

**Module Code & Module Title**

**CS4051NI Fundamentals of Computing**

**Assessment Weightage & Type**

**60% Individual Coursework**

**Year and Semester**

**2020-21 Autumn**

**Student Name:** Bibas Bantawa Rai

**Group:** C13

**London Met ID:** 20049402

**College ID:** NP01CP4S210180

**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.*

**Contents**

[**Introduction** 1](#_Toc82131656)

[**Algorithms** 2](#_Toc82131657)

[**Flow Chart** 5](#_Toc82131658)

[**Pseudocode** 7](#_Toc82131659)

[**Program** 18](#_Toc82131660)

[**Testing** 30](#_Toc82131661)

[**Conclusion** 40](#_Toc82131662)

**Table of Figures**

[Figure 1 Flow chart part 1 5](#_Toc82131695)

[Figure 2 flow chart part 2 6](#_Toc82131696)

[Figure 3 Home menu 18](#_Toc82131697)

[Figure 4 entering wrong choice 18](#_Toc82131698)

[Figure 5 input value as 1 19](#_Toc82131699)

[Figure 6 quitting the program by input as q 19](#_Toc82131700)

[Figure 7 continue to other options by input as c 19](#_Toc82131701)

[Figure 8 borrowing a book by providing a correct value 20](#_Toc82131702)

[Figure 9 books quantity before borrowing 20](#_Toc82131703)

[Figure 10 books quantity after borrowing book Jane Eyre 20](#_Toc82131704)

[Figure 11 when issuing a book to borrow when its quantity is 0 21](#_Toc82131705)

[Figure 12 borrowing book Anna Karenina 21](#_Toc82131706)

[Figure 13 Trying to borrow the same book 21](#_Toc82131707)

[Figure 14 borrowing book Beloved 22](#_Toc82131708)

[Figure 15 Borrowing book The Great Gatsby 22](#_Toc82131709)

[Figure 16 borrowing the book Don Quixote 22](#_Toc82131710)

[Figure 17 Issuing a book not available in the library 23](#_Toc82131711)

[Figure 18 Returning a book without correct value 23](#_Toc82131712)

[Figure 19 Returning book Don Quixote 23](#_Toc82131713)

[Figure 20 Returning book Beloved 24](#_Toc82131714)

[Figure 21 Returning book Anna Karenina 24](#_Toc82131715)

[Figure 22 Returning book The Great Gatsby 24](#_Toc82131716)

[Figure 23 Returning book Jane Eyre 25](#_Toc82131717)

[Figure 24 Returning book To Kill a Mockingbird 25](#_Toc82131718)

[Figure 25 When paying fine entering incorrect values 26](#_Toc82131719)

[Figure 26 Paying a fine without crossing a deadline 26](#_Toc82131720)

[Figure 27 Input value as 5 and find book by book name 27](#_Toc82131721)

[Figure 28 Book name Anna Karenina input 27](#_Toc82131722)

[Figure 29 Choices given yes no 27](#_Toc82131723)

[Figure 30 Choosing choice yes 28](#_Toc82131724)

[Figure 31 Choosing choice 5 to search book by author name 28](#_Toc82131725)

[Figure 32 entering choice as no 28](#_Toc82131726)

[Figure 33 Input choice as no 29](#_Toc82131727)

[Figure 34 running the program 30](#_Toc82131728)

[Figure 35 entering value as negative one 31](#_Toc82131729)

[Figure 36 a message saying enter what you want from above 31](#_Toc82131730)

[Figure 37 message saying the issued book is not in the library. 33](#_Toc82131731)

[Figure 38 entering invalid values when returning books 34](#_Toc82131732)

[Figure 39 message of successful book borrowed 35](#_Toc82131733)

[Figure 40 book returned message 36](#_Toc82131734)

[Figure 41 quantity of books before borrowing 37](#_Toc82131735)

[Figure 42 quantity of books after borrowing 38](#_Toc82131736)

[Figure 43 Quantity of books before returning it 39](#_Toc82131737)

[Figure 44 Quantity of book after returning it 39](#_Toc82131738)

# 

# **Introduction**

Python is a popular high-level programming language for general-purpose applications. Guido van Rossum established it in 1991, and the Python Software Foundation continues to develop it. Its syntax was created with code readability in mind, allowing programmers to communicate their ideas in fewer lines of code.

The project given was to make a library management system. The library management system is a mini project that should be done via object-oriented programming. The project made should be able to carry out some of the given commands such as: When a customer or a member of the library decides to borrow the and take it home appropriate messages should be shown. The books borrowed should be later returned and if certain problem occur and the person is not able to return it under the given dead line, appropriate fine should be paid by the person. The goals and objectives of the project is quite simple. The main goal of the project is to make it user friendly and the objectives are to make the system usable so that borrowing, returning and paying fine for the book would go smoothly without any interruptions.

Some of the platforms that were used to make the project are as follows:

1. IDLE: IDLE is a python dedicated platform. This platform has the ability to be coded in 100% pure python. It accesses other various python modules such as tkinter which is used as a GUI toolkit. IDLE is cross platformed which makes itself be usable in any operating software’s such as Windows, Unix and macOS. Python has other various features as well. Such as colorizing of codes as input, output and error messages. IDLE is used by those who have started to learn python programming as a beginner.
2. MS Word: MS word is a word processing application. It was used in the project to write the report and document the created application. Various tasks that are done on the project is mentioned on the report.
3. Draw.io: Draw.io is an online tool for creating diagrams and other planning’s as well. It was specifically used for the purpose of creating class diagrams and flow charts for this program.

# **Algorithms**

A procedure or set of rules to be followed in calculations or other problem-solving operations, says Algorithm. As a result, an algorithm is a set of rules/instructions that govern how a work is to be conducted step-by-step in order to achieve the desired results. Below is the algorithm of the action performed in steps.

Step 1: Display name of the application as “Welcome to study club library club” and also enter which choice to be made.

Step 2: Start a main loop.

Step 3: Ask the user to enter numbers 1,2,3,4,5,6 to display, borrow, return, pay, search by name, search by author respectively.

Step 4: Verify to enter proper value.

Step 5: When the user enters 1 display all the available books with their information from test.py.

Step 6: Ask the user to enter q or c to continue to main menu or quit accordingly.

Step 7: Ask the user to enter correct value.

Step 8: If the user enters c continue and go to main menu and go to Step 3.

Step 9: If the user enters 2 go to borrow choice.

Step 10: Ask the user to input correct book name and member name.

Step 11: Then if the quantity of books is 0 display message as desired book is out of stock.

