LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC4002NA Information Systems

Assessment Weightage & Type

20% Individual Coursework2

Year and Semester

2018-19 autumn

Student Name: Suraksha Shrestha

London Met ID:

College ID: NP01NT4A180002

Assignment Due Date: 01/18/2019

Assignment Submission Date:01/18/2019

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.

Proposal:

This proposal is written to address the coursework of Information System, given to us as an individual task, which is all about writing a python program that generates the notes of the transaction of books in a library management. The coursework was given in 8th week and the deadline given is 11th week.

Purpose:

The main purpose of this coursework is to write algorithm and pseudocode for the python program to maintain the inventory of the library management. The only purpose was to note the transaction of the books using different python codes and functions. Here, we are just making an efficient way to store the directory of a small management, overcoming the demerits of writing those data manually.

Problem Statement:

The problem given was to create the notes on transaction of books from the library. The part of coding and debugging the errors was time consuming but however by some research, those indentation problems were resolved.

Aims and Objectives:

The main aim of the project is to carry out all the tasks of the coursework in the best way possible. Surfing in the internet, research from different books and journals will only help to accomplish the aims of program by which the computer stores the data of transaction. Within the given period of time, with lots of research and effort, the given coursework can be completed.

Proposed Approach:

In order to perform all the tasks given in the coursework, following approaches will be taken:

- Firstly, lots of research will be done which was related to the topic, like algorithms, flowchart, python programs and testing method of the program.
- In order to compose the program, reasonable data structures will be chosen.

- Algorithm would be written to figure out the flow of coding to achieve the required objectives.
- Pseudocode would be written as the rough sketch of the program.
- By visualizing the algorithm, flowchart would be created to illustrate the way the code would be organized and executed.
- The python program would be written to note the transactions.
- Finally, the written program would be tested so that we can see if the program has errors or bugs.

Scope of Project:

The scope of the project is after the completion of this project, I will be able to know about how to create a program in the python and can learn detail about functions and modules. Moreover, this project would give the idea of effective coding in python. The concept will be pretty clear after reviewing the program. It also illustrates how a good algorithm, flowchart and pseudocode can be prepared and utilized to create the main program. This project can be a great help to the people who want to know the coding for the transaction of any management. Comments are provided in necessary portion by which the code can be understood easily.

Target Audience:

This project is for a small library management system for storing small database. It can utilized by students, lecturers and researches who want to acquire the knowledge of python programming.

Hardware and Software Requirements:

In order to run the program, laptop or PC is needed as hardware and in case of software, python and operating system is required to operate it.

Activity Description and Timeline:

Since this task is time-consuming and requires a lot of research, the project will be carried out in proper manner from the first week.

Activities	Duration
1.Research.	1st week
2.Developing python code.	2nd week
3.Debbuging code and testing it.	3rd week
4.Documentation.	4th week

Table of Contents

Table of Figures:

List of Tables:

Introduction:

This project is designed for the Library Management where users are allowed to either borrow or return the books available in the library. In this century, storing data in a register is not considered to be efficient or safe. If we store the data in a register, as in old times, there is a huge chance of loosing the data. In addition, those data cannot be changed according to the need or we need to rewrite the information again including the previous record. For instance, if we have written the information of person who borrowed the book and if he wants to borrow the another one then, we have to rewrite both, previous and the new information, making a new entry. So, in this century of programming, is it really relevant to use the register to write the information? I don't think so. Ultimately, we have to find an appropriate way of storing the information in versatile manner, where the information can be accessed and changed according to the need.

Using a program, which asks the information of the user, takes their choice to either borrow or return the book and write all the information of the user in a txt file is conversely the best way to manage the data of the library. This python program overcomes all the demerits of storing the data in a register which is briefly described above. The merits of using the code is briefly described as below:

- Data entry: Since, this program asks the information from the user and rewrite it in a txt file, additional data entry is not required (unless the customer is new).
- Saving data: Recording the data in paper work can be erased or damaged after a long time but those data can be stored in secured manner using this program.
- Saving time: In this program, after the user provide his/her information, it is automatically written in a txt file but writing those data is time consuming in paper work.
- Accessing data: While storing the information of a person again and again in register, we should go through all the data entries and search for the

particular person which is time consuming and not relevant. In other hand, while storing the information of a person in python program, naturally it finds the person in a second. For instance, to check the name of the customer, we should go through all the data in a register but in the program, data can be accessed only by entering the name of particular person.

