



slington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Asal Pandey

Group: C7

London Met ID: 20049387

College ID: NP01CP4S210181

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

Introduction	1
Discussion and analysis.....	2
• Algorithm	2
• Flowchart	5
• Pseudocode	7
Pseudocode for main.py	7
Pseudocode for Split.py	9
pseudocode for borrow	10
pseudocode for return	12
Pseudocode for DateTime	14
• program	14
• Data structures	16
Testing	19
Test 1	19
Test 2.....	20
Test 3.....	21
Test 4.....	22
Test 5.....	24
Conclusion.....	27
Bibliography	28
Appendix.....	28
Main.py	28
Borrow.py	30
Return.py	33
Split.py.....	35
DateTime.py	36

List of Figures

Figure 1: flowchart.....	7
Figure 2: example, use of integer datatype in the program in main.py	17
Figure 3: example, use of float datatype in the program in return.py	17
Figure 4:example, use of boolean in borrow.py for success and loop	18
Figure 5: example, implementation of string in the program in return.py	18
Figure 6: provide invalid input and show the message.....	20
Figure 7: testing by providing negative value and non-existent vale as input.....	21
Figure 8: checking the generated file after running the borrowing process	22
Figure 9: running return process	23
Figure 10: generation of return-borrower.txt file	24
Figure 11: initial booklist.....	25
Figure 12:booklist after borrowing	26
Figure 13: booklist after returning.....	27

List of Tables

Test 1: implementation of the try-catch method	19
Test 2: testing borrow and return option.....	21
Test 3: to check the text file generation of borrow	22
Test 4:test, to check the text file generation of return.....	23
Test 5: To check the update in stock.....	24

Introduction

This coursework has been assigned to us to make a library system using the python programming language. The main objective of this project is to summarise the learnings of this module this semester and to build a college library system

The goal of this project is to achieve a user-friendly college library system that stores all the information of the library books and tracks records of every book transaction done in the library. It also posses the capability to calculate the fine for the borrower if returned late.

This report consists of the algorithm, flowchart and pseudocode of the college library system program which we built using python programming language and the summary of the whole program building journey plus testing of the program. This program records the data of all the books in the library, their quantity, prices, borrowed date etc to keep track of its stock and transactions of books. The program is built for the college library.

For ease of reference, algorithm and pseudocode have been written and a flowchart has also been constructed in this report. 'List' data structure has been selected to represent the information to be processed by the program. List comprehension method and indexing have also been used to organize the data in the program.

for easement of us, the program is divided into 5 different modules which are Split, DateTime, Borrow, Return and Main. The user is asked to choose between 4 different choices which are display, borrow, return and exit. the choice display displays all the books available in the library to be borrowed. option borrow is used to borrow the books from the library where data of the borrower is recorded. the return option is used to return the book which was borrowed and also can calculate the fine, if the return of the

borrowed book is delayed then the given time and it can be charged to the borrower. and the option exit the program and if it is selected a thank you message is displayed.

“List” data structure is used in this program to record all the data of the books which are available in the library. “stock.txt” file contains all the data about the books which are available in the library and is accessed by our program and mutated accordingly, according to the changes in the stocks of the library.

Discussion and analysis

- **Algorithm**

Main function

Steps:

1. Print welcome screen
2. Print menu as below

Enter 1 to display

Enter 2 to borrow a book

Enter 3 to return a book

Enter 4 to exit

3. Input 1-4 from the user to choose an option in the menu

4. Choice 1:

a) the input number is 1 to display books

b) Open "stock.txt"

c) Read from "stock.txt" and store the action in a variable "lines"

d) Print "lines"

5. Choice 2:

a) the input number is 2 chosen to borrow books by user

b) input the First name of the borrower

c) input the Last name of the borrower

d) select the name of the book

e) Create a borrower_Borrow.txt file with the borrower's name, book borrowed, cost and stock quantity

f) check if the quantity of stock is more than zero

g) if yes then, append details in the borrow.txt file and add

further information. if no print " the book is not available" and go to "(d)"

h) ask do u want to borrow more books? if yes go to step 4, if no go to ("i")

i) Exit

6. choice 3

a. the input number chosen is 3 to return the borrowed books by the user

b. input the name of the borrower

- c. if the name is valid and has a borrow.txt file, read their respective borrow.txt
 - if not then, print "the borrower name is incorrect" and go back to "b."
 - d. Display the borrow.txt file of the respective borrower
 - e. generate return.txt file and update the book stock file
 - f. ask, have the books return date time expired? if yes go to step 7
 - 1. if no go to step 10
 - 2. late return penalty fine is added
 - g. $\text{fine} = 1.5 * \text{day/s}$
 - h. Add total fine and cost of the books
 - i. text file with the total cost is generated
 - j. End
7. Choice 4
- a. the input number chosen is 4 to end the program
 - b. print "thank you for using our college library system"
 - c. end program

