LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title CS4051NI Fundamentals of Computing

Assessment Weightage & Type 60%

Individual Coursework

Year and Semester 2020-21

Autumn

Student Name: Itesh niroula

Group: L1C10

London Met ID: 20048902

College ID: np01cp4s210307

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

Table of Figure:

List of tables:

INTROCUTION:

Brief introduction of project:

The coursework was assigned to us in the sixth week, and we had to complete it by the 12th week. The course was about creating a Library Management System. IDLE was used to write the program, which was written in Python. Python is a high-level object-oriented programming language with integrated dynamic semantics that is mostly used for web and app development. It is extremely interesting in the subject of Faster Development since it gives dynamic type and dynamic binding capabilities. Like wise IDLE (short for Integrated Development and Learning Environment) is a Python integrated development environment that has been included with the language's default implementation since version 1.5.

Similarly, the Flowchart was created with draw.io. The following project was also produced using IDLE's support. The Project is a simple piece of software that allows users to pick whether they want to borrow or return a book. The application includes a number of functionalities that aid in the accomplishment of various tasks.

goals and objectives:

Objectives of the program are to comprehend the significance of Python as a scripting language. For displaying the codes as flowcharts and algorithms. To create OOPs using Python classes. In order to learn how to develop and write Python apps. Learning how to use tuples, lists, and dictionaries correctly.

Goals of the project is to make library management system. Where the we can borrow the book and return the books. Also, to maintain the price and quantity of the books. If borrower the is unable to return the book within 10days fine will also added. To identify easily unique book id was given to all of book.

Discussion and Analysis:

Algorithm of the Project:

An algorithm (pronounced AL-go-rith-um) is a method or formula for solving a problem that is based on a series of specified steps. A computer program can be thought of as a complex algorithm. An algorithm is a simple technique that solves a recurring problem in mathematics and computer science.

Step 1: Start

Step 2: Press D for display books or
Press L for lend_book books or
Press R for return lensed book or
Press Q for quit the program

Step 3: If user choice ”D”:
The book would be display from the txt file.

Step 4: if user choice “L”:
Step 4: The bookID and Book name will display

Step 4.2: The program will ask the user to Enter the book_ID they want to lend.

Step 4.3: The user has to Enter their Name

Step 4.4: The user has to Enter their Adress

Step 4.5 : The program will ask the user Do you want more books or not.

Step 4.5: If user choice “No”:

Step 4.5.1 The Stock file is updated with the Quantity of the book that is being borrowed.

Step 4.5.1: A suitable message will be displayed, which will say Thank you for lending the book, as well as the title and date of issue.

Step 4.5 2: if user choice “yes”

Step 4.5.2.1: Same process will run again

Step 4.6: If the Book ID does not match, an error message will be displayed.

Step 5 : if user choice “R” :

Step 5.1: The book ID and Book name will display

Step 5.2 : The user has to Enter the number of Days you have borrowed the books for.

Step 5.3 : The user has to Enter the name

Step 5.4: If the book is not returned within 10 days. A specified amount of fine is applied to the overall price.

Step 5.5: A note is formed that contains all of the information saved in the variables.

Step 5..6 : The program will ask Do you have other books to retrun[YES or NO]

Step 5.6.1: if user choice “NO”, A suitable message will displayed asking Book Return is successfully done

Step 5.6.2: if user choice “YES” , The program will run again

Step 5.7 : If the Book ID does not match, an error message will be displayed.

Step 6: if user choice “Q” :

A suitable notice will be displayed, and the program will be terminated.

Flowchart :

A flowchart is a graphical representation of a process, system, or computer algorithm. They are frequently used in a variety of fields to document, analyses, plan, enhance, and convey often complex processes in clear, simple diagrams. (n.d., Lucid chart)


start
if content=="d"
false
if content=="i"
false
if content ==\"r\"
false
if content==q
End
def display_all_books(self):
borrow_book_id
return_book_id
txt= f.read()
print(txt)
borrow_book_id=library_dictionary.key()
borrow_book_id=library_dictionary.key()
Input given to name_of_lender
input given to name_of_lender
Input given to add rece_of_lender
if days>10
if days<10
book_quantity = Quantity-1
fineprice=0
borrow_write.open("BorrowNote.txt","w")
Grand_total= return_total+fineprice
input yes or no
return_write.open("BorrowNote.txt","w")
if asking"==\"yes\"
false
if asking"==\"no\"
true
quantity +1
print"Thanks for lending the book"
if asking"==\"no\"
false
if asking"y"
return Return_book()
true
print("Book return is sucessfully done")

Pseudocode:

Pseudocode is a loose manner of describing programming that does not require any formal programming language syntax or underlying technology considerations. It is used to create a program outline or preliminary draft. Pseudocode summarizes the flow of a program but removes underlying details. System designers generate pseudocode to ensure that programmers grasp the requirements of a software project and align code properly. (Economictimes, undated)

```
CALL Class display:

FUNCTION display_all_books(self):

        DO
        print("------------------List of all bookname , author , quantity and  price ----------
----------")
      with open("List_of_the_books.txt") as f:
         txt = f.read()
      print(txt)
       END DO
END FUNCTION
        Import os
        Import datetime
        Import display



        DELCEAR GLOBAL VARIABLES
         name_of_Books = []
         Total_price = []
         return_price = []
         return_bookname = []
```

```
Class LMS :
CALL  CONSTRUCT __init__(self,name_of_the_books):
DO
INITIALIZE self.name_of_the_books="List_of_the_books.txt"
INITIALIZE self.library_dictionary={}
INITIALIZE book_ID=1
INITIALIZE with open("library.txt","r") as f:
INITIALIZE txt=f.readlines()
FOR  i in txt:
DO
DECLARE content=i.split(",")
self.libary_dictionary.update({str(book_id):{"BookTitle": content[0],
"BookAuthor" : content[1],"Quantity_of_books" : content[2],"Price_of_books" :
content[3]}})
DECLARE book_id=book_id+1
END FOR
END DO
END DO
END CONSTRUCTOR


FUNCTION display_for_lend_and_borrow(self):

DO
print("------------------Book list----------------------------")
print ("BookID","\t","BookTitle")
FOR keys,value in self.liabry_dictionary.items():
print(keys,"\t\t",value.get("BookTitle"))
END FOR
END DO
```

```
END FUNCTION

FUNCTION def lend_book(self):
        DO
     INITIALIZE book_ID=input("Enter the book_ID you wanna lend:  ")
        END DO
IF int(book_id)<0:
        Print(“The book_id is negative”)
ELSE:
  IF book_ID in self.liabry_dictionary.keys():


       IF (int(self.liabry_dictionary[book_ID]["Quantity_of_books"] ) > 0):
            DO
           INITIALIZE name_of_lender= str(input("Enter your name: "))
           INITIALIZE adress_of_lender = input("Enter you adress: ")
           INITIALIZE bookName=self.liabry_dictionary[book_ID]["BookTitle"]
            name_of_BOOKS.append(bookName)
            date_time = datetime.datetime.now()
           INITIALIZE price=int(self.liabry_dictionary[book_ID]["Price_of_books"])
           Total_price.append(price)
           t=0
           FOR price_of_each_book in range(0,len(Total_price)):
                  t=t+Total_price[price_of_each_book]
            END FOR

            DECLARE update_read=open("List_of_the_books.txt","r")
            DECLARE update_write=open("tempory_file.txt","w")

DECLAREQuantity=int(self.liabry_dictionary[book_ID]["Quantity_of_books"])
```

```
            INITIALIZE choose=" "
    WHILE (choose):
        DECLARE choose=update_read.readline()
         INITIALIZE spl=choose.split(",")
         IF len(choose)>0:
                IF (spl[0]== str(self.liabry_dictionary[book_ID]["BookTitle"] )):
                   DECLARE bookname=self.liabry_dictionary[book_ID]["BookTitle"]
                   DECLARE book_author=self.liabry_dictionary[book_ID]["BookAuthor"]
                   DECLARE book_quantity=Quantity-1
                   DECLARE bprice=self.liabry_dictionary[book_ID]["Price_of_books"]
                   WRITE
update_write.write(bookname+","+book_author+","+str(book_quantity)+","+bprice)
               END IF
                 ELSE:
                   update_write.write(choose)
               END ELSE
END IF
            update_read.close()
            update_write.close()
            os.remove("List_of_the_books.txt")
            os.rename("tempory_file.txt","List_of_the_books.txt")
        DECLARE borrow_write=open("BorrowNote.txt","w")
            WRITE borrow.write("--------------------------Notice for lending book----\n")
            WRITE borrow.write("\n")
            WRITE borrow write.write("Name of the person :" +str( name_of_lender)+"\n")
            WRITE borrow.write.write("+++++++++++++++++++++++++++++++++++\n")
            WRITE borrow write.write("Adress_of_lender is :"
+str(address_of_lender)+"\n")
            WRITE borrow_write.write("+++++++++++++++++++++++++++++++++++\n")
            WRITE borrow write.write("Issued date and time is :" +str(idate_time)+"\n")
            WRITE borrow_write.write("+++++++++++++++++++++++++++++++++++\n")
```

```
            WRITE borrow_write.write(Book lend by lender is : “+bookname”)
            WRITE borrow_write.write("================================\n")
            WRITE borrow write.write("Total price of the book you lend is :
        "+str(t)+"$"+"\n")

            WRITE borrow write.write("++++++++++++++++++++++++++++++++++++")
FOR  i in range (len(name_of_BOOKS)):
                WRITE return_write.write(str(name_of_BOOKS[i])+"\n")
END FOR


            return_write.close()
            DECLEARE more=input ("Do you want more Books[Yes or NO]: ")
            DECLEARE asking=more.lower()


IF (asking=="yes"):
                print("Thanks for lending the book "+bookName+"\t on "+str(date_time))
                return self.lend_book()
END IF
            ELIF(asking=="no"):
                print("Thanks for lending the book "+bookName+"\t on "+str(date_time))
            END ELIF
            ELSE:
                print("Error in choosing the value.Please choose carefully")
            END ELSE
END IF
        END IF
        END DO
         ELSE:
            print("The bookName is out of stock")
        END ELSE
      ELSE:
         print("The code you entered is wrong.Please try again")
```

```
        return self.lend_book()
        END ELSE
END IF
END ELSE
END FUNCTION

FUNCTION Return_book(self):

 DECLEARE return_book_ID=input("Please enter the code of the bookName you want to return:")
 DECLEARE Days=int(input("Enter the number of Days you have borrowed the books for:"))
IF int(return_book_ID) <0:
        Print("The book_id you enterd is negative")
ELSE:
IF  return_book_ID in self.liabry_dictionary.keys():
        DECLEARE name_of_lender= str(input("Enter Your Full Name: "))
        DECLEARE bookName=self.liabry_dictionary[return_book_ID]["BookTitle"]
        return_bookName.append(bookName)
        DECLEARE price=int(self.liabry_dictionary[return_book_ID]["Price_of_books"])
        date_time = datetime.datetime.now()
        return_price.append(price)
        INITILIZE reutrn_total=0
        FOR price_of_each_book in range(0,len(return_price)):
            reutrn_total=reutrn_total+return_price[price_of_each_book]
        END FOR
        INITILIZE fineday=0
IF  (Days>10):
            print("Fine is charged because you are late returning the book.")
            DECLEARE fineday=Days-10
            DECLEARE fineprice=(1/10)*reutrn_total*fineday
```

```
        ELIF (Days<=10):
            print("The book was returned on time, and I hope you have a great time.")
            INITILIZE fineprice=0
        ELSE:
            print("The input you have entered is wrong.")
            return self.Return_book()
END IF
END ELIF
END ELSE

      DECLEARE GandTotal=reutrn_total+fineprice

       DECLEARE return_write=open("ReturnNote.txt","w")
        WRITEreturn_write.write("------------------Book Return Notice----------------------\n")
        WRITE return_write.write("\n")
        WRITE return_write.write("Name of the person :" +str( name_of_lender)+"\n")
        WRITE return_write.write("++++++++++++++++++++++++++++++++++++\n")
        WRITE return_write.write("Issued date and time is :" +str(date_time)+"\n")
        WRITE return_write.write("++++++++++++++++++++++\n")
        WRITE return_write.write("Fine for the book is :"+str(fineprice)+"$"+"\n")
        WRITE return_write.write("++++++++++++++++++++++++++++++++++++\n")
        WRITE return_write.write("Total price: "+str(GandTotal)+"$"+"\n")
        WRITE return_write.write("++++++++++++++++++++++++++++++++++++\n")
        WRITE return_write.write("Books Returned by lender is: \n")
        FOR  i in range (len(return_bookName)):
            return_write.write(str(return_bookName[i])+"\n")
        return_write.close()
        END FOR

DECLEARE return_read=open("List_of_the_books.txt","r")
```

```
        DECLEARE return_write=open("tempory_file.txt","w")

DECLEAREQuantity=int(self.liabry_dictionary[return_book_ID]["Quantity_of_books"])
        INITILIZE choose=" "
        WHILE (choose):
           DECLEARE choose=return_read.readline()
           DECLEARE returnsplit=choose.split(",")
           IF  len(choose)>0:
              IF  (returnsplit[0]== str(self.liabry_dictionary[return_book_ID]["BookTitle"] )):
                 DECLEARE
return_book_Name=self.liabry_dictionary[return_book_ID]["BookTitle"]
                 DECLEARE
return_author=self.liabry_dictionary[return_book_ID]["BookAuthor"]
                 DECLEARE return_quantity=Quantity+1
              DECLEARE
return_PRICE=self.liabry_dictionary[return_book_ID]["Price_of_books"]
                 WRITE
return_write.write(return_book_Name+","+return_author+","+str(return_quantity)+","+ret
urn_PRICE)
              ELSE:
                 WRITE return_write.write(choose)
        return_read.close()
        return_write.close()
        os.remove("List_of_the_books.txt")
        os.rename("tempory_file.txt","List_of_the_books.txt")
END IF
END IF
END ELSE
more=input ("Do you have other books to retrun[YES or NO]: ")
        asking=more.lower()
        IF  (asking=="yes"):
           return self.Return_book()
```

```
        ELIF(asking=="no"):
            print("Book Return is sucessfully done\n thank you")
        ELSE:
            print("You made some error while choosing,check it out once")
       END IF
       END ELIF
       END ELSE
END WHILE
     ELSE:
        print("The bookID you are returing is not correct")
        return self.Return_book()
END IF
END ELSE
END IF
END ELSE
END FUNCTION

FUNCTION(main):
while(True):
        print("--------------------------Welcome to Libary Mangement System-----------------------")
        print("-------------------------------------------------------------------------------------------------")
        print("--------------------------Choose press key to perform---------------------------------")
        print("Press D for display Books")
        print("Press L for lend_book books")
        print("Press R for return lended books")
        print("Press Q for quit the program")

        TRY:
            DECLEARE choose=input("Enter the key to choose: ")
```

```
            print()
            DECLEARE content=choose.lower()
            IF  (content=="d"):
              print("You choose to display books")
              CALl constuctor.display_all_books()
            ELIF(content=="l"):
               print("You choose to lend books")
               CALL display.display_all_books(self=

            ELIF(content=="r"):
               print("You choose to return books")
               CALL constuctor.display_for_lend_and_borrow()

              CALL constuctor.Return_book()
            ELIF(content=="q"):

                print("Thnaks for visiting our Libiary. keep coming")
                exit()
            ELSE:
               print("Please Enter the valid press key")
         EXPECT ValueError:
            print("The input is incorrcet.")
            print("Please Enter correct input")
        END IF
        END ELIF
        END ELIF
        END ELIF
        END ELSE
END TRY
END EXPECT
END WHILE
```