Discussion and Analysis:

The program was developed and designed using different tools and functions of the library in python program. Firstly, different modules were made which are imported in the main module using the function 'import'. In addition, different functions were defined for different choices by the user. For example, to display the books available in the library with their book ID, a function (display()) is defined where stock_books i.e.books stored in the dictionary, is passed. Likewise, while storing the information of the user, current date and time is required, which is imported using a default module of the library in python i.e. import datetime. This module ultimately imports the current date and time from the used device. Furthermore, to access and change the data in the txt file of the books in library, such as: number of the books, file can be read or override using the functions of the library. For example, to deduct the number of books (after the person borrows the book) in the txt file LibraryBooks.txt, the function used to access and write in the file is:

```
file=open(file_name,"w")
```

In order to change the data and rewrite it in the txt file, we have introduced a module i.e. DatabaseB(), where a function write_to_file is defined and by passing the parameters i.e. stock_books and borrowed_books, we've written the data in respective txt files.

Algorithm:

The approach I took to run the program is brute force approach i.e. general problem-solving problem. The process of the python program is carried out in following steps:

Step 1. Start.

Step 2. Reading the data of the book from the txt file.

Step 3. Providing choices to the user.

If user chooses to display, continue.

If user chooses to borrow the book, then got to step 5.

If user chooses to return the book, got to step 10.

If user chooses to exit, go to step 14.

Step 4. Display the books of library.

Step 5. Input name.

Step 6. Ask the user for book ID. If given book ID is available in the database, continue. Else, display a suitable output.

Step 7. If the quantity of required book is not 0, continue. Else, display the suitable output.

Step 8. Let the user borrow the book and decrease the quantity by 1.

Step 9. Append the information in a cart.

Step 10. Input name and check whether the name is in borrower's list or not. If yes, continue, else display a suitable output.

Step 11. Ask for the book ID. If book ID is in the borrowed data of the user, continue, else give out a suitable output.

Step 12. Let the user return the book and if the number of days borrowed is more than 10, apply fine of 10% for each day.

Step 13. Increase the quantity of book by 1.

Step 14. Append the information in a list.

Step 15. Print borrow and return notes and rewrite the books.txt file with updated stock.

Step 16. Stop.

Flowchart:

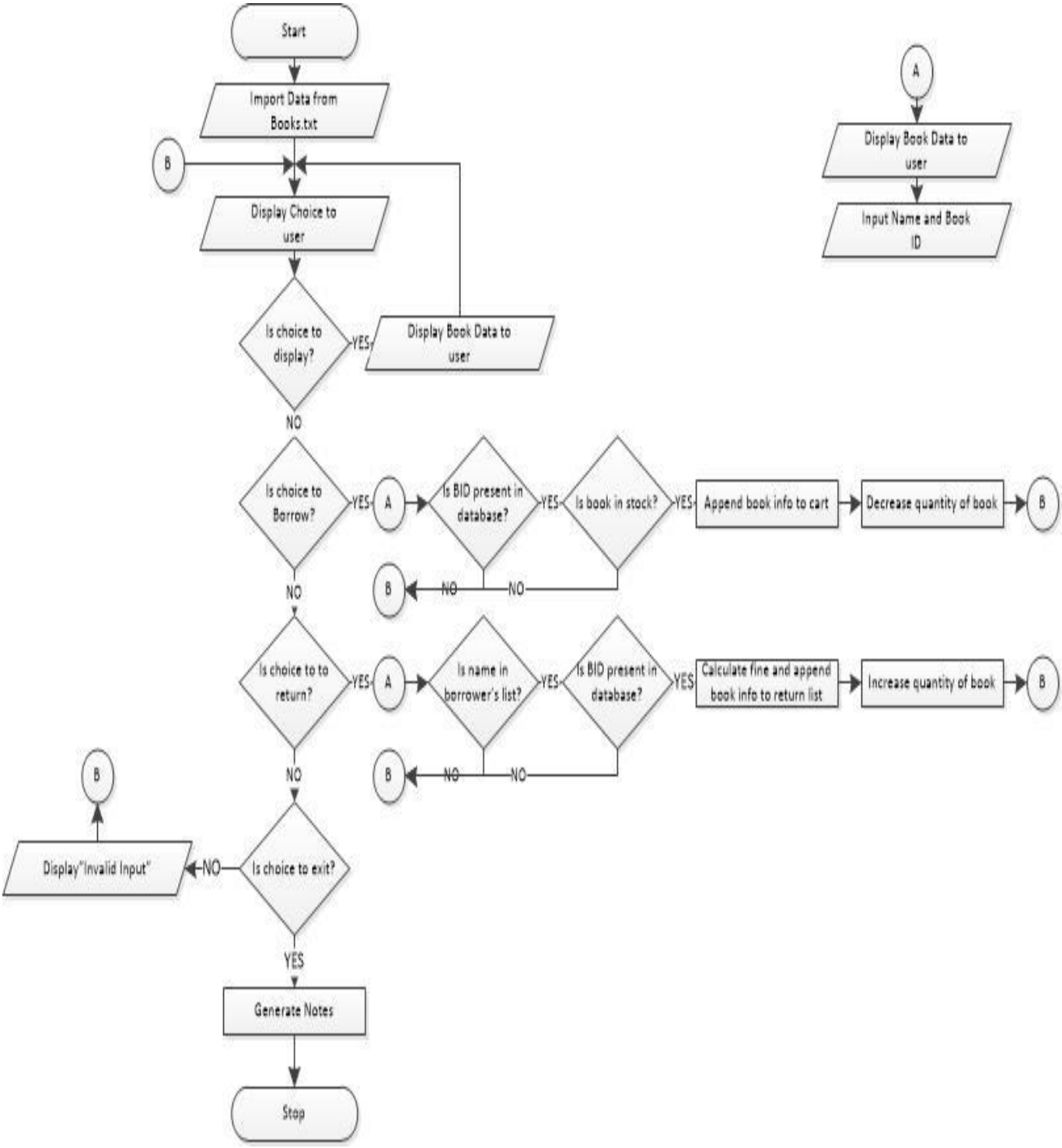


Figure 1: Flowchart of Program

Pseudocode:

```
ALGORITHM main module
        IMPORT borrow_module
        IMPORT return_module
        IMPORT writetofile_module
         READ the txt file
         file=OPEN("library.txt",'r')
        contents=file.read()
        file.close()
        fString=contents.split('\n')
        library={ }
        FOR each in fString:
            line=each.split(",")
            library[line[0]]=line[1:]
        END FOR
          FUNCTION display_books()
            display books
          sorted=True
          WHILE sorted==True:
                    borrowed_database= { }
                    returned database= { }
                    Display menu
```

```
                Input choice
                IF choice==1 THEN
                        Display_books()
                ELSE IF choice==2 THEN
                        INPUT name
                        Display_books(library,name,borrowed_database)
                        borrow_module.borrow()
                ELSE IF choice==3 THEN
                        INPUT name
                          return_module.returns(library,name, borrowed_database,
                          returned_database)
                ELSE IF choice==4 THEN
                        Writetofile_module.write_to_txt("library.txt",library,"l")
                        Writetofile_module.write_to_txt("borrowed.txt",borrowed
                          _database ,"b")
                       Writetofile_module.write_to_txt(""returned_txt",returned
                       _database,"r")
                        EXIT()
                ELSE
                        DISPLAY("The input number does not exists")
                 END IF
          END WHILE
    END FUNCTION
```

```
END main module


ALGORITHM borrow_module
        IMPORT date and time
        FUNCTION borrow
                sorted=True
                WHILE sorted==True:
                    INPUT bookID
                IF bookID=n THEN:
                        RETURN [library,borrowed_database]
                        Break
                END IF
                FOR key in library:
                        IF bookID in key THEN
                            IF bookID[value][2]>0 THEN
                                INPUT date
                                bookID[value][2]-=1
                                IF name in borrowed_database THEN
                                    borrowed_database [name][0].append
                                    (book_name)
                                   borrowed_database [name][2].append
                                   (book_price)
                                ELSE
```

```
                                        borrowed_database[name]=
                                        [[book_name], date,[book_price]
                                    END IF
                                ELSE
                                    DISPLAY("Book not in stock")
                                END IF
                            ELSE
                                DISPLAY("Book ID does not exist in stock")
                        END IF
                    END FOR
                END WHILE
            END FUNCTION
        END module


ALGORITHM return_module
        IMPORT datetime
        FUNCTION returns
                IF !(name in borrowed_database) THEN
                        DISPLAY ("This person has not borrowed any book")
                        RETURN(library,returned_database)
                END IF
                sorted=True
                WHILE sorted==True:
```

```
INPUT bookID
IF bookID==n THEN
        RETURN(library,returned_database)
ELSE
        Return_date=datetime.date,today()
        Days=returned_date-borrowed_date
        IF(library[bookID][0] in book_list) THEN
                IF Days<=10 THEN
                        Fine=0.0
                ELSE
                        Fine=Days*0.1*book_price
                END IF
                IF (name in returned_database) THEN
                        returned_database [name][0].append
                        (book_name)
                        returned_database[name][1].append(book_price)
                        returned_database[name][2].append(Fine)
                ELSE
                        returned_database[name]=[[book_name],
                         [book_price],[Fine]]
                END IF
                library[bookID][2]+=1
        ELSE
```

```
                    print(“The book ID entered can’t be found”)
                ENDIF
            END WHILE
        END FUNCTION
END module


ALGORITHM writetofile_module
    FUNCTION write_to_txt(file_name,data_to_write,file_picker)
        file=open(file_name,”w+”)
        FOR keys, values in data.items():
            IF (file_picker==’l’) THEN
                file.write(keys+","+values[0]+","+values[1]+","+values[2]+","
                 values[3]+"\n")
            ELSE IF (file_picker==’b’) THEN
                FOR x in range(0,len(values[0])):
                    file.write(file.write(str(values[0][x])+"\t\t"+
                    str(values[1])+"\t\t"+"$"+str(values[2][x])+"\n")
                    sum+=values[2][x]
                    file.write("Grand Total: \t\t"+"$"+str(sum)+"\n")
            ELSE
                FOR x in range(0,len(values[0])):
                    file.write(str(values[0][x])+"\t\t"+"$"+
                    str(values[1][x])+"\t\t"+"$"+str(values[2][x])+"\n")
```

```
                    sum+=values[2][x]
                    file.write("Total Fine: \t\t"+"$"+str(sum)+"\n")
                END IF
            END FOR
            file.close()
        END FUNCTION
    END module
```