- **Flowchart**

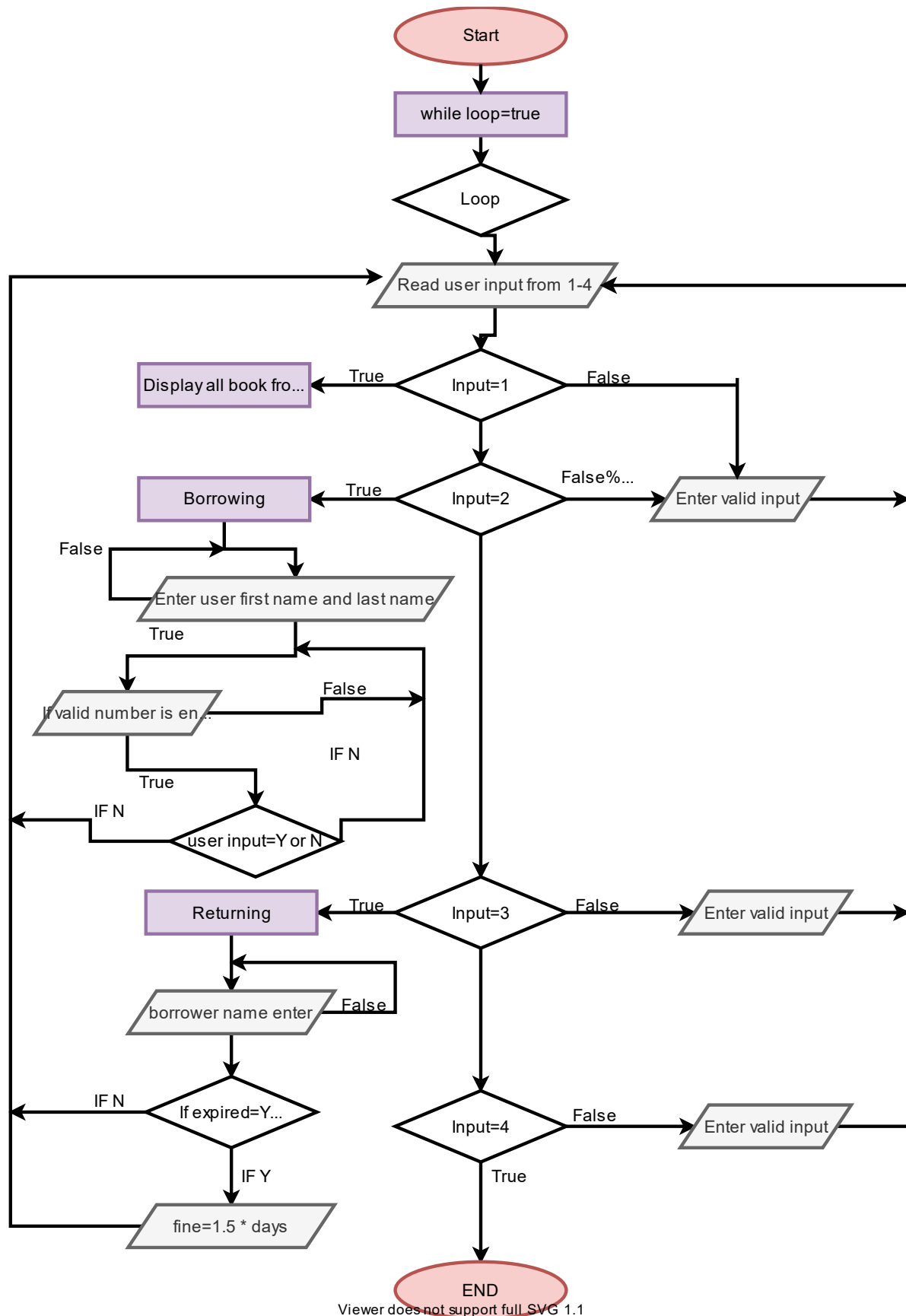


Figure 1: flowchart

- **Pseudocode**

Pseudocode for main.py

IMPORT Return

IMPORT Split

IMPORT DateTime

IMPORT Borrow

CREATE function start

while(True):

 PRINT "Welcome to our college library management system"

 PRINT "-----"

 PRINT "Input 1 . To display the books available"

 PRINT "Input 2 . To borrow a book"

 PRINT "Input 3 -.To return a book"

 PRINT" Input 4 . To exit the library"

 TRY

 INPUT "select a choice from 1 - 4 : " in integer datatype in variable
 "choose"

 IF choose is 1

 CALL listSplit function from Split module

 PRINT "The available books are :"

 OPEN borrow.txt in read mode

 lines=f.read()

 print(lines)

```
PRINT ()
    END IF

    ELIF choose is 2

        CALL listSplit function from Split module
        CALL borrowbook function from Borrow module
    END ELIF

    ELIF choose is 3
        CALL listSplit function from stock module
        CALL returnbook function from Return module
    END ELIF

    ELIF choose is 4
        PRINT "Thank you for using our college library management
        system"
        Break
    END ELIF

    ELSE
        PRINT "Please enter a number from 1 to 4"
    END ELSE

END TRY

EXCEPT ValueError
    PRINT "Please enter a valid choice from 1-4"
```

END EXCEPT

END Function start

Pseudocode for Split.py

Define listSplit function

```
    CREATE empty list for Bookname, Authorname, Quantity, cost
OPEN books.txt
Read book.txt
Strip lines for book.txt
    For i in range(len(lines)):
        count=0
        for a in lines[i].split(','):
            if(count==0):
                bookname.append(a)
        End for
        End if
        ELIF(count==1):
            authorname.append(a)
        End ELIF
        ELIF (count==2):
            quantity.append(a)
        End ELIF
        ELIF (count==3):
            cost.append(a.strip("$"))
```

```
        count=count+1
End ELIF
```

pseudocode for borrow

```
IMPORT DateTime and Split
DEFINE borrowBook FUNCTION():
    INTITALIZE success to False
    WHILE(True):
        INPUT firstname
        break
    PRINT("please input a valid name")
    END IF
    END WHILE
    WHILE(True):
        INPUT lastname
        break
    PRINT("please input a valid name")
    END IF
    END WHILE

    CREATE borrow text file with first name

    WHILE
    initialize success to False:
        PRINT("Please select a option below:")
        FOR I in range(len(Split.bookname)):
            PRINT("Enter", i, "to borrow book", Split.bookname[i])
        END FOR
```

TRY:

initialize int (a)

TRY:

if(int(Split.quantity[a])>0):

 PRINT("Book is available")

 UPDATE borrower textfile

 DECREASE quantity by

 WRITE book text file

initilize loop to True

initilize count to 1

while loop==True:

 INPUT string choice

 if(choice.upper()=="Y"):

 INCREASE count by 1

 PRINT("Please select an option below:")

 for i in range(len(Split.bookname)):

 PRINT("Enter", i, "to borrow book", Split.bookname[i])

 initilize input a

 if(int(Split.quantity[a])>0):

 PRINT("Book is available")

 UPDATE borrower textfile

 DECREASE quantity by

 WRITE book text file

 initilize success to False

 EELSE:

 loop=False

 break

 END IF


```

        END ELSE
    elif (choice.upper()=="N"):
        print ("Thank you for borrowing books from us. ")
        print("")
        initialize loop to False
        initialize success to True
    END ELIF
else:
    print("Please choose as instructed")
    END ELSE

else:
    print("Book is not available now!")
    Call borrowbook function
    initialize success to False
    END ELSE
except IndexError:
    print("")
    print("Please choose book according to their number.")
    END TRY
except ValueError:
    print("")
    print("Please choose as suggested.")
    END TRY
END while