Step 12: If the borrowed book is already issued display message as desired book is already issued and go to Step 7.

Step 13: Get the book to borrow and add it to the members borrow list.

Step 14: Enter c to continue to end borrow choice and go to main menu.

Step 15: If the user inputs value as 3 go to return choice.

Step 16: Ask the user to input correct values.

Step 17: If correct proceed else display message please enter correct username and book name.

Step 18: If book is returned display message book is returned.

Step 19: Remove the returned book from the members borrow list.

Step 20: If user inputs c continue go to Step 3.

Step 21: If the user inputs 4 go to pay fine choice.

Step 22: Enter correct value for the name of the book and borrower name.

Step 23: If the book borrowed crosses 10 days go to check fine and display message the fine to be paid accordingly.

Step 24: Ask if they want to pay the fine.

Step 25: If the user input yes count the number of delay days and multiply it with 10 then, display message as payment is being processed and also successful.

Step 26: Add the returned book to the quantity.

Step 27: If choice is no display message redirecting to home.

Step 28: If there is no fine show message as no fine record to be paid.

Step 29: If the value is incorrect or doesn’t match show message as enter correct information and go to main menu and go to Step 4.

Step 30: If the user inputs 5 go to find book by name choice.

Step 31: Ask to enter name of book.

Step 32: If the value is correct search the book and display message book is being searched.

Step 33: If the value is not correct display message as enter book name available in our library.

Step 34: Go to Step 32.

Step 35: Display the information of the searched book.

Step 36: Ask the user yes or no if they want to proceed.

Step 37: If yes ask the user to continue or not.

Step 38 If no terminate program.

Step 39: Go to Step 3.

Step 10: Ask the user to enter available values.

Step 41: If the user inputs 6 go to find book by author name option.

Step 42: Ask to enter name of author.

Step 43: If the value is correct search the book and display message book is being searched.

Step 44: If the value is not correct display message as enter author name available in our library.

Step 45: Go to Step 43.

Step 46: Display the information of the searched author.

Step 47: Ask the user yes or no if they want to proceed.

Step 48: If yes ask the user to continue or not.

Step 49: If no terminate program.

# **Flow Chart**

A flowchart is a graphical depiction of a series of steps. It is commonly used to present the flow of algorithms, workflows, or processes since it shows steps in a sequential manner. A flowchart often depicts the processes as various types of boxes, with arrows linking them in the proper order. The flow chart of the project is shown as whole below:

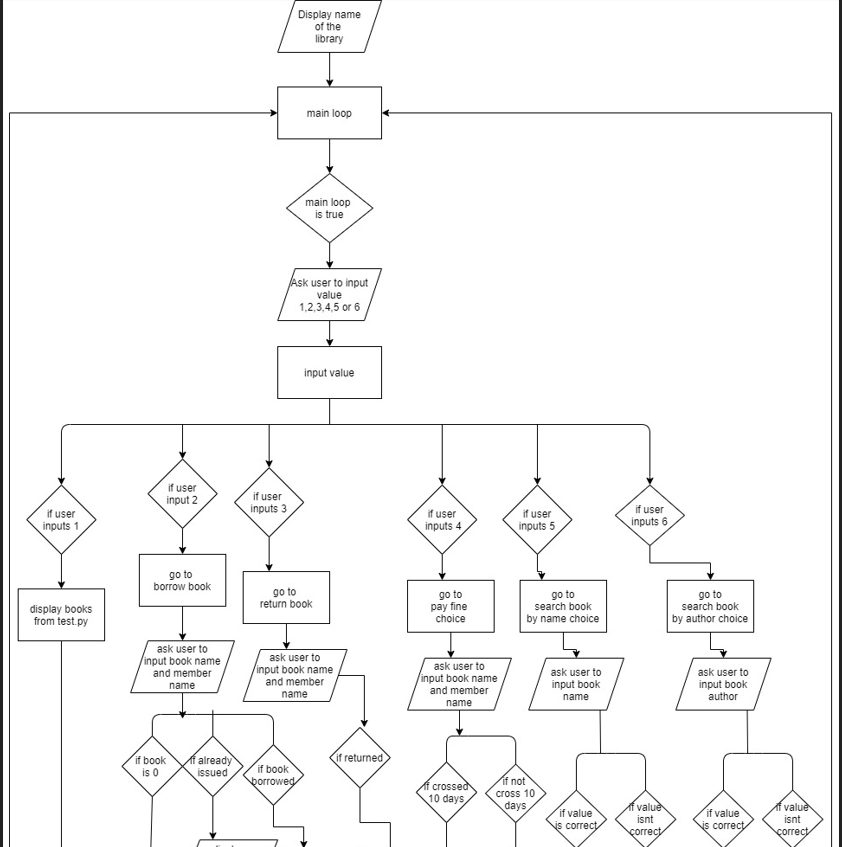


Figure 1 Flow chart part 1



Figure 2 flow chart part 2

# **Pseudocode**

Pseudocode is basically a representation of the implementation of an algorithm. In many instances the algorithms are represented to the programmers with the help of pseudo code since they can be understood by programmers whatever their programming background or knowledge is. Pseudo code improves the readability of any approach so it can be used as starting point for implementation of an algorithm. It uses simple English words so that it is for humans to look into and understand and not for the computer, It cannot be compiled by a computer. The pseudo code can also be considered as a bridge between an algorithm and actual implementation of the algorithm in the program. The pseudo code can also be used as documentation of the feature created by a programmer so that other programmer can easily understand the implementation without having to look into the code itself. It also helps programmer understand the flow of the program from a top-level view.

Pseudocode for the project Library Management System.

