LIBRARY MANAGEMENT SYSTEM

A mini project report submitted by

KULDEEP BISHNOI (20181CSE0358)
K UDITH SIVA SAI RAJU (20181CSE0346)
KARTHIK N (20181CSE0319)
KOULIK SAHA (20181CSE0352)

as part of lab based course Programming in Python, CSE 317
of

BACHELOR OF TECHNOLOGY

in

COMPUTER ENGINEERING

under the supervision of

Dr.Alamelu Mangai, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

PRESIDENCY UNIVERSITY
Itgalpur Rajanakunte, Yelahanka, Bengaluru, Karnataka-560064

November 2019

PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled, " LIBRARY MANAGEMENT SYSTEM" is a bonafide record of Mini Project work done as part of CSE258 Problem Solving Using Python during the academic year 2020-2021 by

KULDEEP BISHNOI (Reg.No:20181CSE0358)
K UDITH SIVA SAI RAJU (20181CSE0346)
KARTHIK N (20181CSE0319)
KOULIK SAHA (20181CSE0352)

Submitted for the Viva Voce held on _____

Examiner

Table of Contents

1. Introduction

This is the report of the coursework designed to produce a system or an application that handles library management system. The program was developed with the precise aid of algorithms and pseudocodes accompanied by the flowcharts.

As the tasks assigned were not so easy. In order to describe our ability to understand and face the questions as well as demonstrate them in an illustrative way as per the requirement requires a lot of firmness. So, perceiving the challenges to complete the tasks in the given period of time and make it beneficial to the targeted audience, the entire efforts are devoted till the final outlook of report is drawn.

This coursework i.e. library management system is similar to the inventory system which will help the user to borrow and return the books available in the library. The library is a place where the user can borrow and return the book with his/her desire. The library lets its users to choose any of the available book to borrow. Well, you may ask why we need to develop this system when there is no need of it. Because the libraries of today are coping well with their old methods i.e. recording the user borrowed or returned history in the paper rather than in software. Even without the help of today's modern software the libraries are managing and are not facing any problems. Yes, it is true but we should always see for the future progress. Because there is technological advances in every sector in today's world.

With the new technologies and system being brought up to life the old methods or techniques should start to change with the passage of time. There is a need for developing this system because our libraries are slowly changing into modern ones and they need the newly developed software. This software or program will help the libraries to keep the proper track of the books borrowed and returned by the user and will help them to ensure there are no loss of books. This system is user friendly and time-efficient and can help to avoid the unnecessary duplication of the data and makes it easier to understand.

The main feature of this project is that it will enable the user to understand the IDLE (Integrated Development and Learning Environment) in python and help them to use the system to the fullest. It will help even those people who don't understand the little about the programming language to use this program to its fullest and without any errors. This program will let the programmer to work with different functions and statements and helps to understand the programming language more in depth.

2.Algorithm

2.1 Stepwise algorithm
Algorithm for borrowing book:

Step 1: Start
Step 2: Input the full name of the borrower.
Step 3: Is the name valid?
If yes, create borrow.txt file and store the borrower details
Then go to step 4
If no, print “Invalid name” then go to step 2
Step 4: Choose any one from the option of the book.
Step 5: Is the book available?
If yes, append further details in the borrow.txt file
Then go to step 6
If no, print “The book is not available.” Then go to step 4
Step 6: Update the stock file from which the book is borrowed.
Step 7: Do you want to borrow the next book?
If yes, go to step 4
If no, go to step 8
Step 8: End

Algorithm for returning book:

Step 1: Start

Step 2: Input the name of borrower.

Step 3: Is the name valid?

If yes, read the borrow.txt file of given name.

Then go to step 4

If no, print “Invalid name” then go to step 2

Step 4: Display the borrow.txt file of given name.

Step 5: Generate return.txt file and update the stock file.

Step 6: Is the book returned date expired?

If yes,

Go to step 7.

If no,

Go to step 9.

Step 7: Add fine

fine= 2*day

Step 8: Add total fine and cost of the book.

Step 9: Output generated text file with the total cost.

Step 10: End

2.2 Pseudocode

Pseudocode for borrowing book:

- input the full name

-if name is valid then

create borrow.txt file and store the borrower details in that file

else

print “Invalid name”

end if

-Choose any one index number from the option of the book.

-if the book available then

append further details in the borrow.txt file

else

print “The book is not available.”

end if

-Update the stock file from which the book is borrowed.

-while borrow book is true

Choose any one index number from the option of the book. if the book available then

append further details in the borrow.txt file

else

print “The book is not available.”

end if

end while

Pseudocode for returning book:

-input the name

- if the name is valid then

read the borrow.txt file of given name.

else

print “Invalid name”

end if

-Display the borrow.txt file of given name.

-Generate return.txt file and update the stock file.