```

pseudocode for return

```

IMPORT Split and DateTime
define returnBook Function():
    INPUT name

```



```
total=total+fine  
END iF
```

```
PRINT("Final Total: "+ "$"+str(total))  
WRITE total in text file
```

CREATE new updated textfile as books

Pseudocode for DateTime

```
CREATE function getdate()  
    IMPORT datetime  
    ASSIGN current date and time to variable now  
    RETURN date from variable now in string datatype  
CREATE function gettime()  
    IMPORT datetime  
    ASSIGN current date and time to variable now  
    RETURN time from variable now in string datatype
```

- **program**

for this project, five .py files are created which are Split.py, main.py, DateTime.py, Borrow.py and Return.py. This program is designed for a library and it can track all the transactions of books and has also capabilities for calculating the fine if the borrower returns the book late.

Main.py file consists of a user-defined function which is start(). In this module of the program, the user is given 4 different choices which are to display book stocks of the library, borrow a book, return a book and end the program. If the user chooses to display, all the books in the library are shown with their cost, author and quantity.

And if the user chooses to borrow, Borrow.py module (which consists of a user-defined borrowbook() function)is called and a text file is generated for the borrower, with the date, cost, quantity of book in the library after the transaction and book name. then the stock is updated accordingly.

And if the user chooses to return, Return.py module (which consist of returnbook() user-defined function) is called and, if the borrow file of the borrower is present in the system, then the user can return the book. The book stock will be updated accordingly in the books.txt file and the user is asked if the return is late or not. If the return of the book is late than the specified date, a fine can be calculated and charged according to how many days it is late

And end the program terminates the program

Invalid input from the user in the choices displays the error message

Split.py module(which contains listSplit() user-defined function) splits the data in text file books.txt to bookname,authorname, quantity and cost and stores them in their respective global variables

DateTime.py module(which consists of getDate() and getTime() user-defined functions) stores the date and time informations and is called by other modules if those data are required.

- **Data structures**

Data structures are a way of organizing and storing data so that they can be accessed and worked with efficiently. They define the relationship between the data and the operations that can be performed on the data. There are many various kinds of data structures defined that make it easier for the data scientists and computer engineers, like to concentrate on the main picture of solving larger problems rather than getting lost in the details of data description and access. (Jaiswal, 2018)

Python provides varieties of data types like integer, string, Boolean, float etc. Some of the data types and data structures that were used while writing the program to store and manipulate the data and perform various operations on them are as follows:

Use of list in the program (non-primitive data structure):

list is used to store values of bookname, authorname, quantity and cost in our program. The list values can be added or removed as per the change in stocks in the library, by the use of methods `append()` and `extend()` which have been successfully included in different modules of our program.

Primitive data structures:

Integer: it is a data type of python that stores integer values. the integers are implemented in multiple places in the program, like when taking the input from the user to select a book to borrow, return or display. When taking input to select a book to borrow. it is used to store the quantity of the books in the variable `quantity[index]` and many more


```

        for i in range(3):
            f.write(Split.bookname[i]+","+Split.authorname[i]+","+str(Split.quantity[i])+","+ "$"+Split.cost[i)+"\n")
            success=False
        else:
            loop=False
            break
    elif (choice.upper()=="N"):
        print ("Thank you for borrowing books from us. ")
        print("")
        loop=False
        success=True
    else:
        print("Please choose as instructed")

    else:
        print("Book is not available now!")
        borrowBook()
        success=False
except IndexError:
    print("")

```

Figure 4: example, use of boolean in borrow.py for success and loop

String: it is a data type that stores values which are in texts. it stores any character as long as it is encoded with “ “. Printed messages in the program use string data type and various operations like choices of the users like yes and no are also encoded with the string data type.

```

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

with open("Books.txt", "w+") as f:
    for i in range(3):
        f.write(Split.bookname[i]+","+Split.authorname[i]+","+str(Split.quantity[i])+","+ "$"+Split.cost[i)+"\n")

```

Figure 5: example, implementation of string in the program in return.py

Testing

The testing process is carried out to check the program is running as expected and to find any bugs and correct if there is any

Test 1

Objective	To check whether the implementation of the try-catch method is valid or not
Action	Provide string input
Expected results	It should be able to handle any invalid input and display a "please enter a valid choice from 1-4" message
Actual results	It handled the exception and showed a "please enter a valid choice from 1-4" message
Conclusion	The test was successful