1. Module: catalog

**START**

**CREATE** catalog

**DEFINE** searchbyname.

**WHILE** mainloop is True

**DO**

**FOR IF** book\_name in library.count\_book\_dict.keys()

**PRINT** The book is being searched.

**PRINT** \n

**PRINT** book name, quantity, author name, rach number, pages

**PRINT** \n

**PRINT**  choose what you wan to proceed

**CHOICE**  input yes or no

**IF** input is yes

**BREAK**

**ELSE**

**PRINT** system exit

**END IF**

**ELSE PRINT** enter the book name available in our library

**END IF**

**END DO**

**DEFINE** searchbyauthor

**WHILE** mainloop is true

**DO**

**IF** value of author is 1

**PRINT** The book is being searched.

**PRINT** \n

**PRINT** book name, quantity, author name, rach number, pages

**PRINT** \n

**PRINT**  choose what you wan to proceed

**CHOICE**  input yes or no

**IF** input is yes

**BREAK**

**ELSE**

**PRINT** system exit

**END IF**

**ELSE PRINT** enter the book present in our library

**END IF**

**END DO**

1. Module: library

**START**

**CREATE** library

**CREATE** function init(self,count\_book,name)

self.count\_book\_dict = count\_book

self.name = name

self.bookdict = {}

self.max\_book\_count = {}

self.count\_book\_dict

**CREATE** function display\_books(self)

**DO**

**FOR** books in self.count\_book\_dict\_items()

**PRINT** books

**END DO**

1. Module: Test.py

**START**

**INITIALIZE** library books information as ({"Anna Karenina":[6, "Leo Tolsty", "12H", 1878, 510], "To Kill a Mockingbird":[3, "Harper Lee", "12H", 1960, 475], "The Great Gatsby":[5, "F.Scott Fitzgerald", "12H", 1920, 337], "Don Quixote":[2, "Miguel de Cervantes", "12H", 1615, 337], "Beloved":[3, "Toni Morrison", "12H", 1987, 437],"Jane Eyre":[1, "Charlotte Bronte", "12H", 1847, 427]},"Study Club")

**INITIALIZE** members information as "@bibas123", "bibas123", "bibas", "bibasrai68@gmail.com", "9982030680", "NA"," 100")

**SET** catalog as catalog()

**WHILE** loop as true

**DO**

**PRINT** Welcome to libarary(). Enter your choice.format(library.name)

**PRINT** 1, display books

**PRINT** 2, borrow books

**PRINT** 3, return books

**PRINT** 4, pay fine

**PRINT** 5, find book by author

**PRINT** 6, find book by book name

**PRINT** \n

**SET** user\_choice = input()

**IF** user\_choice is not 1,2,3,4,5,6

**PRINT** enter what you want from above

**CONTINUE**

**ELSE SET** user\_choice is int(user\_choice)

**IF** user\_choice is 1

**PRINT** the available books in our library.format(library.name)

Library.display\_books()

**PRINT** \n

**ELIF** user\_choice is 2

**INPUT** book(enter name of book you want to borrow)

**INPUT** user(enter your name)

m1.lend\_book(library,book,user)

**PRINT** \n

**ELIF** user\_choice is 3

**INPUT** book(enter name of book you want to return)

**INPUT** user(enter your name)

m1.return\_book(library,book,user)

**PRINT** \n

**ELIF** user\_choice is 4

**INPUT** book(enter name of book for fine)

**INPUT** user(enter your name)

m1.check\_fine(library,book,user)

**PRINT** \n

**ELIF** user\_choice is 5

**INPUT** book(enter name of book)

**INPUT** user(enter your name)

Catalog.searchbyname(library,book\_name)

**PRINT** \n

**ELIF** user\_choice is 6

**INPUT** book(enter name of author)

**INPUT** user(enter your name)

Catalog.searchbyauthor (library,book\_name)

**PRINT** \n

**END IF**

**PRINT** press q to quit or c to continue

**SET** choice as “”

**WHILE** choice is either c or q

**DO**

**INPUT** choice()

**IF** choice is c

**CONTINUE**

**ELIF** choice is q

Sys.exit

**END IF**

**END DO**

**END IF**

**END DO**

1. Module: User

**START**

**CREATE** user

**CREATE** function init(self, username, password, user, email, mobile, address)

self.username = username

self.password = password

self.user = user

self.email = email

self.mobile = mobile

self.address = address

**CREATE** member(user)

**CREATE**  init (self, username, password, user, email, mobile, address, mber\_id)

super().\_\_init\_\_(username, password, user, email, mobile, address)

self.mber\_id = mber\_id

self.max\_book\_count = {}

self.bookdict = {}

**CREATE** function count\_book(self, user,book)

**IF** user not in self.max\_book\_count

**RETURN** len(self.max\_book\_count[user])

**ELSE**

**IF** len(self.max\_book\_count[user]) <= 4

**RETURN** len(slef.max\_book\_count(user))

**ELSE**

**RETURN** len(slef.max\_book\_count(user))

**END IF**

**END IF**

**CREATE** function lend\_book(self, library, book, user)

**IF** book in library.count\_book\_dict.keys()

**IF** library.count\_book\_dict(book)(0) is >=1

**IF** (book,user) is not in self.bookdict,keys()

**IF** self.count\_book(user,book) is <=5

lend\_date is date.today()

return\_date is sum of date.today() and timedelta(days=10)

self.book.update(return\_date)

library book quantity is reduced by 1

**FOR** value of self.bookdict.keys()

**IF** value of (book, user)

**PRINT** {} book is issued to your name {} on {}. the book can be lended".format(value[0], value[1], lend\_date

**PRINT**  please read the book under 10 days and return it or else fine should be paid

**PRINT** self.max\_book\_count

**END IF**

**ELSE**

**PRINT** maximum number of books that can be issed is 5 only.

**ELSE**

**PRINT** only after paying the fine you can borrow a book

**ELSE**

**PRINT** the issued book is already issued.

**ELSE**

**PRINT** the desired book is put of stock

**ELSE**

**PRINT** the book issued is not in the library

**END IF**

**END DO**

**CREATE** function check\_previous\_fine(self,book,user)

**SET** current\_date as date.today()

**FOR** key,value in slef.bookdict.items()

**IF** key value is 1 equal to user

Return\_date becomes self.bookdict(key)

**IF** current date is greater than return\_date

Delay\_days is currentdate subtracted to return date

Total fine becomes delay\_date multiplied by ten

**RETURN** total\_fine

**ELSE**

**RETURN** 0

**END IF**

**END IF**

**CREATE** function return\_book(self, library, book, user)

**IF** book in library.count\_book\_dict.keys()

Return\_date is self.bookdict(book,user)

**SET** current date to date.today()

**IF** current\_date is greater than return\_date

Total fine becomes delay\_date multiplied by ten

**PRINT** the fine to be paid.format(total\_fine)

**PRINT** redirecting to payment

**ELSE**

Self.max\_book\_count(user).remove(book)

Slef.bookdict.pop(book,user)

Library book quantatity is added by 1

**PRINT** book returned

**END IF**

**ELSE**

**PRINT** Please netr corrct username and book name

**END IF**

**CREATE** function payment(self,library,book,user)

**PRINT** please choose yes if you want to pay fine

**INPUT** choice(yes or no)

**IF** choice is yes

**PRINT** payment is being processed

**PRINT** payment successful

self.max\_book\_count(user).remove(book)

self.bookdict.pop(book,user)

add book count by 1

**ELSE**

**PRINT** redirecting to home

**END IF**

**CREATE** function check fine(self,library,book,user)

**DO IF** book user is self.bookdict.keys()

Return\_date is equal to self.bbokdict(book,user)

Current\_date becomes rdate.today()

**IF** currnt\_date is greater than return\_date

Delay\_days becomes the subtraction of current date and return date

Total\_finr becomes multiple of delay\_days and ten

**INPUT** choice(yes or no)

**IF** choice is yes

Self.payment(library,book,user)

**ELSE**

**PRINT** redirecting to home

**END IF**

**END IF**

**ELSE**

**PRINT** no fine record to be paid

**END IF**

**END DO**

# **Program**

The execution of the program is shown below along with their screenshots as proof after running it.