END FUNCTION

INITILIZE constuctor=LMS("List_of_the_books.txt")

CALL LMS.MAIN()

Data Structure:

Data structure is the particular way of organizing or managing the data in a computer so that it can be used successfully.

The types of data structures used in the program are as follows:

String:

It is made up of a series of characters that can also include spaces and numerals.

```
++++++++++++++++++++++++++++
book is :"+str(fineprice)+"$"
++++++++++++++++++++++++++++
"+str(GandTotal)+"$"+"\n")
```

Figure 1 : screenshot of showing the use of string.

As we can see in the screenshot above, the amount of fine price and grand total is in int. We converted int to string using str.

Int:

Int, which stands for "integer," is a basic variable type built into the compiler that is used to build numeric variables that carry whole integers.

```
Days=int(input("Enter the number of I
if return book ID in self.liabry dict
```

Figure 2 : screenshot of showing the use of int.

As we can see in the screenshot above, that the days have been defined using the int class.

Dictionary:

Dictionaries are Python's implementation of an associative array, which is a data structure. A dictionary is made up of a set of key-value pairs. Each key-value combination corresponds to a key and its corresponding value. A dictionary is defined by enclosing a comma-separated list of key-value pairs in curly braces (). Each key is separated from its associated value by a colon (:):The values in the dictionary can be duplicates but the keys cannot be duplicates.

```
class LMS():
    """LMS class is made for managing a List_of_the_books
    #Constructor is called and the data in stock file is
    def __init__(self,name_of_the_books):
        self.name_of_the_books="List_of_the_books.txt"
        self.liabry_dictionary={}
```

Figure 3 : screenshot of showing the use of dictionary.

As we can see in the screenshot above, we created a empty dictionary.
The values and keys from the text file are added to the dictionary.

List:

Lists are used to hold a number of elements in a single variable. A list is generated in Python programming by putting all of the items (elements) inside square brackets [], separated by commas.

It can include an unlimited number of elements of various categories (integer, float, string etc.). for example [1,2,3.4,"itesh]

```
name_of_BOOKS=[]
Total_price=[]
return_price=[]
return_bookName=[]
```

Figure 4 : screenshot of showing the use of list.

As soon in above screenshot, To add an element to the list, an empty list is created by declaring the variables name of books, total price, return books, and return price.

Tuples:

A tuple is a collection of ordered and immutable items. Tuples, like lists, are sequences. Tuples and lists differ in that tuple cannot be changed but list can be.

For example:

tuples=(10,20,30,40)

Set:

Sets are collections of elements that are not ordered and always contain unique elements.

Curly brackets, similar to dictionary, are used to represent sets. Sets, on the other hand, can only store unique elements and do not support duplicate elements.

