



islington college
(इस्लिंग्टन कॉलेज)

Module Code & Module Title

CC4051N FUNDAMENTALS OF COMPUTING

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-2021 Autumn

Student Name: SALAJ SUBEDI

London Met ID: 20049007

College ID: np01cp4s210311

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 mark of zero will be awarded.

Table of Contents

1. Introduction	1
2. Discussion and Analysis.....	2
2.1 ALGORITHM	2
2.2 FLOWCHART	3
2.3 PSEUDOCODE	4
2.4 DATA STRUCTURES.....	14
3.Program	20
4. Testing	31
Test 1.....	31
Test 2.....	32
Test 3.....	34
Test 4.....	36
Test 5.....	38
Conclusion	40
Appendix	41

List of Figures

Figure 1: String Data in Python	14
Figure 2 Creating lists	15
Figure 3 Tuple creation	16
Figure 4 Creating sets	17
Figure 5 Dictionary creation	18
Figure 6: BOOKs stock.....	21
Figure 7: MAIN.py	22
Figure 8: Date.py.....	23
Figure 9: Lists.py	23
Figure 10 : Borrow.py	24
Figure 11: contd. Borrow	25
Figure 12: Borrow process in output along with generated .txt file	26
Figure 13: Return.py.....	27
Figure 14: contd. Return.py	28
Figure 15: Return process along with generated .txt file	29
Figure 16: Termination after choosing an option.	30
Figure 17: TEST 1; INVALID DATA ENTRY	31
Figure 19: invalid data entry	32
Figure 20: Non-existent data entry	33
Figure 21 BORROWING BOOKS.....	34
Figure 22: BORROWING BOOKS WITH TEXT FILE.....	35
Figure 23: RETURNING BOOKS	36
Figure 24: Returning books with Txt file	37
Figure 25: The book Stock	38
Figure 26: Book quantity reduced after borrowing.....	39
Figure 27: Book quantity being added after returning books	39

List of Tables

Table 1: TEST 1	31
Table 2: TEST 2	32
Table 3: TEST 3	34
Table 4 : Test 4	36
Table 5: Test 5	38

1. Introduction

Library management system is a project for managing borrowing and returning of book from a small library. The management of books from displaying borrowing returning and keeping ledger of borrowed books and returned books of each person is done using python codes. This application shows books available to borrow by a user. Handling invalid inputs and waiting for a valid input from user is also to be included in this project.

After each transaction of book, the book stock database is required to be updated. For returning the borrowed book user can proceed the book returning process using return command. Similar to the book borrowing process, after returning the book an update is to be made to the book database along with appropriate message. The cost of borrowing a book should be shown to user and appropriate fine should be added if book is not returned within return time. This application displays the list of books available to borrow. If the user selected book is available to borrow, book borrowing process is conducted. A message is generated displaying details. To return a borrowed book, return command is provided to the program which enables the book returning process. After completion of either book borrowing or book returning the main menu displays the books available to borrow and waits for user command to either burrow further or to return or to completely exit a program. To exit the program, exit command can be entered. This project was developed in order to complete the first coursework of Fundamentals of Computing. There were a lot of hardships and sleepless nights to go through in order to complete the project in time. Lots of research, discussion, analysis and follow up meetings with teachers were required to be done in order to develop a library management system as required in the course work.

2. Discussion and Analysis

2.1 ALGORITHM

An Algorithm is step wise procedure to be followed to solve a problem. Developing algorithm is first step towards programming. It defines the steps to be taken which later leads the way towards creating a software. An algorithm is not limited to one platform. In fact, same algorithm can be used to develop programs in almost all programming language available till date. Here's the algorithm for this project:

Step1: Start

Step2: set main loop true, with in main loop do the following

Step3: Read a text files containing book stock and display 4 options to display,
borrow Return, display, terminate

Step4: Request user input in for decision making of either burrowing, displaying
Books, returning books or exiting the program as a whole

Step5: Check for user input validate. if invalid, display appropriate message and
Loop to step.4 to get user input again,

Step6: Check if user input is valid: 2 for burrowing book and proceed accordingly

Step7: Check user input if user wants to burrow more books. If yes go to step:6 finalize
The borrow book process by issuing a borrow note details of borrowed books.

Step8: from step 4 check if user input if valid: 3 return books and proceed accordingly
and issue a return bill containing all details of cost fine and books returned

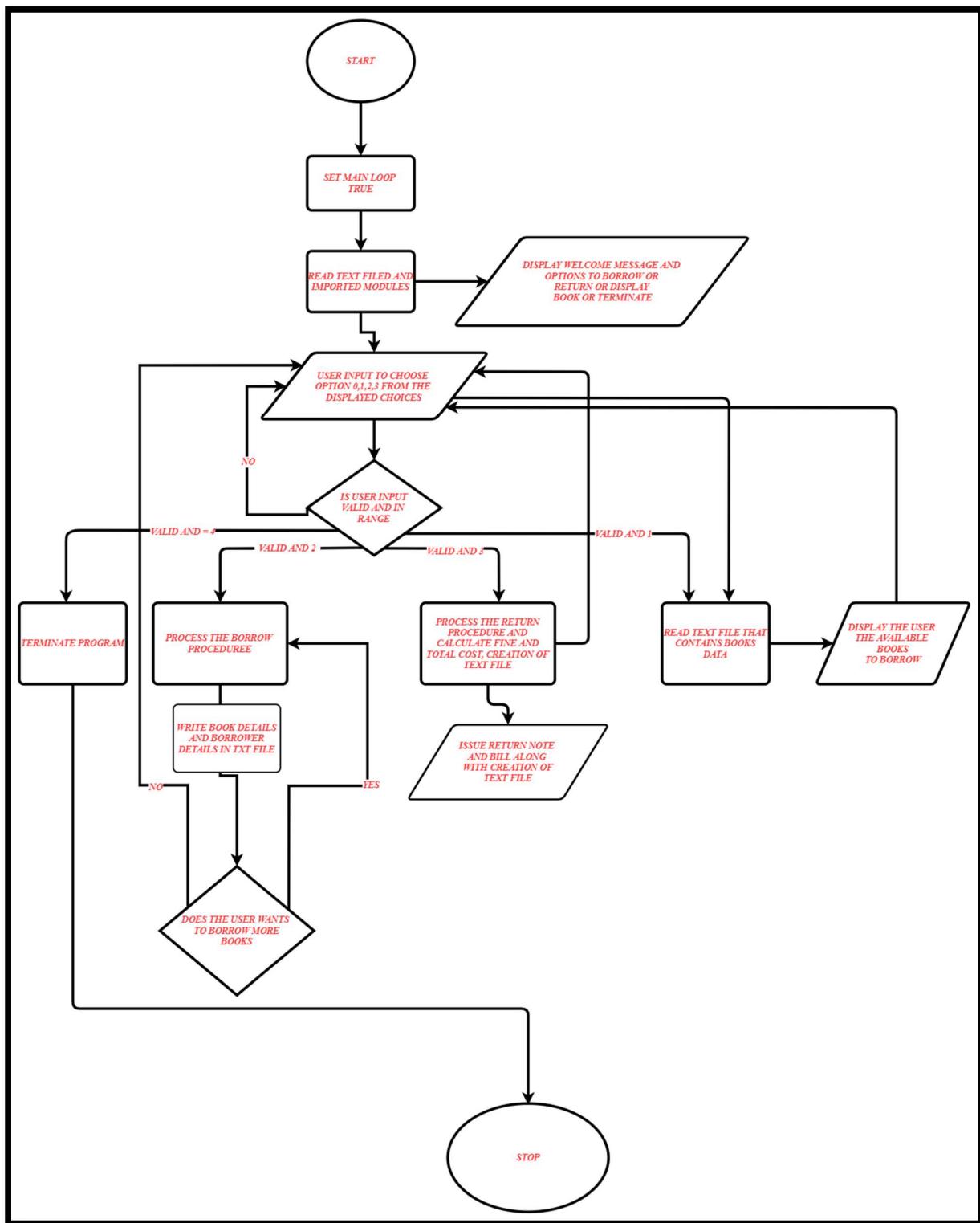
Step9: go to step 4 for further processing

Step10: from step 4 Check for valid data: 0 and display books if user requests it
And go to step 4

Step11: from step 4 Accept valid data value 4 for exiting program and terminate the
Program entirely.

Step 12: STOP

2.2 FLOWCHART



2.3 PSEUDOCODE

```
>>> MAIN.py

Start

import BorrowBook
import ReturnBook
import Lists
import Date

define main
    looping=True
    while looping==True
        #display choice
        display  \n
        display  -----Welcome to our Library-----
        display  Please Enter  1  to Display the books
        display  Please Enter  2  to Borrow our books
        display  Please Enter  3  to Return a borrowed book
        display  To Exit Please Enter  4
        display  -----
        # try block to look for errors in
        try
            ##Asking input
            id= int  input  Provide your input here -
            #Using the if and else to process further.
            if  id==1
                #opening a text file and displaying its content
                display  \n
                display  books here are displayed in NAME AUTHOR QUANTITY
                BORROWING PRICE format
                file=open  Books.txt  r
```

```
read=file.read
display  read
file.close
else if  id==2
    #Calling Lists and BorrowBook module
    Lists.lists
    BorrowBook.borrow
else if  id==3
    #Calling Lists and ReturnBook module
    Lists.lists
    ReturnBook.Return
else if  id==4
    #Displaying
    display  Thank you for choosing us!
    break
else
    # else block
    display  Please enter a number from the given choices.
END if
except ValueError
    #except block if error occurs
    display  Invalid! only numeric data that is within the given choices are
accepted.
close main
STOP
```

BorrowBooks.py

```
import Date
import Lists

def borrow():
    IsBorrowed=False
    #Using loops
    while(True):
        # input values
        FirstName=input( Please Enter the first name of the borrower: )
        LastName=input( Please Enter the last name of the borrower: )
        print( \n )
        break
    # creating a txt file on borrower
    record= Borrowed by  FirstName .txt
    file=open(record w )
    file.write( \t\t\tDetails of the Borrower and the book borrowed\n )
    file.write( Borrowed by: FirstName   LastName \n )
    file.write( Date and Time of borrowing: Date.datetime() \n\n )
    file.write( \t\tS.N. \t\tName of the Book \t\tAuthor\n )
    file.close()
    while IsBorrowed==False:
        print( Enter an number from below to borrow a book: )
        for i in range(len(Lists.Books)):
            print( To borrow  Lists.Books[i] Enter the number i)
        try:
            num=int(input( Enter the number: ))
        try:
```

```

# writing the borrowed book to txt file

if(int(Lists.Quantity[num])>0):

    print( The book is available )

    file=open(record a)

    file.write( \t\t 1. \t\t Lists.Books[num] \t\t Lists.Author[num] \n )

    file.close()

    Lists.Quantity[num]=int(Lists.Quantity[num])-1

    file=open( Books.txt w )

    for i in range(4):

        file.write(Lists.Books[i] Lists.Author[i] str(Lists.Quantity[i]) $

Lists.Price[i] \n )

    file.close()

borrowmultiple=True

counter=1

while borrowmultiple==True:

    # processing multiple borrowingg

    Ask=input( Do you wish to borrow another book? Type y for yes and n

for no ).lower()

    if(Ask== y ):

        counter=counter 1

        print( Enter the books corresponding number to borrow ti: )

        for i in range(len(Lists.Books)):

            print( If your book choice is Lists.Books[i] Enter the number i)

        try:

            check=int(input( Please Enter the number: ))

        try:

            if(int(Lists.Quantity[check])>0):

                print( This book is available )

                file=open(record a)

                file.write( \t\t str(counter) . \t\t Lists.Books[check] \t\t

Lists.Author[check] \n )

```

```
file.close()

Lists.Quantity[check]=int(Lists.Quantity[check])-1

file=open( Books.txt w )

for i in range(4):

    file.write(Lists.Books[i]    Lists.Author[i]
str(Lists.Quantity[i])    $ Lists.Price[i] \n )

    file.close()

else:

    borrowmultiple=False

    break

# tracing and warning for invalid data input and errors.

except IndexError:

    print( Choose the books according to their number )

except ValueError:

    print( Choose as suggested )

else:

    print( Mind that returning date is 10 days from the date of borrow
and fine will be levied for late return of books. Thank you for borrowing books from
us.\n\n )

    borrowmultiple=False

    IsBorrowed=True

except IndexError:

    print( Choose the books according to their number )

except ValueError:

    print( Please Choose within the given options only )
```

ReturnBooks.py

START

import Date

import Lists

DEFINE borrow

IsBorrowed=False

#Using loops

while True

input values

FirstName=input Please Enter the first name of the borrower

LastName=input Please Enter the last name of the borrower

print \n

break

creating a txt file on borrower

record= Borrowed by FirstName .txt

file=open record w

file.write \t\t\tDetails of the Borrower and the book borrowed\n

file.write Borrowed by FirstName LastName \n

file.write Date and Time of borrowing Date.datetime \n\n

file.write \t\tS.N. \t\tName of the Book \t\tAuthor\n

file.close

while IsBorrowed==False

DISPLAY Enter an number from below to borrow a book

for i in range len Lists.Books

DISPLAY To borrow Lists.Books[i] Enter the number i

try

num=int input Enter the number

try

```

# writing the borrowed book to txt file
if int Lists.Quantity[num] >0 then
    DISPLAY The book is available
    file=open record a
    file.write \t\ t 1. \t\ t Lists.Books[num] \t\ t Lists.Author[num] \n
    file.close
    Lists.Quantity[num]=int Lists.Quantity[num] -1
    file=open Books.txt w
    for i in range 4
        file.write Lists.Books[i]   Lists.Author[i]   str Lists.Quantity[i]   $ Lists.Price[i] \n
        file.close

borrowmultiple=True
counter=1
while borrowmultiple==True
    # processing multiple borrowingg

Ask=input Do you wish to borrow another book? Type y for yes and n for no .lower
    if Ask== y then
        counter=counter 1
        DISPLAY Enter the books corresponding number to borrow ti
        for i in range len Lists.Books
DISPLAY If your book choice is   Lists.Books[i] Enter the number i
try
    check=int input Please Enter the number
try
    if int Lists.Quantity[check] >0 then
        DISPLAY This book is available
        file=open record a
        file.write \t\ t str counter . \t\ t Lists.Books[check] \t\ t Lists.Author[check] \n
        file.close