- while returned date expired is true

Add fine

Fine = 2*day

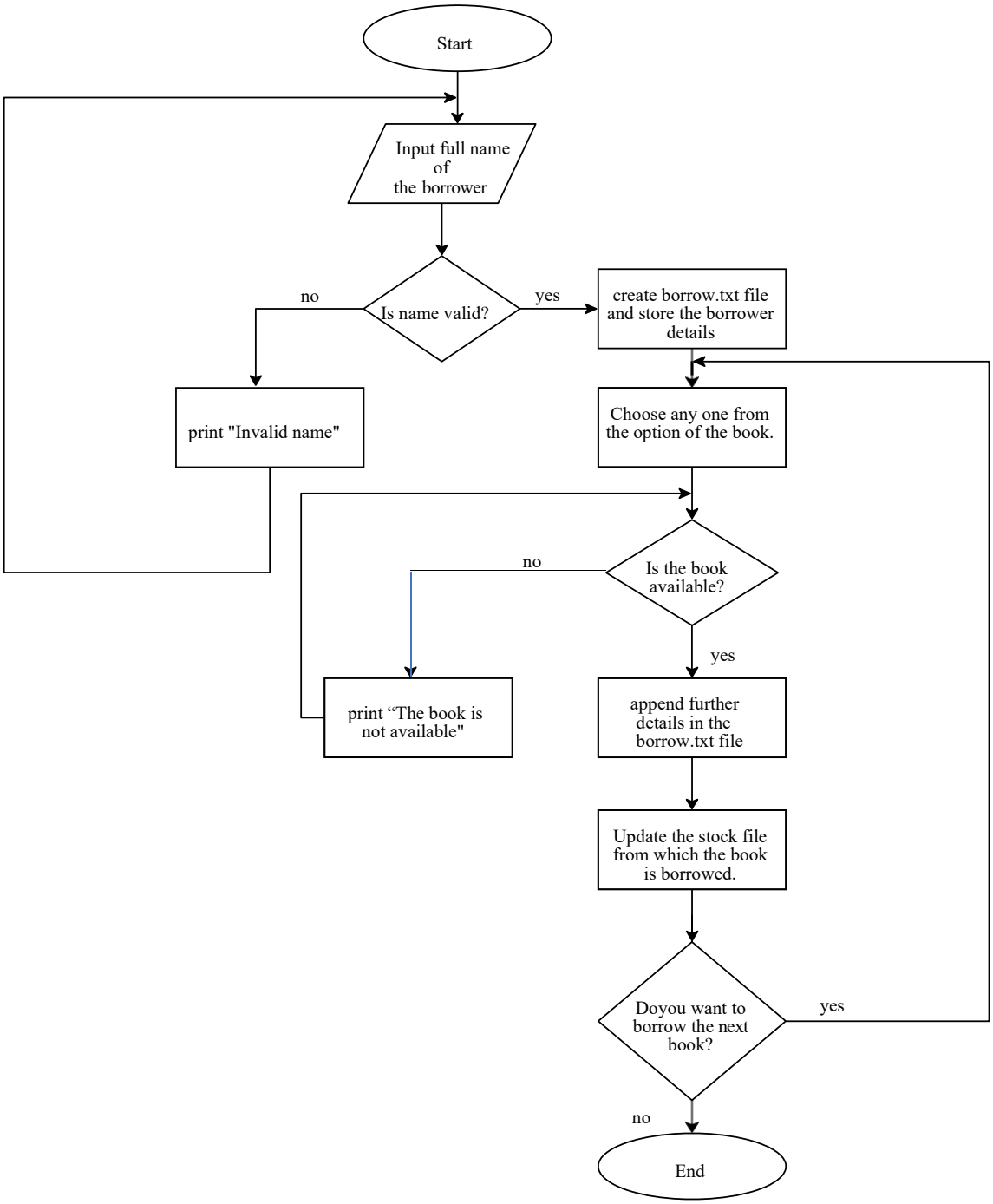
end while

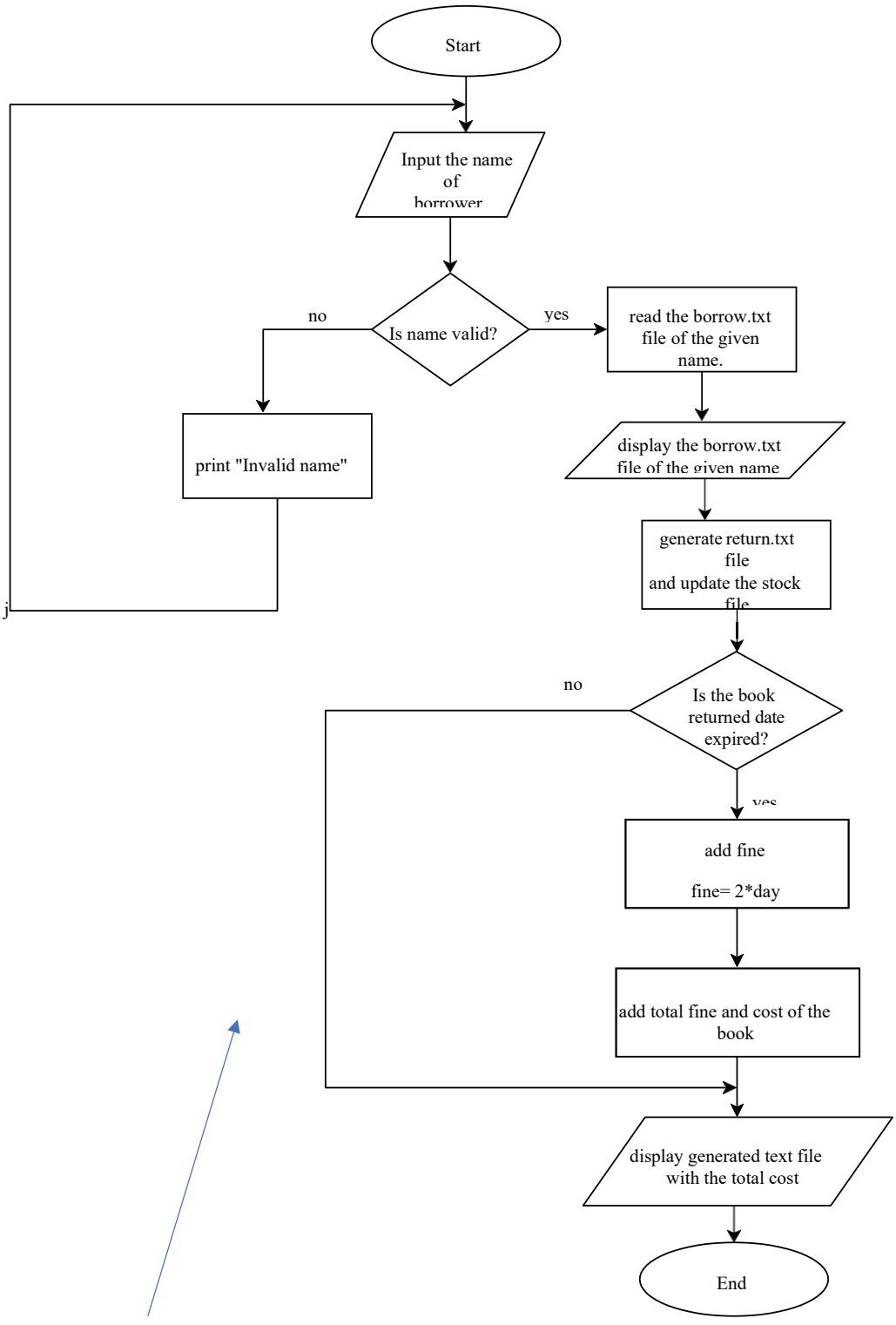
- Add total fine and cost of the book.

-display generated text file with the total cost.