test 1: implementation of the try-catch method

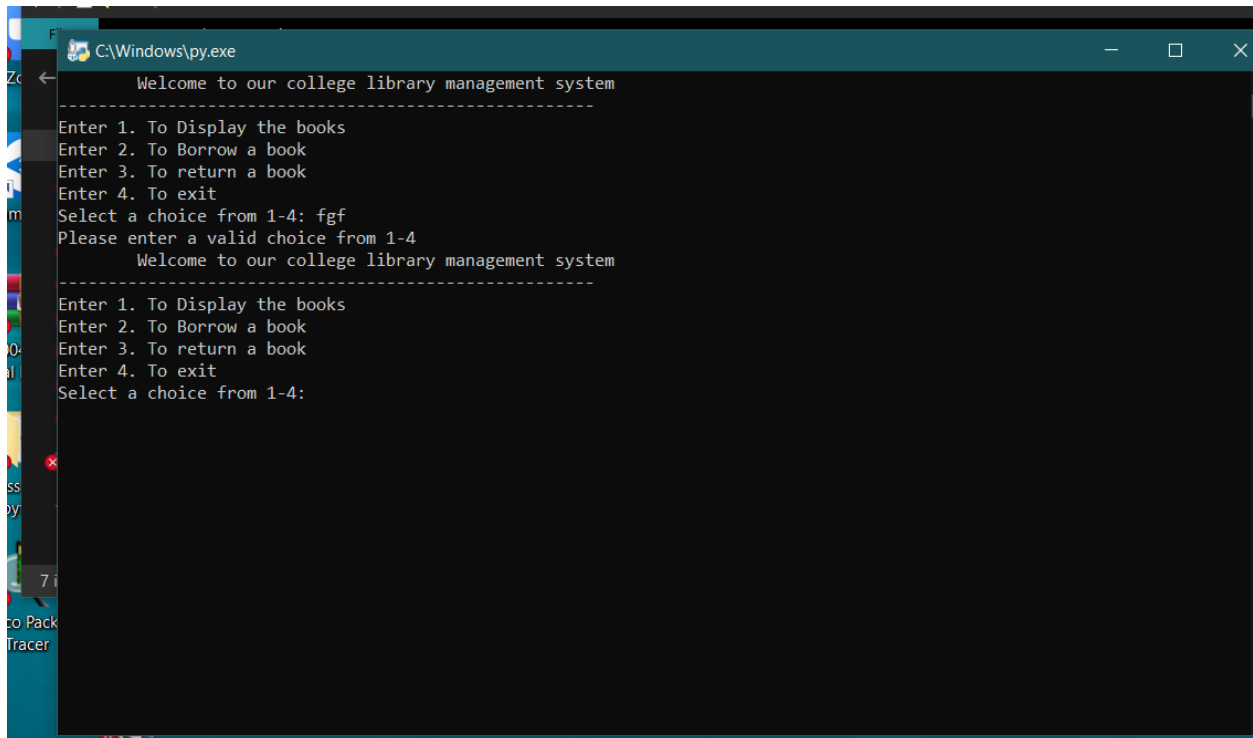


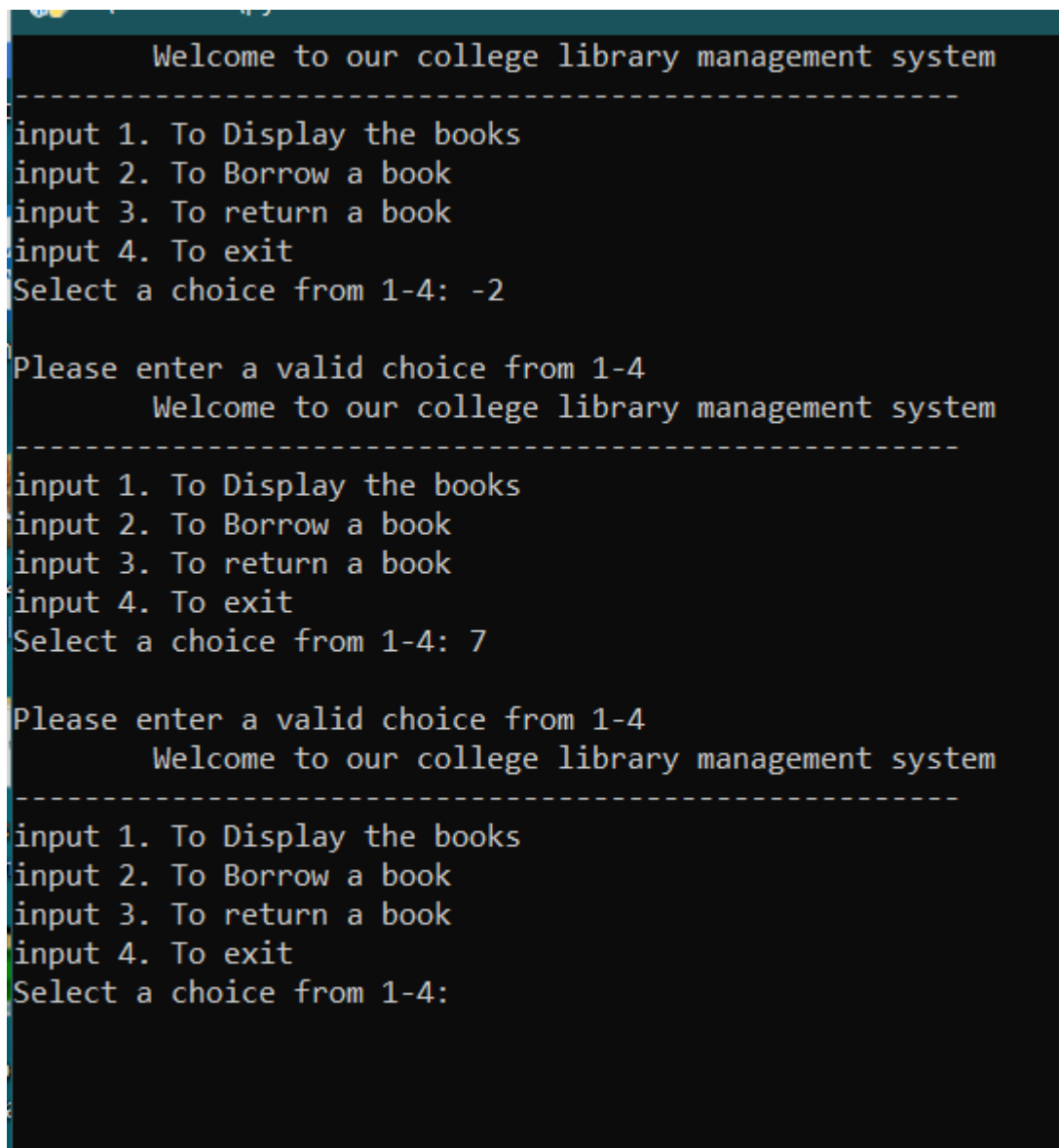
Figure 6: provide invalid input and show the message

Test 2

Objective	Testing selection of borrow and return option
Action	Provide a negative value as input and Provide the random non-existent value as input
Expected results	Negative value and non-existence values show an error message

Actual results	Negative value and non-existence values show an error message
Conclusion	Test was successful

Test 2: testing borrow and return option



```

Welcome to our college library management system
-----
input 1. To Display the books
input 2. To Borrow a book
input 3. To return a book
input 4. To exit
Select a choice from 1-4: -2

Please enter a valid choice from 1-4
Welcome to our college library management system
-----
input 1. To Display the books
input 2. To Borrow a book
input 3. To return a book
input 4. To exit
Select a choice from 1-4: 7

Please enter a valid choice from 1-4
Welcome to our college library management system
-----
input 1. To Display the books
input 2. To Borrow a book
input 3. To return a book
input 4. To exit
Select a choice from 1-4:

```

Figure 7: testing by providing negative value and non-existent value as input

Test 3

Objective	To check text file generation of borrow
-----------	---

Action	Run the borrowing process and check the generated text file
Expected results	Borrow file of the borrower is generated
Actual results	Borrow file of the borrower is generated
Conclusion	Test was successful

test 3: to check the text file generation of borrow

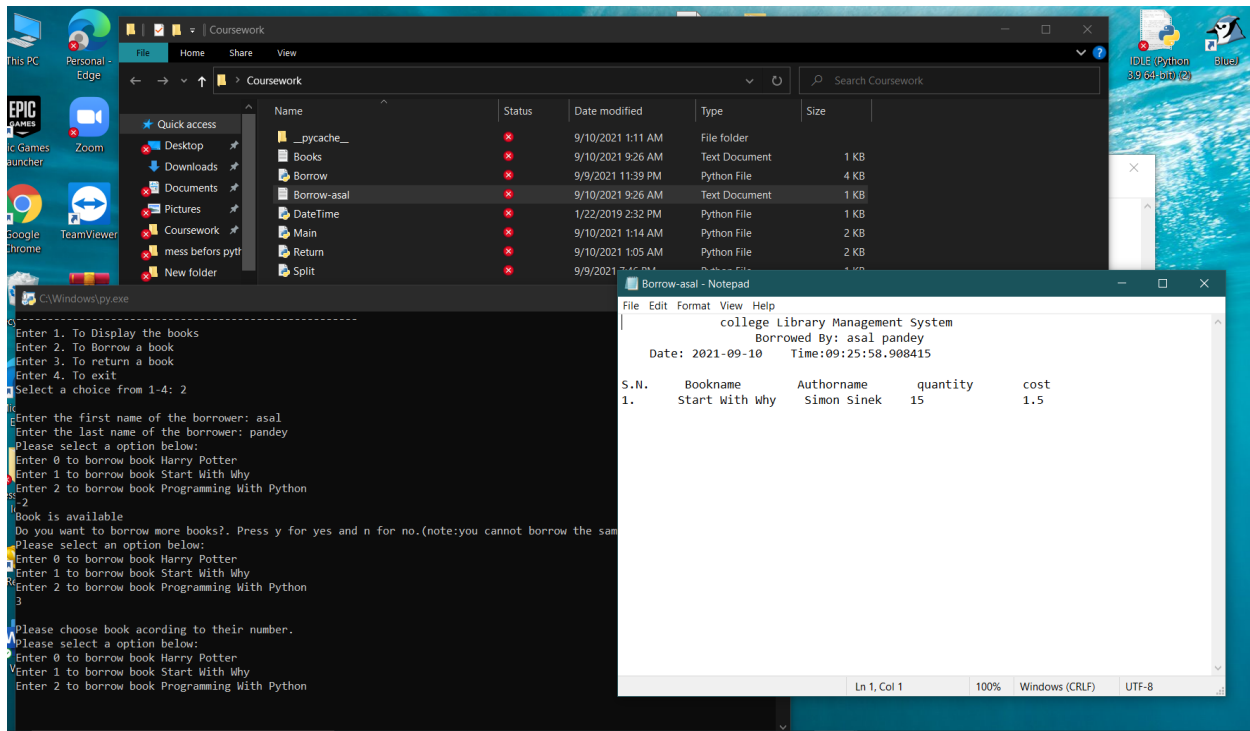


Figure 8: checking the generated file after running the borrowing process

Test 4

Objective	To check text file generation of return
Action	Run the returning process and check the generated text file
Expected results	return file of the borrower is generated
Actual results	return file of the borrower is generated

Conclusion

The test was successful

Test 4: test, to check the text file generation of return

```
C:\Windows\py.exe
Thank you for borrowing books from us.

Welcome to our college library management system
-----
input 1. To Display the books
input 2. To Borrow a book
input 3. To return a book
input 4. To exit
Select a choice from 1-4: 3

Enter name of borrower: asal
college Library Management System
Borrowed By: asal pandey
Date: 2021-09-10 Time:13:10:48.193983

S.N.    Bookname    Authorname    quantity    cost
1.      Start With Why    Simon Sinek    16          1.5
                                           $1.5

Is the book return date expired?
Press Y for Yes and N for No
n
Final Total: $1.5
Welcome to our college library management system
-----
input 1. To Display the books
input 2. To Borrow a book
input 3. To return a book
input 4. To exit
Select a choice from 1-4: _
```