```

```

Lists.Quantity[check]=int Lists.Quantity[check] -1
file=open Books.txt w
for i in range 4
file.write Lists.Books[i] Lists.Author[i] str Lists.Quantity[i] $ Lists.Price[i] \n
file.close
else
borrowmultiple=False
break
End IF
# tracing and warning for invalid data input and errors.
except IndexError
DISPLAY Choose the books according to their number
except ValueError
DISPLAY Choose as suggested
else
DISPLAY Mind that returning date is 10 days from the date of borrow and fine will
be levied for late return of books. Thank you for borrowing books from us.\n\n
borrowmultiple=False
IsBorrowed=True
END IF
except IndexError
DISPLAY Choose the books according to their number
except ValueError
DISPLAY Please Choose within the given options only

END IF

STOP

```

Lists.py

START

Books = []

Author = []

Quantity = []

Price = []

DEFINE lists

```
file=open Books.txt r
read=file.readlines
read=[a.strip \n  for a in read]
for i in range len read
    a=0
    for j in read[i].split
        #DISPLAYing j
        if a==0
            Books.append j
            #DISPLAYing Books
        else if a==1
            Author.append j
        else if a==2
            Quantity.append j
        else if a==3
            Price.append j.strip $
    a=a 1
```

STOP

Date.py

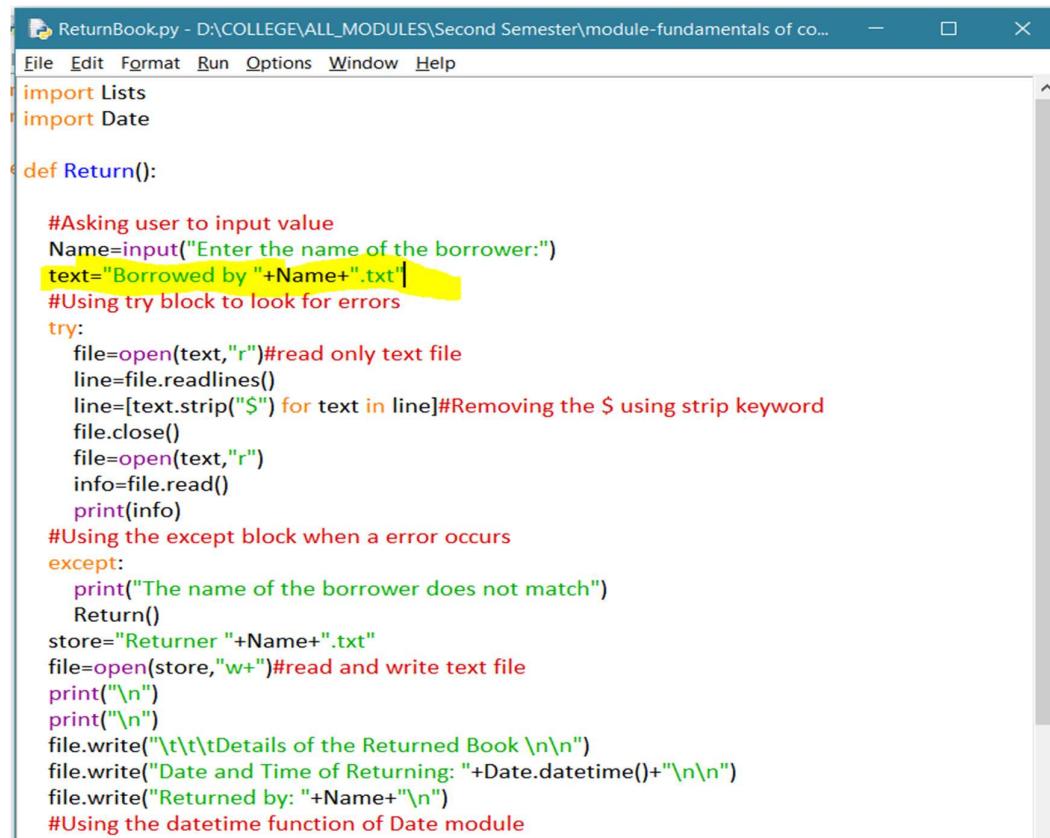
```
def datetime
    import datetime
    dandt=datetime.datetime.now
    #DISPLAY Date and Time dandt
    return str dandt
```

2.4 DATA STRUCTURES

Python provides variety of data types that can make handling groups of data much simpler. The data types in Python are strings, tuples, lists and dictionaries. These data types collectively are called data structures. We can introduce these data types briefly like strings are pieces of text, tuples and lists are ordered groups of individual data items. Dictionaries are groups of key-value pairs. Strings, tuples and lists have same process of accessing data in sequence. The data types are also classified accordingly to the availability of modification. Strings and tuples cannot be modified so they are called immutable whereas lists are mutable meaning we can modify items from a list. (Stacey, 2009)

Strings:

Strings are simply pieces of text. Each character of the string can be accessed by their index. The index is zero-based and it is specified as a whole number in square brackets following the variable name like sample[i] will retrieve the item at position of sequence sample. Strings can be used anytime when a text is required in a program.



```

ReturnBook.py - D:\COLLEGE\ALL MODULES\Second Semester\module-fundamentals of co...
File Edit Format Run Options Window Help
import Lists
import Date

def Return():

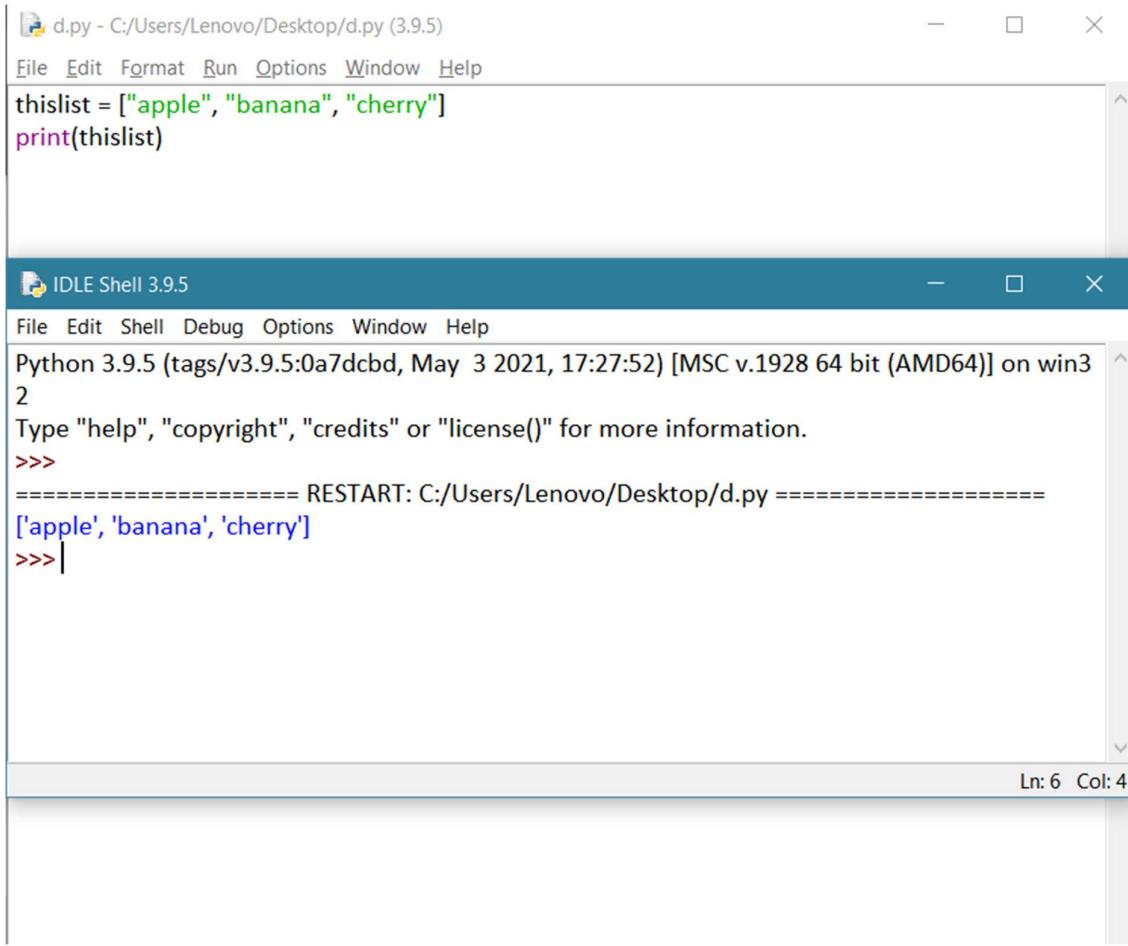
    #Asking user to input value
    Name=input("Enter the name of the borrower:")
    text="Borrowed by "+Name+".txt"
    #Using try block to look for errors
    try:
        file=open(text,"r")#read only text file
        line=file.readlines()
        line=[text.strip("$") for text in line]#Removing the $ using strip keyword
        file.close()
        file=open(text,"r")
        info=file.read()
        print(info)
    #Using the except block when a error occurs
    except:
        print("The name of the borrower does not match")
        Return()
    store="Returner "+Name+".txt"
    file=open(store,"w+")#read and write text file
    print("\n")
    print("\n")
    file.write("\t\t\tDetails of the Returned Book \n\n")
    file.write("Date and Time of Returning: "+Date.datetime()+"\n\n")
    file.write("Returned by: "+Name+"\n")
    #Using the datetime function of Date module

```

Figure 1: String Data in Python

List:

A list is an ordered, comma-separated list of items not necessarily of same type enclosed in square brackets. A list can contain another list to form a multidimensional list. List can be sliced, concatenated, indexed and update individual item in the list. List can be used efficiently while working with stacks, queues, sorting items, matrixes and working with large amount of data. (Stacey, 2009)



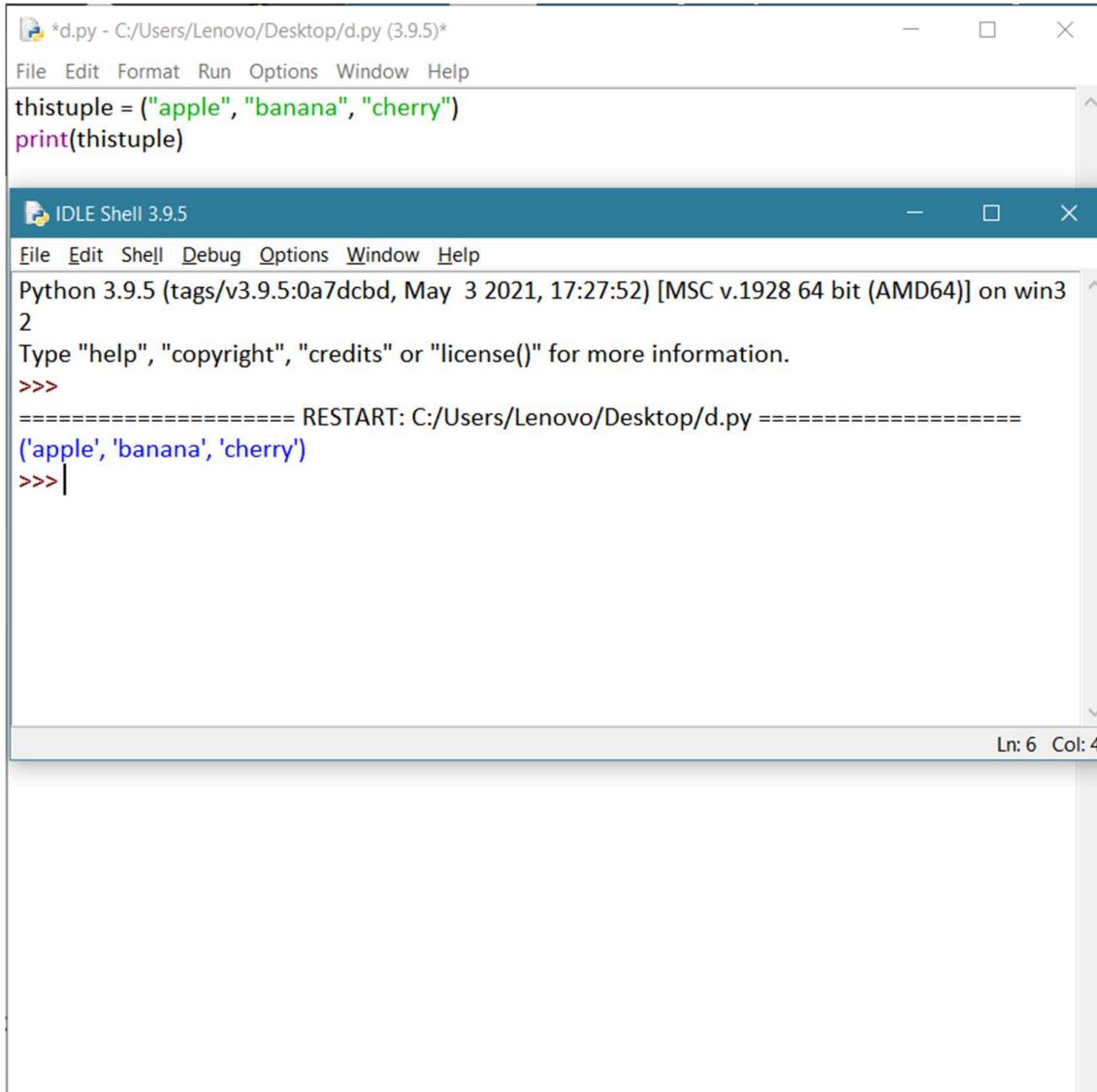
The image shows two windows from the Python IDLE environment. The top window is a script editor titled 'd.py - C:/Users/Lenovo/Desktop/d.py (3.9.5)'. It contains the following code:```pythonthislist = ["apple", "banana", "cherry"]print(thislist)```

```
The bottom window is the 'IDLE Shell 3.9.5' window. It displays the output of running the script. The output starts with the Python version and build information, followed by the command-line prompt 'Type "help", "copyright", "credits" or "license()" for more information.', then 'RESTART: C:/Users/Lenovo/Desktop/d.py', and finally the printed list: '['apple', 'banana', 'cherry']'. The status bar at the bottom right of the shell window shows 'Ln: 6 Col: 4'.
```

Figure 2 Creating lists

Tuples:

A tuple is an immutable ordered group of items. Tuples are sealed packed once created meaning that they cannot be modified once created but can be overwritten. Tuples can be used efficiently whenever a fixed value are to be used repeatedly in a program.



