- Set

A Set is an unordered collection data type that is iterable, mutable and has no duplicate elements. The set class in Python represents the mathematical concept of a set. The primary advantage of using a set over a list is that it provides a highly efficient method for determining if a given element is in the set. This is based on a hash table, which is a type of data structure. We can't use indexes to access items in sets since they're not ordered like lists.

- Dictionary

A colon (:) separate each key from its value, commas separate the elements, and the whole thing is wrapped in curly braces{}. With just two curly braces, an empty dictionary with no items is represented as follows:

Values may not be unique within a dictionary, but keys are. A dictionary's values can be any data type, but the keys must be immutable data types such characters, integers, or tuples.

CHAPTER 4: ALGORITHM

An algorithm is a sequence of procedures for completing the task or solving a problem. A recipe, which consists of specific instructions for preparing a dish or meal, is a common example of an algorithm. Algorithms are used by every digital device to carry out its own tasks. In this project I used algorithm for a library management system. The algorithm of library management system is divided into two parts:

- Borrow book
- Return book

i. Algorithm for borrowing book is given below:

Step 1: Start

Step 2: Input the full name of the borrower.

Step 3: Is the name valid?

If yes break

Then go to step 4

Else print “"Please input alphabet from A-Z" and go to step 2.

Step 4: Create borrow by (full name).txt file and store the borrower details

Step 5: if borrow is false print(“Please select the book given below,”)

Step 6: show the detail information of available books and ask user to borrow books according to its number.

Step 7: enter the book number

Step 8: if (selection of book < 0)

Then, print(“Please input positive number.”)

Else go to step 9

Step 9: if (selection of book > 0)

Then, print(“The book is available”)

Step 10: Then, append further details in the borrow.txt file

Step 11: Open stock file, Update the stock file from which the book is borrowed.

Step 12: create a while loop and assign it true to run and infinity loop

Step 13: Ask user to borrow the next book?

Step 14: if yes then increase the count and go to step 5

Else, print(“Thank you for borrowing books from us. Have a good day.”)

Step 8: End

ii. Algorithm for returning book:

Step 1: Start

Step 2: Input the name of borrower.

Step 3: Open try and create a borrowed by (full name).txt file

And store the detail of borrower.

Step 4: open borrow.txt file and read it.

Step 5: create a text name and add borrow.txt file and strip “$” from the cost

Step 6: Again open borrow.txt file and read it.

Step 7: if try catch exception then print(“The borrower name is not available, Please input another name.”) and go to step 2

Step 8: Create a borrow variable and store returned by (fullname).txt file in it

Step 9: Assign total amount as 0

Step 10: Open borrow and append the value in it

Step 11: Update the quantity of book after it is returned and increase total amount from detail

Step 12: Print total amount

Step 13: Ask if book expire date is expire or not.

Step 14: if yes,

Then, fine amount = 2*day

Step 15: Add total fine amount and cost of the book.

Step 16: Output generated text file with the total amount.

Step 17: End

CHAPTER 5: FLOWCHART

A flowchart is a graphic representation of a method, system, or computer algorithm. They're widely used in a variety of areas to document, study, plan, develop, and describe often complex processes in clear, simple diagrams. Flowcharts, also known as flow charts, are diagrams that use rectangles, ovals, diamonds, and possibly other shapes to show the kind of step, as well as connecting arrows to show flow and sequence. They can be as simple as hand-drawn diagrams or as complex as computer-drawn diagrams displaying multiple steps and pathways.

Flowchart of borrow:

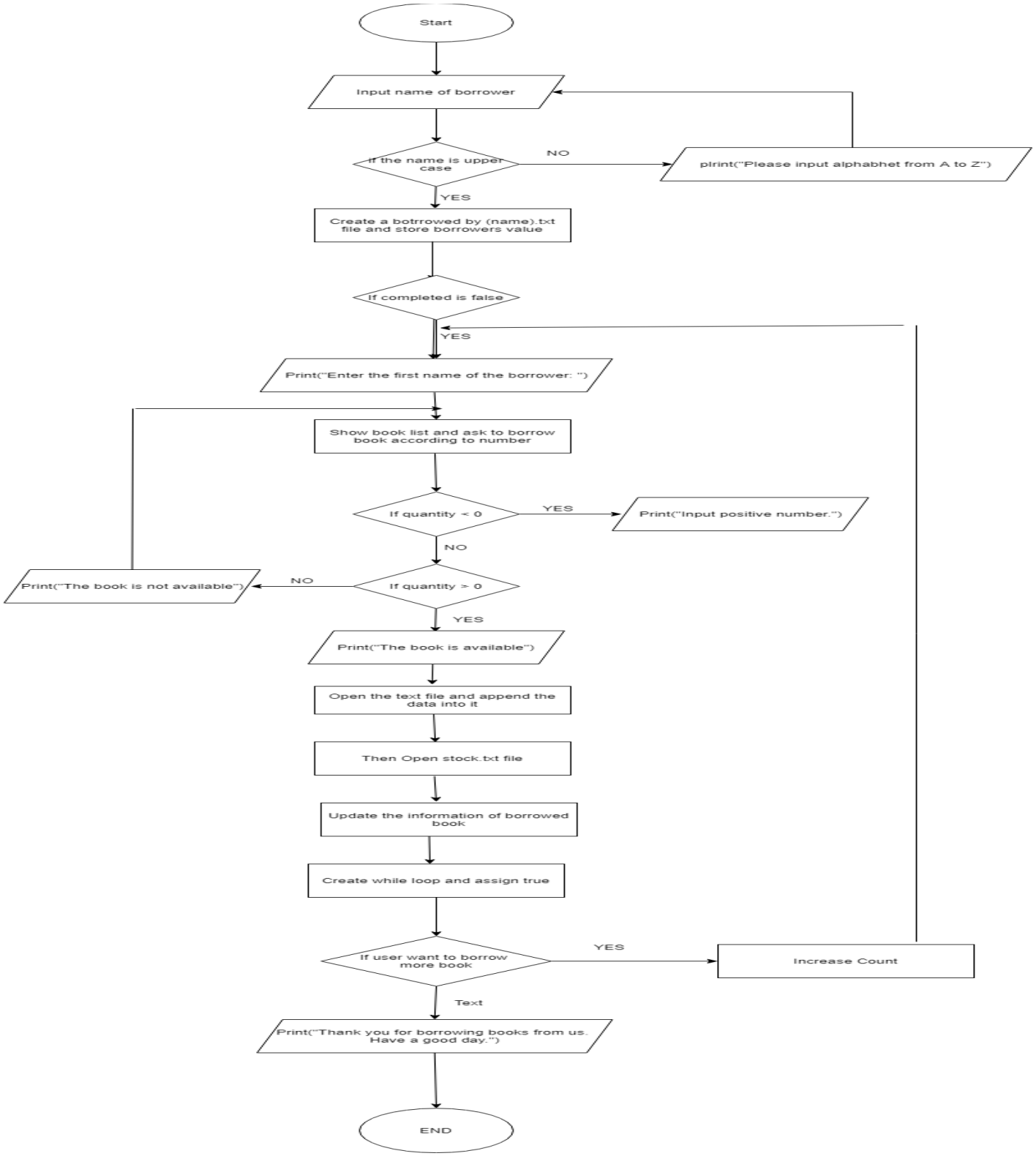


Figure 5: Flowchart of borrow

Flowchart of Return method:

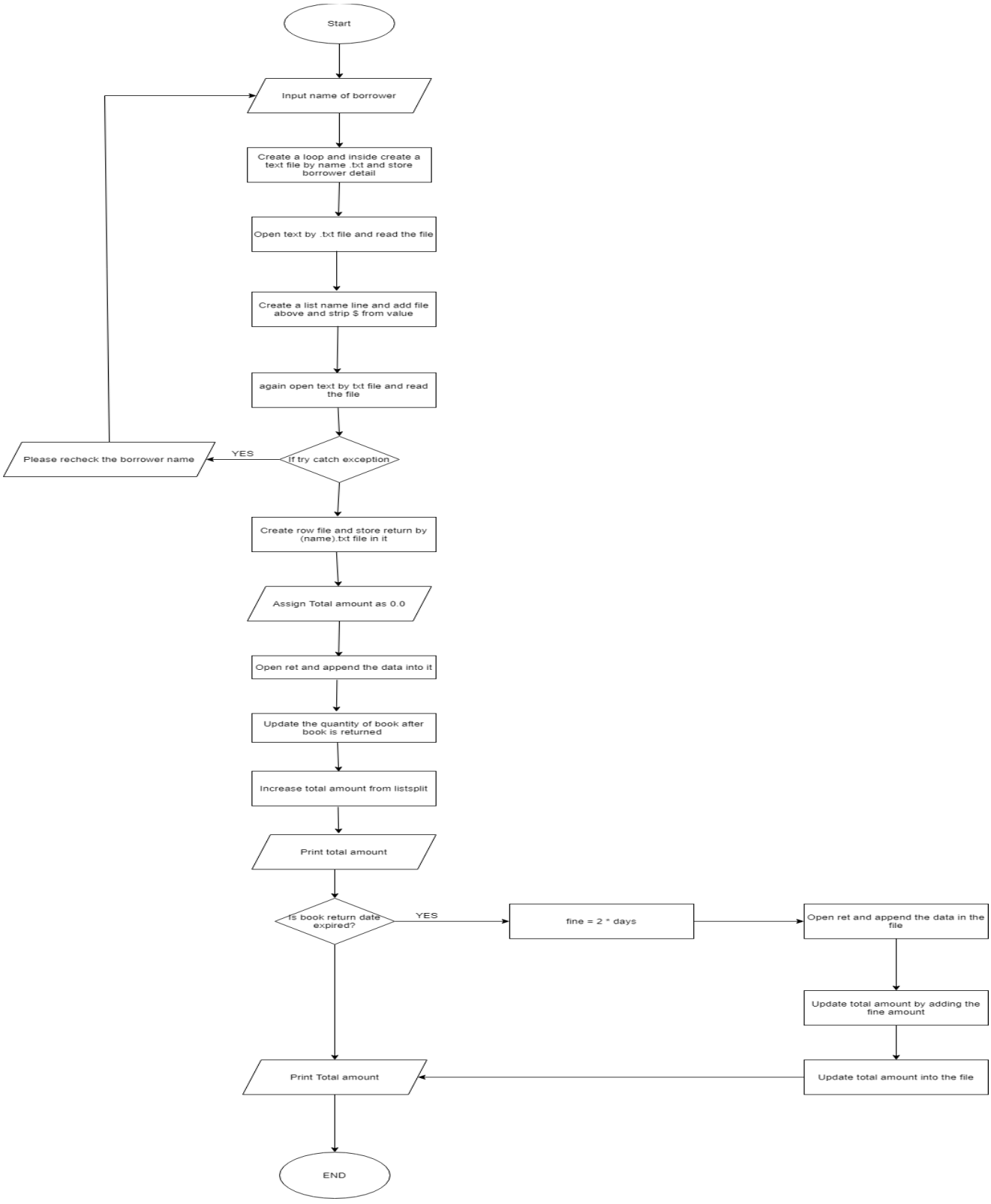


Figure 6: Flowchart of return method

CHAPTER 6: PSEUDOCODE

Pseudocode is a programming methodology that allows a programmer to represent an algorithm's implementation. Pseudocode is an unstructured technique of describing a program, not a programming language. It serves as a basic representation of a program's functions rather than having specific syntax. Since pseudocode is an informal language, it is mostly used to create a program blueprint or rough draft. Pseudocode cannot be compiled into an executable program since it is not a programming language. As a result, if pseudocode is to become a workable application, it must be translated into a specific programming language. The pseudocode of the program is given below:

a) Peudocode for main class

import listsplit
import borrow
import Return
while(True):
    Display (" Welcome to the Islington library management system ")
    Display ("<--->")
    Display ("Press 1 For Display")
    Display ("Press 2 For Borrowing a book")
    Display ("Press 3 For Returning book")
    Display ("Press 4 For Exit")
try
    Create a variable c
    Input c
    c **Display** ("Select a number from 1-4: ")
    Display ()

    if (c == 1) **then**
    Create an object named as lines file.
    Open stock.txt file and read the text into file object.
    Create a variable lines and add the text of file object.
    close object file.

```
        Else if (c== 2) then
                Call listsplit () function from listsplit file.
                Call borrowBook() function from borrowfile.
        Display ()

        Else if (c == 3) then
                Call listsplit () function from listsplit file.
                Call returnBook() function from Return file.
        Display ()

        Else if (c == 4) then
                Display (" Thank you For visiting Islington library. Have a good day.")
                break

        Else
                Display (" Please enter a valid number from 1-4")
               Display(“”)
               Display(“”)

         End if

except
        Display ("Please Input suggested number only.")
        Display ("")
        Display ("")
Display ("--------------------------------------------------------------------------------")
 End while
 End class
```

b) Pseudocode of borrow:

```
import Listsplit
import dt

borrowBook()
        Assign a new variable complete as false

while (True)
        Create a variable first_name.
                Input firstName
                firstName Display (“Enter the first name of the borrower.”)
        if firstName has alphabets then
                break

                Display ("Please Input valid characters from A to Z")
        End if
while (True)
        Create a variable lastName.
                Input lastName
                lastName Display (“Enter the last name of the borrower.
        if lastName has alphabets then
                break
                Display ("Please Input valid characters from A to Z")
         End if

    Create text variable and store Borrowed by- “name” .txt file
                Open text in Write data mode and Create an object “obj” in file
                Write (“  Islington  Library Management System  “) into object obj
                Write (“Borrowed By:” firstName from above and lastName from above) into
                object obj
                Write (“Date:” date from getDate () function and time from getTime () function )
                 into object obj
```

Write (“Name of the book” with some spaces and Write “Author Name”)

while completed == False

Display (“----------”)

Display ("Please select a book given below: ")

For i. in range length of bookName

Display (“Enter”, [i] , “to borrow book”, book from bookName [i])

try

Create a variable bo.

Input bo

bo **Display** (“Enter the book number: ")

if num < 0 **then**

Display ("Please enter the positive value.")

Else

try

if quantity[bo] from text file > 0 **then**

Display ("The book is borrowed")

Open text in appEnd mode in object “obj”

Write (“1. bookName[bo] from Listsplit and authurName[bo] from Listsplit**)** into object obj.

Decrease the quantity and update

Open stock.txt in Write mode in object “obj”

For I in range 3

Write(bookName[i] from Listsplit, author[i] from Listsplit, quantity [i] from Listsplit and cost [i] from Listsplit) into object obj

Assign newly Created variable loop as True

Assign newly Created variable count as 1

while loop = True

Create a variable.

Input opt

num **Display** (“"Do you want to borrow more books?. Press y For yes and n For no. Instruction: You cannot borrow same book.: ")

if opt == “Y” or “y” **then**

count

Display ("Please select an option given below”)

For i range length(bookName)

Display (“Enter” , [i] , “to borrow book”, book from bookName [i])

Create a variable val.

Input val

val **Display** (“Enter the book number:")

if (bo == val) **then**

Display (“Same book cannot be borrowed twice.

Elif quantity[val] from Listsplit > 0 **then**

Display (“The book is available and borrowed.”)

Open text in appEnd mode in object “obj”

Write (count in strings, bookName[val] from Listsplit and authorName[val] from Listsplit**)** into object obj.

Decrease the quantity and update

Open stock.txt in Write mode in object “obj”

For i in range 3

Write(bookName[i] from Listsplit, authorName[i] from listsplit, quantity[i] from Listsplit and cost [i] from Listsplit) into object obj

Assign complete as False

Else

Assign loop as False

break

Else if opt == “n’ or “N” **then**

Display (" Thank you For borrowing books from us. Have a good day.")

Assign loop as False

Assign complete as True

Else

Display ("Sorry! Invaid option.Please choose the given option only.")

Display (“”)

Display (“”)

End if

Else

Display ("The book is not available now. Thanks For visiting us.")

Call borrowBook () function

Assign complete as False

End if

except IndexError **then**

Display (“”)

Display (“Please choose given number only.”)

End try

except ValueError **then**

Display (“”)

Display (“Please choose the suggested number.”)

End try

End borrowBook ()

- **Pseudo code for return function:**

```
Import List
Import dt

Algorithm returnBook
    input firstName
    create an object named text

    try
        open the object text and read the data
        read lines of the text file as lines
        Then strip the $ from the lines

        open the object text and read the data
        read the data as value from the text
        display the value
    except
        display " The borrower name is not available, Please input another name."
        call returnBook

    create an object ret for return txt file
        open the ret and write in txt file
        write("              Library Management System \n "
        write("          Returned By: "+ name + "\n")
        write("     Date: " + dt.getDate() + "\t “Time:”  +dt.getDate() " +"\n")
        write("S.N. \t\t Book Name \t        Cost \n")

        assign variable totalamount as 0.0
 for i in range(len(3))
        if List.bookname[i] in value then
        open the ret and append the data in file
        write(str(i+1)+"\t\t"+Listsplit.bookName[i]+"\t\t$"+Listsplit.cost[i]+"\n")
        Increase the quantity of book by Listsplit.quantity[i] = Listsplit.quantity[i] + 1
        Add the totalamount of the book by totalamount = totalamount +
float(List.totalamount[i])

        Then,
```

```
        Display("Total amount: " + "$" + str(fineamount))

        display("Did the book return date expired?")
        display "Enter y for Yes and n for No: " and take input in opn

         if opn.upper() == 'Y' then
                display " By how many days was the book returned was late? " and take input
in day
                fineamount = 2 * day

        open the ret and append the data in the txt l
        write("\t\t\t\t\t"+"Fine amount: $"+ str(fineamount)+"\n")
                totalamount = totalamounta + fineamount
                display ("Total amount: "+ "$"+str(totalamount))
        open the ret and append the value in the txt file
        write("\t\t\t\t\t"+"Total: $"+ str(total_cost))
        display " The book is returned"
        display ("   ----------   ")


    open the stock file and update the data in the txt file
       for i in range(3)
          write(Listsplit.bookName[i]+","+Listsplit.authorName[i]+","+str(Listsplit.quantity[i])+","+
"$"+Listsplit.cost[i]+"\n")
```

- **Pseudo code for list function:**

```
Algortihm listsplit
    Declare global variable bookName
    Declare global variable authorName
    Declare global variable quantity
    Declare global variable cost

    Create a list of the varibale bookName
    Create a list of the varibale authorName
    Create a list of the varibale cost
    Create a list of the varibale quantity
```

```
    Open the stock.txt file and read the data as object file
    readlines of the file as line1
    strip the '\n' from the line1

    for i in range(len(line1))
if index == 0 then
                append the data in the bookName listsplit
        elif index == 1 then
                append the data in the authorName listsplit
        elif index == 2 then
                append the data in the quantity listsplit
        elif index == 3 then
                append the data in the cost listsplit by stripping the $ from it
                index = index + 1
```

. CHAPTER 8: PROGRAM:

- Display Process

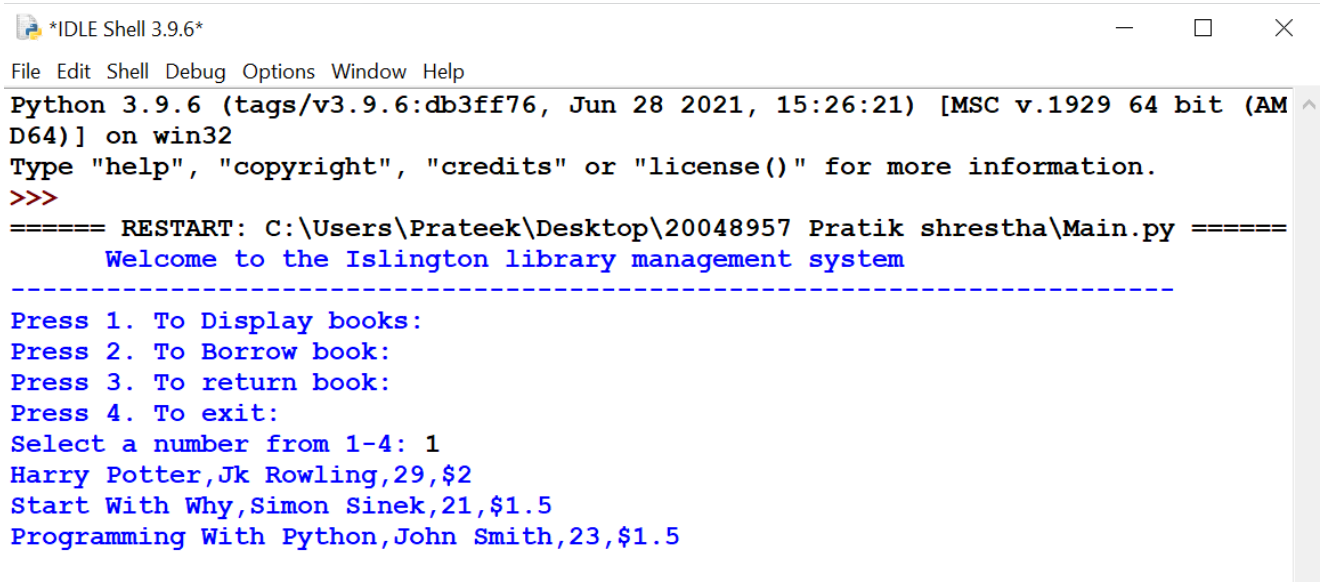


Figure 7: Display Process

First, the main function was executed, and it will request you for input for the task you want to do. Enter 1 to see a list of the library's available books.