For example:

X = {1,3,7,8,9}

About the program

The program's goal was to develop a library management system. When the code is executed, the program's main interface is displayed. The program then prompts the user to select a key to press in order to complete the task.

```
--------------------------Welcome to Libary Mangement System----------
----------
------------------------------------------------------------------------
-----------------------------
--------------------------Choose press key to perform-----------------
----------
Press D for display Books
Press L for lend_book books
Press R for return lended books
Press Q for quit the program
Enter the key to choose:
```

Figure 5 : Showing the main interface.

After running the code and obtaining the main interface, the user must select the press key. To begin, the user must select d to see all of the books. as well as the price, quantity, and author

```
You choose to display books
------------------List of all bookname , author , quantity and  price -
--------------------
The Hunger Games, Suzanne Collins,28, 20
The Fault In Our Stars, John Green,7, 11
The Notebook, Nicholas Sparks,29, 11
Harry Potter, JK Rowling,29, 2
Start With Why, Simon Sinek, 10, 1.
Programming With Python, John Smith, 20, 1
A Better India:A Better World, Narayana Murthy, 40, 5
A Passage to India, E.M. Foster, 17, 15
A Revenue Stamp, Amrita Pritam, 12, 13
A Woman's Life, Guy de Maupassant,44, 10
```

Figure 6 : displaying books

After the display procedure was completed, the main interface would reappear. If the user chooses to "L" to lend the book. The book ID and title will be displayed, and the user must enter a valid book ID. If the user enters an invalid Book id, an appropriate notice will appear. After entering the right book ID, the program will prompt the user to enter their name and address. The program will then ask the user if they want to lend another book [YES or NO]. If the user selects no, the program will stop saying Thank you for lending the book, as well as your name and the date you borrowed book Similarly, if the user selects "yes," the borrow function will be executed once again. The quantity of the book would be decrease when user borrow the book.

```
You choose to lend books
------------------Book list----------------------------
BookID   BookTitle
1001             The Hunger Games
1002             The Fault In Our Stars
1003             The Notebook
1004             Harry Potter
1005             Start With Why
1006             Programming With Python
1007             A Better India:A Better World
1008             A Passage to India
1009             A Revenue Stamp
1010             A Woman's Life
Enter the book_ID you wanna lend:  1001
Enter your name: Itesh niroula
Enter you adress: damak jhapa
Do you want more Books[Yes or NO]: no
Thanks for lending the book The Hunger Games     on 2021-09-09 12:46:03.979577
```

Figure 7: complete borrow process.

As previously said, the main interface will reappear after the entire lending procedure is completed. If the user selects "r," the book ID and title will appear, and the program will prompt the user to enter the book Id you wish to return, the days we kept the book, and the name of the lender. If the number of days is less than 10, no additional fee is applied to the price; however, if the number of days exceeds 10, an additional fine is applied to the price. The application will then ask the user [YES or NO] if they wish to lend another book. If the user selects no, a notification stating that the book return was successful will appear. Similarly, if the user selects "yes," the return function is called once again. The quantity of the book will increase when user return the book.

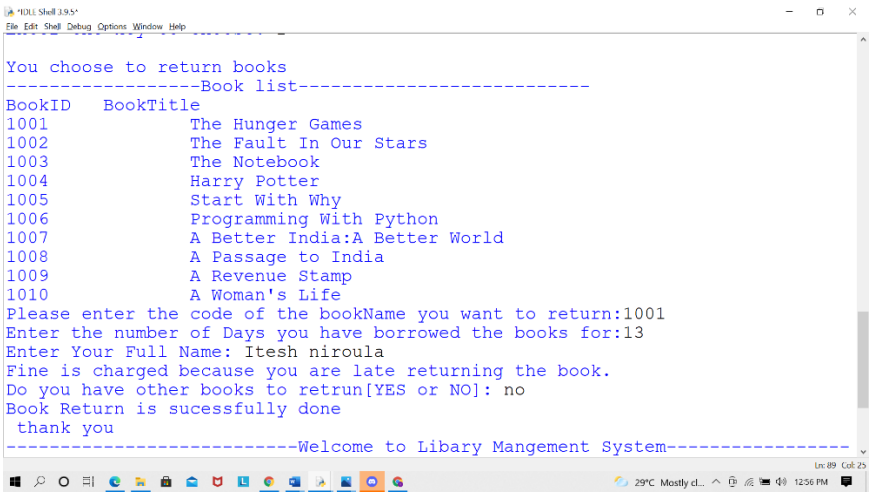

Figure 8: Complete return process.

```
Name of the person :Itesh niroula
++++++++++++++++++++++++++++++++++++++++++++++++++
Adress of lender is: damak jhapa
++++++++++++++++++++++++++++++++++++++++++++++++++
Issued date and time is :2021-09-09 12:46:03.979577
++++++++++++++++++++++++++++++++++++++++++++++++++
==========================================================
Total price of the book you lend is : 20$
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++The Hunger Games
```

Figure 9: BorrowNote.

ReturnNote - Notepad

File Edit Format View Help

```
--------------------------Book Return Notice-------------------------------------------

Name of the person :Itesh niroula
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Issued date and time is :2021-09-09 12:52:35.013708
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Fine for the book is :6.0$
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Total price: 26.0$
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Books Returned by lender is:
The Hunger Games
```

Figure 10 : ReturnNote.

Similarly, After finishing the return and borrow process, the user can close the program by selection “Q”. And a suitable message will display in the terminal.

```
Press D for display Books
Press L for lend_book books
Press R for return lended books
Press Q for quit the program
Enter the key to choose: q

Thnaks for visiting our Libiary. keep coming
```

Figure 11: showing the termination of program as well after selecting an option q

Testing:

Test 1: Show implementation of try, except

Objective	To Show Implementation of try, except
Action	1. Run the program 2. L was selected as press key 3. Z was return as input
Expected Results	Exception will be occurred.