3.Flowchart

Flowchart of borrowing book





Flow chart for returning the book

4.Data Structures

To carry out different operations in Python for input/output and for data storage this project makes the maximum utilization of the collection data type list. Python also provides varieties of data types like integer, string, Boolean, float etc. Some of the data types and data structures were used while writing the program to store and manipulate the data and perform various operations on them. The data types and structures used in the program are:

1. Integer
2. float
3. String
4. Boolean
5. List

Integer data type was used in the program to store all the numerical values provided by the users as well as the results developed from the different operations. For instance, it is used to store the quantity of the books in the variable quantity[index].

Similarly, float data type is used to store decimal values in the program. It is used to store the total cost price of the books in the variable total.

Correspondingly, string data type is used to store textual data having sequence of one or more characters. Various operations like choice, stat are carried out in string. Most of the printed statement in our .py file are often concatenated and of data type string. In the different function of our .py file string data type are used for storing the decision of user.

Likewise, Boolean is used to store one character (either true or false). It is used in the program to check conditions and stored in variables like loop, success.

Moreover, list is used to store values on bookname, authorname, quantity and cost in our .py file. As the list is mutable collection data types, the value on the list can be added or removed as per the user needs by the use of methods append() and extend() which have been successful to include in different module of our .py folder.

5.code

```
import Return

import ListSplit

import dt

import Borrow



def start():

    while (True):

        print("        Welcome to the library management system      ")

        print("-----------------------------------------------------")

        print("Enter 1. To Display")

        print("Enter 2. To Borrow a book")

        print("Enter 3. To return a book")

        print("Enter 4. To exit")

        try:

            a = int(input("Select a choice from 1-4: "))

            print()

            if (a == 1):

                with open("stock.txt", "r") as f:

                    lines = f.read()

                    print(lines)

                    print()
```

```
            elif (a == 2):
                ListSplit.listSplit()
                Borrow.borrowBook()
            elif (a == 3):
                ListSplit.listSplit()
                Return.returnBook()
            elif (a == 4):
                print("Thank you for using library management system")
                break
            else:
                print("Please enter a valid choice from 1-4")
        except ValueError:
            print("Please input as suggested.")



start()
import dt
import ListSplit



def borrowBook():
    success = False
    while (True):
        firstName = input("Enter the first name of the borrower: ")
```

```
    if firstName.isalpha():
        break
    print("please input alphabet from A-Z")
while (True):
    lastName = input("Enter the last name of the borrower: ")
    if lastName.isalpha():
        break
    print("please input alphabet from A-Z")


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


while success == False:
    print("Please select a option below:")
    for i in range(len(ListSplit.bookname)):
        print("Enter", i, "to borrow book", ListSplit.bookname[i])


    try:
        a = int(input())
        try:
```

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

                ListSplit.quantity[a] = int(ListSplit.quantity[a]) - 1
                with open("Stock.txt", "w+") as f:
                    for i in range(3):
                        f.write(ListSplit.bookname[i] + "," + ListSplit.authorname[i] + "," +
str(
                            ListSplit.quantity[i]) + "," + "$" + ListSplit.cost[i] + "\n")

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

```
                    a = int(input())
                    if (int(ListSplit.quantity[a]) > 0):
                        print("Book is available")
                        with open(t, "a") as f:
                            f.write(
                                str(count) + ". \t\t" + ListSplit.bookname[a] + "\t\t   " +
ListSplit.authorname[
                                    a] + "\n")

                        ListSplit.quantity[a] = int(ListSplit.quantity[a]) - 1
                        with open("Stock.txt", "w+") as f:
                            for i in range(3):
                                f.write(ListSplit.bookname[i] + "," + ListSplit.authorname[i]
+ "," + str(
                                    ListSplit.quantity[i]) + "," + "$" + ListSplit.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")
            borrowBook()
            success = False
    except IndexError:
        print("")
        print("Please choose book acording to their number.")
except ValueError:
    print("")
    print("Please choose as suggested.")


    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())
```

```
def listSplit():
    global bookname
    global authorname
    global quantity
    global cost
    bookname = []
    authorname = []
    quantity = []
    cost = []
    with open("stock.txt", "r") as f:

        lines = f.readlines()
        lines = [x.strip('\n') for x in lines]
        for i in range(len(lines)):
            ind = 0
            for a in lines[i].split(','):
                if (ind == 0):
                    bookname.append(a)
                elif (ind == 1):
                    authorname.append(a)
                elif (ind == 2):
                    quantity.append(a)
```

```
elif (ind == 3):
    cost.append(a.strip("$"))
ind += 1
import ListSplit
import dt
def returnBook():
    name = input("Enter name of borrower: ")
    a = "Borrow-" + name + ".txt"
    try:
        with open(a, "r") as f:
            lines = f.readlines()
            lines = [a.strip("$") 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("              Library Management System \n")
        f.write("                Returned By: " + name + "\n")
```

```
            f.write("    Date: " + dt.getDate() + "    Time:" + dt.getTime() + "\n\n")
            f.write("S.N.\t\tBookname\t\tCost\n")


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


        print("\t\t\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 = 2 * day
            with open(b, "a")as f:
                f.write("\t\t\t\tFine: $" + str(fine) + "\n")
            total = total + fine
```