- Borrow Process

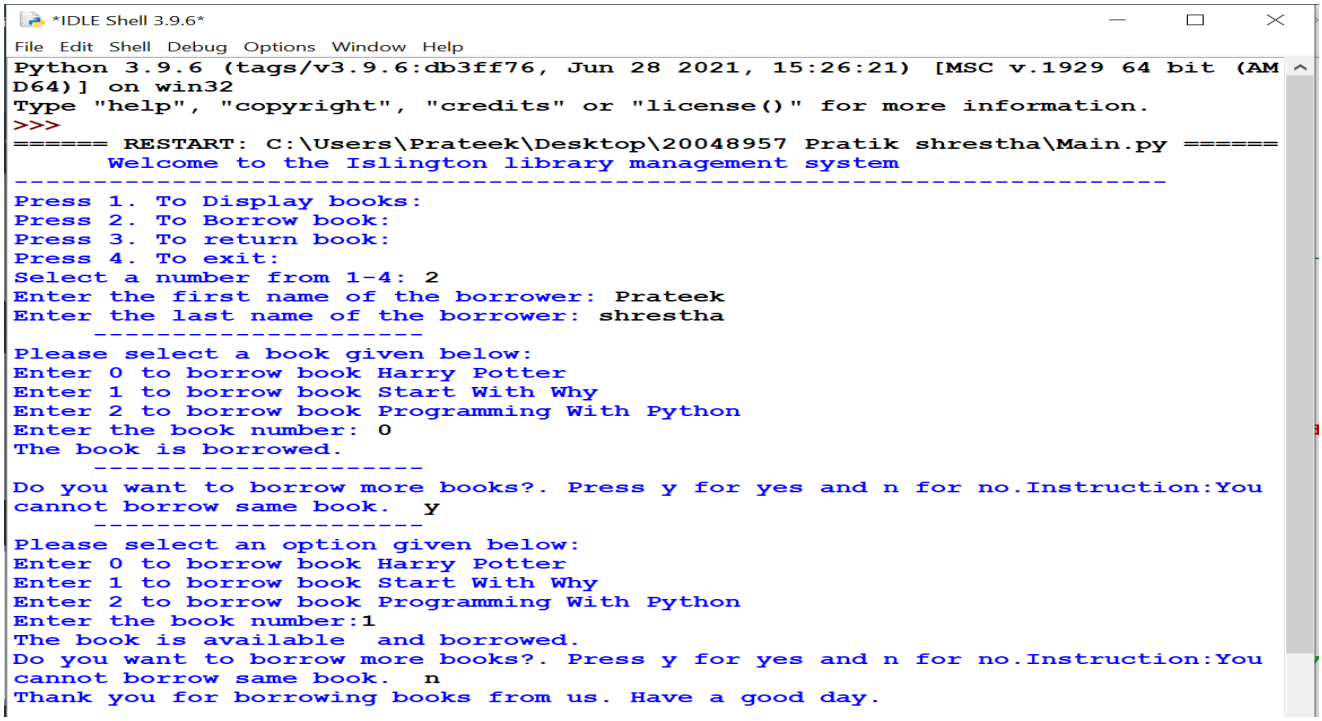


Figure 8: Borrow process

First and initially, the main function was run, and it requested for input for the task which you wished to do. Enter 2 if you'd like to borrow a book from the library. Then it will ask for your first and last names, following which it will ask for an input for the book you wish to borrow. Enter the book's index, and the software will produce a new text file including the essential information. The software will then ask whether you would like to borrow another book. If you input y, the program will continue the borrowing loop by adding data to the text file, but if you enter n, the program will terminate.

- Return Process

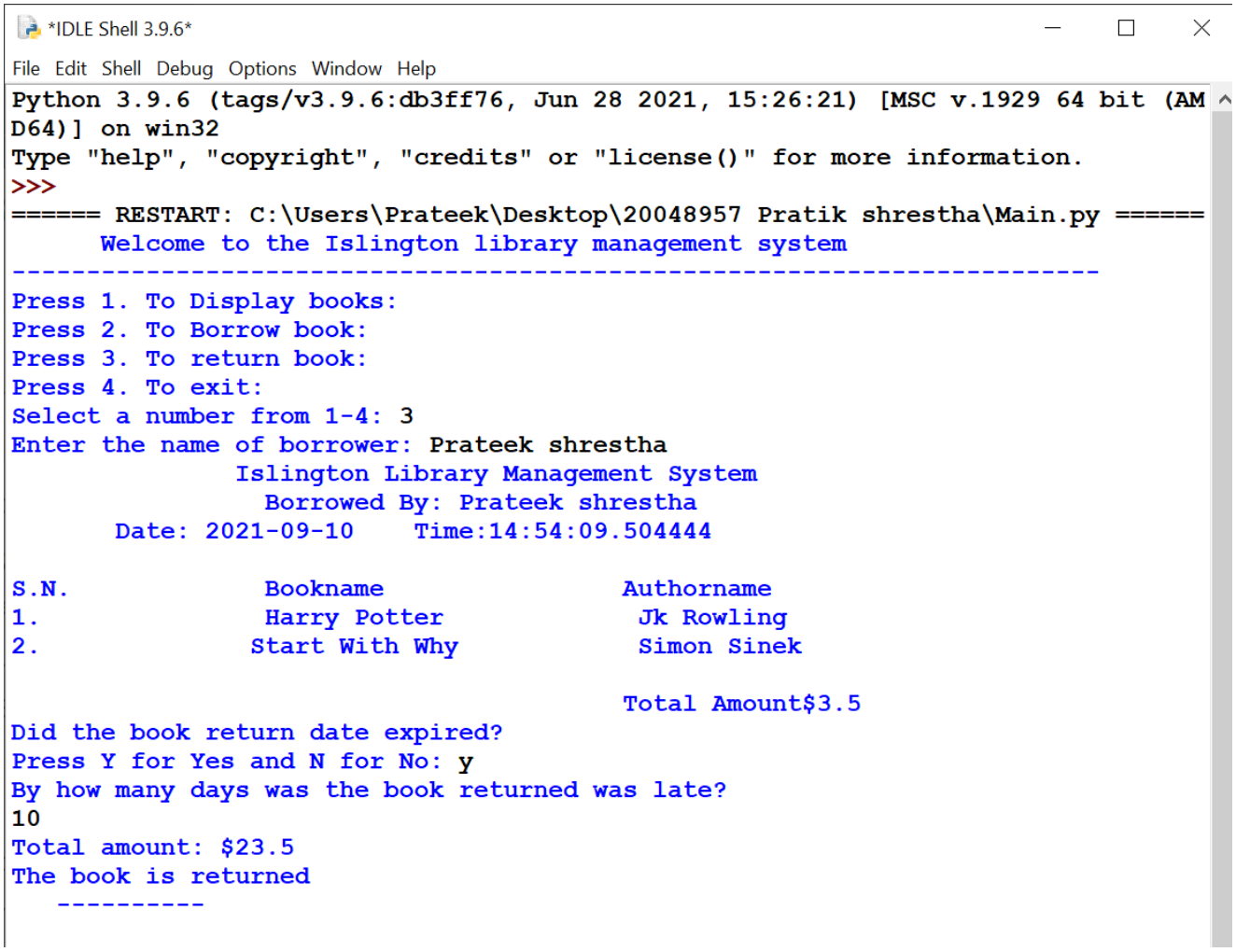
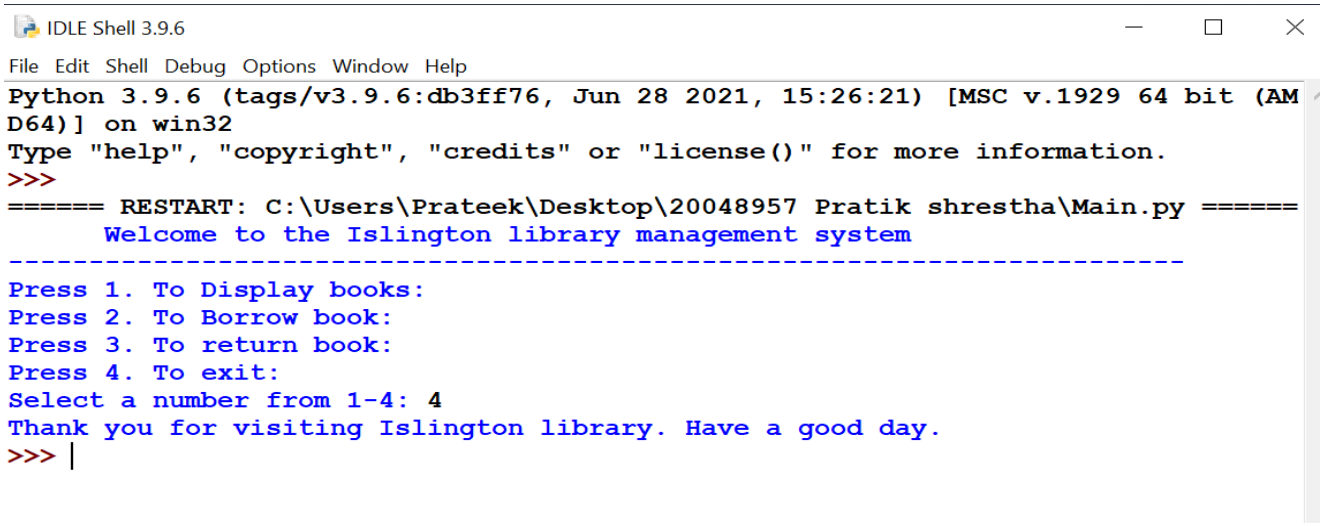


Figure 9: Return process

First, the main function was executed, and it will prompt you for input for the task you desire to do. To return a book to the library, enter 3. After that, it will ask for the borrower's name. It will show the information from the borrower file with the same name. It will then ask as to whether the return date has passed. If you input y, it will ask how many days it has been delayed, and the fine will be computed based on the number of days it has been delayed. If you enter n, I'll figure out how much the book will cost in total.