Figure 9: running return process

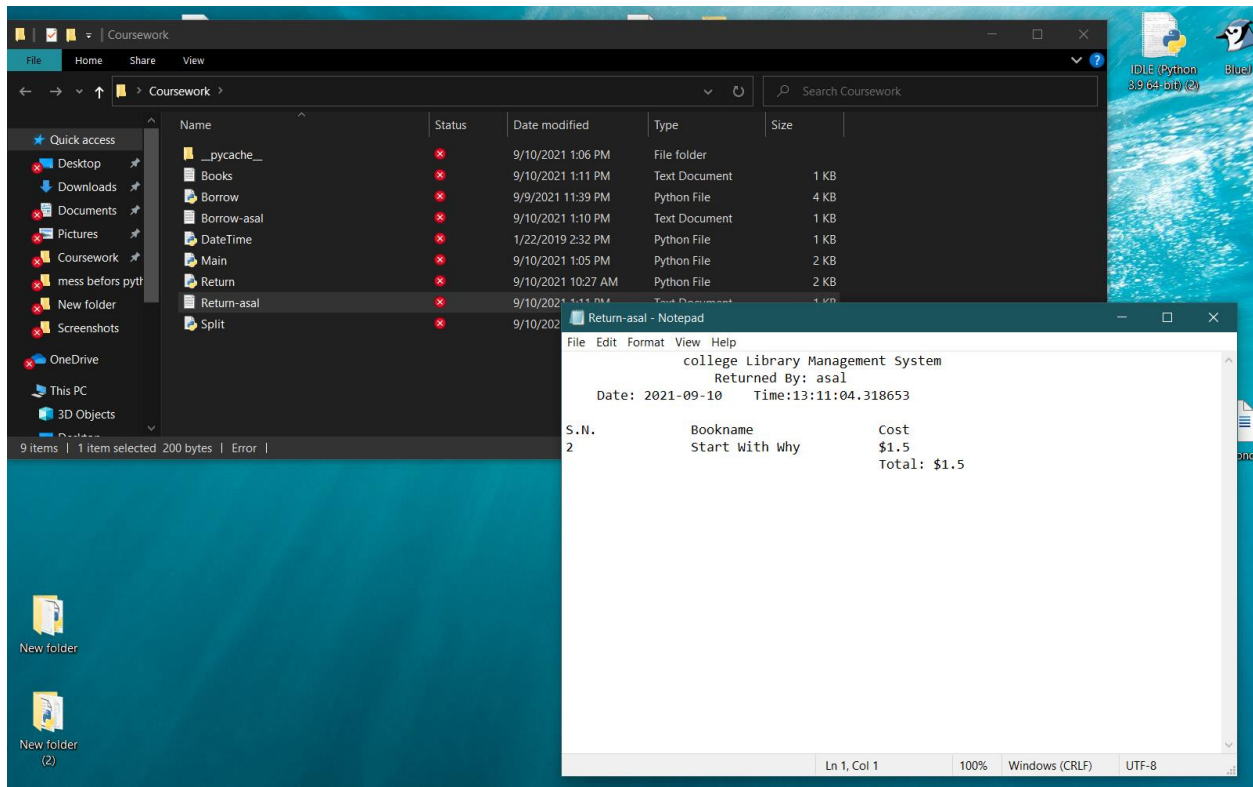


Figure 10: generation of return-borrower.txt file

Test 5

Objective	To check whether the library stocks will be updated by the program or not
Action	Check initial book textfile Check book text file after borrowing Check book text file after returning
Expected results	Quantity decreases after borrowing and increases after returning
Actual results	Quantity decreases after borrowing and increases after returning
Conclusion	The test was successful