The screenshot shows the Python IDLE environment. At the top, there is a code editor window titled "d.py - C:/Users/Lenovo/Desktop/d.py (3.9.5)*". The code inside the editor is:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

Below the code editor is the IDLE Shell window, titled "IDLE Shell 3.9.5". The shell output is:

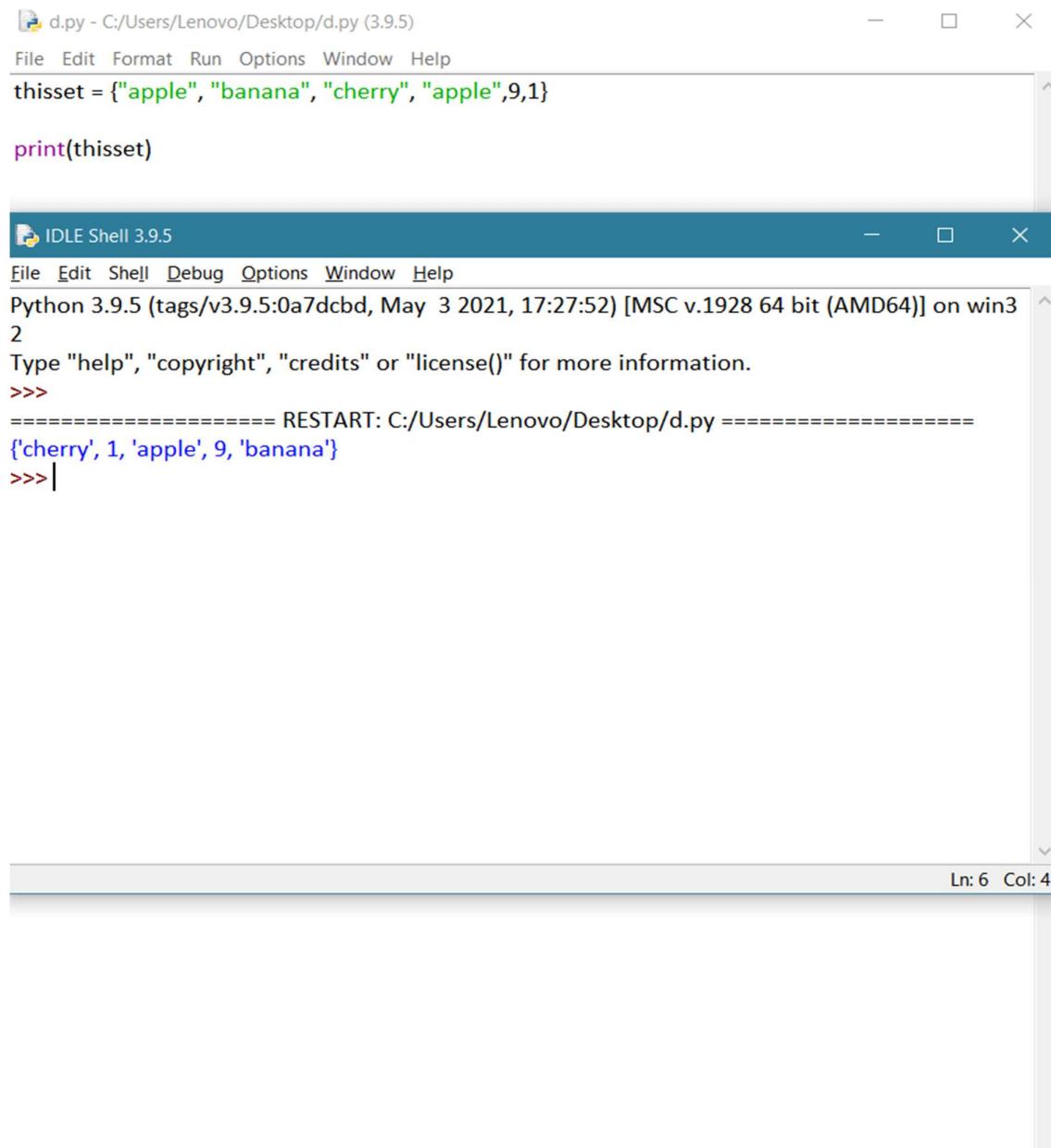
```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Lenovo/Desktop/d.py =====
('apple', 'banana', 'cherry')
>>> |
```

The bottom status bar indicates "Ln: 6 Col: 4".

Figure 3 Tuple creation

Sets:

Sets are used to test for membership, to get single item from duplicate entries and to find out what is common or different between two sets of items. We can perform all mathematical set operations in Python using sets.



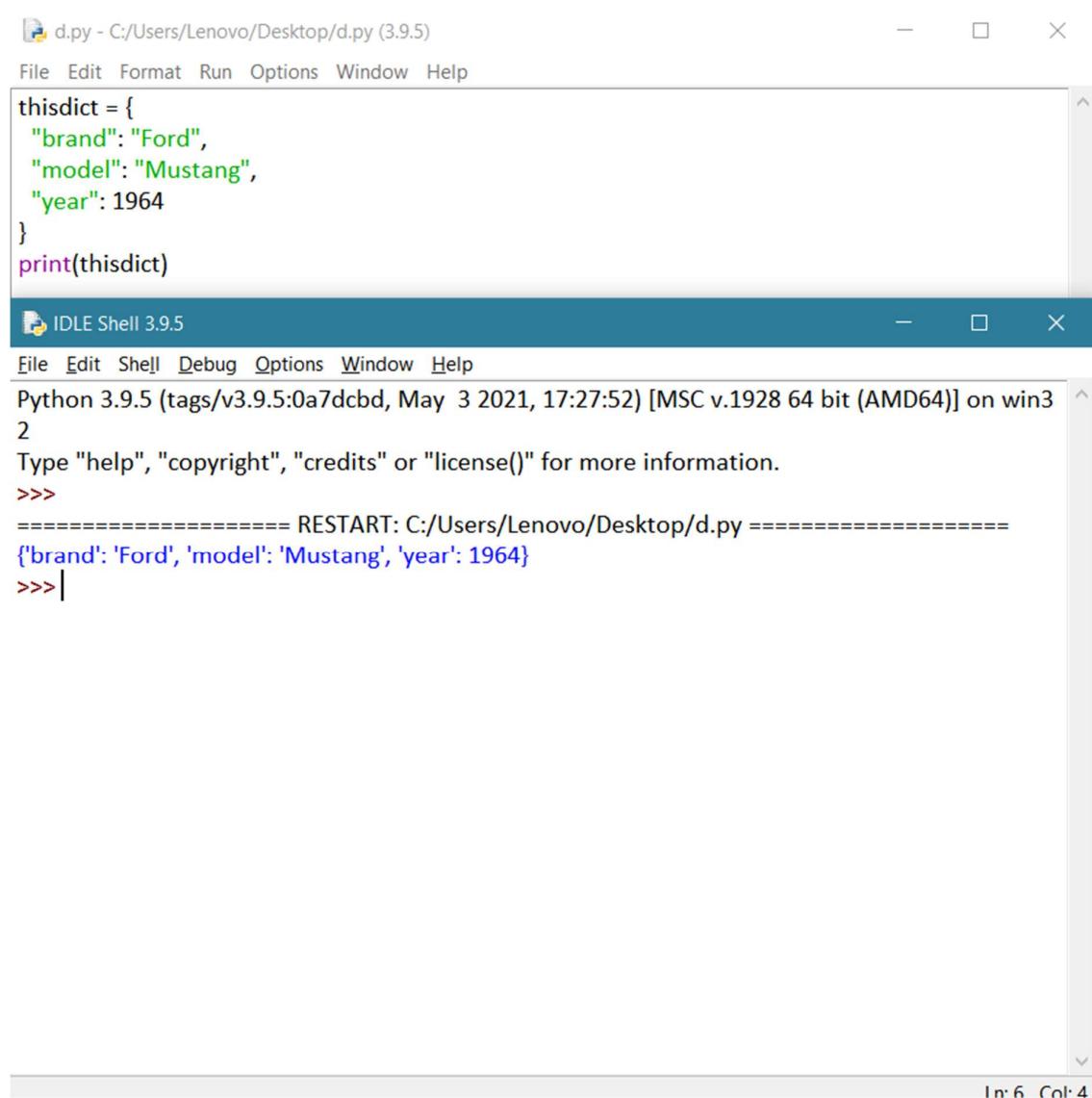
The screenshot shows a Windows desktop environment. In the foreground, there is an 'IDLE Shell 3.9.5' window with a dark blue header bar. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. Below the menu, it says 'Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32'. It also displays the message 'Type "help", "copyright", "credits" or "license()" for more information.' A command prompt shows 'RESTART: C:/Users/Lenovo/Desktop/d.py' followed by the output '{'cherry', 1, 'apple', 9, 'banana'}'. The cursor is positioned at the end of the command line. In the background, there is a code editor window titled 'd.py - C:/Users/Lenovo/Desktop/d.py (3.9.5)'. The code in the editor is:

```
thisset = {"apple", "banana", "cherry", "apple", 9, 1}  
print(thisset)
```

Figure 4 Creating sets

Dictionary:

Dictionary is an unsorted collection of items. Dictionaries are created by specifying key-value pairs separated by colons inside curly braces. The values of dictionaries can be accessed using keys. The key must be of an immutable type like string, number or tuple. The value can be anything. Dictionary is a mutable data type, thus we can add, remove and modify key-value pairs. Dictionaries can be used efficiently to add values like details of a person/item in keys like name of the person/item. (Stacey, 2009)



The screenshot shows two windows from the Python IDLE environment. The top window is titled 'd.py - C:/Users/Lenovo/Desktop/d.py (3.9.5)' and contains the following Python code:

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
print(thisdict)
```

The bottom window is titled 'IDLE Shell 3.9.5' and contains the following output:

```
File Edit Shell Debug Options Window Help  
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win3  
2  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/Lenovo/Desktop/d.py ======  
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}  
>>>|
```

The status bar at the bottom right of the shell window indicates 'In: 6 Col: 4'.

Figure 5 Dictionary creation

Integer:

Integers data type are whole numbers containing number 0 and other numbers in both positive and negative. Such as ...-2, -1, 0, 1, 2, 3 and so on.

Boolean

Boolean data types are those types of data which hold only either True or False value. Example: access = true

It is used in different parts program to check user input password, ask whether the user wants to continue or not.

Float

Float data type is that type of type which can hold numbers with decimal point value in them. Example: 12.5, 12.32, 0.2.

In the program float is used to hold the data of price of the books.

3.Program

All of the files are necessary for the program to run smoothly. Firstly, all the required modules are imported in main.py that contains the main frame of the program. It imports List.py for creating storing and importing lists, BorrowBook.py that contains the code for borrowing a book, ReturnBook.py that contains code for returning a borrowed book, Date.py that returns the dates and time used to borrow and return a book and finally the Books.txt file that is the stock of books that this program will be working with. A message is displayed welcoming the user along with 4 options i.e. to display available books, to borrow a book, to return a book, to terminate a program whole. Then the program requests user to choose from these 4 options and displays an error message if the user inputs invalid data. If the user chooses Display option, the books in the Books.txt file is read and displayed to the user and provided with the same 4 options as earlier. If the user inputs option to borrow he is asked his full name and displayed the books with option to choose between them which he wants to borrow. Once valid data is entered and book is selected a new .txt file is created with the user's name stating the books that is borrowed. The user again is asked if he wants to borrow more books If yes he is again given options of books to choose from and this process loops unless the user decides to not borrow anymore. If the user chooses to return books. He is asked for his name and if it matched with the borrowed txt file from previous users A return note is generated with a Bill he'd have to pay for borrowing the book. He is given his bill and charged with fine if he testifies returning the book late. If the name doesn't match an error message is displayed that alerts the user regarding their incorrect input. If the user chooses the 4th option that is to terminate program, A thank you message is displayed and the program terminates. The user is provided with the same 4 options from the beginning until the user decides to terminate the program.



Figure 6: BOOKs stock

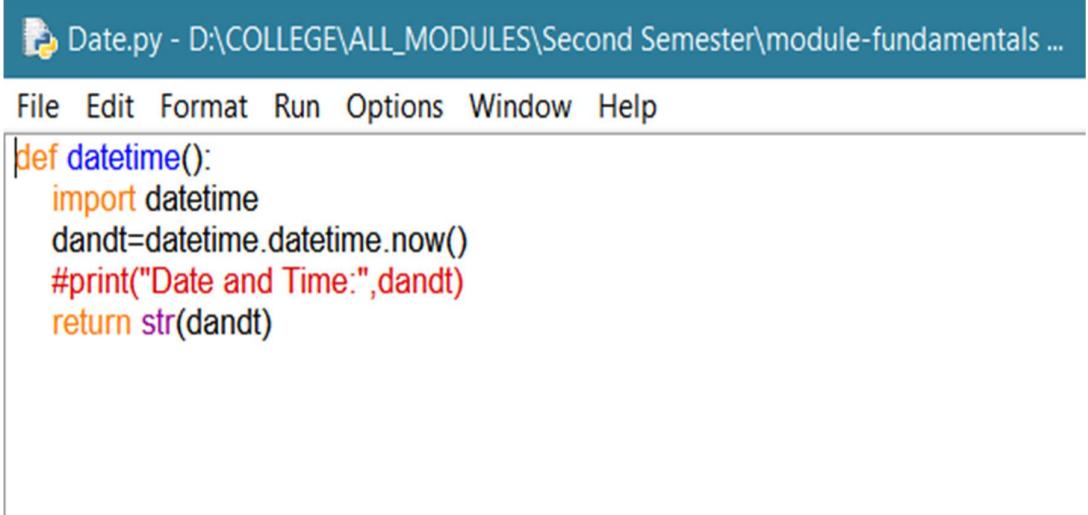
```

Main.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy>Main.py (3.9.5)
File Edit Format Run Options Window Help
import BorrowBook
import ReturnBook
import Lists
import Date

def main():
    looping=True
    while(looping==True):
        #display choice
        print("\n")
        print("-----Welcome to our Library-----")
        print(" Please Enter +" + "1" + " to Display the books")
        print(" Please Enter +" + "2" + " to Borrow our books")
        print(" Please Enter +" + "3" + " to Return a borrowed book")
        print(" To Exit, Please Enter +" + "4")
        print("-----")
        # try block to look for errors in
        try:
            ##Asking input
            id=int(input("Provide your input here:- "))
            #Using the if and else to process further.
            if(id==1):
                #opening a text file and printing its content
                print("\n")
                print("books here are displayed in NAME , AUTHOR , QUANTITY , BORROWING PRICE format")
                file=open("Books.txt", "r")
                read=file.read()
                print(read)
                file.close()
            elif(id==2):
                #Calling Lists and BorrowBook module
                Lists.lists()
                BorrowBook.borrow()
            elif(id==3):
                #Calling Lists and ReturnBook module
                Lists.lists()
                ReturnBook.Return()
            elif(id==4):
                #Displaying
                print("Thank you for choosing us!")
                break
            else:
                # else block
                print("Please enter a number from the given choices.")
        except ValueError:
            #except block if error occurs
            print("Invalid! only numeric data that is within the given choices are accepted.")
    main()

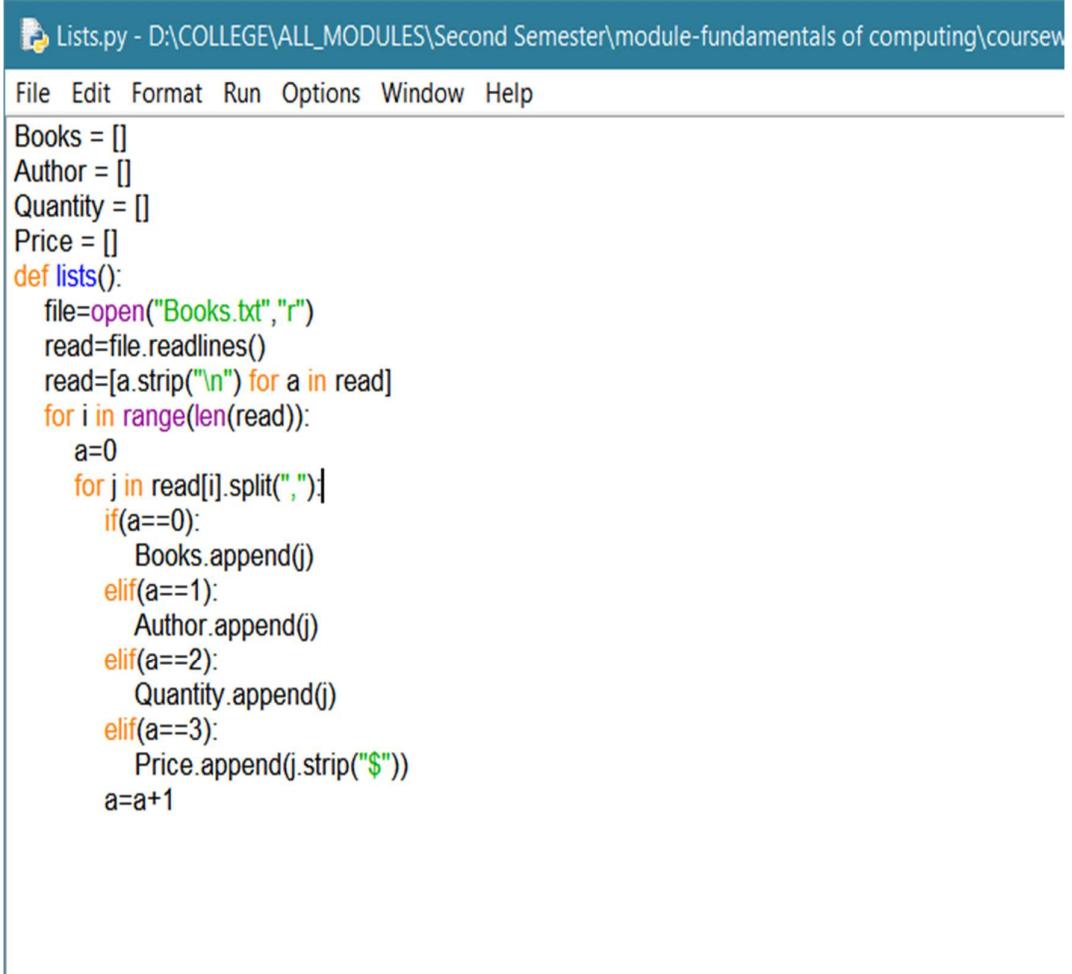
```

Figure 7: MAIN.py



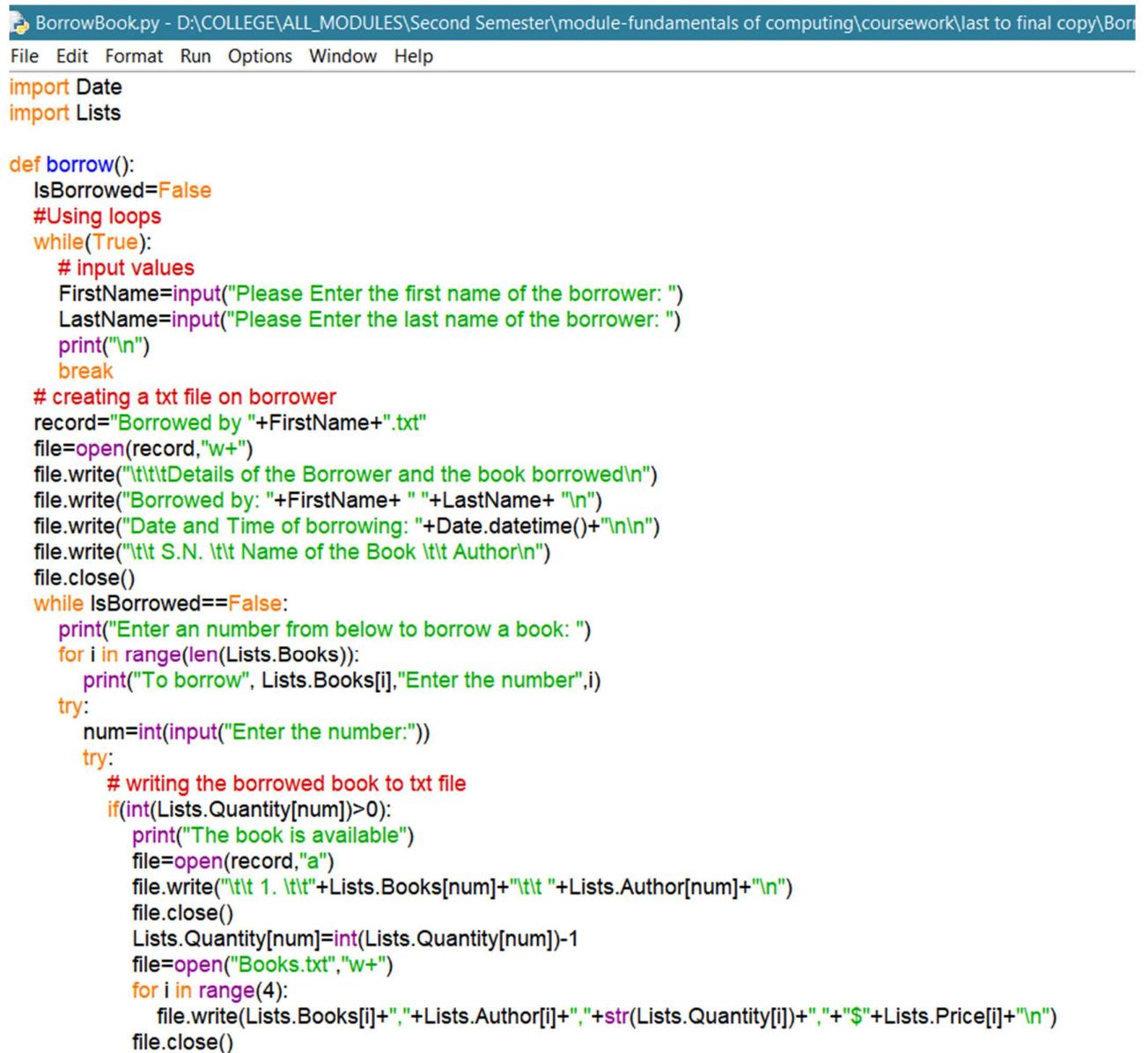
```
def datetime():
    import datetime
    dandt=datetime.datetime.now()
    #print("Date and Time:",dandt)
    return str(dandt)
```

Figure 8: Date.py



```
Books = []
Author = []
Quantity = []
Price = []
def lists():
    file=open("Books.txt","r")
    read=file.readlines()
    read=[a.strip("\n") for a in read]
    for i in range(len(read)):
        a=0
        for j in read[i].split(","):
            if(a==0):
                Books.append(j)
            elif(a==1):
                Author.append(j)
            elif(a==2):
                Quantity.append(j)
            elif(a==3):
                Price.append(j.strip("$"))
            a=a+1
```

Figure 9: Lists.py