Actual Results	Exception occurred successfully.
Conclusion	Test is done successfully

Table 1: Test 1.

```
Press D for display Books
Press L for lend_book books
Press R for return lended books
Press Q for quit the program
Enter the key to choose: Z

Please Enter the valid press key
```

Figure 12: showing try, expect

Test 2: Selection borrow and return option

Test 2. 1: Provide negative value as input

Objective	To provide negative value as input , show selection borrow and return option.
Action	1. Run the program 2. Select L as key press 3. Return negative value -1001
Expected Results	Suitable message will be appear saying the book id you entered is negative.
Actual Results	Suitable message appeared saying the book id you entered is negative.
Conclusion	Test is done successfully

Test 2.1 : Test 2

```
You choose to lend books
------------------Book list----------------------------
BookID   BookTitle
1001             The Hunger Games
1002             The Fault In Our Stars
1003             The Notebook
1004             Harry Potter
1005             Start With Why
1006             Programming With Python
1007             A Better India:A Better World
1008             A Passage to India
1009             A Revenue Stamp
1010             A Woman's Life
Enter the book_ID you wanna lend:  -1001
the book id you entered is negative.
```

Figure 13 : Providing negative value as input

Test 2.2 : Provide nonexistent value as input

Objective	To provide non existent as input , show selection borrow and return option.
Action	1. Run the program 2. Select L as key press 3. Return non existent value 56
Expected Results	Suitable message will be appear saying The code you entered is wrong.Please try again
Actual Results	Suitable message appeared saying The code you entered is wrong.Please try again
Conclusion	Test is done successfully

Test 2.2: Test 2

```
You choose to lend books
------------------Book list--------------------------
BookID   BookTitle
1001              The Hunger Games
1002              The Fault In Our Stars
1003              The Notebook
1004              Harry Potter
1005              Start With Why
1006              Programming With Python
1007              A Better India:A Better World
1008              A Passage to India
1009              A Revenue Stamp
1010              A Woman's Life
Enter the book_ID you wanna lend:  56
The code you entered is wrong.Please try again
```

Figure 14 : Providing nonexistent value as input

Test 3: To show the File generation of borrow

Show complete borrow process
Show output in the shell as well
Finally show the borrow note in txt file

Objective	To show the complete borrow process and output in the shell and finally the borrow note in txt file.
Action	1. Run the program 2. Select L as key press 3. 1010 as book ID 4. Itesh Niroula as name 5. Damak jhapa as address 6. Only one book is selected 7. Suitable note will display in terminal
Expected Results	A Text File with the book information would be generated.
Actual Results	A Text File with the book information generated.
Conclusion	Test is done successfully

Test 3: Test 3

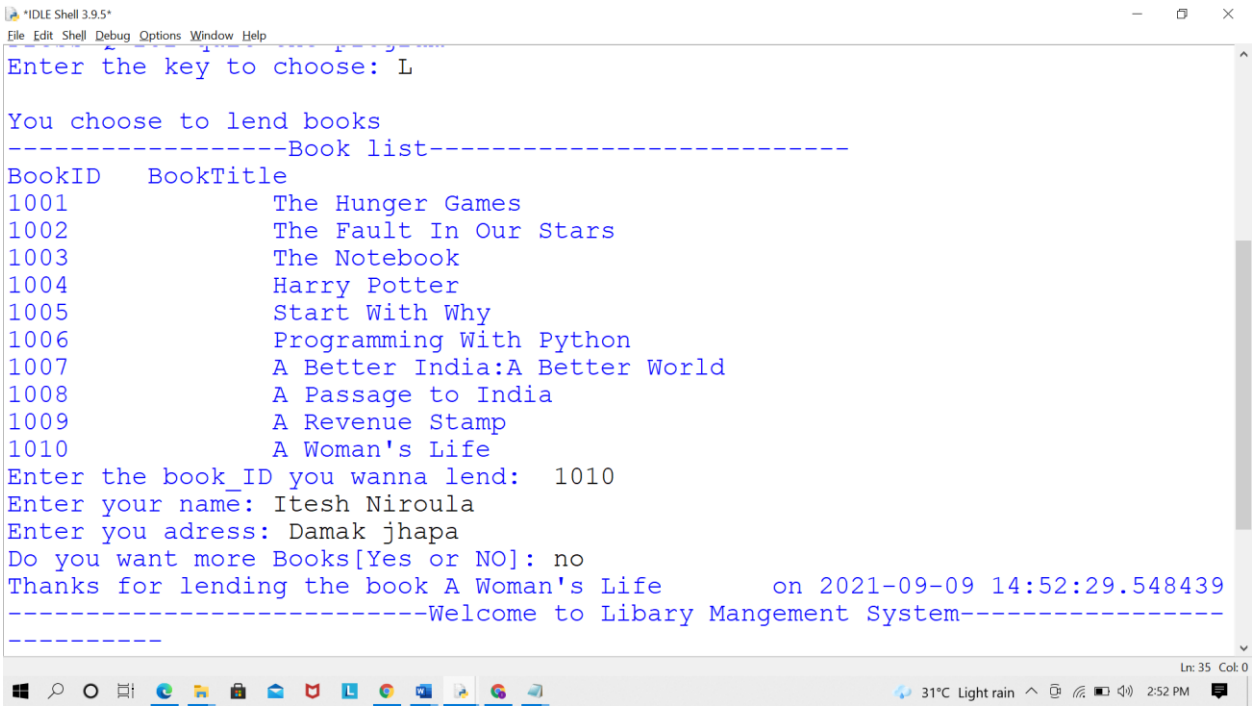

Figure 15 : screen shot of all borrow process in terminal.

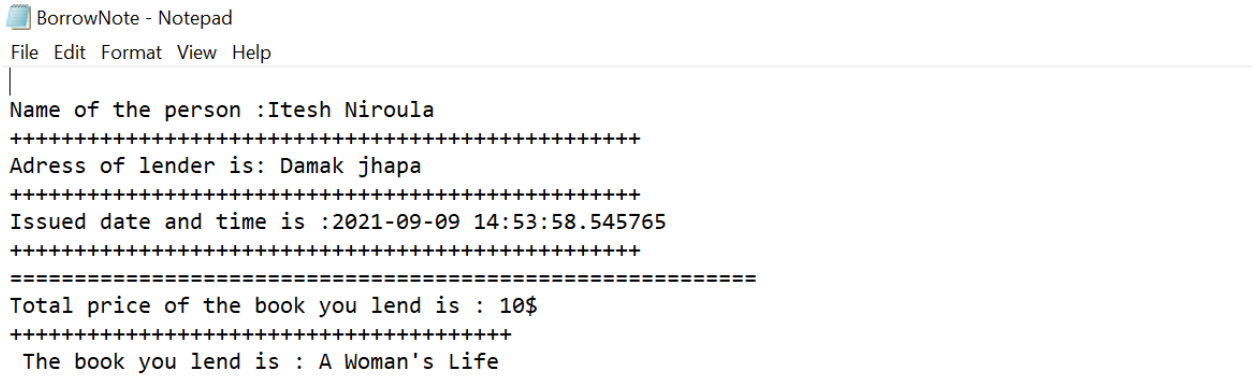

Figure 16 : screenshot of borrow note in text file.