- Terminate Process



After the borrowing or returning process has been completed. The software will ask if you want to perform any other tasks; if you don't, enter 4; otherwise, the program will end.

CHAPTER 9: TEST CODE

Test 1: To test the program shows a message or not if the user inputs invalid input.

Objective:	To test if the program shows a message or not if the user inputs invalid input option.
Action:	➔ Firstly, opening the main file and running the file.

	➔ Then, enter the number which isn't on the list.
Expected Result:	A message should be displayed
Actual Result:	The required message was shown.
Conclusion:	The test was successful.

Table 1: To test user invalid input option

Action:

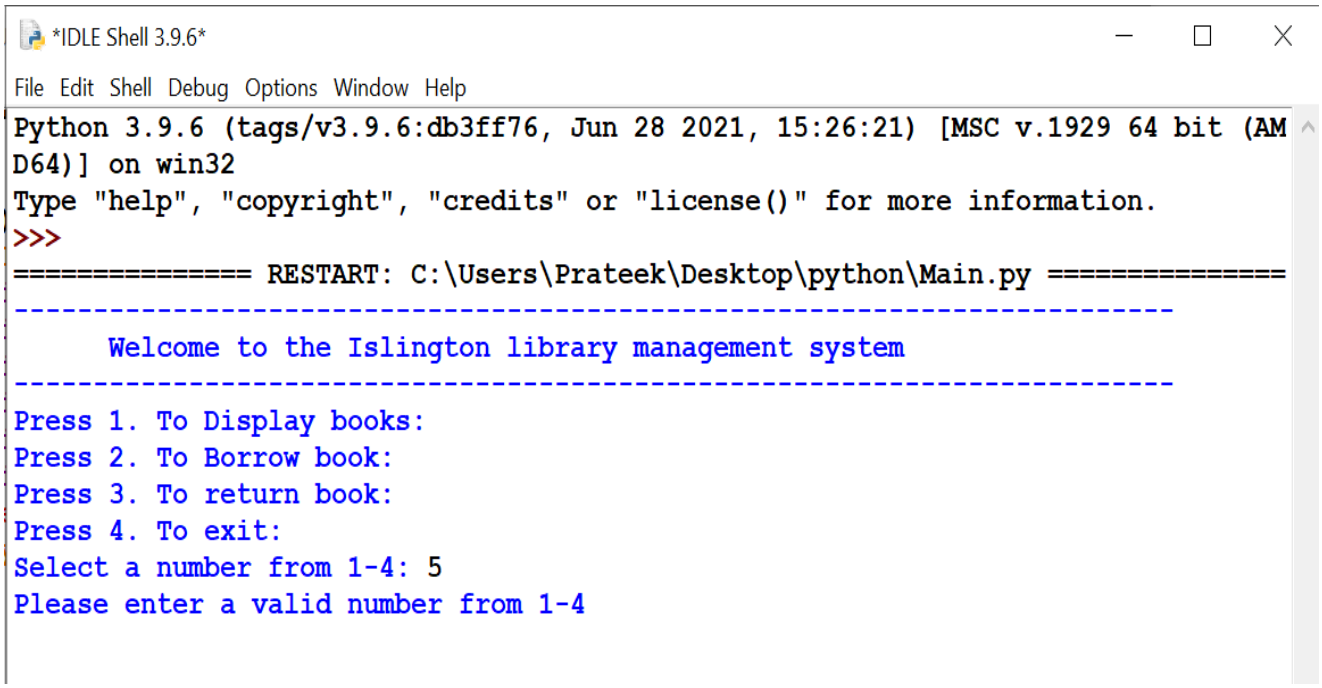


Figure 10: To test user invalid input option

Test 2: To test if the program shows a message or not if the user inputs negative value and non-existed value in borrow method

Objective:	To test if the program shows a message or not if the user inputs negative value and non-existed value in borrow method
Action:	➔ Firstly, opening the main file and running the file. ➔ Then select option 2 to borrow book ➔ Then, enter the negative number ➔ Finally, enter the number which isn't on the list.
Expected Result:	A message should be displayed If negative number: "Please enter the positive value." If non-exist number: "Please choose given number only."
Actual Result:	The required message was shown.
Conclusion:	The test was successful.

Table 2: To test whether it shows a message or not if the user inputs negative value and non-existed value in borrow method

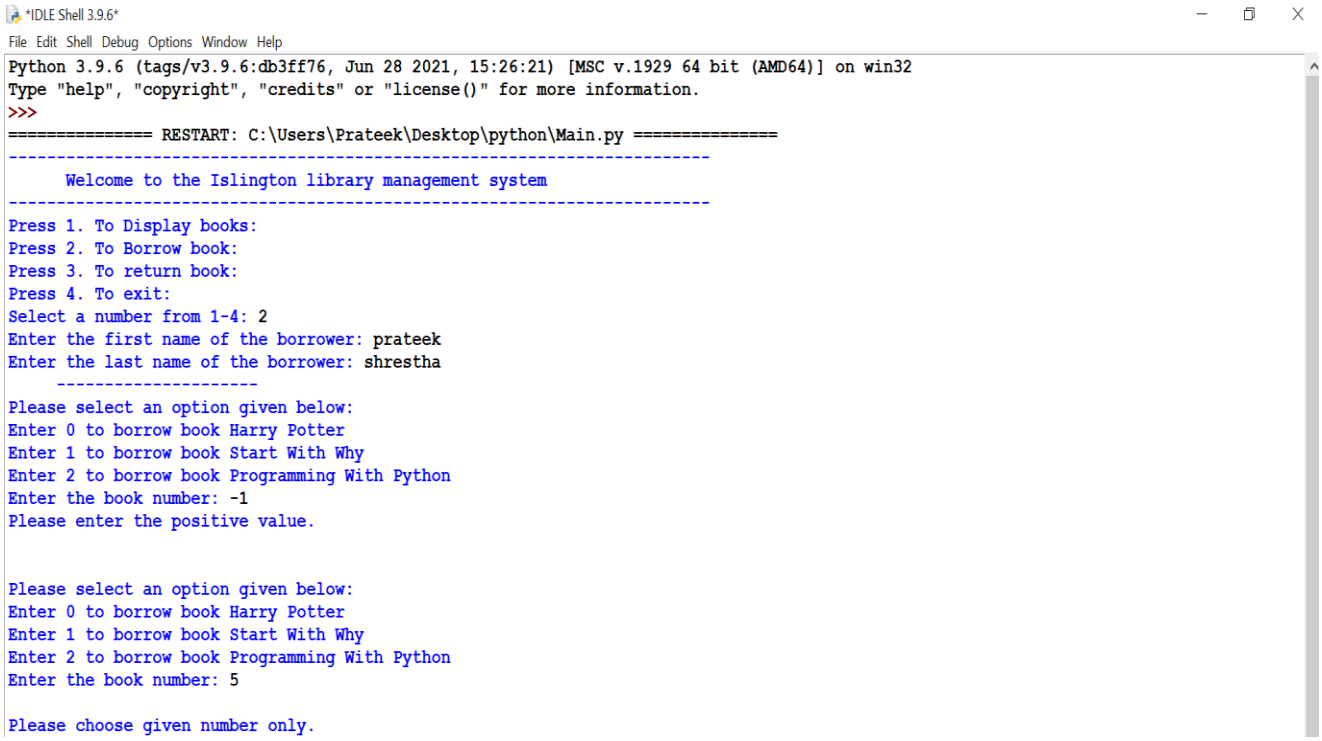


Figure 11: To test whether it shows a message or not if the user inputs negative value and non-existed value in borrow method

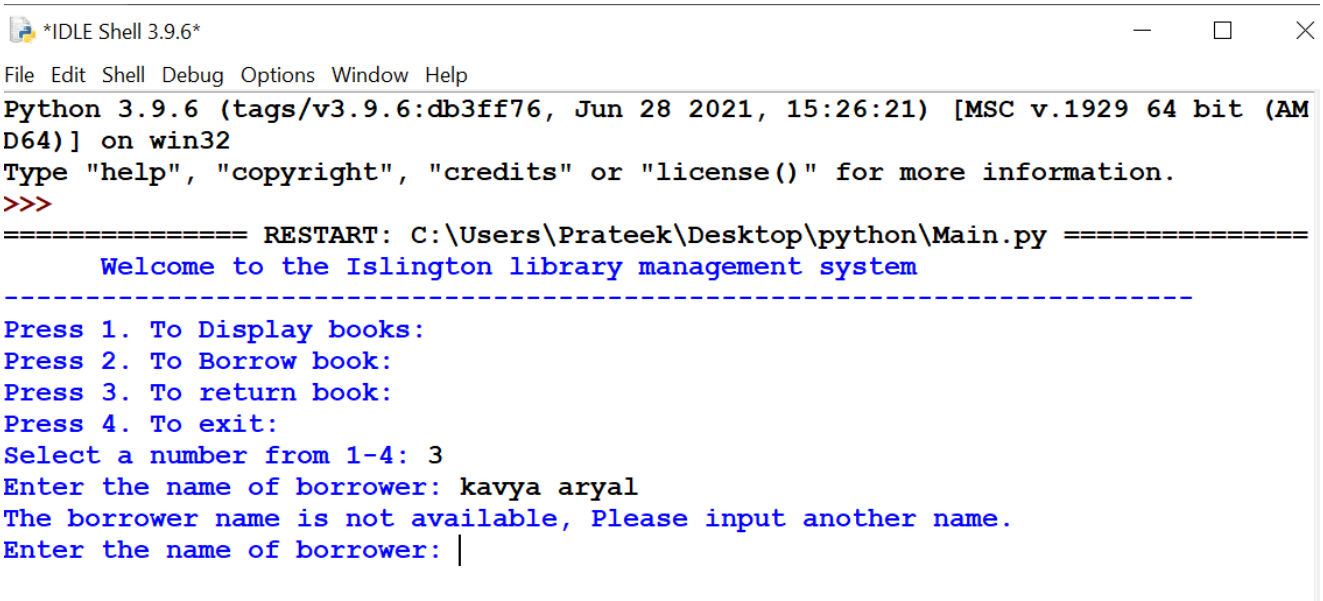


Figure 12: To test whether it shows a message or not if the user non-existed value in return.