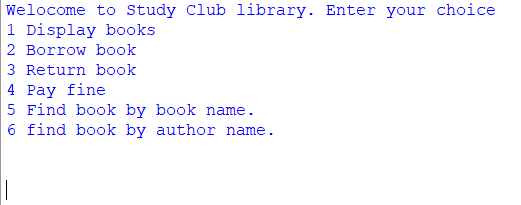


Figure 3 Home menu

When the program is executed, a welcome message appears which choices to choose. In which choosing 1 displays book, 2 borrows book, 3 returns book, 4 pay fine, 5 find book by book name and 6 find book by author name.

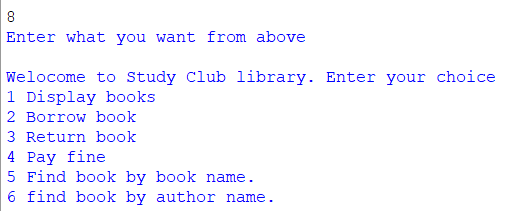


Figure 4 entering wrong choice

If the user inputs other choices besides the shown above, a message is given by the application saying enter what you want from above.

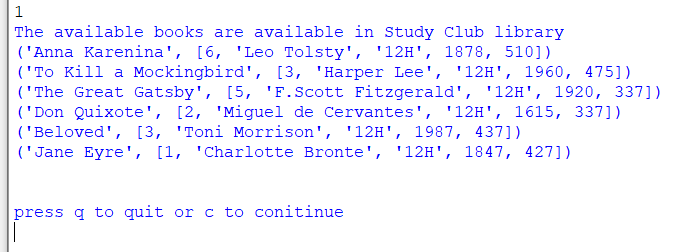


Figure 5 input value as 1

When the choice was input as 1 the books available were displayed. And another choice was given to the user to press q to terminate the program or by pressing q to continue to proceed for other actions.

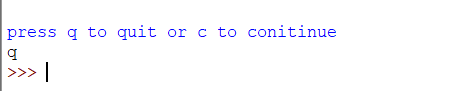


Figure 6 quitting the program by input as q

When given the choice and the value was input as q the program terminated.

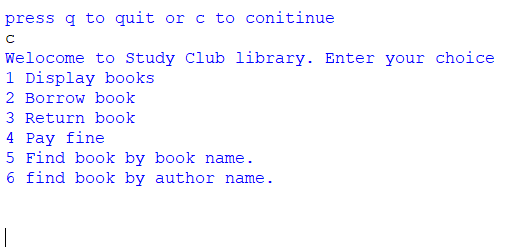
****

Figure 7 continue to other options by input as c

When given the choice when the input was given as c the program was directed to the home menu where other choices were available.

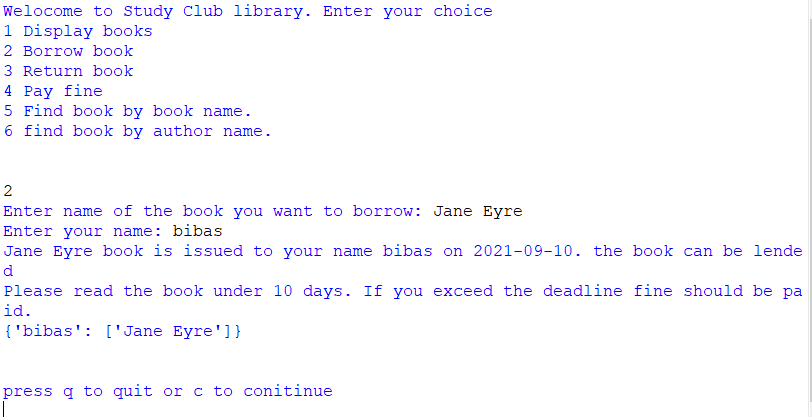


Figure 8 borrowing a book by providing a correct value

When borrowing a book by entering correct values such as book name: Jane Eyre and name of borrower: bibas the book was issued and a date was shown along with a message saying to read the book under 10 days and return it or a fine should be paid.

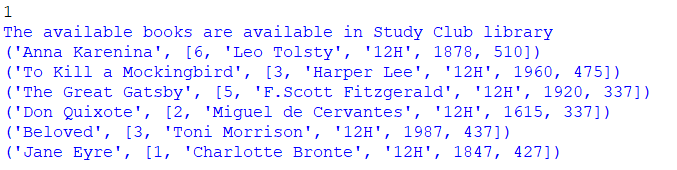


Figure 9 books quantity before borrowing

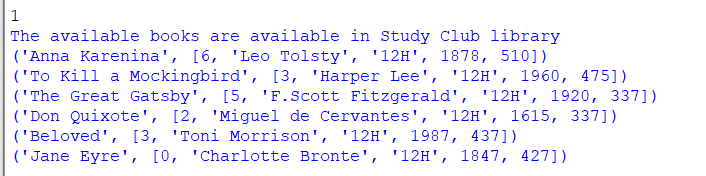


Figure 10 books quantity after borrowing book Jane Eyre

When after the book Jane Eyre was borrowed the quantity dropped from 1 to 0. Which shows the borrowing was successful.

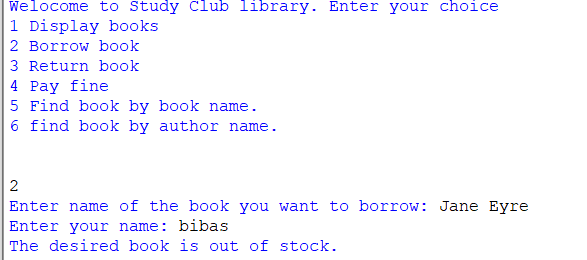


Figure 11 when issuing a book to borrow when its quantity is 0

When issuing the book whose quantity was 0 a message was shown to us saying the desired book is out of stock.