Test 4 : File generation of return

Objective	To show the complete return process and output in the shell and finally the return note in txt file.
Action	1. Run the program 2. Select r as key press 3. 1010 as book ID 4. 13 as a day 5. Itesh Niroula as name 6. Only one book is selected 7. Suitable note will display in terminal
Expected Results	A Text File with the book information would be generated.
Actual Results	A Text File with the book information generated.
Conclusion	Test is done successfully

Test 4 : Test 4

```
------------------Book list--------------------------
BookID   BookTitle
1001             The Hunger Games
1002             The Fault In Our Stars
1003             The Notebook
1004             Harry Potter
1005             Start With Why
1006             Programming With Python
1007             A Better India:A Better World
1008             A Passage to India
1009             A Revenue Stamp
1010             A Woman's Life
Please enter the code of the bookName you want to return:1010
Enter the number of Days you have borrowed the books for:13
Enter Your Full Name: Itesh Niroula
Fine is charged because you are late returning the book.
Do you have other books to retrun[YES or NO]: no
Book Return is sucessfully done
 thank you
```

Figure 17 : screen shot of all return process in terminal.

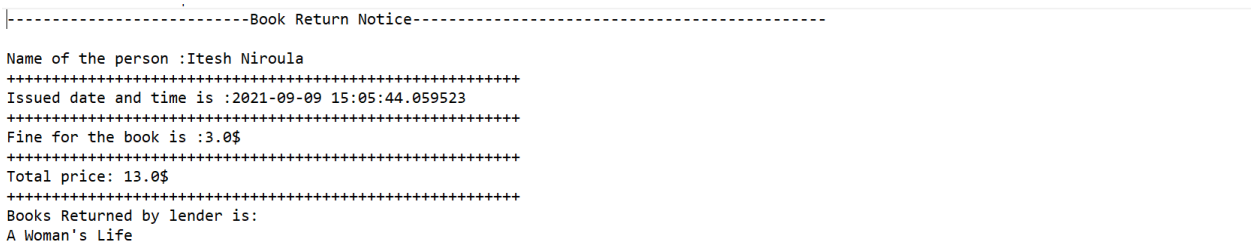

Figure 18 : screenshot of borrow note in text file.

Test 5 : To show the update in stock file

Test 5.1 : To show the quantity being deducted while borrowing the book.

Objective	To show the quantity being deducted while borrowing the book.
Action	1. Run the program 2. Select l as key press 3. 1001 as book ID 4. Itesh Niroula as name 5. Damak jhapa as adress 6. Only one book is selected
Expected Results	The quantity of the book_ID would be decreased by 1
Actual Results	The quantity of the book_ID decreased by 1
Conclusion	Test is done successfully

Test 5.1 : Test 5.1

List_of_the_books - Notepad

File Edit Format View Help

```
The Hunger Games, Suzanne Collins,28, 20
The Fault In Our Stars, John Green,7, 11
The Notebook, Nicholas Sparks,29, 11
Harry Potter, JK Rowling,29, 2
Start With Why, Simon Sinek, 10, 1.
Programming With Python, John Smith, 20, 1
A Better India:A Better World, Narayana Murthy, 40, 5
A Passage to India, E.M. Foster, 17, 15
A Revenue Stamp, Amrita Pritam, 12, 13
A Woman's Life, Guy de Maupassant,44, 10
```

Figure 19: screenshot before borrowing the book.

IDLE Shell 3.9.5

File Edit Shell Debug Options Window Help

```
You choose to lend books
------------------Book list----------------------------
BookID   BookTitle
1001             The Hunger Games
1002             The Fault In Our Stars
1003             The Notebook
1004             Harry Potter
1005             Start With Why
1006             Programming With Python
1007             A Better India:A Better World
1008             A Passage to India
1009             A Revenue Stamp
1010             A Woman's Life
Enter the book_ID you wanna lend:  1001
Enter your name: Itesh Niroula
Enter you adress: damak jhapa
Do you want more Books[Yes or NO]: no
Thanks for lending the book The Hunger Games     on 2021-09-09 15:18:32.890051
```

Figure 19: screenshot of complete borrowing the book.

List_of_the_books - Notepad

File Edit Format View Help

```
The Hunger Games, Suzanne Collins,27, 20
The Fault In Our Stars, John Green,7, 11
The Notebook, Nicholas Sparks,29, 11
Harry Potter, JK Rowling,29, 2
Start With Why, Simon Sinek, 10, 1.
Programming With Python, John Smith, 20, 1
A Better India:A Better World, Narayana Murthy, 40, 5
A Passage to India, E.M. Foster, 17, 15
A Revenue Stamp, Amrita Pritam, 12, 13
A Woman's Life, Guy de Maupassant,44, 10
```

Figure 20: screenshot after borrowing the book.

Objective	To show the quantity being added while returing the book.
Action	1. Run the program 2. Select r as key press 3. 1001 as book ID 4. 9 as days 5. Itesh Niroula as name 6. Only one book is selected
Expected Results	The quantity of the book_ID would be increased by 1
Actual Results	The quantity of the book_ID increased by 1
Conclusion	Test is done successfully

Test 5.2 : Test 5.2

```
You choose to return books
------------------Book list----------------------------
BookID   BookTitle
1001             The Hunger Games
1002             The Fault In Our Stars
1003             The Notebook
1004             Harry Potter
1005             Start With Why
1006             Programming With Python
1007             A Better India:A Better World
1008             A Passage to India
1009             A Revenue Stamp
1010             A Woman's Life
Please enter the code of the bookName you want to return:1001
Enter the number of Days you have borrowed the books for:9
Enter Your Full Name: Itesh Niroula
The book was returned on time, and I hope you have a great time.
Do you have other books to retrun[YES or NO]: no
Book Return is sucessfully done
 thank you
```

Figure 21: screenshot of complete returning the book.