Test 3: To test the borrow process and show borrow note.

Objective:	To test the borrow process and show borrow note.
Action:	➔ Firstly, opening the main file and running the file. ➔ Then select option 2 to borrow book ➔ Then, enter the first name and last name of user ➔ Then select the book number you want to buy ➔ Then select “y” for yes if you want to borrow more book. ➔ Finally select the book number you want to buy
Expected Result:	A message should be displayed i.e “The book is borrowed”.
Actual Result:	The required message was shown.
Conclusion:	The test was successful.

Table 3: To test the borrow process and show borrow note.

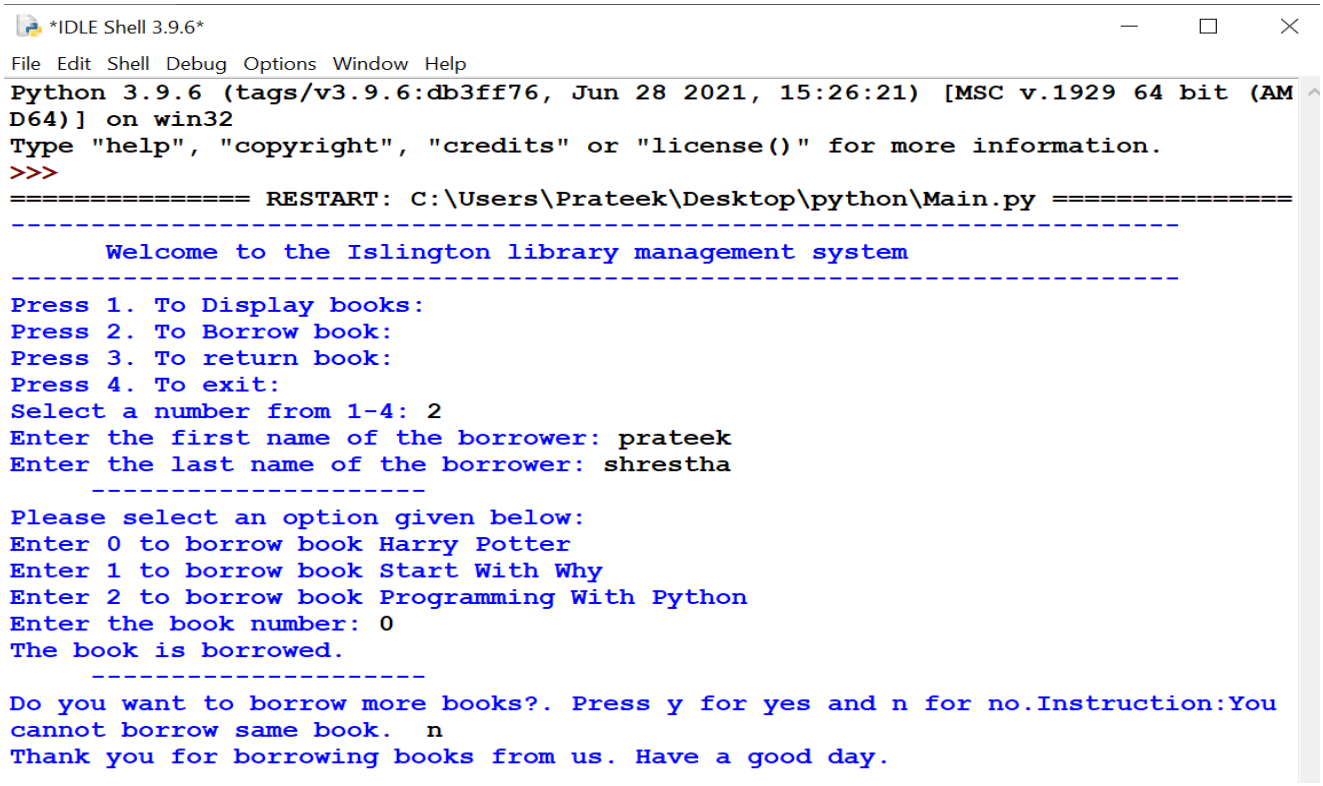


Figure 13: To test the borrow process and show borrow note in IDLE.

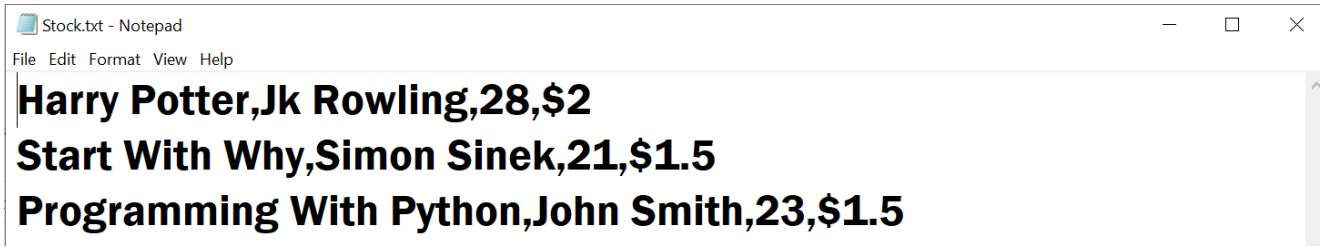


Figure 14: To test the borrow process and show borrow note.

Test4: To test the return process and show return note.

Objective:	To test the return process and show return note.
Action:	➔ Firstly, opening the main file and running the file. ➔ Then select option 3 to return book ➔ Then, enter the first name and last name of user ➔ Finally select “y” for yes if you delay submission or “n” for no if you submit on time
Expected Result:	A message should be displayed i.e “The book is borrowed”.
Actual Result:	The required message was shown .ie. “The book is returned.”
Conclusion:	The test was successful.

Table 4:To test the user inputs invalid input.

```
      Welcome to the Islington library management system
------------------------------------------------------------------------------
Press 1. To Display books:
Press 2. To Borrow book:
Press 3. To return book:
Press 4. To exit:
Select a number from 1-4: 3
Enter the name of borrower: prateek shrestha
               Islington Library Management System
                 Borrowed By: prateek shrestha
       Date: 2021-09-09    Time:19:11:57.734454

S.N.             Bookname                Authorname
1.               Harry Potter             Jk Rowling

                                         Total Amount$2.0
Did the book return date expired?
Press Y for Yes and N for No: y
By how many days was the book returned was late?
19
Total amount: $40.0
   ----------
```

Figure 15:The code for user inputs invalid input.

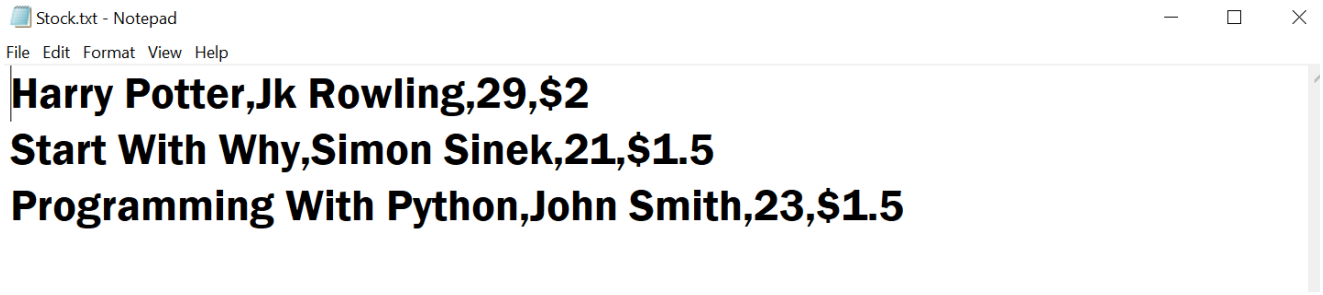


Figure 16: To test the update of stock.txt after return

Test 5: To test if the program stock is update (deducted or added).

Objective:	To test if the program stock is update (deducted or added).
Action:	➔ Firstly, opening the main file and running the file. ➔ Then, borrow the book by providing detail of borrower. ➔ Finally, return the borrow by providing necessary detail.
Expected Result:	The data is deducted and added successfully
Actual Result:	The data was deducted and added successfully
Conclusion:	The test was successful.