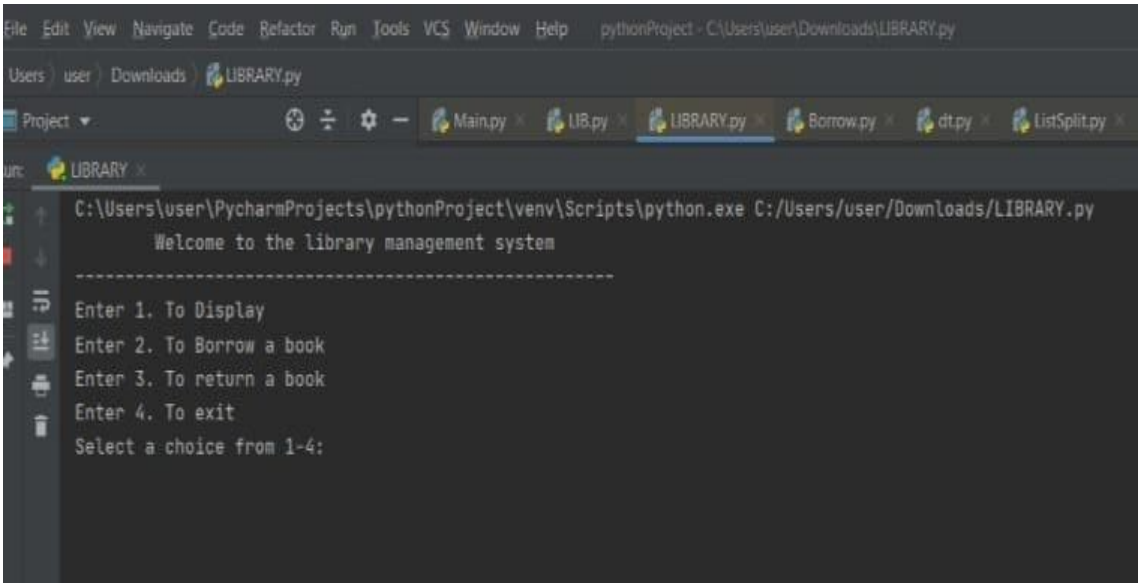
```
            print("Final Total: " + "$" + str(total))
            with open(b, "a")as f:
                f.write("\t\t\t\t\tTotal: $" + str(total))


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

6.Testing

Testing is an important portion that is to be carried out while developing any program as it helps to find the error if there is any error and helps to debug the program. Testing helps us to get familiar with the type of error and handle the errors with the necessary formatting in the program.

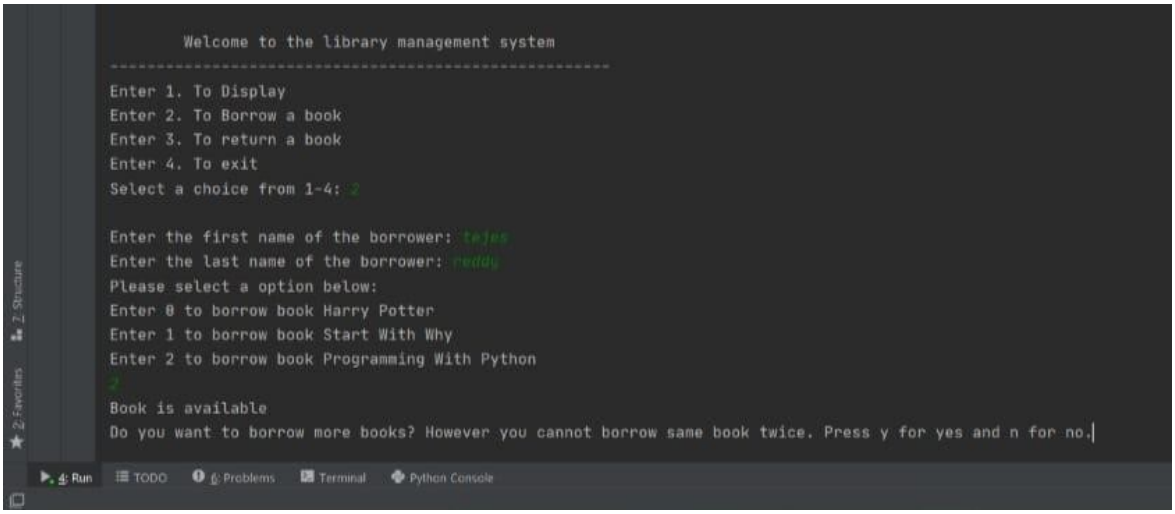
Test-1

Action	To check whether it can handle exceptions or not
Expected output	It should be able to handle any sum of input of user
Actual output	The program was able to handle even the wrong input of user
Test result	The program runs in its flow until user closes it.



Test-2

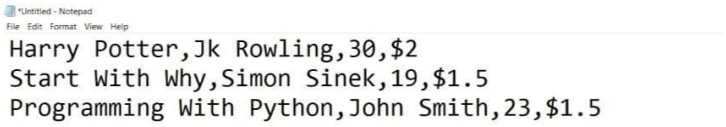
Action	To check whether the program can borrow multiple books or not.
Expected output	It should be able to ask the user whether to borrow more books or not.
Actual output	The program was able to ask the user whether to borrow more books or not.
Test result	Pass



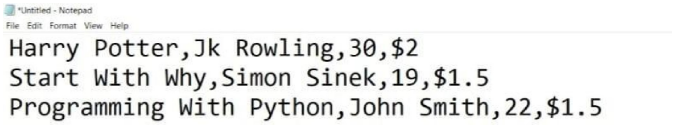
Test-3

Action	To check whether the quantity of books in the stock file while borrowing is reduced by 1 or not
Expected output	It should be able to reduce the quantity of books by 1.
Actual output	The program was able to reduce the quantity of books by 1.

Test result	Pass



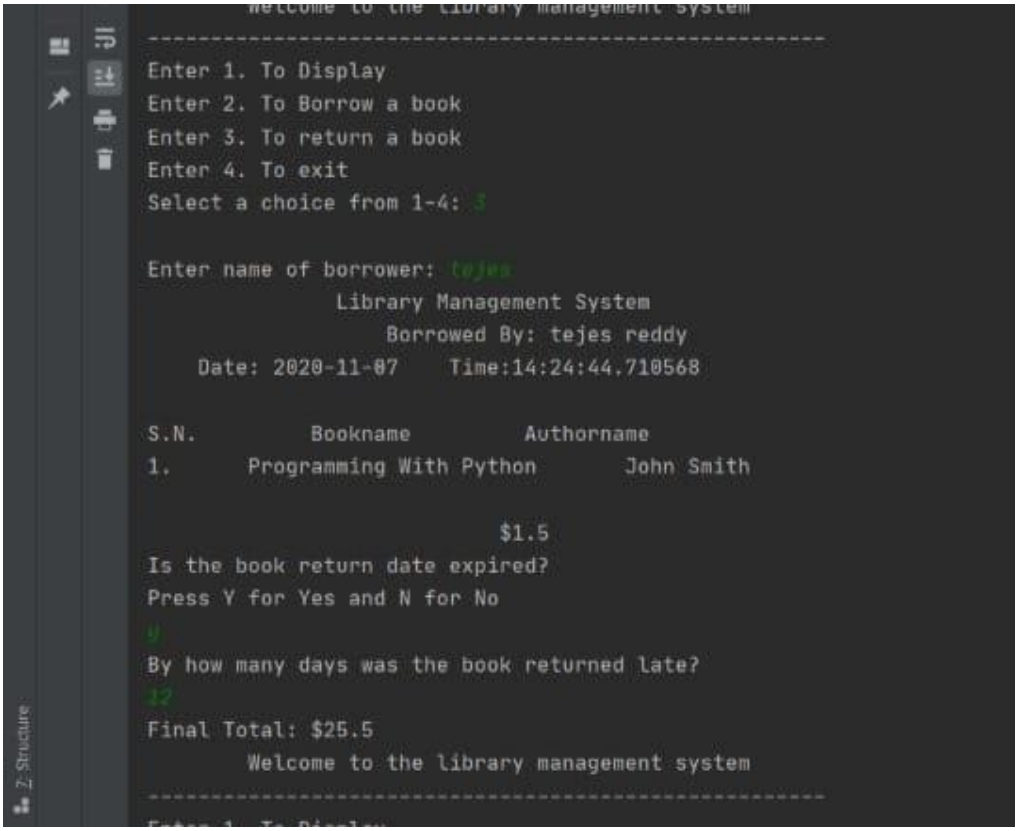
Before borrowing book



After borrowing book

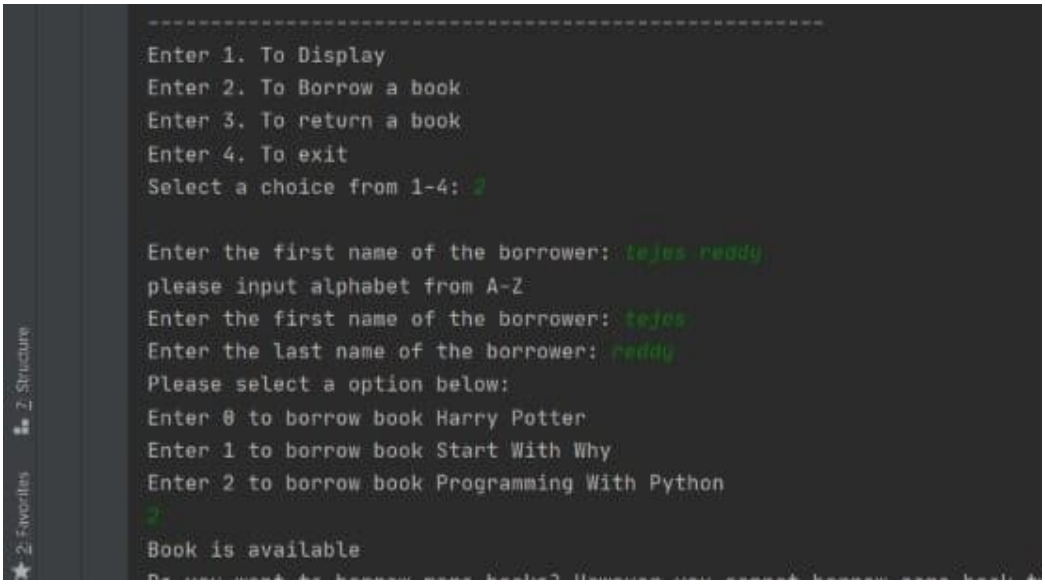
Test-4

Action	To check whether the fine is added after returning book or not.
Expected output	It should be able to display the returning book with the total cost.
Actual output	The program was able to display the returning book with the total cost.
Test result	Pass

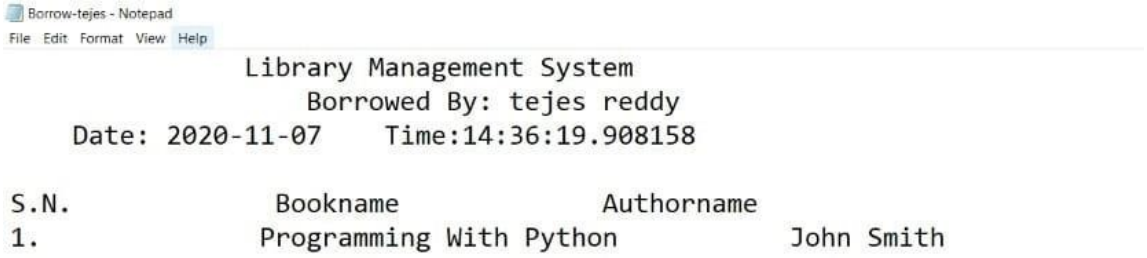