```

BorrowBook.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\Bor
File Edit Format Run Options Window Help
import Date
import Lists

def borrow():
    IsBorrowed=False
    #Using loops
    while(True):
        # input values
        FirstName=input("Please Enter the first name of the borrower: ")
        LastName=input("Please Enter the last name of the borrower: ")
        print("\n")
        break
    # creating a txt file on borrower
    record="Borrowed by "+FirstName+".txt"
    file=open(record,"w+")
    file.write("\t\t\tDetails of the Borrower and the book borrowed\n")
    file.write("Borrowed by: "+FirstName+ " "+LastName+ "\n")
    file.write("Date and Time of borrowing: "+Date.datetime()+"\n\n")
    file.write("\t\t S.N. \t\t Name of the Book \t\t Author\n")
    file.close()
    while IsBorrowed==False:
        print("Enter an number from below to borrow a book: ")
        for i in range(len(Lists.Books)):
            print("To borrow", Lists.Books[i],"Enter the number",i)
        try:
            num=int(input("Enter the number:"))
            try:
                # writing the borrowed book to txt file
                if(int(Lists.Quantity[num])>0):
                    print("The book is available")
                    file=open(record,"a")
                    file.write("\t\t 1. \t\t"+Lists.Books[num]+\t\t+Lists.Author[num]+\n")
                    file.close()
                    Lists.Quantity[num]=int(Lists.Quantity[num])-1
                    file=open("Books.txt","w+")
                    for i in range(4):
                        file.write(Lists.Books[i]+","+Lists.Author[i]+","+str(Lists.Quantity[i])+","+"$"+Lists.Price[i]+\n")
                    file.close()
            except:
                print("Book not available")
        except:
            print("Invalid input")

```

Figure 10 : Borrow.py

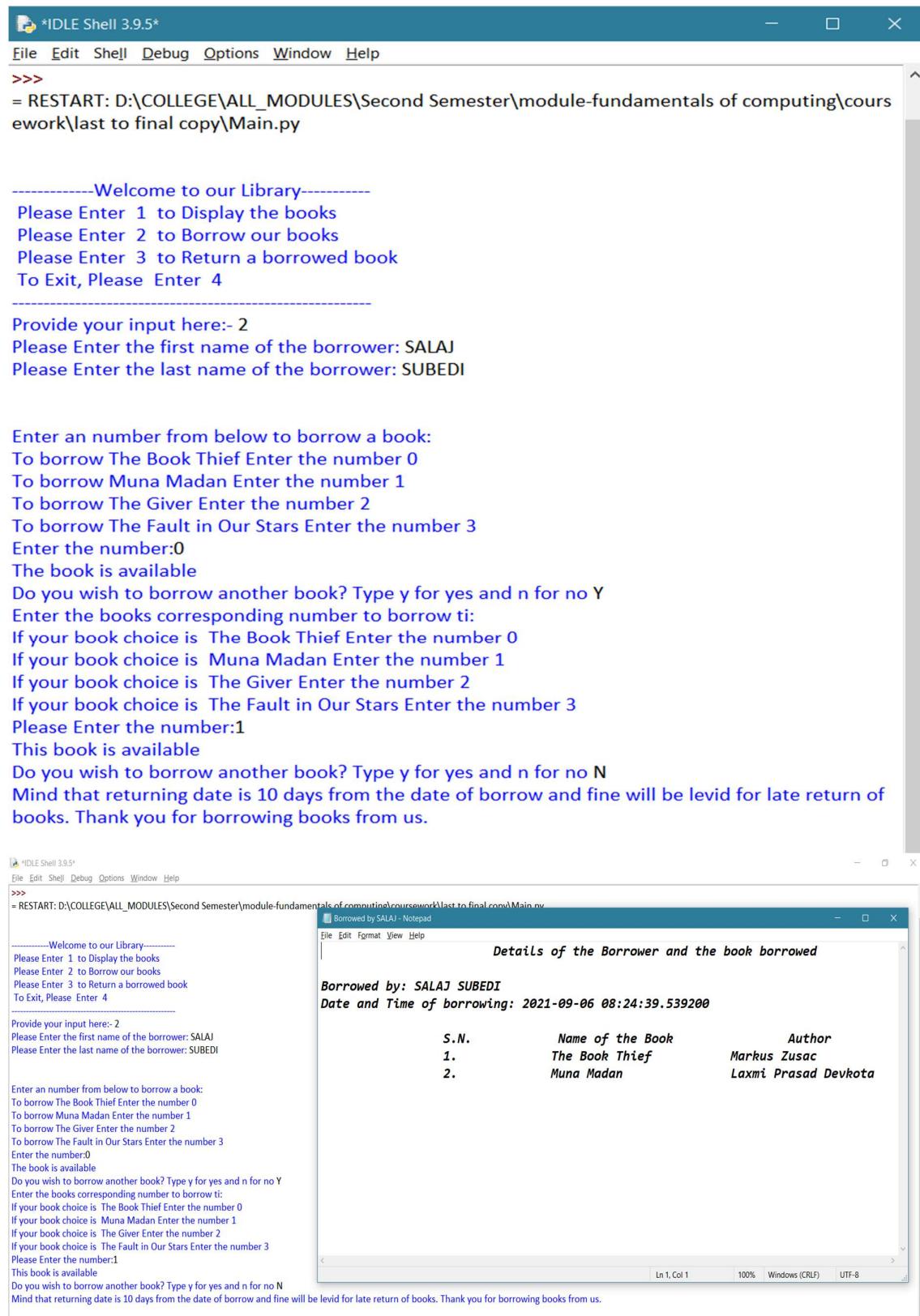
```

BorrowBook.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\BorrowBook.py (3.9.5)
File Edit Format Run Options Window Help
file=open("Books.txt","w+")
for i in range(4):
    file.write(Lists.Books[i]+","+Lists.Author[i]+","+str(Lists.Quantity[i])+","+"$"+Lists.Price[i]+\n")
file.close()

borrowmultiple=True
counter=1
while borrowmultiple==True:
    # processing multiple borrowingg
    Ask=input("Do you wish to borrow another book? Type y for yes and n for no ").lower()
    if(Ask=="y"):
        counter=counter+1
        print("Enter the books corresponding number to borrow ti: ")
        for i in range(len(Lists.Books)):
            print("If your book choice is ", Lists.Books[i],"Enter the number",i)
        try:
            check=int(input("Please Enter the number:"))
            try:
                if(int(Lists.Quantity[check])>0):
                    print("This book is available")
                    file=open(record,"a")
                    file.write("\t"+str(counter)+"."+ "\t"+Lists.Books[check]+\t+Lists.Author[check]+\n")
                    file.close()
                    Lists.Quantity[check]=int(Lists.Quantity[check])-1
                    file=open("Books.txt","w+")
                    for i in range(4):
                        file.write(Lists.Books[i]+","+Lists.Author[i]+","+str(Lists.Quantity[i])+","+"$"+Lists.Price[i]+\n")
                    file.close()
            else:
                borrowmultiple=False
                break
            # tracing and warning for invalid data input and errors|
        except IndexError:
            print("Choose the books according to their number")
        except ValueError:
            print("Choose as suggested")
    else:
        print("Mind that returning date is 10 days from the date of borrow and fine will be levied for late return of books. Thank you for borrowing books from us.\n\n")
        borrowmultiple=False
        IsBorrowed=True
    except IndexError:
        print("Choose the books according to their number")
    except ValueError:
        print("Please Choose within the given options only")

```

Figure 11: contd. Borrow



The screenshot shows two windows. The top window is the IDLE Shell 3.9.5 interface with the following text:

```
>>>
= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\Main.py

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 2
Please Enter the first name of the borrower: SALAJ
Please Enter the last name of the borrower: SUBEDI

Enter an number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number:0
The book is available
Do you wish to borrow another book? Type y for yes and n for no Y
Enter the books corresponding number to borrow ti:
If your book choice is The Book Thief Enter the number 0
If your book choice is Muna Madan Enter the number 1
If your book choice is The Giver Enter the number 2
If your book choice is The Fault in Our Stars Enter the number 3
Please Enter the number:1
This book is available
Do you wish to borrow another book? Type y for yes and n for no N
Mind that returning date is 10 days from the date of borrow and fine will be levied for late return of
books. Thank you for borrowing books from us.
```

The bottom window is a Notepad titled "Borrowed by SALAJ - Notepad" containing the following text:

```
Details of the Borrower and the book borrowed

Borrowed by: SALAJ SUBEDI
Date and Time of borrowing: 2021-09-06 08:24:39.539200

S.N. Name of the Book Author
1. The Book Thief Markus Zusac
2. Muna Madan Laxmi Prasad Devkota
```

Figure 12: Borrow process in output along with generated .txt file

```

ReturnBook.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\ReturnBook
File Edit Format Run Options Window Help
import Lists
import Date

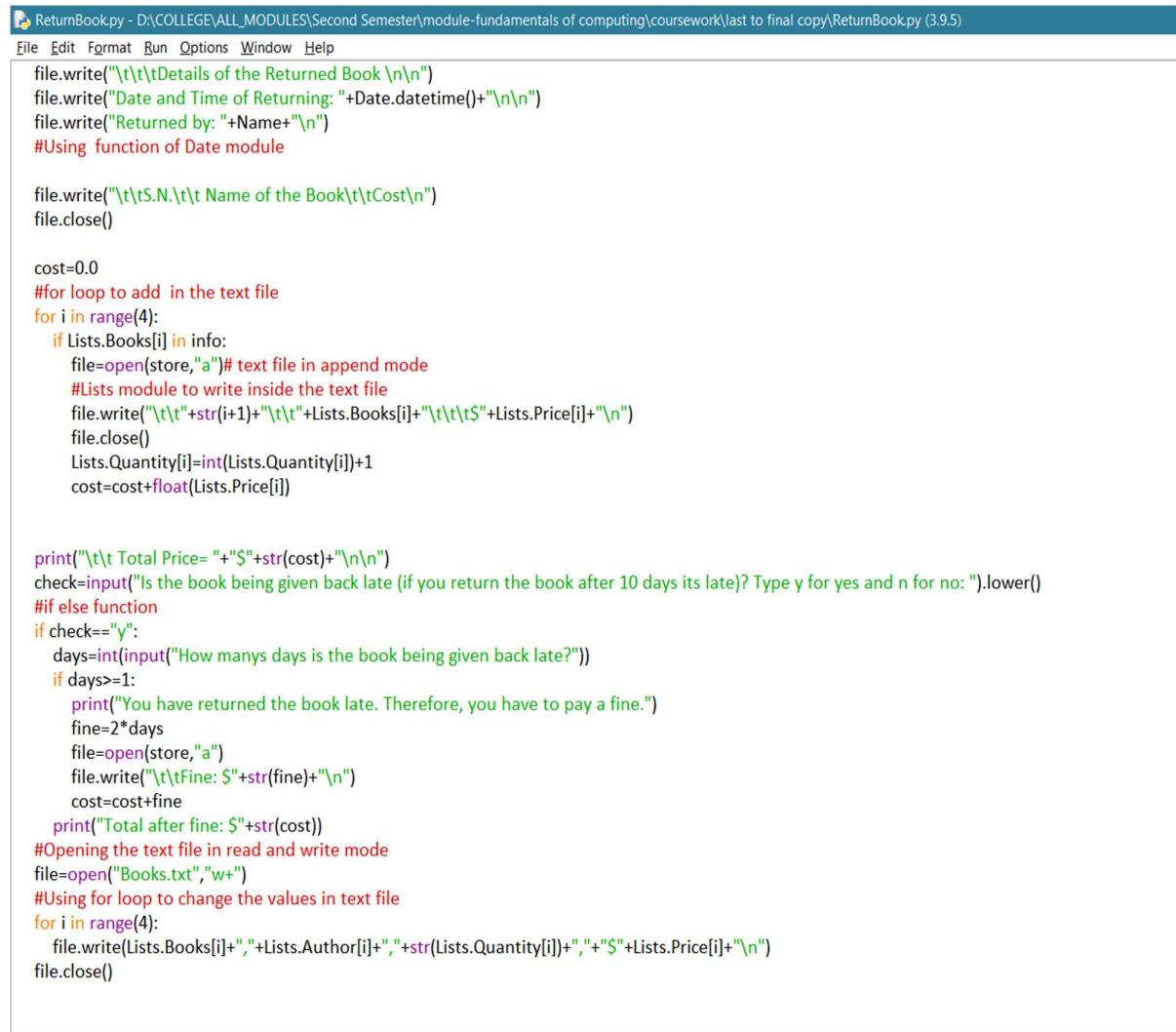
def Return():

    #input value
    Name=input("Enter the name of the borrower:")
    text="Borrowed by "+Name+".txt"
    # try block for errors
    try:
        file=open(text,"r")#read text file
        line=file.readlines()
        line=[text.strip("$") for text in line]#Removing the $ using strip keyword
        file.close()
        file=open(text,"r")
        info=file.read()
        print(info)
    #Using the except block for error
    except:
        print("The name of the borrower does not match")
        Return()
    store="Returner "+Name+".txt"
    file=open(store,"w+")#read and write text file
    print("\n")
    print("\n")
    file.write("\t\t\tDetails of the Returned Book \n\n")
    file.write("Date and Time of Returning: "+Date.datetime()+"\n\n")
    file.write("Returned by: "+Name+"\n")
    #Using function of Date module

    file.write("\t\tS.N.\t\t Name of the Book\t\tCost\n")
    file.close()

```

Figure 13: Return.py



```

ReturnBook.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\ReturnBook.py (3.9.5)
File Edit Format Run Options Window Help
file.write("\t\t\tDetails of the Returned Book \n\n")
file.write("Date and Time of Returning: "+Date.datetime()+"\n\n")
file.write("Returned by: "+Name+"\n")
#Using function of Date module