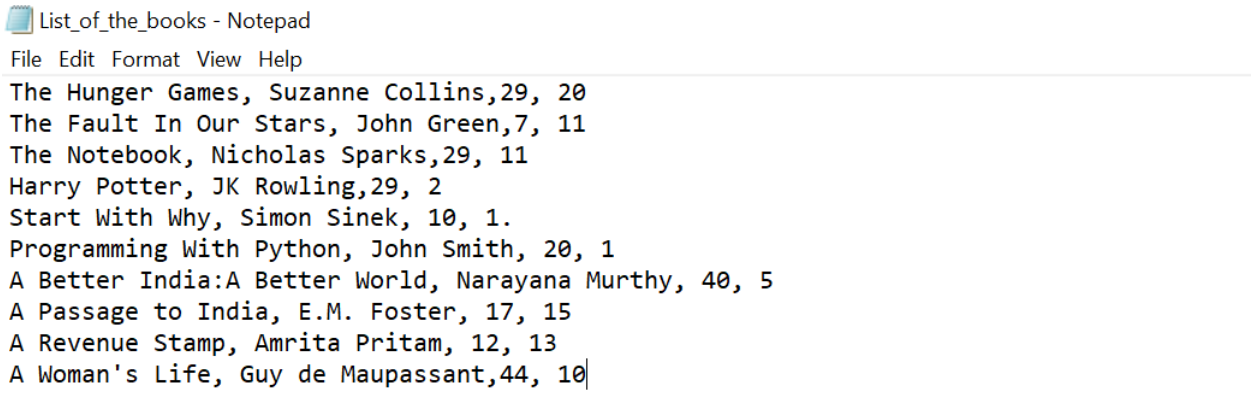

Figure 22: screenshot of returning the book after borrowing.

Conclusion:

This document is a step forward in the development of software that can display a book's cover and return it to the library. I wrote the algorithm and pseudocode for the specified software, as well as the flowchart for the specific program. Pseudocode is designed for nonprogrammers; thus, it has a simple structure. As a result, it has become much easier for the person to recognize the software.

Such courses will surely be quite beneficial in creating a career as a professional programmer. And I was getting assistance from my teacher and seniors. I put forth a lot of effort and gave it my all to finish this.

Overall, I believe that completing this course work was a pleasure for me because I learnt something new, and programming is more than simply writing code; it is also about logical thinking and knowledge of code operations.

Appendix:

Appendix for display:

```
class Display():
    def display_all_books(self):
        print("------------------List of all bookname , author , quantity and  price ----------
--------")
        with open("List_of_the_books.txt") as f:
            txt = f.read()
        print(txt)
```