Test-5

Action	To check whether the data is stored in text file or not.
Expected output	It should be able to store data in text file.
Actual output	The program was able to store data in stock file.
Test result	Pass



Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
Select a choice from 1-4: 2
Enter the first name of the borrower: tejas reddy
please input alphabet from A-Z
Enter the first name of the borrower: tejas
Enter the last name of the borrower: reddy
Please select a option below:
Enter 0 to borrow book Harry Potter
Enter 1 to borrow book Start With Why
Enter 2 to borrow book Programming With Python
2
Book is available
2: Favorites
2: Structure


Borrow-tejes - Notepad
File Edit Format View Help
Library Management System
Borrowed By: tejes reddy
Date: 2020-11-07 Time:14:36:19.908158
S.N. Bookname Authorname
1. Programming With Python John Smith

7.Research

The given coursework was completed with lots of researches which made the coursework comparatively easier. The steady effort and boundless research related to different themes made the tasks less difficult to carry out. After various research and practise we were able to know how exactly the program runs and what are the correct codes for the program. Lots of websites, journals and books were consulted for gaining information on various important topics. The research not only enhanced our knowledge to complete the coursework, it also sharpened our knowledge and understanding about various topics in python.

7.1 Websites

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

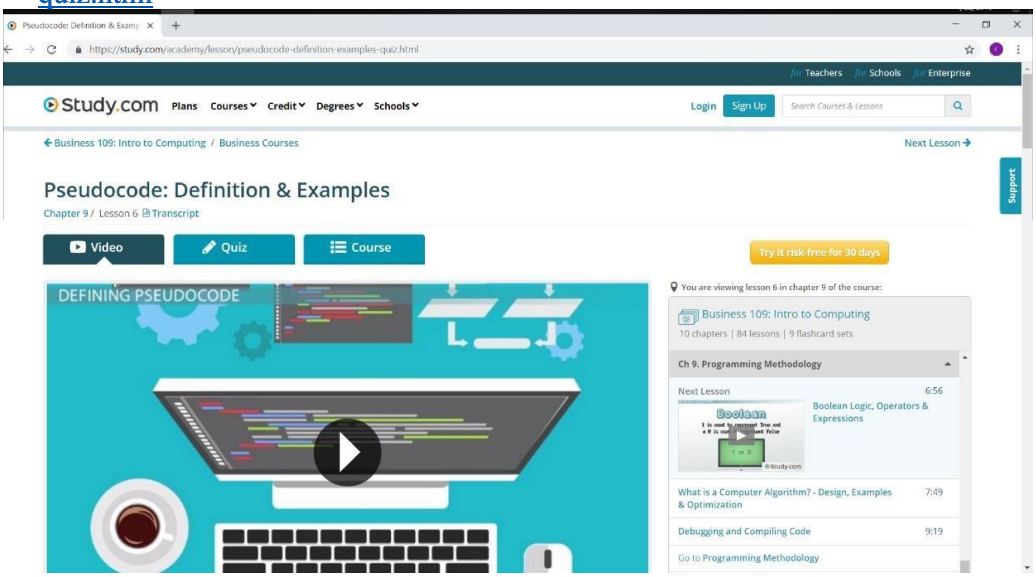


Figure 12: Research for pseudocode

I used this site to learn about the pseudocode to be written for the assigned program.

- https://www.learnpython.org/en/Loops

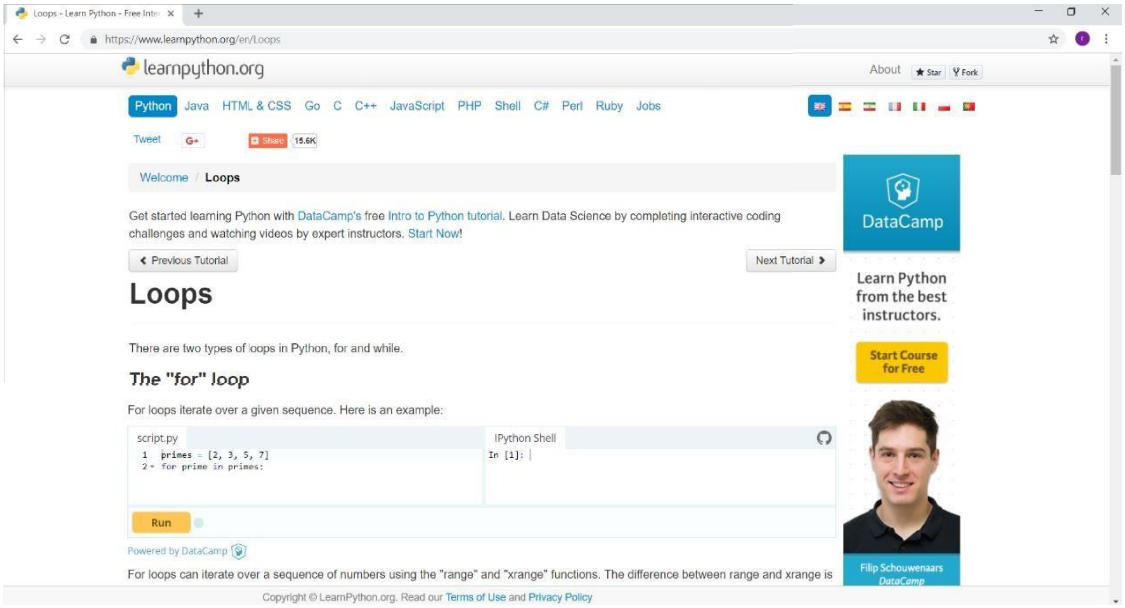


Figure 13: Research for loops

I used this website to have a sound knowledge on python loops and how they are used and operated.