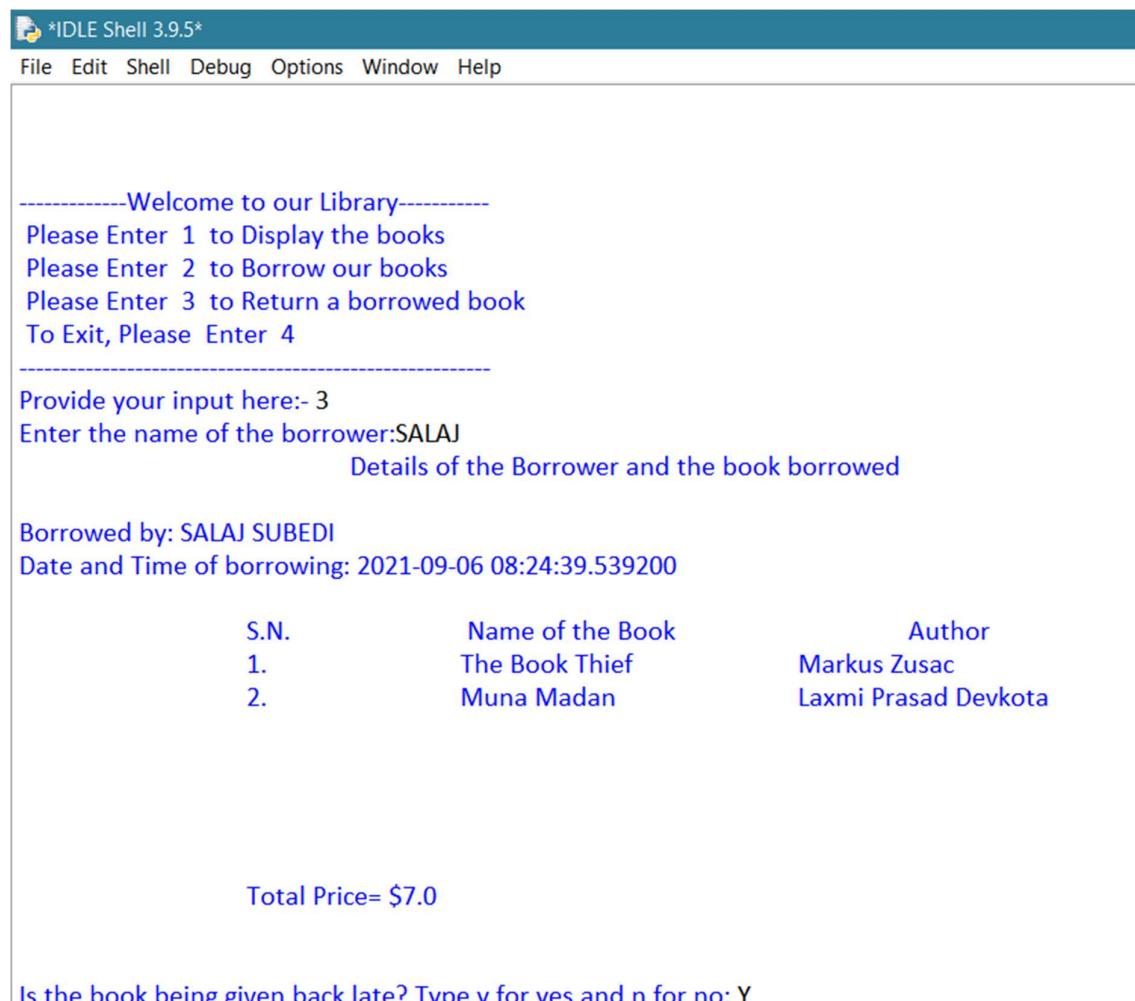
file.write("\t\tS.N.\t\t Name of the Book\t\tCost\n")
file.close()

cost=0.0
#for loop to add in the text file
for i in range(4):
    if Lists.Books[i] in info:
        file=open(store,"a")# text file in append mode
        #Lists module to write inside the text file
        file.write("\t\t"+str(i+1)+"\t\t"+Lists.Books[i]+"\t\t$"+Lists.Price[i]+"\n")
        file.close()
        Lists.Quantity[i]=int(Lists.Quantity[i])+1
        cost=cost+float(Lists.Price[i])

print("\t\t Total Price= $" +str(cost)+"\n\n")
check=input("Is the book being given back late (if you return the book after 10 days its late)? Type y for yes and n for no: ").lower()
#if else function
if check=="y":
    days=int(input("How many days is the book being given back late?"))
    if days>=1:
        print("You have returned the book late. Therefore, you have to pay a fine.")
        fine=2*days
        file=open(store,"a")
        file.write("\t\tFine: $" +str(fine)+"\n")
        cost=cost+fine
        print("Total after fine: $" +str(cost))
#Opening the text file in read and write mode
file=open("Books.txt","w+")
#Using for loop to change the values in text file
for i in range(4):
    file.write(Lists.Books[i]+","+Lists.Author[i]+","+str(Lists.Quantity[i])+","+ "$"+Lists.Price[i]+\n")
file.close()

```

Figure 14: contd. Return.py



The screenshot shows the Python IDLE Shell interface. The window title is '*IDLE Shell 3.9.5*'. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays a library management program:

```

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 3
Enter the name of the borrower:SALAJ
                                Details of the Borrower and the book borrowed

Borrowed by: SALAJ SUBEDI
Date and Time of borrowing: 2021-09-06 08:24:39.539200

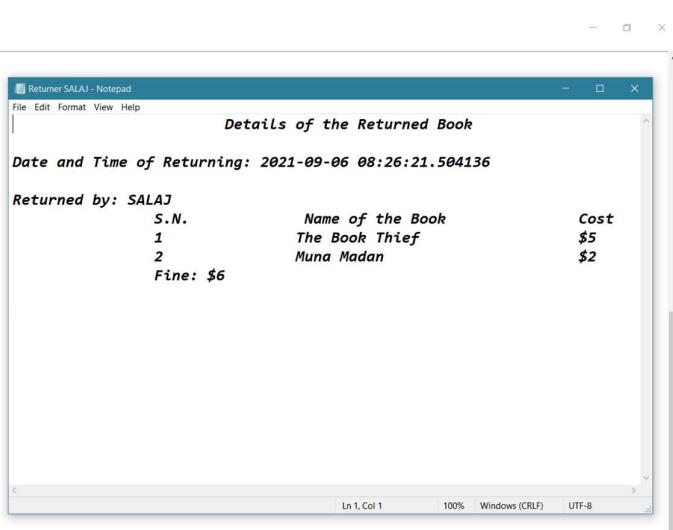
      S.N.          Name of the Book          Author
      1.            The Book Thief        Markus Zusac
      2.            Muna Madan          Laxmi Prasad Devkota

Total Price= $7.0

Is the book being given back late? Type y for yes and n for no: Y
How many days is the book being given back late?3
You have returned the book late. Therefore, you have to pay a fine.
Total after fine: $13.0

```

The program asks for input (3), identifies the borrower (SALAJ), and lists the borrowed books. It calculates a total price of \$7.0 and adds a fine of \$6 due to the late return, resulting in a total of \$13.0.



The Notepad window has the title 'Returner SALAJ - Notepad'. The menu bar includes File, Edit, Format, View, and Help. The window contains the following text:

```

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 3
Enter the name of the borrower:SALAJ
                                Details of the Borrower and the book borrowed

Borrowed by: SALAJ SUBEDI
Date and Time of borrowing: 2021-09-06 08:24:39.539200

      S.N.          Name of the Book          Author
      1.            The Book Thief        Markus Zusac
      2.            Muna Madan          Laxmi Prasad Devkota

Total Price= $7.0

Is the book being given back late? Type y for yes and n for no: Y
How many days is the book being given back late?3
You have returned the book late. Therefore, you have to pay a fine.
Total after fine: $13.0

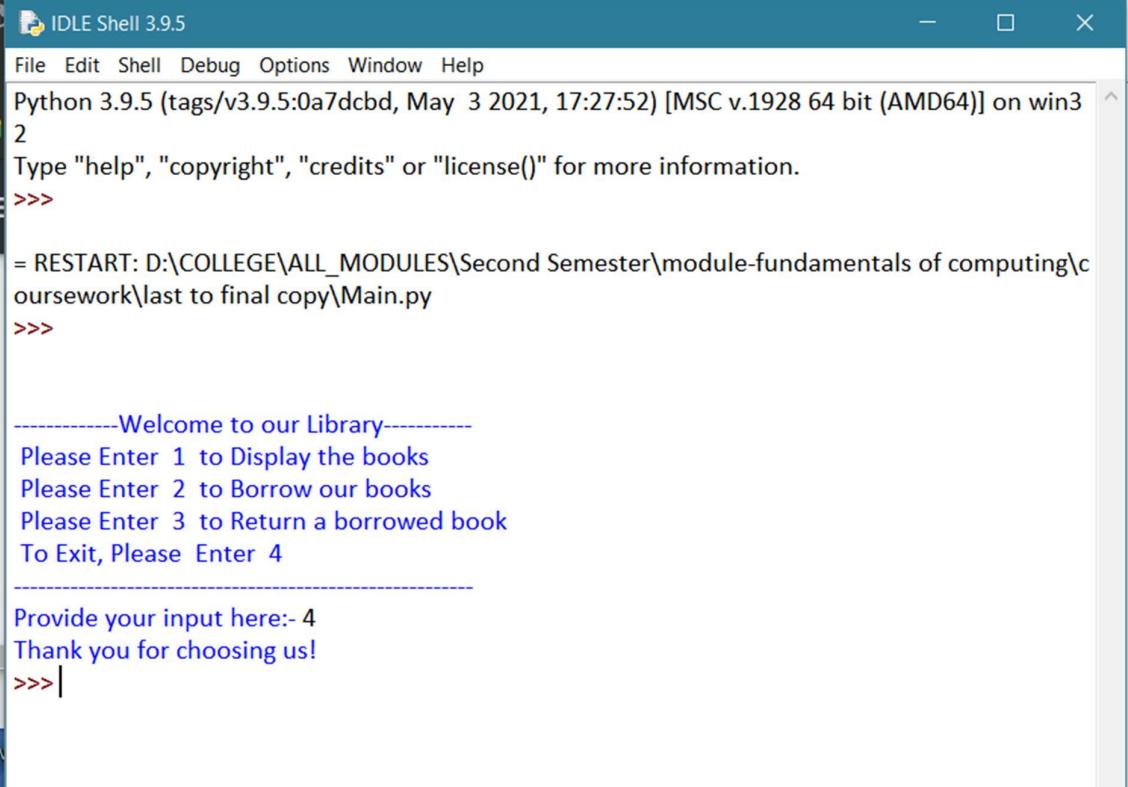
-----Welcome to our Library-----

```

The right side of the window shows a table titled 'Details of the Returned Book' with the following data:

Details of the Returned Book		
Date and Time of Returning: 2021-09-06 08:26:21.504136		
Returned by: SALAJ		
S.N.	Name of the Book	Cost
1	The Book Thief	\$5
2	Muna Madan	\$2
Fine: \$6		

Figure 15: Return process along with generated .txt file



IDLE Shell 3.9.5

File Edit Shell Debug Options Window Help

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\Main.py

>>>

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 4
Thank you for choosing us!
>>>|

Figure 16: Termination after choosing an option.

4. Testing

Test 1

Objectives	To check for try and accept formats .
Action	To enter invalid data in various inputs and check for validation notes from the program
Expected Result	Error messages are to be shown at appropriate valid inputs
Actual Result	Error messages are to be shown at appropriate valid inputs
Conclusion	Test successful

Table 1: TEST 1

```
-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 6
Please enter a number from the given choices.

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- f
Invalid! only numeric data that is within the given choices are accepted.

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

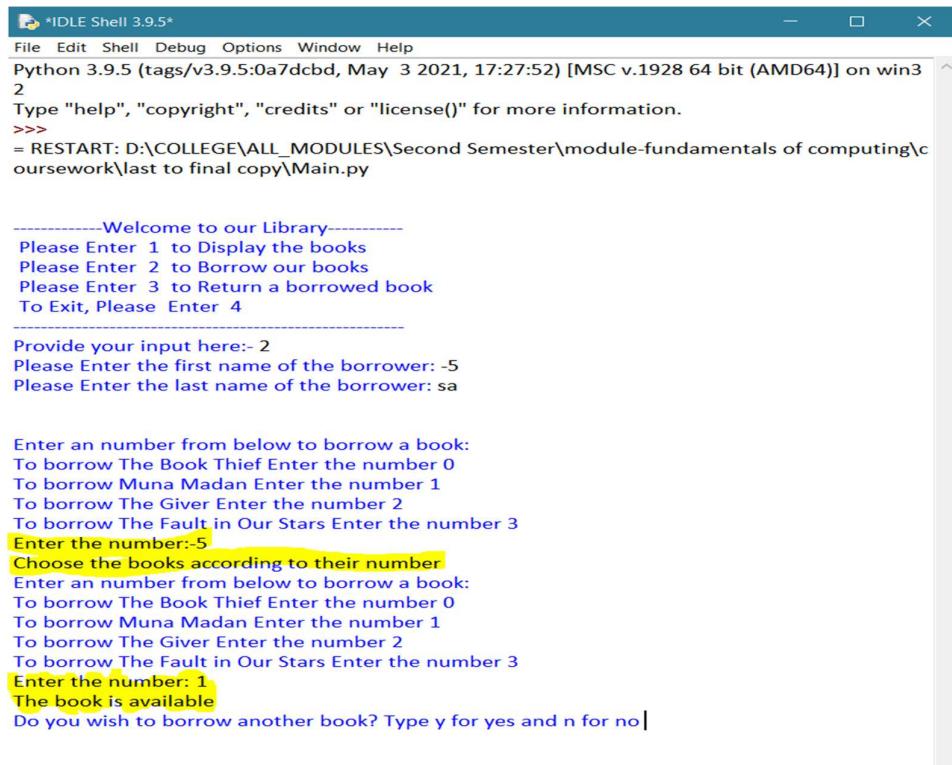
Provide your input here:- |
```

Figure 17: TEST 1; INVALID DATA ENTRY

Test 2

Objectives	TO provide negative and non-existed value as input and check validation for borrow and return option
Action	Invalid data is input in borrow and return option to check for borrow and return exception handling
Expected Result	Warning and error messages to be shown
Actual Result	Error messages were shown
Conclusion	Test successful

Table 2: TEST 2



```

IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\c
oursework\last to final copy\Main.py