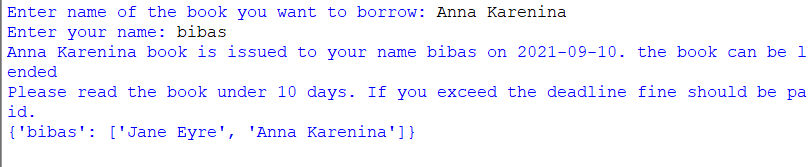


Figure 12 borrowing book Anna Karenina

When borrowing the book Anna Karenina the task was successful.

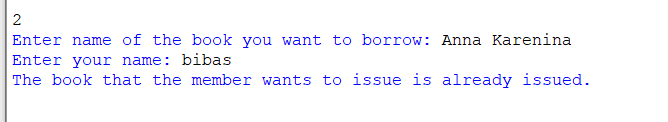


Figure 13 Trying to borrow the same book

When trying to borrow the same book Anna Karenina which had been already issued to bibas a message was shown saying the book that the member wants to issue is already issued.

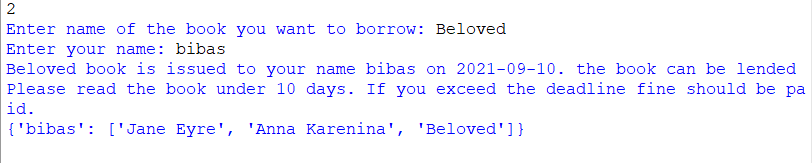


Figure 14 borrowing book Beloved

When borrowing the book Beloved, the task was successful and the book has been issued to the member.

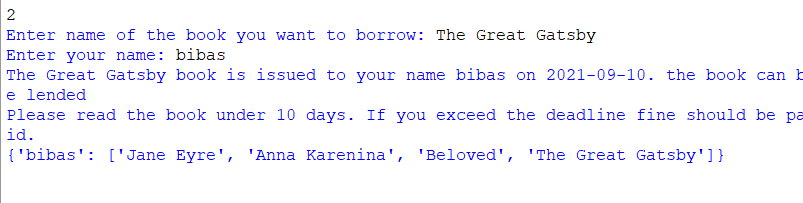


Figure 15 Borrowing book The Great Gatsby

When borrowing the book, The Great Gatsby, the task was successful and the book has been borrowed by the member.

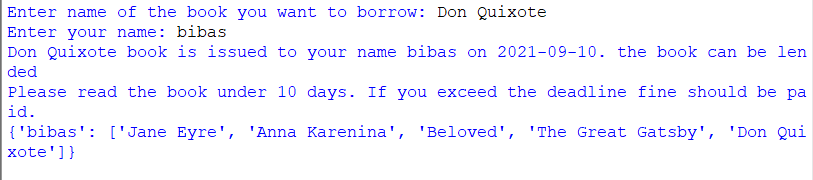


Figure 16 borrowing the book Don Quixote

When borrowing the book Don Quixote the task was successful without any problems.

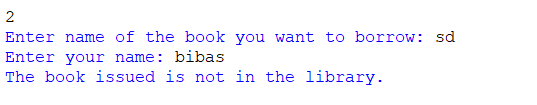


Figure 17 Issuing a book not available in the library

When issuing a book that is not available in the library or putting wrong values displays a message saying the book issued is not in the library.

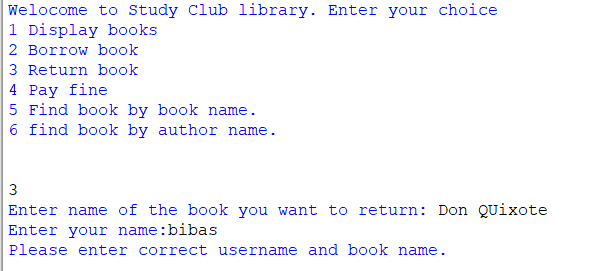


Figure 18 Returning a book without correct value

When returning a book, the incorrect book name was written as a result the application displayed a message say please enter correct username and book name.