Table 5: To test stock update

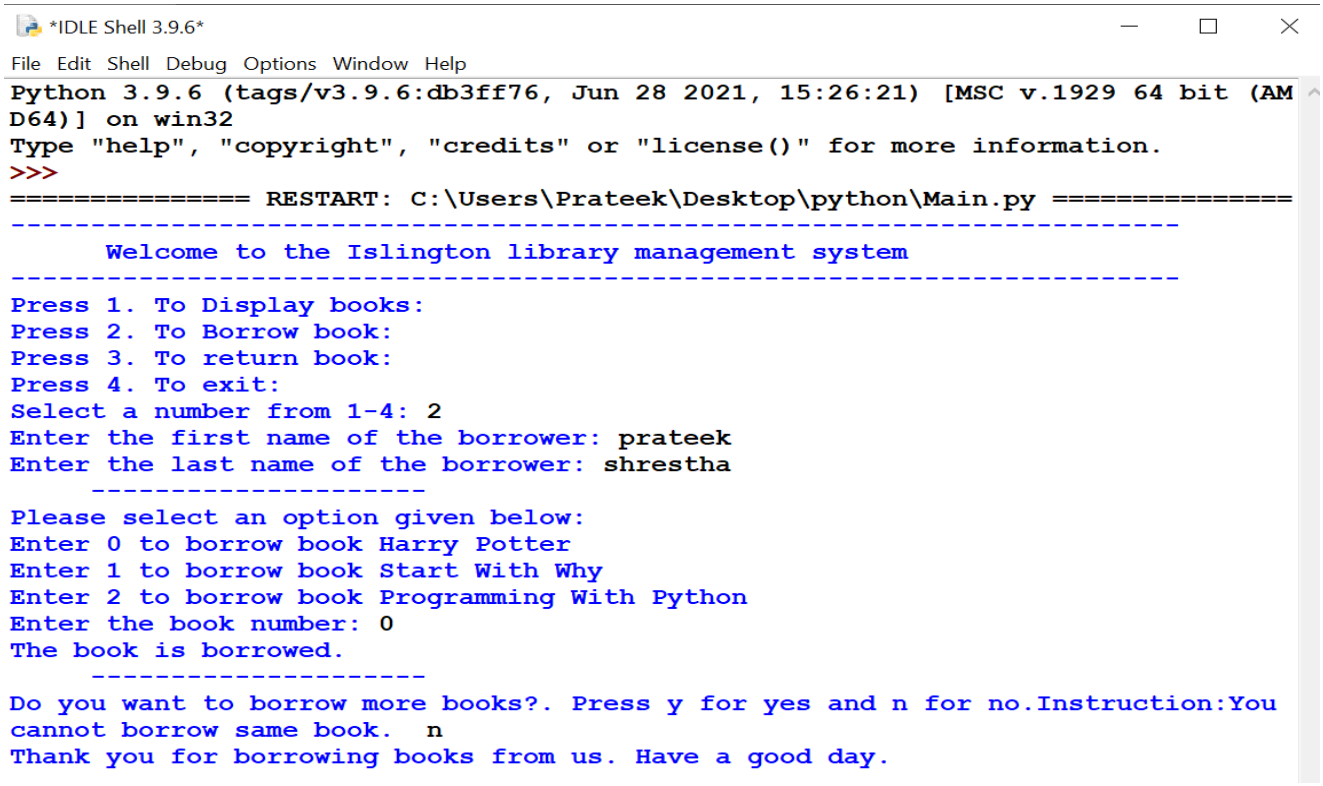


Figure 17: To test borrower process

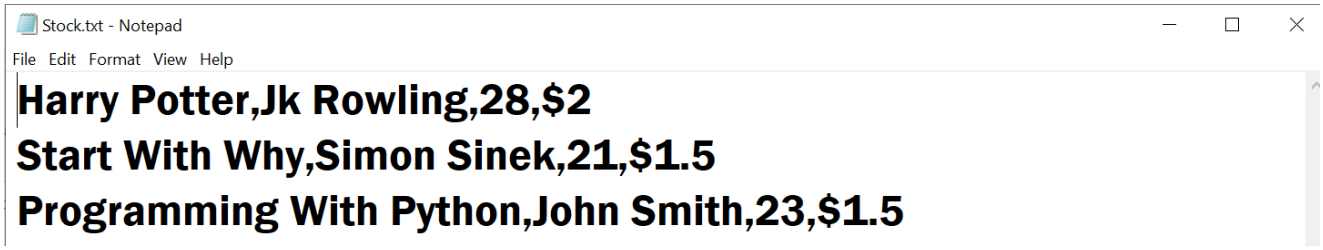


Figure 18: To test the update of stock.txt after borrow

```
      Welcome to the Islington library management system
------------------------------------------------------------------------------
Press 1. To Display books:
Press 2. To Borrow book:
Press 3. To return book:
Press 4. To exit:
Select a number from 1-4: 3
Enter the name of borrower: prateek shrestha
               Islington Library Management System
                 Borrowed By: prateek shrestha
       Date: 2021-09-09    Time:19:11:57.734454

S.N.             Bookname                Authorname
1.               Harry Potter             Jk Rowling

                                         Total Amount$2.0
Did the book return date expired?
Press Y for Yes and N for No: y
By how many days was the book returned was late?
19
Total amount: $40.0
   ----------
```

Figure 19: To test return process

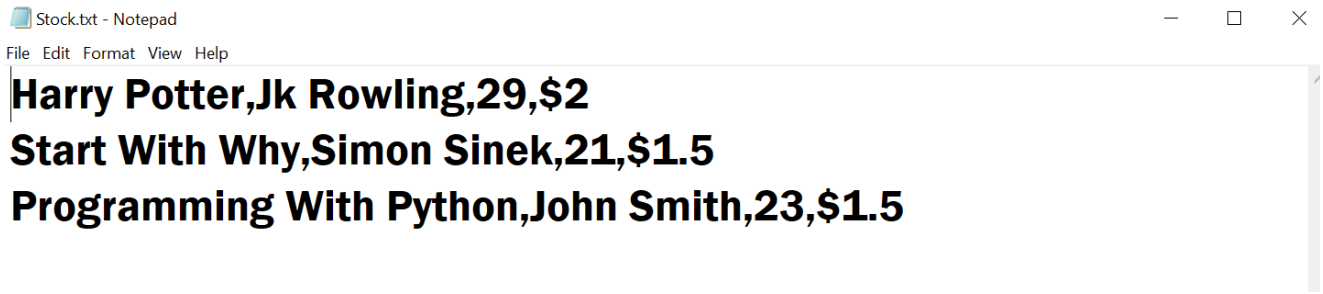


Figure 20: To test the update of stock.txt after return

CHAPTER 8: CONCLUSION

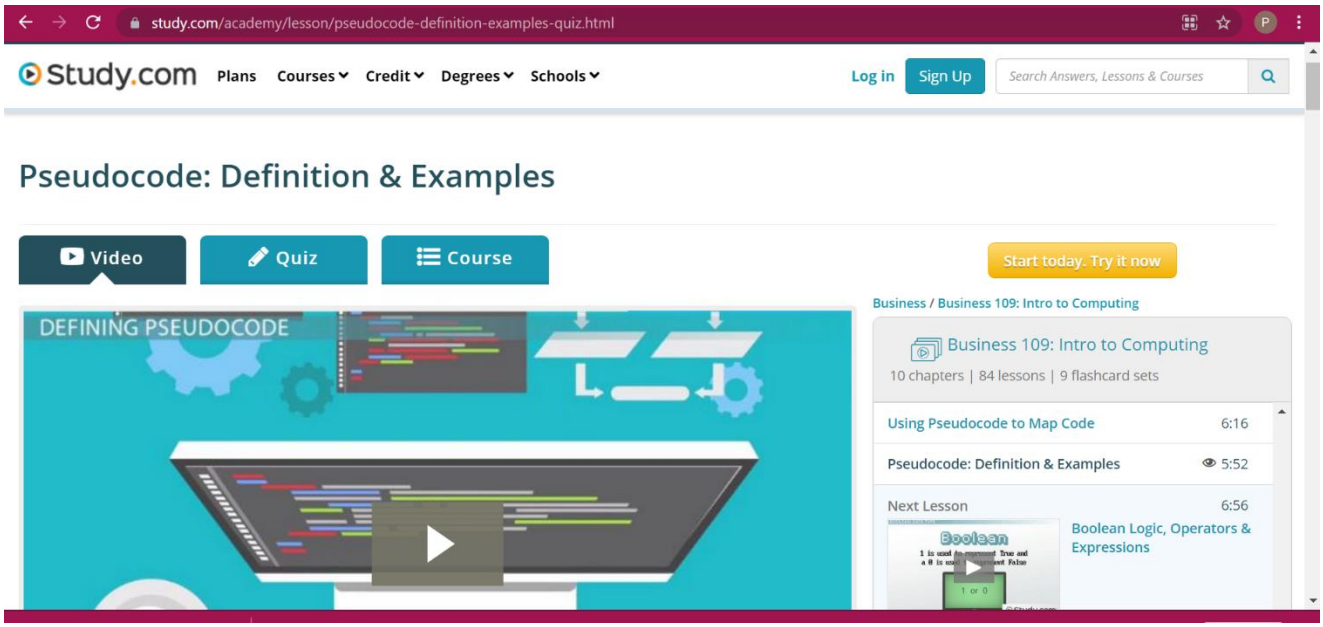
This coursework was successfully completed after a lot of hard work and several research projects on related topics such flowcharts, algorithms, and pseudocode. The coursework assignments were not easy at all. The code was created and tested to make sure that it was rid of bugs and errors and that it produced the preferred outcome. Each task was carried out in steps to ensure that all the tasks were completed successfully, making the task smoother.

This coursework demonstrated how to perform a task in a fast and organized manner, as well as how to develop skills that will be useful in the future. While working on the project, I learned a lot about Python, including its data structures, built-in functions, comments, while and for loops, if/else conditionals, and more. This assignment gave me the opportunity to obtain significant experience. Overall, spite of the fact that the tasks were difficult and needed days of hard work and effort, completing them was a lot of fun.