- https://docs.python.org/3/tutorial/modules.html

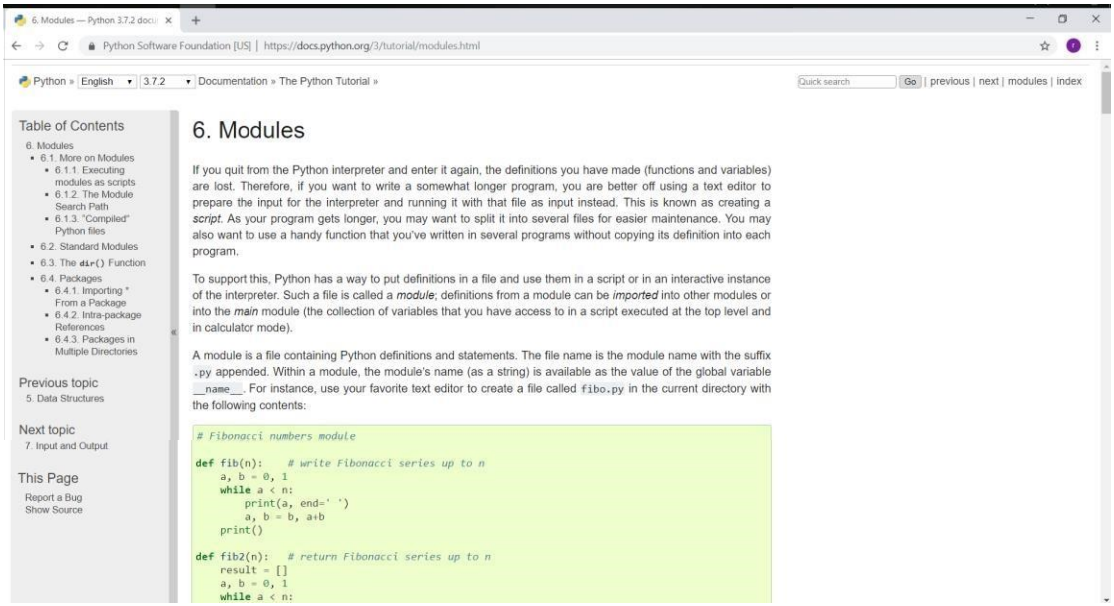


Figure 14: Research for modules

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

- https://www.tutorialspoint.com/python/python_exceptions.htm

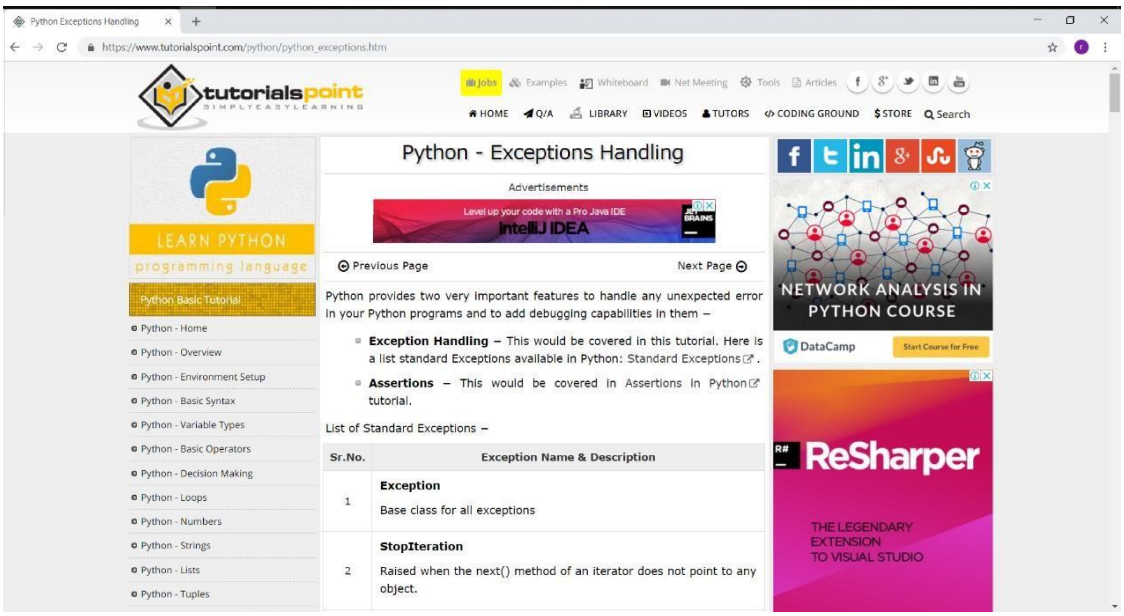


Figure 15: Research for exceptional handling

I used this website to handle exceptions in the assigned python programming.

- https://www.datacamp.com/community/tutorials/data-structures-python

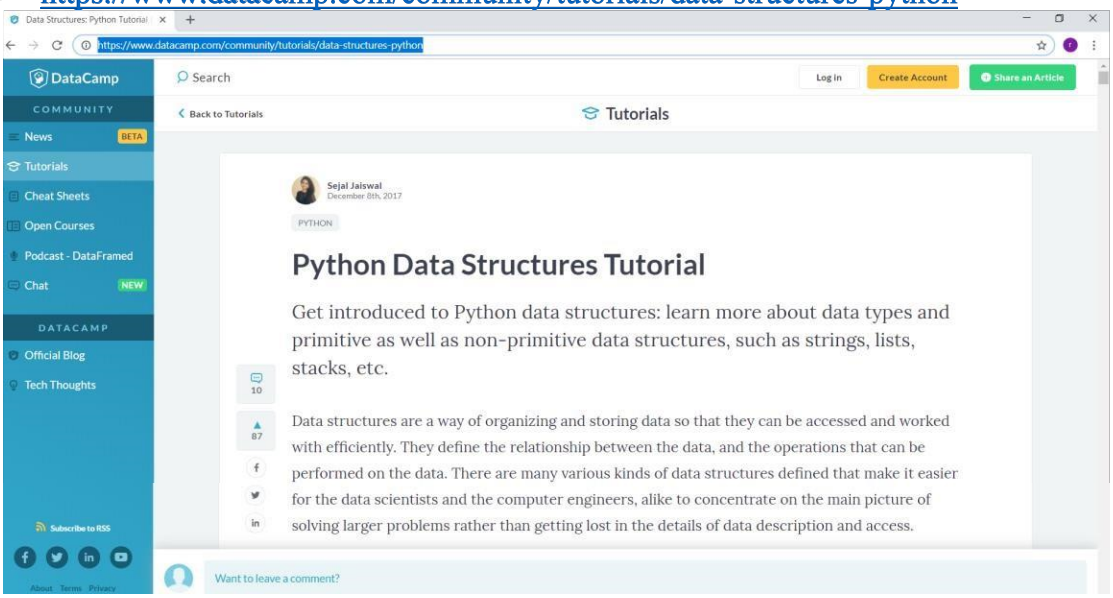


Figure 16: Research for data structures

This site was used for gaining effective knowledge about the data structures. I became familiar with lots of terms related to types of data structures and their functions and advantages.

7.2 Books

Dive into python

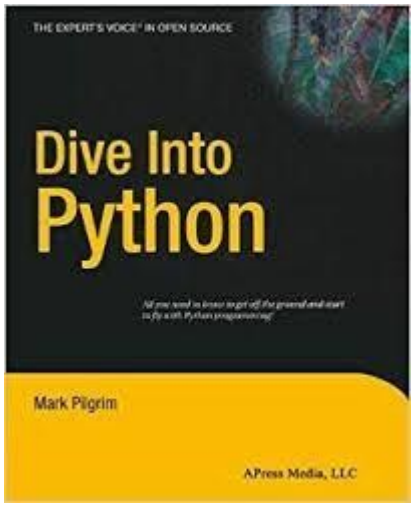


Figure 17: Dive into python

By the help of this book I was able to learn about various data types in python programming language like lists, tuples, strings, etc.

Python succinctly

Figure 18: Python succinctly

By the aid of this book, I learnt about python modules and packages and how to use them.

Beginning python learning from novice to professional

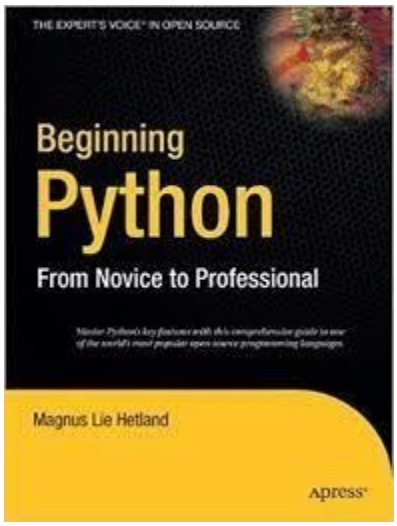


Figure 19: Beginning python learning from novice to professional

This book helped me to understand about loops, conditional statements, break and continue statements.

Think python

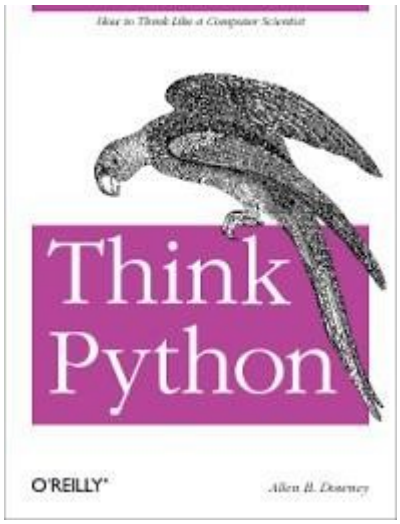


Figure 20: Think python

From this book, I gained sound knowledge on python programming and use of comments.

Fluent Python

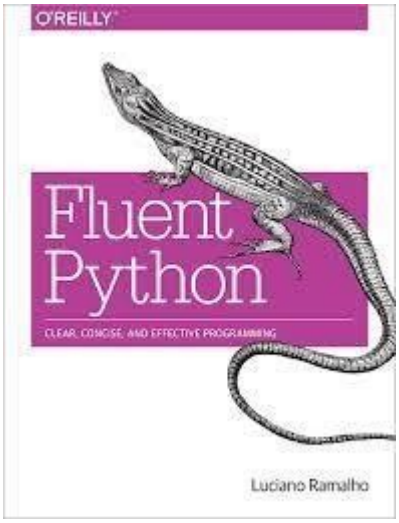
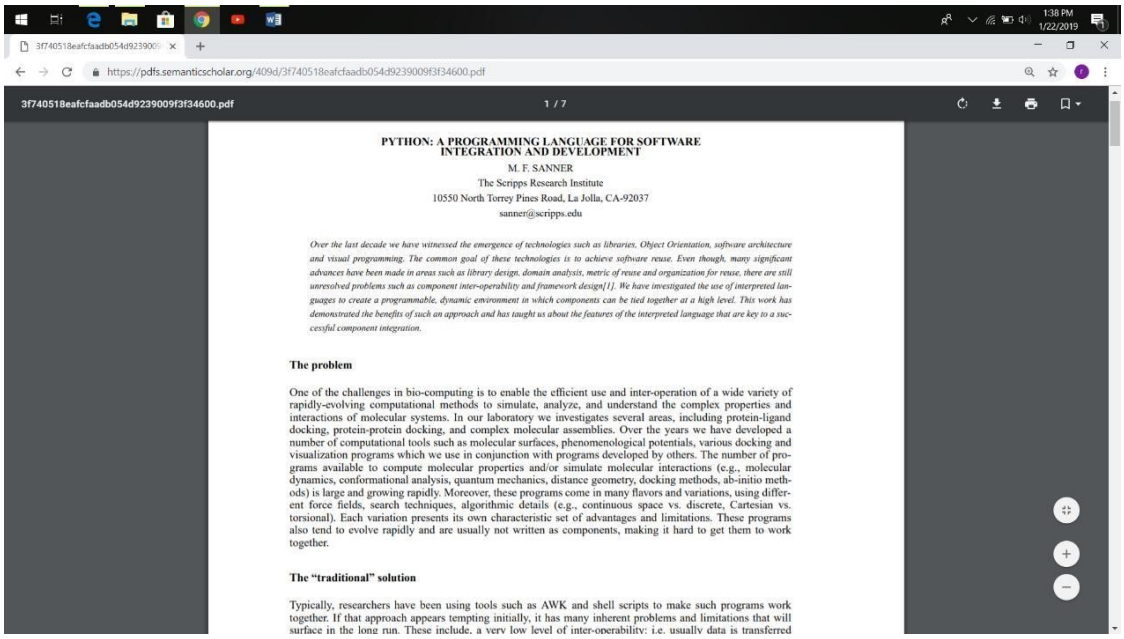


Figure 21: Fluent python

By the aid of this book, I was able to define various functions like borrowbook, returnbook, listsplit, etc.

7.3 Journals

Python: A programming language for software integration and development