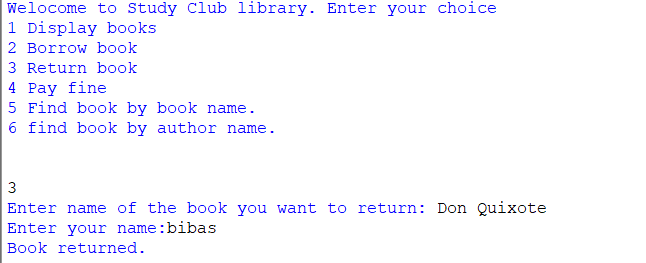


Figure 19 Returning book Don Quixote

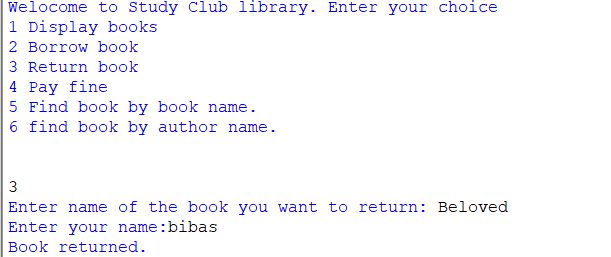


Figure 20 Returning book Beloved

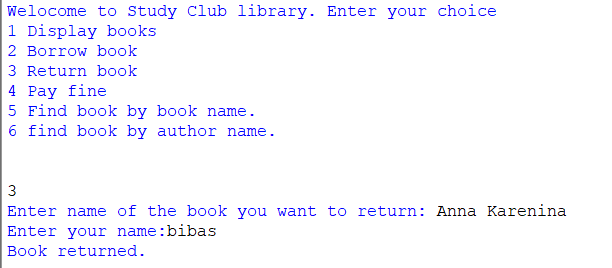


Figure 21 Returning book Anna Karenina

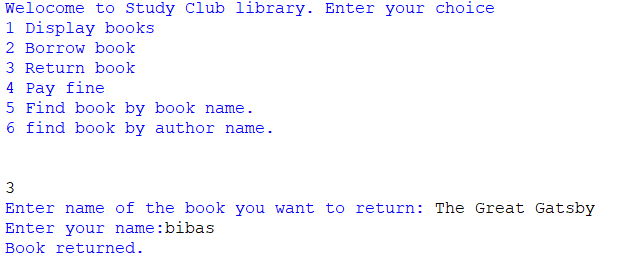


Figure 22 Returning book The Great Gatsby

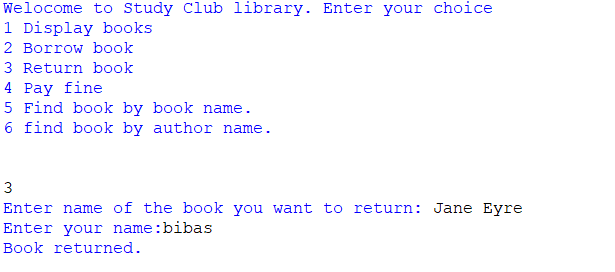


Figure 23 Returning book Jane Eyre

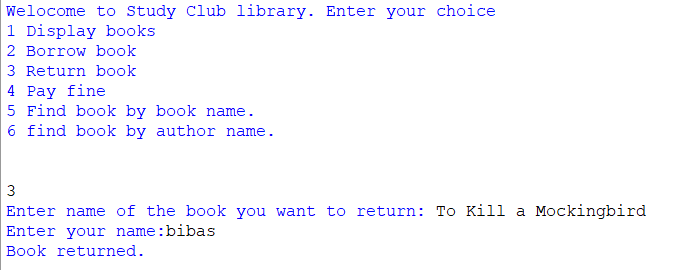


Figure 24 Returning book To Kill a Mockingbird

When returning the borrowed books Anna Karenina, Jane Eyre, To Kill a Mockingbird, Don Quixote and The Great Gatsby the task was handled smoothly and all the books that had been taken were returned properly.

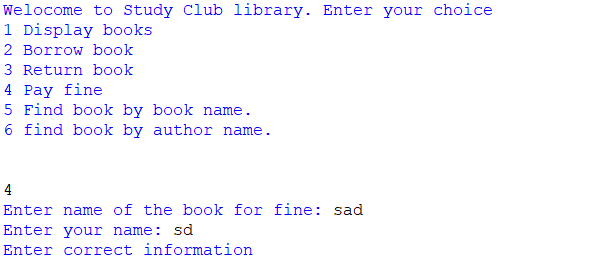


Figure 25 When paying fine entering incorrect values

When paying the fine by choosing option 4 and entering wrong values a message was displayed saying enter the correct information.

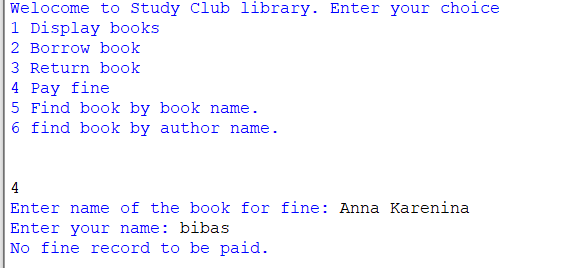


Figure 26 Paying a fine without crossing a deadline

When paying a fine by choosing choice 4 and by not crossing the borrowing period of ten days. The application showed a message saying no fine record to be paid.

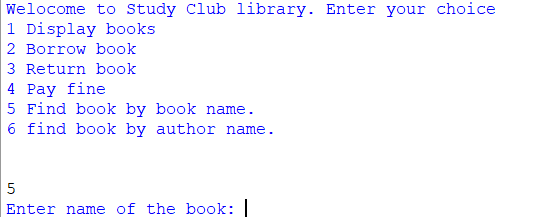


Figure 27 Input value as 5 and find book by book name

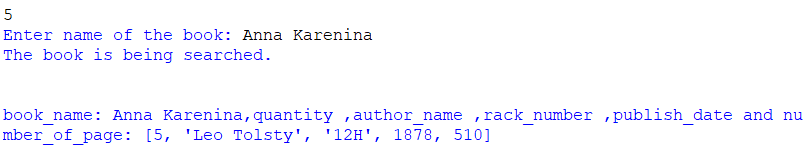


Figure 28 Book name Anna Karenina input

The book name Anna Karenina was entered. Then a message was displayed saying the book is being searched. After the book was found the information about the book was also given below.



Figure 29 Choices given yes no

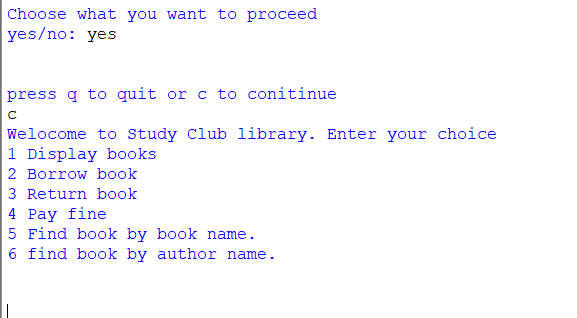


Figure 30 Choosing choice yes

After choosing the option as yes it showed another option to continue or quit.

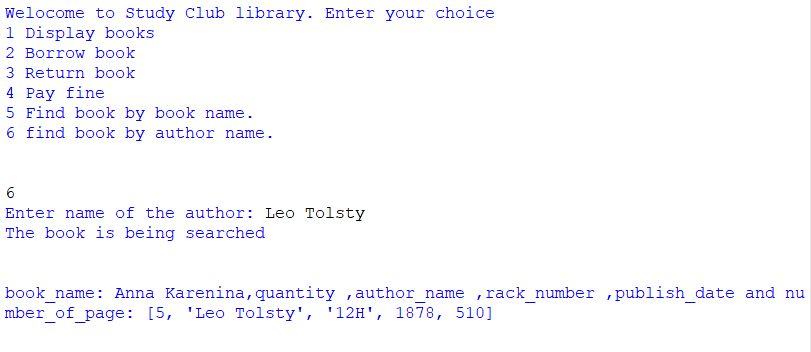


Figure 31 Choosing choice 5 to search book by author name

When choosing option 5 it opened enter name of author. After entering the name of the author as Leo Tolsty the book was searched and the information was displayed properly.



Figure 32 entering choice as no

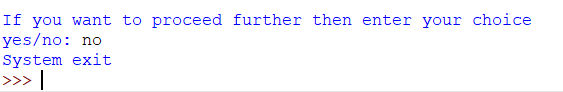


Figure 33 Input choice as no

After choosing the option as no the application terminated itself by displaying a message as System exit.