CHAPTER 9: RESEARCH

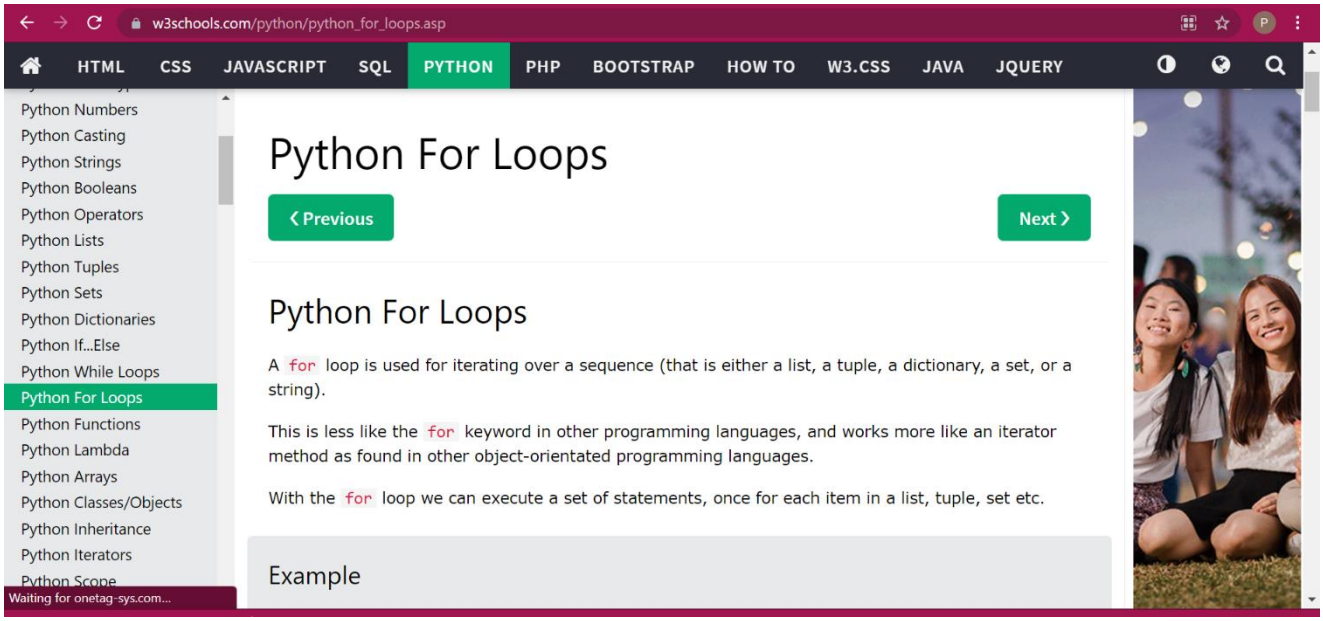
The assigned coursework was completed with a lot of researches, which made it a simpler. The assignments were made much easier to complete thanks to consistent effort and continuous research on different topics. We could figure out how all the software works and what the required codes are after a lot of research and effort. For information on various key aspects, a variety of websites, magazines, and books were also used. The research not only enhanced our power to achieve the coursework, but it also boosted our knowledge and understanding of a lot of Python concepts.

➔ **https://study.com/academy/lesson/pseudocode-definition-examples-quiz.html**



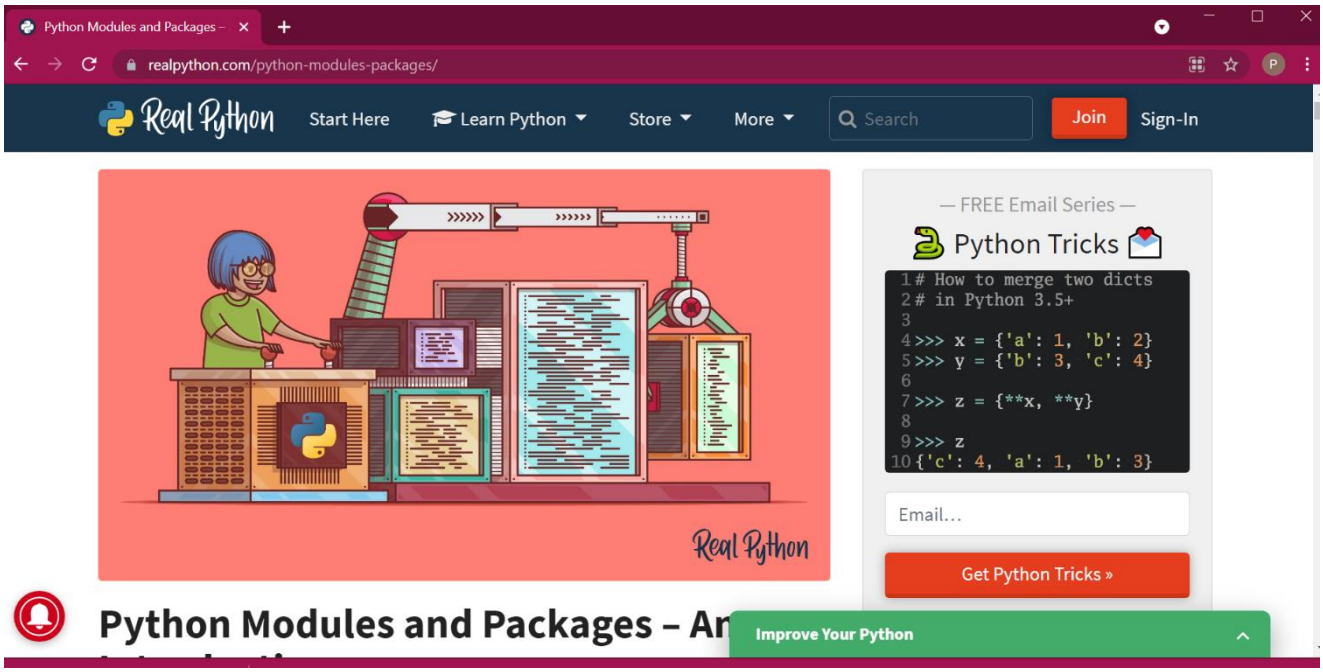
This website helped me learn about the pseudocode I tried to write for the assigned program.

➔ **https://www.w3schools.com/python/python_for_loops.asp**



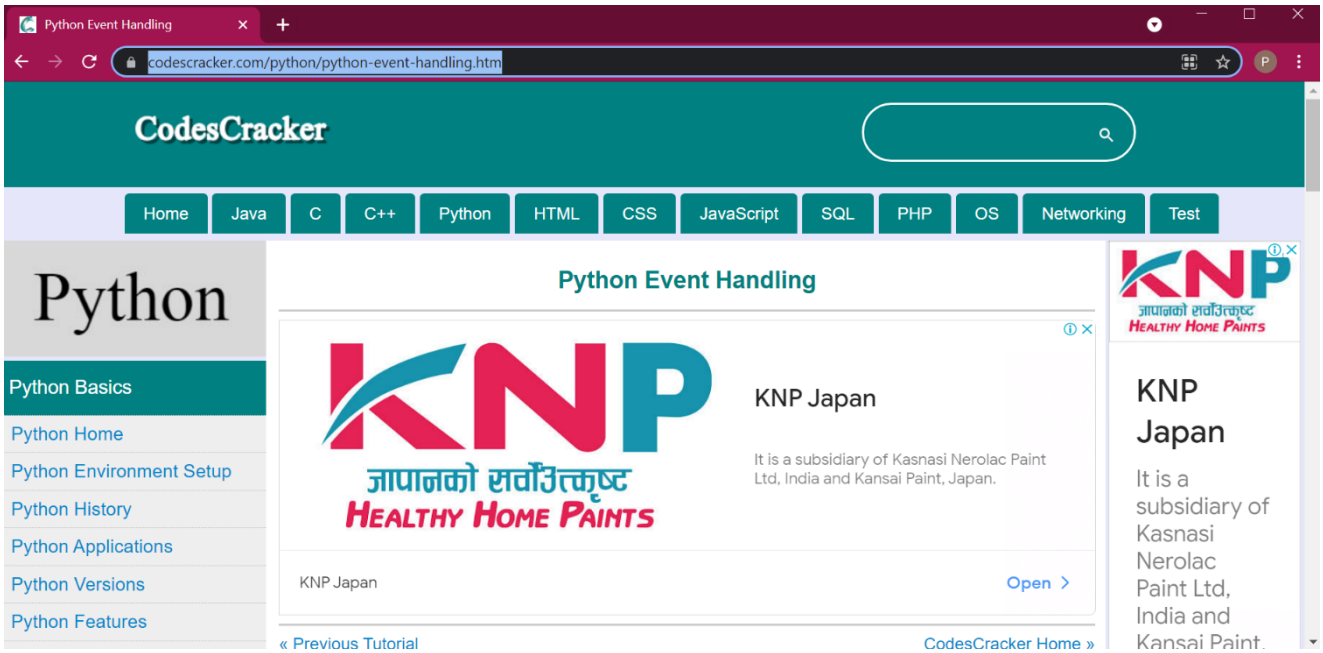
I visited this website to develop a full understanding of Python loops and how to use and operate them.