PYTHON: A PROGRAMMING LANGUAGE FOR SOFTWARE INTEGRATION AND DEVELOPMENT

M. F. SANNER

The Scripps Research Institute
10550 North Torrey Pines Road, La Jolla, CA-92037
sanner@scripps.edu

Over the last decade we have witnessed the emergence of technologies such as libraries, Object Orientation, software architecture and visual programming. The common goal of these technologies is to achieve software reuse. Even though, many significant advances have been made in areas such as library design, domain analysis, metric of reuse and organization for reuse, there are still unresolved problems such as component inter-operability and framework design[1]. We have investigated the use of interpreted languages to create a programmable, dynamic environment in which components can be tied together at a high level. This work has demonstrated the benefits of such an approach and has taught us about the features of the interpreted language that are key to a successful component integration.

The problem

One of the challenges in bio-computing is to enable the efficient use and inter-operation of a wide variety of rapidly-evolving computational methods to simulate, analyze, and understand the complex properties and interactions of molecular systems. In our laboratory we investigates several areas, including protein-ligand docking, protein-protein docking, and complex molecular assemblies. Over the years we have developed a number of computational tools such as molecular surfaces, phenomenological potentials, various docking and visualization programs which we use in conjunction with programs developed by others. The number of programs available to compute molecular properties and/or simulate molecular interactions (e.g., molecular dynamics, conformational analysis, quantum mechanics, distance geometry, docking methods, ab-initio methods) is large and growing rapidly. Moreover, these programs come in many flavors and variations, using different force