# **Testing**

Testing is done in a program to ensure that a program is bug free and performs as expected. In our system various tests for the programs were performed so as to verify that the behavior of the developed application is as expected. The tests performed are listed below:

Test-1: Providing invalid or negative values to the program and looking for appropriate messages.

|  |  |
| --- | --- |
| Test No | 1 |
| Objective | To test the program when giving invalid or negative values. |
| Action | 1. Run the program by pressing F5. 2. After the options are shown input, the values not included in the program. |
| Expected Result | A message to shown as saying enter what you want from above. |
| Actual Result | A message is shown as saying to enter what you want from above. |
| Conclusion | The test was successful. |

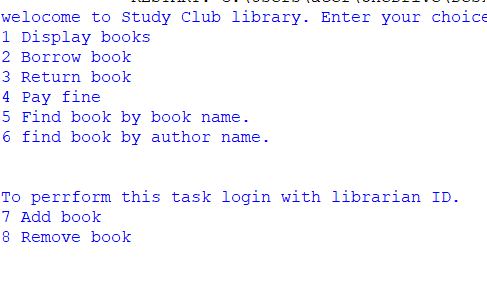


Figure 34 running the program

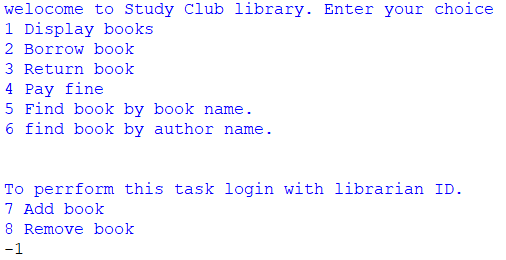


Figure 35 entering value as negative one

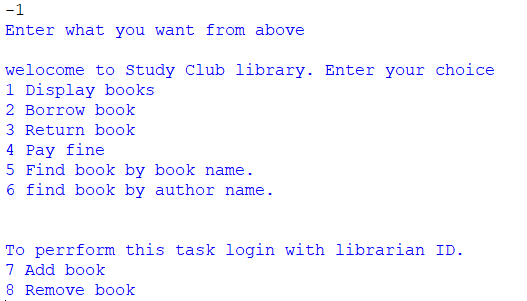


Figure 36 a message saying enter what you want from above

Test 2.1: When borrowing a book input negative or non-existing value.

|  |  |
| --- | --- |
| Test No | 2.1 |
| Objective | To test when borrowing a book while giving negative or non-existing value. |
| Action | 1. Choose the option 1 to display books. 2. A message appears saying enter press q to quit c to continue. 3. Then after pressing c it will take you to the main part. 4. After enter 2 to borrow books. 5. Enter book name will be shown. 6. Enter non existing values and press enter. 7. Enter name will be shown. 8. Enter your name and press enter. |
| Expected Result | A message saying the book is not issued in the library should be generated. |
| Actual Result | A message is shown as saying the book issued is not in the library. |
| Conclusion | The test was successful. |

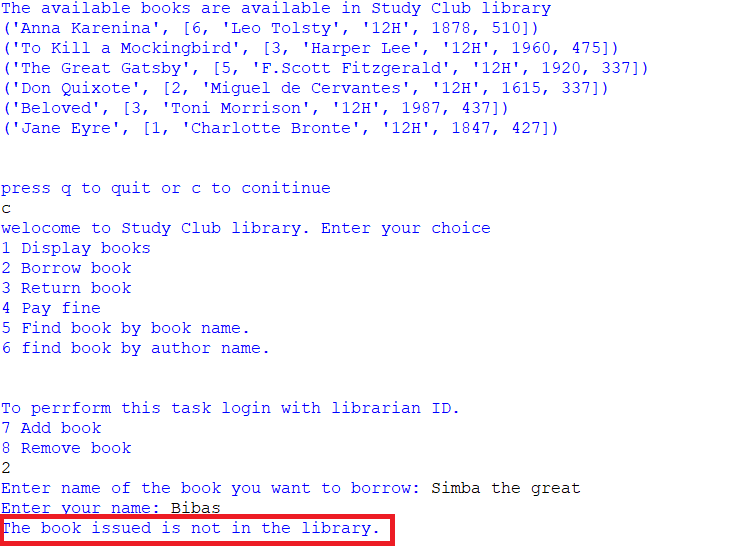


Figure 37 message saying the issued book is not in the library.

Test-2.2: When returning a book input negative or non-existing value.

|  |  |
| --- | --- |
| Test No | 2.2 |
| Objective | To test when returning a book while giving negative or non-existing value. |
| Action | 1. Choose the option 2 to display books. 2. A message appears saying enter press q to quit c to continue. 3. Then after pressing c it will take you to the main part. 4. After enter 3 to return books. 5. Enter book name will be shown. 6. Enter Anna corft as book name and bibas as username and press enter. 7. Enter name will be shown. 8. Enter your name and press enter. |
| Expected Result | A message saying please enter correct username and book name. |
| Actual Result | A message is shown as saying please enter correct username and book name. |
| Conclusion | The test was successful. |

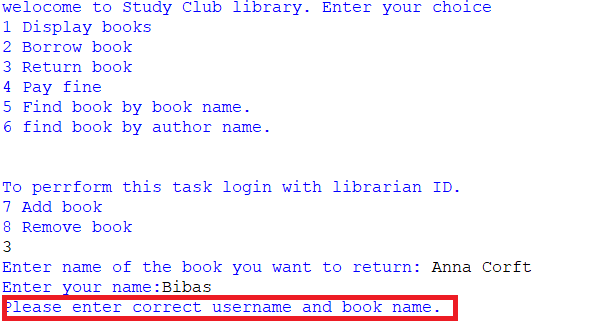


Figure 38 entering invalid values when returning books

Test-3.1: To test whether the book borrowing works or not.

|  |  |
| --- | --- |
| Test No | 3.1 |
| Objective | To test whether the book borrowing works or not. |
| Action | 1. Enter c to continue. 2. Enter number 2 for borrowing a book. 3. Then enter book name Anna Karenina. 4. Also enter the name of person the book is issued to. |
| Expected Result | A message should show up saying the book has been issued to the person along with other information. |
| Actual Result | Message is shown up saying the book have been issued to the given name along with other information. |
| Conclusion | The test was successful. |

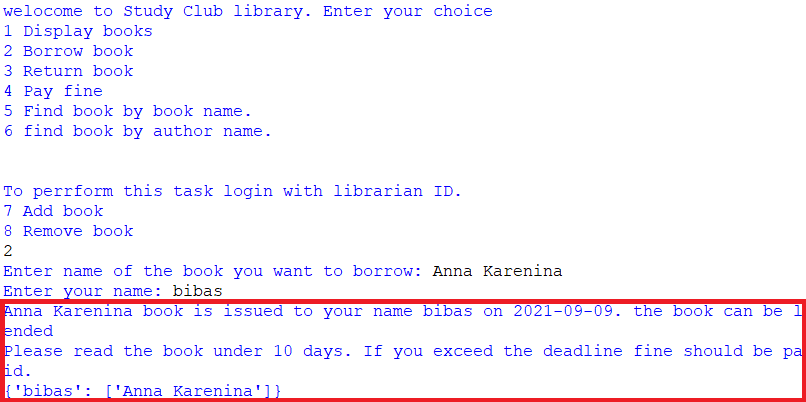


Figure 39 message of successful book borrowed

Test-4: To test whether the book returning works or not.