➔ **https://realpython.com/python-modules-packages/**



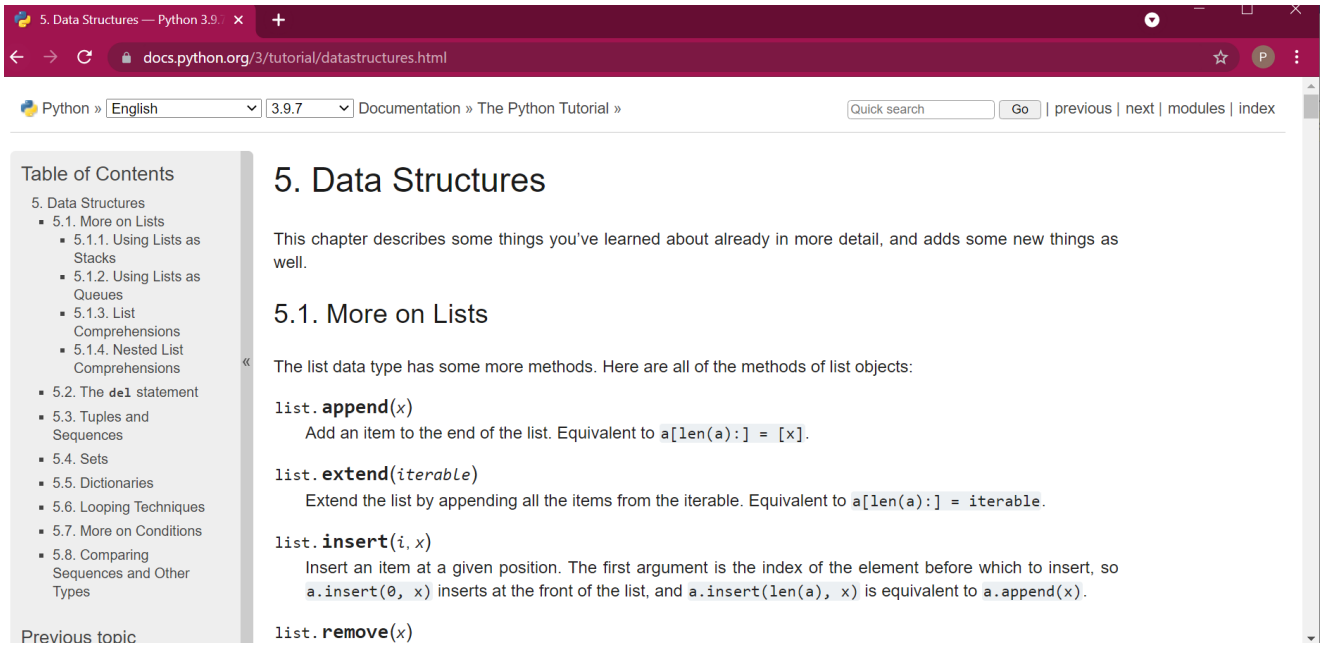
I used this website to increase my understanding on python modules and packages.

➔ **https://codescracker.com/python/python-event-handling.html**



I used this website to increase my understanding on python modules and packages.

➔ **https://docs.python.org/3/tutorial/datastructures.html**



This website was used to gain a basic knowledge of data structures. I learned many things terminology related to different kinds of data structures, as well as their purposes and effects.

CHAPTER 10: APPENDIX

Main Function:

```
# import file
import Return
import Listsplit
import Borrow

# creating start method
class start():
    while(True):
        print("      Welcome to the Islington library management system      ")
        print("-------------------------------------------------------------------------")
        print("Press 1. To Display books: ")
        print("Press 2. To Borrow book: ")
        print("Press 3. To return book: ")
        print("Press 4. To exit: ")
        # exception handling
        try:
            c = int(input("Select a number from 1-4: "))
            if(c==1):
                with open("stock.txt","r") as l: # open stock text file and read
                    lines=l.read()
                    print(lines)
                    print ()
              #condition check
             elif(c==2):
                Listsplit.listsplit()
```

```
            Borrow.borrowBook()
        elif(c==3):
            Listsplit.listsplit()
            Return.returnBook()
        elif(c==4):
            print("Thank you for visiting Islington library. Have a good day.")
            break
        else:
            print("Please enter a valid number from 1-4")
            print("")
            print("")
    except ValueError: # value error handling
        print("Please input suggested number only.")
        print("--------------------------------------------------------------------------")
```

Borrow Function:

```
# import dt and listsplit file
import dt
import Listsplit

# creating borrowbook method to borrow book
def borrowBook():
    complete = False
    while(True):
        firstName=input("Enter the first name of the borrower: ")
        if firstName.isalpha(): # isaplha method checks wheather the name is alphabhet or not.
            break
        print("Please input alphabet from A-Z")
    while(True):
        lastName = input("Enter the last name of the borrower: ")
        if lastName.isalpha():  # isaplha method checks wheather the name is alphabhet or not.
```

```
            break
        print("please input any alphabet from A-Z")

    text = "Borrow:" +firstName+" "+lastName+ ".txt" # creating text file to store borrow detail
    with open(text,"w") as l: # open text file to write
        l.write("              Islington Library Management System  \n")
        l.write("               Borrowed By: "+ firstName+" "+lastName+"\n")
        l.write("       Date: " + dt.getDate()+"    Time:"+ dt.getTime()+"\n\n")
        l.write("S.N. \t\t Bookname \t\t Authorname \n" )
    print("     ----------   ")

    while complete == False:
        print("Please select a book given below: ")
        for i in range(len(Listsplit.bookName )):
            print("Enter", i, "to borrow book", Listsplit.bookName[i])

        try:
            bo = int(input("Enter the book number: "))
            if(bo<0):
                print("Please enter the positive value.")
                print("")
                print("")
            else:
                try:
                    if(int(Listsplit.quantity[bo])>0):
                        print("The book is borrowed.")
                        print("     ----------   ")
                        with open(text,"a") as l:
                            l.write("1.              \t\t          "+          Listsplit.bookName[bo]+"\t\t
"+Listsplit.authorName[bo]+"\n")

                        Listsplit.quantity[bo]=int(Listsplit.quantity[bo])-1
```

```
            with open("Stock.txt","w") as l:
                for i in range(3):

l.write(Listsplit.bookName[i]+","+Listsplit.authorName[i]+","+str(Listsplit.quantity[i])+","+"$"+Li
stsplit.cost[i]+"\n")



            # Code to borrow multiple books
            loop=True
            count=1
            # loop for borrowing multiple book
            while loop == True:
                opt = str(input("Do you want to borrow more books?. Press y for yes and n
for no.Instruction:You cannot borrow same book.  "))
                if(opt.upper()=="Y"): # upper method return uppercase string from given
string
                    count=count+1
                    print("     ----------   ")
                    print("Please select an option given below:")
                    # for loop
                    for i in range(len(Listsplit.bookName)):
                        print("Enter", i, "to borrow book", Listsplit.bookName[i])
                    val = int(input("Enter the book number:")) # user input for book number
                    if (bo==val):
                        print("Same book cannot be borrowed twice.")
                    elif(int(Listsplit.quantity[val])>0):
                        print("The book is available  and borrowed.") # book is borrowed
                        with open(text,"a") as l:
                            l.write(str(count)      +".      \t\t"+      Listsplit.bookName[val]+"\t\t
"+Listsplit.authorName[val]+"\n") # storing borrowed book detail.

                        Listsplit.quantity[val]=int(Listsplit.quantity[val])-1 # decreasing quantity
```

```
of book after borrowed
                        with open("Stock.txt","w") as l:
                            for i in range(3):

l.write(Listsplit.bookName[i]+","+Listsplit.authorName[i]+","+str(Listsplit.quantity[i])+","+"$"+Li
stsplit.cost[i]+"\n")
                                complete=False
                    else:
                        loop=False
                        break
                elif (opt.upper()=="N"): # upper method return uppercase string from given
string
                    print ("Thank you for borrowing books from us. Have a good day. ")
                    print("")
                    loop=False
                    complete=True
                else:
                    print("Sorry! Invaid option.Please choose the given option only.")
                    print("")
                    print("")


            else:
                print("The book is not available now. Thanks for visiting us.") # print if book is
not available
                borrowBook()
                complete=False
        except IndexError: # index error exception handling
            print("")
            print("Please choose given number only.")
    except ValueError: # value error exception handling
        print("")
        print("Please choose the suggested number.")
```

Return Function:

```
import Listsplit
import dt
def returnBook():
    name = input("Enter the name of borrower: ")
    text = "Borrow:" +name+ ".txt"
    try:
        with open(text,"r") as l:
            lines=l.readlines()
            lines=[text.strip("$") for text in lines]

        with open(text,"r") as l:
            value = l.read()
            print(value)
    except:
        print("The borrower name is not available, Please input another name.")
        returnBook()

    ret = "Return: "+name+".txt"
    with open(ret,"w")as l:
        l.write("               Library Management System \n")
        l.write("                   Returned By: "+ name+"\n")
        l.write("       Date: " + dt.getDate()+"    Time:"+ dt.getTime()+"\n\n")
        l.write("S.N.\t\t\t\tBookName\t\t\t\tCost\n")



    totalamt = 0.0
    for i in range(3):
        if Listsplit.bookName[i] in value:
            with open(ret,"a") as l:
                l.write(str(i+1)+"\t"+Listsplit.bookName[i]+"\t$"+Listsplit.cost[i]+"\n")
```

```
            Listsplit.quantity[i]=int(Listsplit.quantity[i])+1
        totalamt += float(Listsplit.cost[i])

    print("\t\t\t\t\t Total Amount"+"$"+str(totalamt))
    print("Did the book return date expired? ")
    opn = input("Press Y for Yes and N for No: ")
    if(opn.upper()=="Y"):
        print("By how many days was the book returned was late? ")
        day = int(input())
        fineamt = 2*day
        with open(ret,"a")as l:
            l.write("\t\t\t Fine amount: $"+ str(fineamt)+"\n")
        totalamt = totalamt + fineamt
    print("Total amount: "+ "$"+str(totalamt))
    with open(ret,"a")as l:
        l.write("\t\t\t\t\tTotal amount: $"+ str(totalamt))
    print("The book is returned")
    print ("   ----------   ")
    print("")
    print("")
    with open("Stock.txt","w") as l:
            for i in range(3):

l.write(Listsplit.bookName[i]+","+Listsplit.authorName[i]+","+str(Listsplit.quantity[i])+","+"$"+Li
stsplit.cost[i]+"\n")
```

Listsplit method:

```
# defining method
def listsplit():
    # defining global variable
    global bookName
    global authorName
```

```
    global quantity
    global cost

    # creating list of variables
    bookName=[]
    authorName=[]
    quantity=[]
    cost=[]
    with open("Stock.txt","r") as l: # open stock.txt file
        line1=l.readlines()
        line1=[line.strip('\n') for line in line1] # strip \n from list
        for i in range(len(line1)):
            index=0
            for c in line1[i].split(','):
                if(index==0): # condition 1
                    bookName.append(c)
                elif(index==1): # condition 2
                    authorName.append(c)
                elif(index==2): # condition 3
                    quantity.append(c)
                elif(index==3): # conditon 4
                    cost.append(c.strip("$")) # strip $ from cost
                index+=1
```

Datetime function:

```
# defining get method for date
def getDate():
    import datetime
    now=datetime.datetime.now
    #print("Date: ",now().date())
    return str(now().date())
# defining get method for time
def getTime():
    import datetime
    now=datetime.datetime.now
    #print("Time: ",now().time())
    return str(now().time())
```