Test 5: To check the update in stock

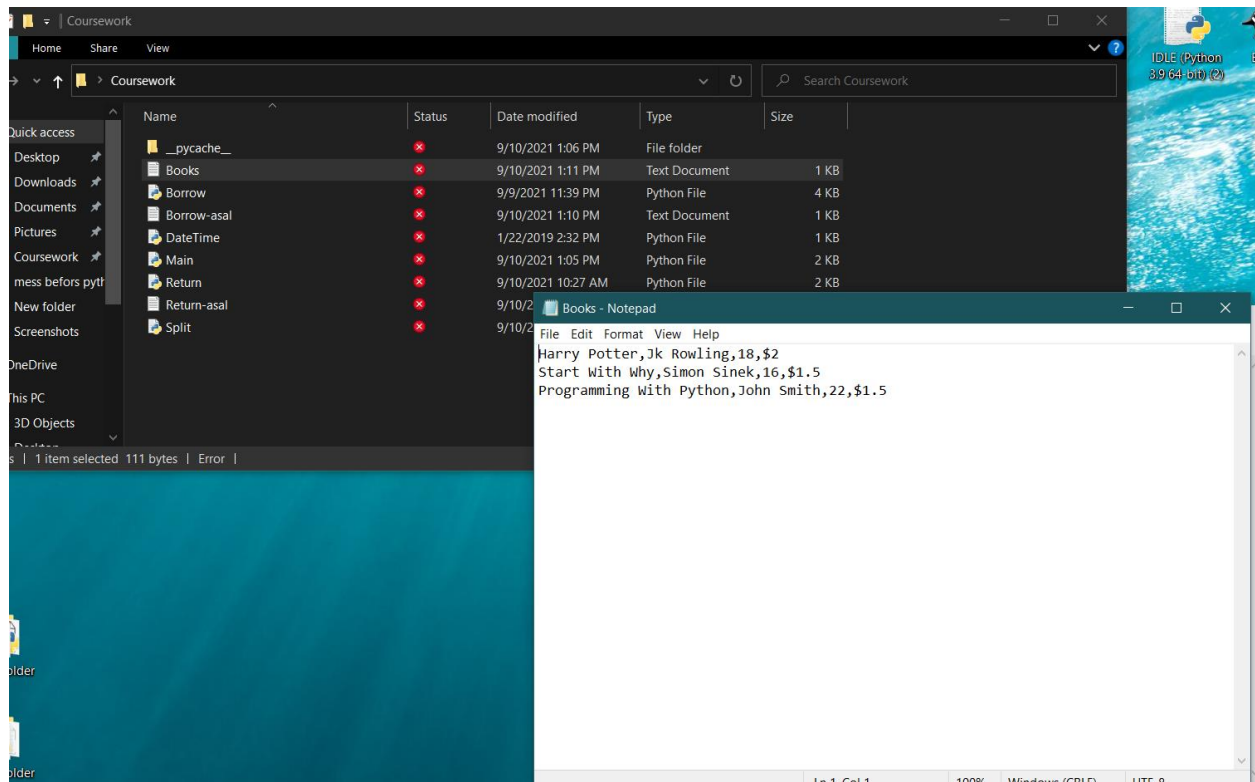


Figure 11: initial booklist

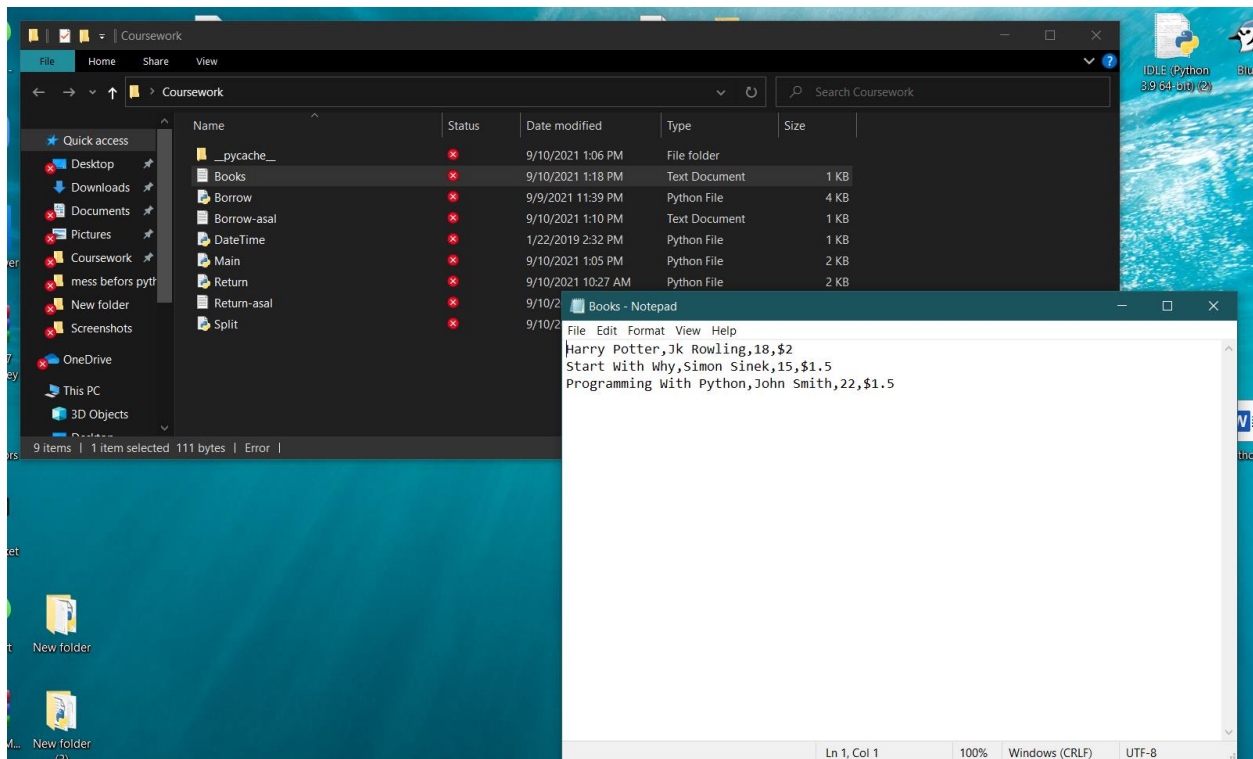


Figure 12:booklist after borrowing

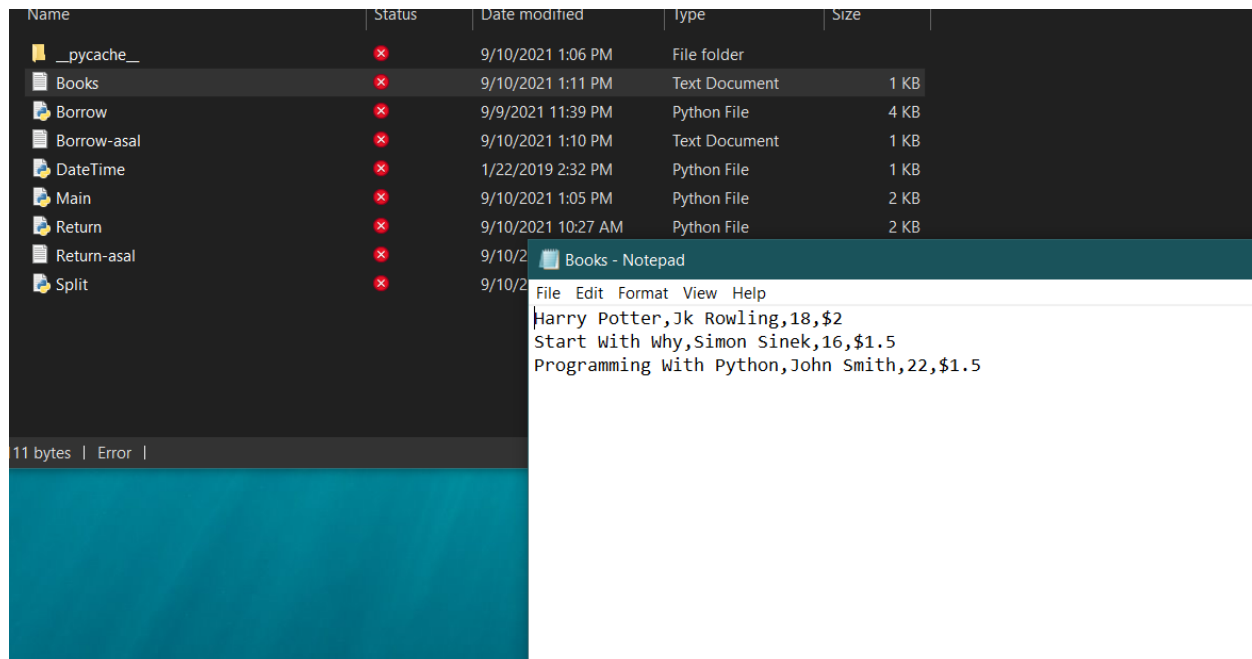


Figure 13: booklist after returning

Conclusion

A lot of hard work and effort was consumed while doing this coursework. While doing this coursework, I had to encounter many bugs and errors. A lot of research was done through the internet, and python related books. Lecture slides also helped a lot.

In this coursework, we were asked to build a college library system using the python programming language. This was my first time doing a real-life project using python. So, I was excited and the coursework was completed easily.

I had faced lots of difficulties while performing this task. The topic was new to me, as I mentioned before. Plus due to the current pandemic situation and discontinuation of physical classes, completing the coursework in time was comparatively difficult. It was hard for me to perform this coursework. But with the help of the algorithm, pseudocode, flowchart, researches and module leader... I was able to complete this coursework with ease. I came across some bugs while doing this coursework but with the help of our

Module Leader and friends, I was able to debug it. This coursework has made me fluent with the concepts of python, functions, its implementations, lists and many more this project was due submitted before 10th September 2021 and was done very cautiously and effectively. the program was intended to satisfy the needs of Islington college library. a lot of research was also done to complete this project fruitfully.