|  |  |
| --- | --- |
| Test No | 4 |
| Objective | To test whether the book returning works or not. |
| Action | 1. Enter c to continue. 2. Enter number 3 for borrowing a book. 3. Then enter book name Anna Karenina. 4. Also enter the name of person the book is issued to. |
| Expected Result | A message should show up saying the book has been returned. |
| Actual Result | Message is shown up saying the book has been returned. |
| Conclusion | The test was successful. |

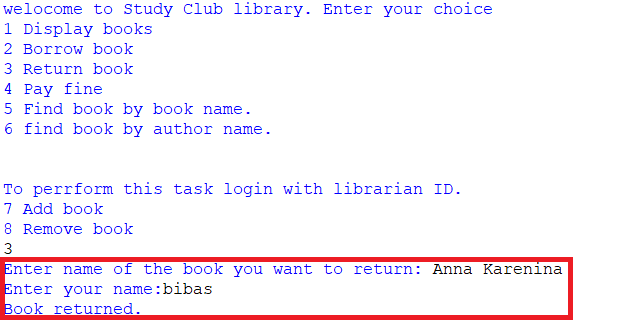


Figure 40 book returned message

Test-5.1: When borrowing a book, the quantity of books should be decreased.

|  |  |
| --- | --- |
| Test No | 5.1 |
| Objective | To test when borrowing a book, the quantity of books should be decreased. |
| Action | 1. Enter c to continue. 2. Enter number 2 for borrowing a book. 3. Then enter book name Anna Karenina. 4. Also enter the name of person the book is issued to. 5. Enter c to continue. 6. Then enter 1 to display books. |
| Expected Result | The total number quantity of books should be decreased by one. |
| Actual Result | The quantity of books is decreased. |
| Conclusion | The test was successful. |

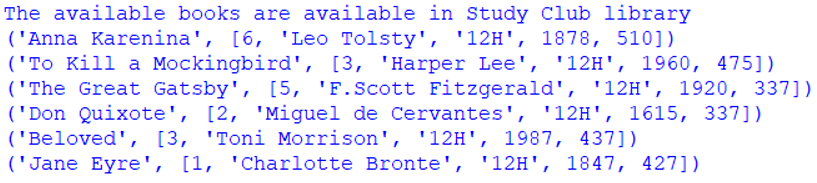


Figure 41 quantity of books before borrowing

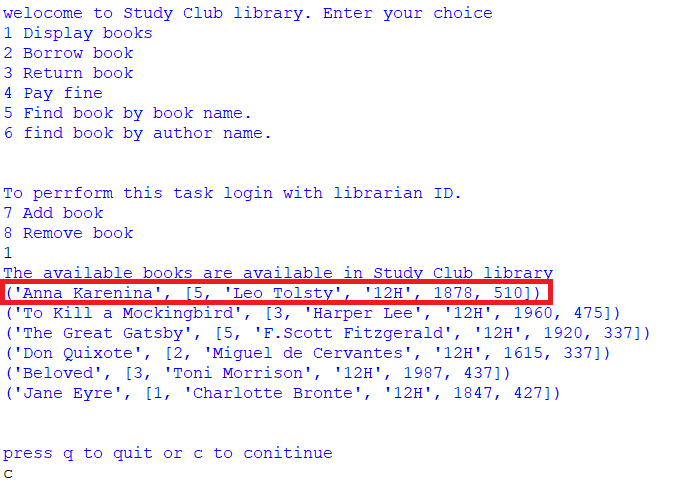


Figure 42 quantity of books after borrowing

Test-5.2: When returning a book, the quantity of books should be go back to its original number.

|  |  |
| --- | --- |
| Test No | 5.2 |
| Objective | To test when returning a book, the quantity of books should go back to its original number. |
| Action | 1. Enter c to continue. 2. Enter number 3 for returning a book. 3. Then enter book name Anna Karenina. 4. Also enter the name of person the book is issued to. 5. Enter c to continue. 6. Then enter 1 to display books. |
| Expected Result | The total number quantity of books should increase to its original number. |
| Actual Result | The quantity of books is increased. |
| Conclusion | The test was successful. |

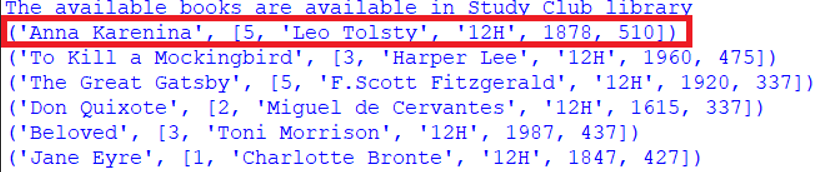


Figure 43 Quantity of books before returning it

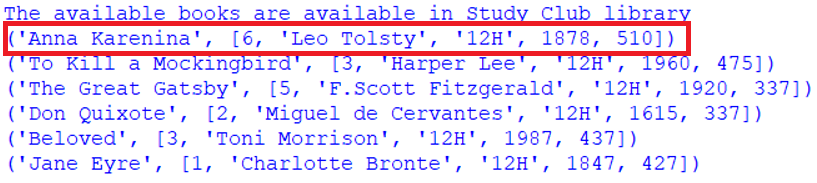


Figure 44 Quantity of book after returning it

# **Conclusion**

The project given to us in the subject Fundamentals Computing was to produce an application. The produced application named as Library Management System. The library Management system should have been able to display, borrow, return, pay fine for the books. The project was similar to that of other subject programming project in which we were to make a similar project. The application making was hard enough. The application used many loops for continuously executing the application until the satisfied conclusion and answer was found. The application used many unique keys words and other functions as well. Making a function was quite easy and using different data structures as well. The term heard before was to making the sequence of data store properly and make it easy for the user. Some of the data structures used were order, tuple and dictionary. These data structures were learned and used properly in each of the places to make the application run properly. The application used many exceptions as well. As for such as if the application didn’t like the entered value which would be incorrect or different from the desired one, the application would immediately give a not satisfied message saying enter correct value or enter value that matches. The books stored in the library of the system made had novel books which were matched with many peoples tastes in reading. The project made this time made was only possible due to learning the things taught overtime to our superiors. The application programming this time showed the significance of itself that it would implement itself into daily life of people and organizations which are big or small with being versatile as well. The application designed would be helpful in libraries where it would keep records of many people without altering them until the conditions such as borrowing, returning, and fine payment has been properly done.