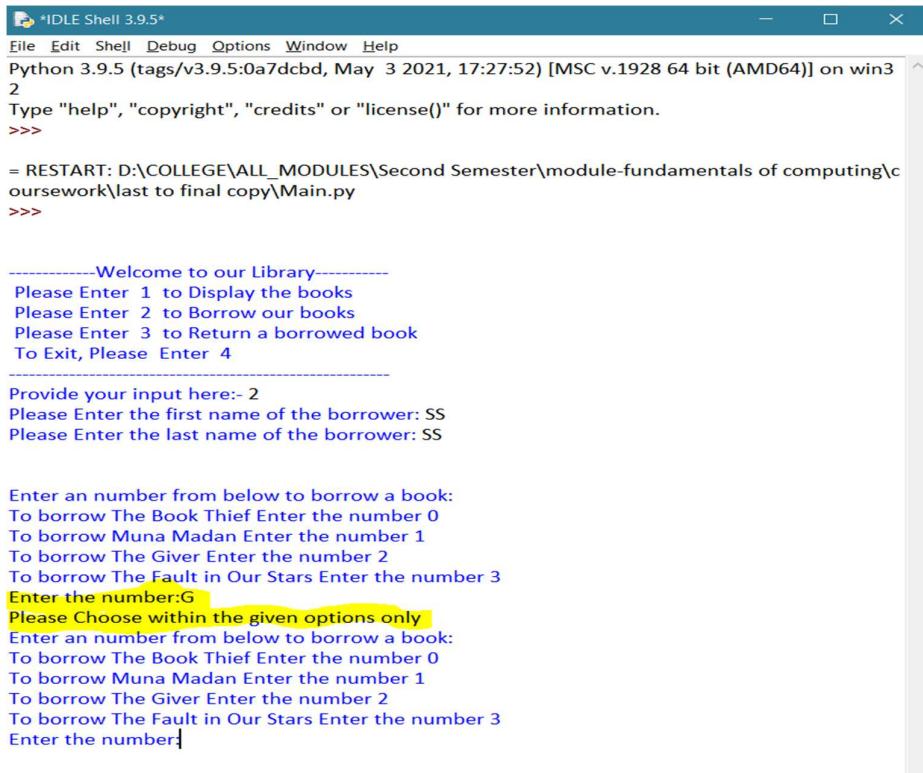
-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 2
Please Enter the first name of the borrower: -5
Please Enter the last name of the borrower: sa

Enter a number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number: -5
Choose the books according to their number
Enter a number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number: 1
The book is available
Do you wish to borrow another book? Type y for yes and n for no|

```

Figure 19: invalid data entry



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license()" for more information.
>>>

= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\Main.py
>>>

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4
-----
Provide your input here:- 2
Please Enter the first name of the borrower: SS
Please Enter the last name of the borrower: SS

Enter an number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number:G
Please Choose within the given options only
Enter an number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number:|
```

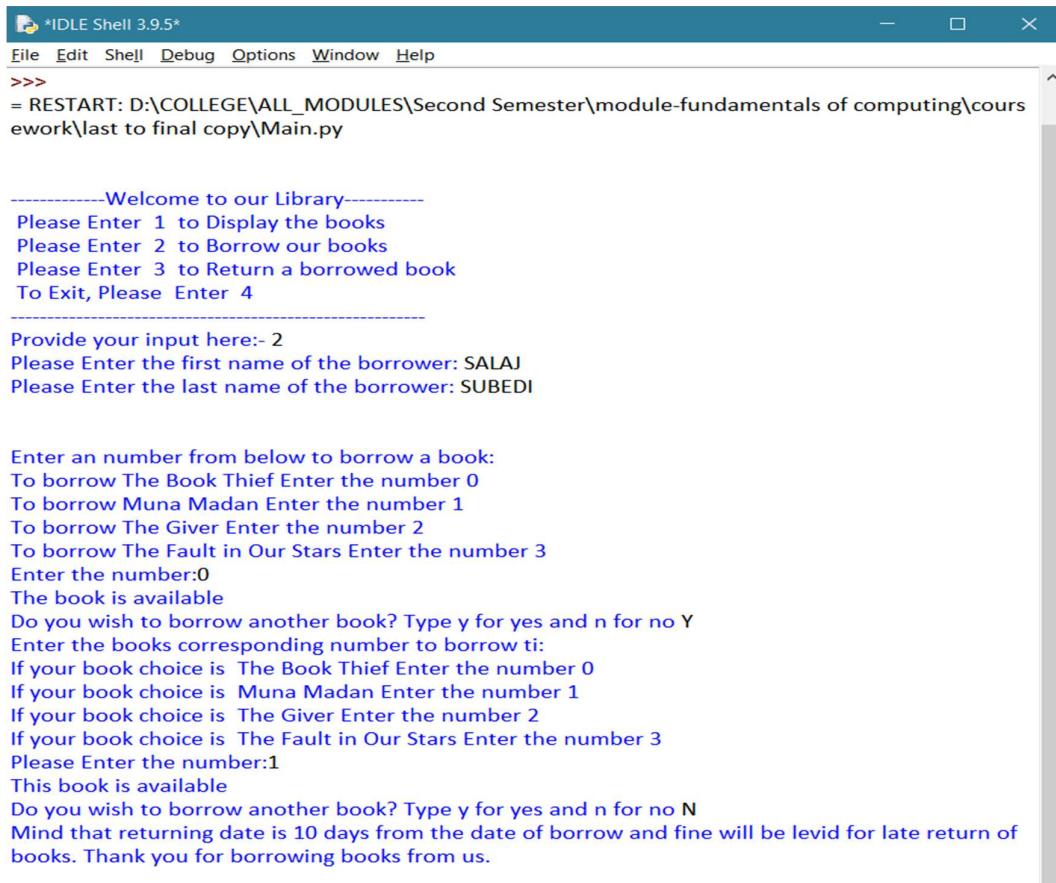
Figure 20: Non-existent data entry

Test 3

Table of Test 3

Objectives	Borrow process and file generation
Action	To enter data and Show complete borrow process along with output in shell and borrow note in txt file
Expected Result	Data is entered book is borrowed and a borrow note is generated
Actual Result	Data is entered book is borrowed and a borrow note is generated
Conclusion	Test successful

Table 3: TEST 3



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
>>>
= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\coursework\last to final copy\Main.py

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 2
Please Enter the first name of the borrower: SALAJ
Please Enter the last name of the borrower: SUBEDI

Enter a number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number:0
The book is available
Do you wish to borrow another book? Type y for yes and n for no Y
Enter the books corresponding number to borrow it:
If your book choice is The Book Thief Enter the number 0
If your book choice is Muna Madan Enter the number 1
If your book choice is The Giver Enter the number 2
If your book choice is The Fault in Our Stars Enter the number 3
Please Enter the number:1
This book is available
Do you wish to borrow another book? Type y for yes and n for no N
Mind that returning date is 10 days from the date of borrow and fine will be levied for late return of
books. Thank you for borrowing books from us.
```

Figure 21 BORROWING BOOKS

The screenshot shows two windows. On the left is a terminal window titled 'IDLE Shell 3.9.5' with the following text:

```
>>>
= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals.of.computing\coursework\last to final\conv\Main.py
-----Welcome to our Library-----
Please Enter 1 To Display the books
Please Enter 2 To Borrow our books
Please Enter 3 To Return a borrowed book
To Exit, Please Enter 4

Provide your input here: 2
Please Enter the first name of the borrower: SALAJ
Please Enter the last name of the borrower: SUBEDI

Enter an number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number:0
The book is available
Do you wish to borrow another book? Type y for yes and n for no Y
Enter the books corresponding number to borrow t:
If your book choice is The Book Thief Enter the number 0
If your book choice is Muna Madan Enter the number 1
If your book choice is The Giver Enter the number 2
If your book choice is The Fault in Our Stars Enter the number 3
Please Enter the number:1
This book is available
Do you wish to borrow another book? Type y for yes and n for no N
Mind that returning date is 10 days from the date of borrow and fine will be levied for late return of books. Thank you for borrowing books from us.
```

On the right is a Notepad window titled 'Borrowed by SALAJ - Notepad' with the following text:

Details of the Borrower and the book borrowed

Borrowed by: SALAJ SUBEDI

Date and Time of borrowing: 2021-09-06 08:24:39.539200

S.N.	Name of the Book	Author
1.	The Book Thief	Markus Zusac
2.	Muna Madan	Laxmi Prasad Devkota

Figure 22: BORROWING BOOKS WITH TEXT FILE

Test 4

Table of Test 4

Objectives	Return process and file generation
Action	Data is entered to return book appropriate message is shown with return fines if any
Expected Result	Data is entered book is returned and a return note is generated
Actual Result	Data is entered book is returned and a return note is generated
Conclusion	Test successful

Table 4 : Test 4

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 3
Enter the name of the borrower:SALAJ
Details of the Borrower and the book borrowed

Borrowed by: SALAJ SUBEDI
Date and Time of borrowing: 2021-09-06 08:24:39.539200

S.N. Name of the Book Author
1. The Book Thief Markus Zusac
2. Muna Madan Laxmi Prasad Devkota

Total Price= $7.0

Is the book being given back late? Type y for yes and n for no: Y
How many days is the book being given back late?3
You have returned the book late. Therefore, you have to pay a fine.
Total after fine: $13.0

```

Figure 23: RETURNING BOOKS

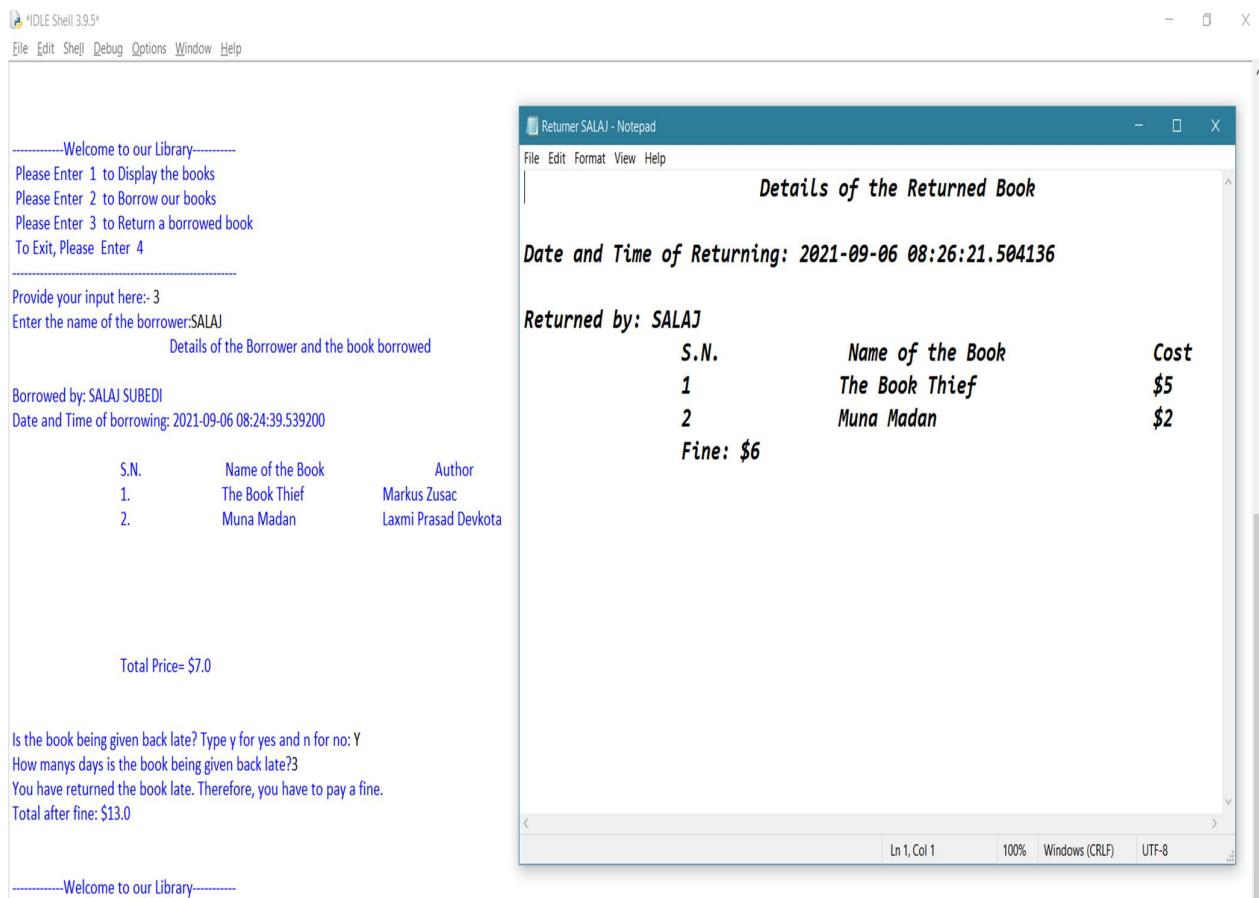


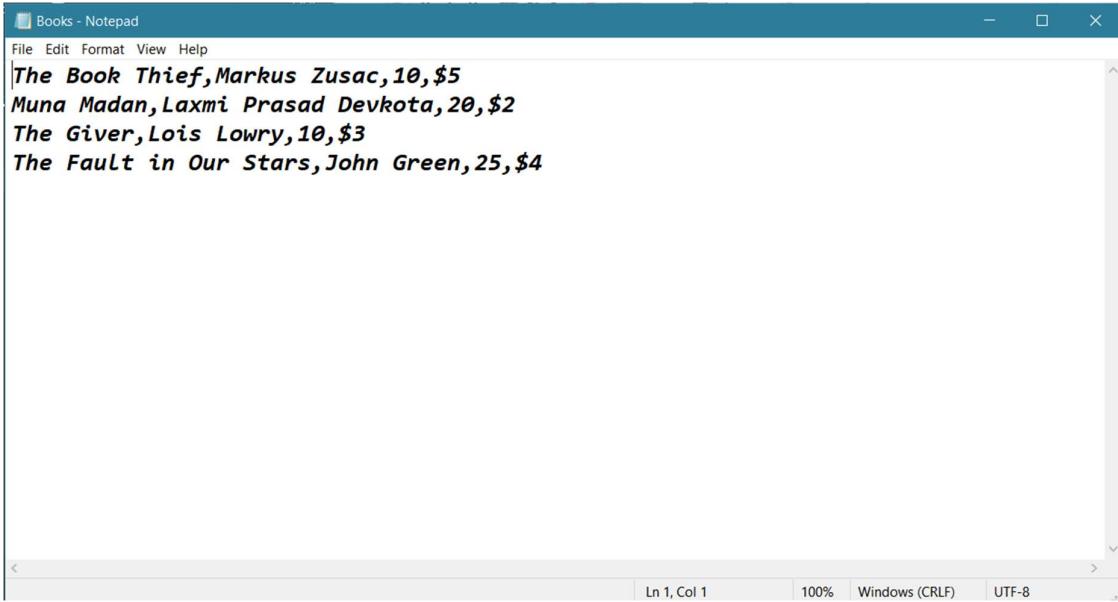
Figure 24: Returning books with Txt file

Test 5

Table of Test 5

Objectives	To show quantity being deducted while burrowing the book and quantity being added when returning the book
Action	Burrow a book and check for the quantity in stock file. Return the book and recheck for addition in the quantity
Expected Result	Quantity is deducted when burrowing and added when returning
Actual Result	Quantity is deducted when burrowing and added when returning
Conclusion	Test successful

Table 5: Test 5



```

Books - Notepad
File Edit Format View Help
The Book Thief, Markus Zusac, 10, $5
Muna Madan, Laxmi Prasad Devkota, 20, $2
The Giver, Lois Lowry, 10, $3
The Fault in Our Stars, John Green, 25, $4

```

Figure 25: The book Stock

The screenshot shows two windows side-by-side. On the left is the Python IDLE Shell 3.9.5 window, and on the right is a Windows Notepad window titled 'Books - Notepad'.