Bibliography

Jaiswal, S., 2018. *python datastructure tutorial*. [Online]

Available at: https://www.datacamp.com/community/tutorials/data-structures-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adposition=&utm_creative=278443377095&utm_targetid=ds

[Accessed 09 09 2021].

programiz, n.d. *python lists*. [Online]

Available at: <https://www.programiz.com/python-programming/list>

[Accessed 06 09 2021].

w3schools, n.d. *w3schools*. [Online]

Available at: www.w3schools.com

[Accessed 06 09 2001].

Appendix

Main.py

```
import Return
```

```
import Split
import DateTime
import Borrow

def start():
    while(True):
        print("      Welcome to our college library management system      ")
        print("-----")
        print("input 1. To Display the books")
        print("input 2. To Borrow a book")
        print("input 3. To return a book")
        print("input 4. To exit")
        try:
            choose=int(input("Select a choice from 1-4: "))
            print()
            if(choose==1):
                with open("Books.txt","r") as f:
                    lines=f.read()
                    print(lines)
                    print ()

            elif(choose==2):
                Split.listSplit()
                Borrow.borrowBook()
            elif(choose==3):
                Split.listSplit()
                Return.returnBook()
            elif(choose==4):
                print("Thank you for using college library management system")
                break
        else:
```

```
        print("Please enter a valid choice from 1-4")
    except ValueError:
        print("Please enter a valid choice from 1-4")
start()

Borrow.py
import DateTime
import Split

def borrowBook():
    success=False
    while(True):
        firstName=input("Enter the first name of the borrower: ")
        if firstName.isalpha():
            break
        print("please input a valid name")
    while(True):
        lastName=input("Enter the last name of the borrower: ")
        if lastName.isalpha():
            break
        print("please input a valid name")

    t="Borrow-"+firstName+".txt"
    with open(t,"w+") as f:
        f.write("          college Library Management System \n")
        f.write("          Borrowed By: "+ firstName+" "+lastName+"\n")
        f.write("    Date: " + DateTime.getDate()+"    Time:"+ DateTime.getTime()+"\n\n")
        f.write("S.N. \t Bookname \t Authername \t quantity \t cost \n")

    while success==False:
        print("Please select a option below:")
        for i in range(len(Split.bookname)):
```

```

print("Enter", i, "to borrow book", Split.bookname[i])

try:
    a=int(input())
    try:
        if(int(Split.quantity[a])>0):
            print("Book is available")
            with open(t,"a") as f:
                f.write("1. \t"+ Split.bookname[a]+" \t "+Split.authorname[a]+" \t
"+Split.quantity[a]+" \t \t "+Split.cost[a]+" \n")

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

f.write(Split.bookname[i]+",""+Split.authorname[i]+",""+str(Split.quantity[i])+",""+ "$"+Split.co
st[i]+" \n")

#code for borrowing more than one book.
loop=True
count=1
while loop==True:
    choice=str(input("Do you want to borrow more books?. Press y for yes
and n for no.(note:you cannot borrow the same book twice)"))
    if(choice.upper()=="Y"):
        count=count+1
        print("Please select an option below:")
        for i in range(len(Split.bookname)):
            print("Enter", i, "to borrow book", Split.bookname[i])
            a=int(input())

```

```

        if(int(Split.quantity[a])>0):
            print("Book is available")
            with open(t,"a") as f:
                f.write(str(count) +". \t"+ Split.bookname[a]+" \t"
"+Split.authername[a]+" \t "+Split.quantity[a]+" \t\t "+Split.cost[a]+" \n")

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

f.write(Split.bookname[i]+", "+Split.authername[i]+", "+str(Split.quantity[i])+", "+ "$"+Split.co
st[i]+" \n")

                success=False
            else:
                loop=False
                break
        elif (choice.upper()=="N"):
            print ("Thank you for borrowing books from us. ")
            print("")
            loop=False
            success=True
        else:
            print("Please choose as instructed")

    else:
        print("Book is not available now!")
        borrowBook()
        success=False
except IndexError:
    print("")
    print("Please choose book according to their number.")

```

```
except ValueError:
    print("")
    print("Please choose as suggested.")
```

Return.py

```
import Split
import DateTime
def returnBook():
    name=input("Enter name of borrower: ")
    a="Borrow-"+name+".txt"
    try:

        with open(a,"r") as f:
            lines=f.readlines()
            lines=[a.strip("$") for a in lines]

        with open(a,"r") as f:
            data=f.read()
            print(data)
    except:
        print("The borrower name is incorrect")
        returnBook()

    b="Return-"+name+".txt"
    with open(b,"w+")as f:
        f.write("          college Library Management System \n")
        f.write("          Returned By: "+ name+"\n")
        f.write("    Date: " + DateTime.getDate()+"    Time:"+ DateTime.getTime()+"\n\n")
        f.write("S.N.\t\tBookname\t\tCost\n")
```

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

print("\t\t\t\t\t"+ "$"+str(total))
print("Is the book return date expired?")
print("Press Y for Yes and N for No")
stat=input()
if(stat.upper()=="Y"):
    print("By how many days was the book returned late?")
    day=int(input())
    fine=1.5*day

    with open(b,"a")as f:
        f.write("\t\t\t\t\tFine: $"+ str(fine)+"\\n")
    total=total+fine

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

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

```
f.write(Split.bookname[i]+","+Split.authorname[i]+","+str(Split.quantity[i])+","+ "$"+Split.co  
st[i]+"\\n")
```

Split.py

```
def listSplit():  
    global bookname  
    global authorname  
    global quantity  
    global cost  
    bookname=[]  
    authorname=[]  
    quantity=[]  
    cost=[]  
    with open("Books.txt","r") as f:  
  
        lines=f.readlines()  
        lines=[x.strip('\\n') for x in lines]  
        for i in range(len(lines)):  
            count=0  
            for a in lines[i].split(','):  
                if(count==0):  
                    bookname.append(a)  
                elif(count==1):  
                    authorname.append(a)  
                elif(count==2):  
                    quantity.append(a)  
                elif(count==3):  
                    cost.append(a.strip("$"))  
            count=count+1
```


DateTime.py

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