fields, search techniques, algorithmic details (e.g., continuous space vs. discrete, Cartesian vs. torsional). Each variation presents its own characteristic set of advantages and limitations. These programs also tend to evolve rapidly and are usually not written as components, making it hard to get them to work together.

The "traditional" solution

Typically, researchers have been using tools such as AWK and shell scripts to make such programs work together. If that approach appears tempting initially, it has many inherent problems and limitations that will surface in the long run. These include, a very low level of inter-operability: i.e. usually data is transferred

Figure 22: Journal for python programming language

This journal is written by M.F. SANNER which describes about python, python numeric extensions, etc. I used this website to learn about the python programming language and its features over other programming languages.

8.Conclusion

This coursework was finally completed with lots of hardship, several research on the related topics such as flowcharts, algorithm, pseudocode and so on. The tasks assigned in the coursework were not easy at all. The code was written and tested to confirm that it had no bugs and errors and delivered the accurate result. For the successful completion of all the tasks, each task was carried out in steps, which made the task easier.

This coursework helped the way to do a task in a certain time period in a managed way and also helped to develop skills which will be very helpful in the future as well. While doing the project I gained sound knowledge of python, its data structures, its various in-built functions, comments, while and for loops, if/else conditionals and many more. This project helped to gain valuable experience. All in all, although the tasks were tough and required nights of hard work and labor, successfully completing those tough tasks was a great fun.