IDLE Shell (Left):

```

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 2
Please Enter the first name of the borrower: ALPHA
Please Enter the last name of the borrower: SUBEDI

Enter an number from below to borrow a book:
To borrow The Book Thief Enter the number 0
To borrow Muna Madan Enter the number 1
To borrow The Giver Enter the number 2
To borrow The Fault in Our Stars Enter the number 3
Enter the number:0
The book is available
Do you wish to borrow another book? Type y for yes and n for no Y
Enter the books corresponding number to borrow t:
If your book choice is The Book Thief Enter the number 0
If your book choice is Muna Madan Enter the number 1
If your book choice is The Giver Enter the number 2
If your book choice is The Fault in Our Stars Enter the number 3
Please Enter the number:1
This book is available
Do you wish to borrow another book? Type y for yes and n for no N
Mind that returning date is 10 days from the date of borrow and fine will be levied for late return of books. Thank you for borrowing books from us.

```

Notepad (Right):

```

File Edit Format View Help
The Book Thief,Markus Zusac,9,$5
Muna Madan,Laxmi Prasad Devkota,19,$2
The Giver,Lois Lowry,10,$3
The Fault in Our Stars,John Green,25,$4

```

Figure 26: Book quantity reduced after borrowing

The screenshot shows two windows side-by-side. On the left is the Python IDLE Shell 3.9.5 window, and on the right is a Windows Notepad window titled 'Books - Notepad'.

IDLE Shell (Left):

```

-----Welcome to our Library-----
Please Enter 1 to Display the books
Please Enter 2 to Borrow our books
Please Enter 3 to Return a borrowed book
To Exit, Please Enter 4

Provide your input here:- 3
Enter the name of the borrower:ALPHA
Details of the Borrower and the book borrowed

Borrowed by: ALPHA SUBEDI
Date and Time of borrowing: 2021-09-06 08:30:06.862418

S.N. Name of the Book Author
1. The Book Thief Markus Zusac
2. Muna Madan Laxmi Prasad Devkota

Total Price= $7.0

Is the book being given back late? Type y for yes and n for no: Y
How many days is the book being given back late?1
You have returned the book late. Therefore, you have to pay a fine.
Total after fine: $9.0

```

Notepad (Right):

```

File Edit Format View Help
The Book Thief,Markus Zusac,10,$5
Muna Madan,Laxmi Prasad Devkota,20,$2
The Giver,Lois Lowry,10,$3
The Fault in Our Stars,John Green,25,$4

```

Figure 27: Book quantity being added after returning books

Conclusion

The course work was finished with bunches of exertion and difficulties. Huge loads of research and analysis were directed to traverse each progression of the work. Routine meetups with educators were conducted where reformist work was shown and significant ideas were embraced. With cycle of the above advances over and over, at last, an executable program playing out the allotted task was created. This course work helped a great deal in transforming my stereotype on programmers and the programming world. This assessment has developed exploration and investigation abilities to cut strides for development of a program, choosing the most ideal approach to foster the program, troubleshooting mistakes produced during compilation. The coursework likewise required an appropriately recorded report of the relative multitude of the program developed and ideas used in development of a functional library management program. As documentation of any task serves to outline and comprehend the assignment, it was exceptionally valuable and supportive in building our expertise of report creation in a standard practiced globally. Using time productively and multitasking to finish allotted coursework before the cut-off time was perhaps the main lesson gained from this assessment. The course work was a strong errand to be finished under set time. All necessary exploration, examination, arranging, improvement, testing, and reporting the interaction was to be done inside that time span. It was an exceptionally close timetable as course work of other subject were likewise to be finished in those equivalent weeks. There were restless evenings and steady programming musings going through my brain in those weeks. Notwithstanding lingering and overpowering inclination, I had to keep my focus on completions and place procrastination aside to focus only on my work. Looking behind starting here, it causes me to feel great that I have finished the job on schedule. To summarize the 1st coursework of Fundamentals of computing where a library executive's framework was to be created utilizing Python programming language, was effectively finished. All things considered, this coursework gave a brief look of what being a future software engineer or a programmer will resemble.

Appendix

Main.py

```
import BorrowBook
import ReturnBook
import Lists
import Date

def main():
    looping=True
    while(looping==True):
        #display choice
        print("\n")
        print("-----Welcome to our Library-----")
        print(" Please Enter +"+"1"+ " to Display the books")
        print(" Please Enter +"+"2"+ " to Borrow our books")
        print(" Please Enter +"+"3"+ " to Return a borrowed book")
        print(" To Exit, Please Enter +"+"4"+")
        print("-----")
        # try block to look for errors in
        try:
            ##Asking input
            id=int(input("Provide your input here:- "))
            #Using the if and else to process further.
            if(id==1):
                #opening a text file and printing its content
                print("\n")
                print("books here are displayed in NAME , AUTHOR , QUANTITY ,
BORROWING PRICE format")
                file=open("Books.txt","r")
                read=file.read()
```

```
    print(read)
    file.close()
elif(id==2):
    #Calling Lists and BorrowBook module
    Lists.lists()
    BorrowBook.borrow()
elif(id==3):
    #Calling Lists and ReturnBook module
    Lists.lists()
    ReturnBook.Return()
elif(id==4):
    #Displaying
    print("Thank you for choosing us!")
    break
else:
    # else block
    print("Please enter a number from the given choices.")
except ValueError:
    #except block if error occurs
    print("Invalid! only numeric data that is within the given choices are accepted.")
main()
```

BorrowBooks.py

```
import Date
import Lists

def borrow():
    IsBorrowed=False
    #Using loops
    while(True):
        # input values
        FirstName=input("Please Enter the first name of the borrower: ")
        LastName=input("Please Enter the last name of the borrower: ")
        print("\n")
        break
    # creating a txt file on borrower
    record="Borrowed by "+FirstName+".txt"
    file=open(record,"w+")
    file.write("\t\t\tDetails of the Borrower and the book borrowed\n")
    file.write("Borrowed by: "+FirstName+ " "+LastName+ "\n")
    file.write("Date and Time of borrowing: "+Date.datetime()+"\n\n")
    file.write("\t\t S.N. \t\t Name of the Book \t\t Author\n")
    file.close()
    while IsBorrowed==False:
        print("Enter an number from below to borrow a book: ")
        for i in range(len(Lists.Books)):
            print("To borrow", Lists.Books[i],"Enter the number",i)
        try:
            num=int(input("Enter the number:"))
            try:
                # writing the borrowed book to txt file
                if(int(Lists.Quantity[num])>0):
```

```

print("The book is available")
file=open(record,"a")
file.write("\t\t 1. \t\t"+Lists.Books[num]+\t\t+Lists.Author[num]+\n")
file.close()
Lists.Quantity[num]=int(Lists.Quantity[num])-1
file=open("Books.txt","w+")
for i in range(4):

file.write(Lists.Books[i]+","+Lists.Author[i]+","+str(Lists.Quantity[i])+",\"$"+Lists.Price
[i]+\n")
file.close()

borrowmultiple=True
counter=1
while borrowmultiple==True:
    # processing multiple borrowingg
    Ask=input("Do you wish to borrow another book? Type y for yes and
n for no ").lower()
    if(Ask=="y"):
        counter=counter+1
        print("Enter the books corresponding number to borrow ti: ")
        for i in range(len(Lists.Books)):
            print("If your book choice is ", Lists.Books[i],"Enter the number",i)
        try:
            check=int(input("Please Enter the number:"))
            try:
                if(int(Lists.Quantity[check])>0):
                    print("This book is available")
                    file=open(record,"a")
                    file.write("\t\t
"+str(counter)+".\t\t"+Lists.Books[check]+\t\t+Lists.Author[check]+\n")
                    file.close()
            except:
                print("Book not available")
        except:
            print("Please enter a valid number")
    else:
        break

```

```
Lists.Quantity[check]=int(Lists.Quantity[check])-1
file=open("Books.txt","w+")
for i in range(4):
    file.write(Lists.Books[i]+","+Lists.Author[i]+","+str(Lists.Quantity[i])+"."++$"+Lists.Price[i]+"\n")
    file.close()
else:
    borrowmultiple=False
    break
# tracing and warning for invalid data input and errors.
except IndexError:
    print("Choose the books according to their number")
except ValueError:
    print("Choose as suggested")
else:
    print("Mind that returning date is 10 days from the date of borrow and fine will be levied for late return of books. Thank you for borrowing books from us.\n\n")
    borrowmultiple=False
    IsBorrowed=True
except IndexError:
    print("Choose the books according to their number")
except ValueError:
    print("Please Choose within the given options only")
```

ReturnBooks.py

```
import Lists
import Date

def Return():

    #input value
    Name=input("Enter the name of the borrower:")
    text="Borrowed by "+Name+".txt"

    # try block for errors
    try:

        file=open(text,"r")#read text file
        line=file.readlines()
        line=[text.strip("$") for text in line]#Removing the $ using strip keyword
        file.close()

        file=open(text,"r")
        info=file.read()
        print(info)

    #Using the except block for error
    except:
        print("The name of the borrower does not match")
        Return()

    store="Returner "+Name+".txt"
    file=open(store,"w+")#read and write text file
    print("\n")
    print("\n")
    file.write("\t\t\tDetails of the Returned Book \n\n")
    file.write("Date and Time of Returning: "+Date.datetime()+"\n\n")
    file.write("Returned by: "+Name+"\n")
```

```
#Using function of Date module
```

```
file.write("\t\tS.N.\t\t Name of the Book\t\tCost\n")
```

```
file.close()
```

```
cost=0.0
```

```
#for loop to add in the text file
```

```
for i in range(4):
```

```
    if Lists.Books[i] in info:
```

```
        file=open(store,"a")# text file in append mode
```

```
        #Lists module to write inside the text file
```

```
        file.write("\t\t"+str(i+1)+"\t"+Lists.Books[i]+\t\t$"+Lists.Price[i]+\n")
```

```
        file.close()
```

```
    Lists.Quantity[i]=int(Lists.Quantity[i])+1
```

```
    cost=cost+float(Lists.Price[i])
```

```
print("\t\t Total Price= "+"$"+str(cost)+"\n\n")
```

```
check=input("Is the book being given back late (if you return the book after 10 days its late)? Type y for yes and n for no: ").lower()
```

```
#if else function
```

```
if check=="y":
```

```
    days=int(input("How manys days is the book being given back late?"))
```

```
    if days>=1:
```

```
        print("You have returned the book late. Therefore, you have to pay a fine.")
```

```
        fine=2*days
```

```
        file=open(store,"a")
```

```
        file.write("\t\tFine: $" +str(fine)+ "\n")
```

```
        cost=cost+fine
```

```
        print("Total after fine: $" +str(cost))
```

```
#Opening the text file in read and write mode
```

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

```
#Using for loop to change the values in text file
```

```
for i in range(4):
```

```
    file.write(Lists.Books[i] + " " + Lists.Author[i] + " " + str(Lists.Quantity[i]) + " $" + Lists.Price[i] + "\n")
```

```
file.close()
```

Lists.py

```
Books = []
Author = []
Quantity = []
Price = []

def lists():
    file=open("Books.txt","r")
    read=file.readlines()
    read=[a.strip("\n") for a in read]
    for i in range(len(read)):
        a=0
        for j in read[i].split(","):
            #printing(j)
            if(a==0):
                Books.append(j)
            #printing(Books)
            elif(a==1):
                Author.append(j)
            elif(a==2):
                Quantity.append(j)
            elif(a==3):
                Price.append(j.strip("$"))
        a=a+1
```

Date.py

```
def datetime():
    import datetime
    dandt=datetime.datetime.now()
    #print("Date and Time:",dandt)
    return str(dandt)
```