Appendix for LMS:

```
import os
import datetime
import display



name_of_BOOKS=[]    # creating empty list
Total_price=[]
return_price=[]
return_bookName=[]
# creating class
class LMS(display.Display): # using interface


    def __init__(self,name_of_the_books):  # creating constructor
        self.name_of_the_books="List_of_the_books.txt"
        self.liabry_dictionary={}              # creating empty dictionary to store
booname,bookauthor,quantity,price.
        book_id=1001  # declearing Id for unique book_id
        with open("List_of_the_books.txt","r") as f:   # opening txt file
            txt=f.readlines()  # reading text file and readlines convert txt in list
        for i in txt:
```

```
            content=i.split(",") # spliting txt by comma
            self.liabry_dictionary.update({str(book_id):{"BookTitle": content[0], "BookAuthor" :
content[1],"Quantity_of_books" : content[2],"Price_of_books" : content[3]}})
            book_id=book_id+1




    def display_for_lend_and_borrow(self):   ## displaying book for lend and borrow
        print("------------------Book list--------------------------")
        print ("BookID","\t","BookTitle")
        for keys,value in self.liabry_dictionary.items():
            print(keys,"\t\t",value.get("BookTitle"))



    def lend_book(self):  # creating function to lend books
        book_ID=input("Enter the book_ID you wanna lend:  ")
        if int(book_ID) <0:  # checking whether the book id is negative or not
            print("The book_id you entered is negative")
        else:
            if book_ID in self.liabry_dictionary.keys():
                if (int(self.liabry_dictionary[book_ID]["Quantity_of_books"] ) > 0):
                    name_of_lender= str(input("Enter your name: "))
                    adress_of_lender = input("Enter you adress: ")
                    bookName=self.liabry_dictionary[book_ID]["BookTitle"]
                    name_of_BOOKS.append(bookName)
                    date_time = datetime.datetime.now()
                    price=int(self.liabry_dictionary[book_ID]["Price_of_books"])
                    Total_price.append(price)
```

```
            t=0
            for price_of_each_book in range(0,len(Total_price)):
                t=t+Total_price[price_of_each_book]



            update_read=open("List_of_the_books.txt","r")
            update_write=open("tempory_file.txt","w")
            Quantity=int(self.liabry_dictionary[book_ID]["Quantity_of_books"])
            choose=" "
            while(choose):
                choose=update_read.readline()
                spl=choose.split(",")
                if len(choose)>0:
                    if (spl[0]== str(self.liabry_dictionary[book_ID]["BookTitle"] )):
                        bookname=self.liabry_dictionary[book_ID]["BookTitle"]
                        book_author=self.liabry_dictionary[book_ID]["BookAuthor"]
                        book_quantity=Quantity-1
                        bprice=self.liabry_dictionary[book_ID]["Price_of_books"]

update_write.write(bookname+","+book_author+","+str(book_quantity)+","+bprice)
                    else:
                        update_write.write(choose)
            update_read.close()
            update_write.close()
            os.remove("List_of_the_books.txt")
            os.rename("tempory_file.txt","List_of_the_books.txt")
        #A Borrow Note is provided after borrowing the books

            borrow_write=open("BorrowNote.txt","w")
            borrow_write.write("\n")
```

```
            borrow_write.write("Name of the person :" +str( name_of_lender)+"\n")

borrow_write.write("++++++++++++++++++++++++++++++++++++++++++++++++++++\n")
            borrow_write.write("Adress of lender is: "+ str(adress_of_lender)+"\n")

borrow_write.write("++++++++++++++++++++++++++++++++++++++++++++++++++++\n")
            borrow_write.write("Issued date and time is :" + str(date_time)+"\n")

borrow_write.write("++++++++++++++++++++++++++++++++++++++++++++++++++++\n")

borrow_write.write("==================================================
=======\n")
            borrow_write.write("Total price of the book you lend is : "+str(t)+"$"+"\n")
            borrow_write.write("++++++++++++++++++++++++++++++++++++\n
The book you lend is : ")
            for i in range (len(name_of_BOOKS)):
                borrow_write.write(str(name_of_BOOKS[i])+"\n")
            borrow_write.close()
            more=input ("Do you want more Books[Yes or NO]: ")
            asking=more.lower()
            if (asking=="yes"):
                print("Thanks for lending the book "+bookName+"\t on "+str(date_time))
                return self.lend_book()
            elif(asking=="no"):
                print("Thanks for lending the book "+bookName+"\t on "+str(date_time))
            else:
                print("Error in choosing the value.Please choose carefully")


        else:
            print("The bookName is out of stock")
    else:
```

```
            print("The code you entered is wrong.Please try again")
            return self.lend_book()


# creating function to perform return_book

    def Return_book(self):
        return_book_ID=input("Please enter the code of the bookName you want to return:")
        Days=int(input("Enter the number of Days you have borrowed the books for:"))
        if int(return_book_ID)<0:
            print("The book_id you entered is negative")
        else:
            if return_book_ID in self.liabry_dictionary.keys():
                name_of_lender= str(input("Enter Your Full Name: "))
                bookName=self.liabry_dictionary[return_book_ID]["BookTitle"]
                return_bookName.append(bookName)
                price=int(self.liabry_dictionary[return_book_ID]["Price_of_books"])
                date_time = datetime.datetime.now()
                return_price.append(price)
                reutrn_total=0
                for price_of_each_book in range(0,len(return_price)):
                    reutrn_total=reutrn_total+return_price[price_of_each_book]
                # for fine
                fineday=0
                if (Days>10):
                    print("Fine is charged because you are late returning the book.")
                    fineday=Days-10
                    fineprice=(1/10)*reutrn_total*fineday
                elif (Days<=10):
                    print("The book was returned on time, and I hope you have a great time.")
                    fineprice=0
```

```
            else:
                print("The input you have entered is wrong.")
                return self.Return_book()

            GandTotal=reutrn_total+fineprice

            return_write=open("ReturnNote.txt","w")
            return_write.write("-------------------------Book Return Notice----------------------
--------------------\n")
            return_write.write("\n")
            return_write.write("Name of the person :" +str( name_of_lender)+"\n")

return_write.write("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++\n")
            return_write.write("Issued date and time is :" +str(date_time)+"\n")

return_write.write("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++\n")
            return_write.write("Fine for the book is :"+str(fineprice)+"$"+"\n")

return_write.write("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++\n")
            return_write.write("Total price: "+str(GandTotal)+"$"+"\n")

return_write.write("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++\n")
            return_write.write("Books Returned by lender is: \n")
            for i in range (len(return_bookName)):
                return_write.write(str(return_bookName[i])+"\n")
            return_write.close()
  ## updating the quantity
```

```
            return_read=open("List_of_the_books.txt","r")
            return_write=open("tempory_file.txt","w")
            Quantity=int(self.liabry_dictionary[return_book_ID]["Quantity_of_books"])
            choose=" "
            while(choose):
                choose=return_read.readline()
                returnsplit=choose.split(",")
                if len(choose)>0:
                    if  (returnsplit[0]==  str(self.liabry_dictionary[return_book_ID]["BookTitle"]
)):
                        return_book_Name=self.liabry_dictionary[return_book_ID]["BookTitle"]
                        return_author=self.liabry_dictionary[return_book_ID]["BookAuthor"]
                        return_quantity=Quantity+1

return_PRICE=self.liabry_dictionary[return_book_ID]["Price_of_books"]

return_write.write(return_book_Name+","+return_author+","+str(return_quantity)+","+ret
urn_PRICE)
                    else:
                        return_write.write(choose)
            return_read.close()
            return_write.close()
            os.remove("List_of_the_books.txt")
            os.rename("tempory_file.txt","List_of_the_books.txt")

            more=input ("Do you have other books to retrun[YES or NO]: ")
            asking=more.lower()
            if (asking=="yes"):
                return self.Return_book()
            elif(asking=="no"):
                print("Book Return is sucessfully done\n thank you")
```

```
            else:
                print("You made some error while choosing,check it out once")
        else:
            print("The bookID you are returing is not correct")
            return self.Return_book()

# main function
    def main():
        while(True):
            print("--------------------------Welcome to Libary Mangement System----------------------")
            print("-------------------------------------------------------------------------------------------------------")
            print("--------------------------Choose press key to perform---------------------------------")
            print("Press D for display Books")
            print("Press L for lend_book books")
            print("Press R for return lended books")
            print("Press Q for quit the program")
     # try and expept
            try:
                choose=input("Enter the key to choose: ")
                print()
                content=choose.lower()
                if (content=="d"):
                   print("You choose to display books")
                   display.Display.display_all_books(self=display.Display)
                elif(content=="l"):
                    print("You choose to lend books")
                    constuctor.display_for_lend_and_borrow()
                    constuctor.lend_book()
```

```
            elif (content=="r"):
                print("You choose to return books")
                constuctor.display_for_lend_and_borrow()

                constuctor.Return_book()
            elif(content=="q"):

                print("Thnaks for visiting our Libiary. keep coming")
                exit()
            else:
                print("Please Enter the valid press key")
        except ValueError:
            print("The input is incorrcet.")

            print("Please Enter correct input")
# calling constuctor
constuctor=LMS("List_of_the_books.txt")



# call main function
LMS.main()
```

References:

https://www.w3schools.com/python/

https://www.tutorialspoint.com/python/index.htm