Data Structures:

The collection data types required in this program are:

1. List: List is the collection data type which can store different data separated by comma sign and enclosed between the big brackets. Every element can be fetched using their index which starts from zero. Here, in this program, list is used to store the data from the txt file and then it is changed into dictionary. (Anon., 2019)

```
['B001', 'Harry Poter', 'JK Rowling', '29', '$2']
['B002', 'Start With Why', 'Simon Sinek', '0', '$1.5']
['B003', 'Programming With Python', 'John Smith', '20', '$1.5']
['B004', 'Summer Love', 'Subin Bhattarai', '15', '$3']
['B005', 'Romeo and Juliet', 'Shakespeare', '40', '$4']
['B006', 'Famous Last Words', 'Katie Alender', '25', '$2.5']
['B007', 'Great Expectations', 'Charles Dickens', '30', '$3']
```

Figure 2:List demonstration

2. Dictionary: Dictionary is the best data structure that can be used in this project. Dictionary is unordered collection of the data, so there is no concept of indexing. The data in dictionaries can be easily fetched and are mutable as well. In more precise way, the data can be added or deleted easily. So, while using it in this project, the transaction data can be changed according to the need. Dictionaries are denoted by curly brackets { }. The data does not have definite order.

Dictionaries store data in the form of key-value, the keys are unique. Because of this feature of dictionary, we can make Book Id as key and other information as keys. (Anon., 2019)

```
{'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

Figure 3: Demonstration of Dictionary

Program:

In the given coursework the program was given to generate invoice of a library and show the database of books, borrower and returner as well. The coursework consists of four modules and their codes are shown as below:

Main module(final.py):

```
#importing the modules
import borrow_module
import return_module
import writetofile module
```

Figure 4:Importing Module

Simply, the child modules are imported to the main module.

```
#Reading the database from library.txt file
file=open("library.txt","r")
fString=file.read()
file.close()
fString=fString.split("\n")
# Dictionaries to store the information about borrowed and returned books
borrowed_database={}
returned_database={}
#Putting database information to a python dictionary named as "library"
library={}
chk=True
for each in fString:
    line=each.split(",")
    library[line[0]]=line[1:]
    try:
        library.pop("") <-- In order to remove '/n' from the tst file if necesssary.
    except:
        pass
```

Figure 5:Reading txt file

This is the first part of the program where the database of library books present in the txt file is changed into a dictionary where the book ID is the key and a list of book's information is the value. Here, pop function is used which removes the error caused by '\ n' in the program.

```
#Fuction to display the library database
def display_books(database):
    print ("Book id\t\tBook name\t\tQuantity\t\tPrice")
    for keys,values in database.items():
        print(keys+"\t\t"+values[0]+"\t\t"+values[2]+"\t\t"+values[3])
    print("\n")
```

Figure 6:Function to Display

This function(display_database) is introduced in the main module and it is called if we need to display the books available in the bookstore. The formatting of the function is made to make it look simpler and effective. Not all the information is displayed, only the necessary information is provided i.e. book ID, book name and its price.

```
sorted=True
while sorted==True:
    ask=int(input("1) Display Books 2) Borrow Books 3) Return Book 4) Quit\n"))
    if ask==1:
        display_books(library)
    elif ask==2:
        name=input("Enter your name: ")
        bList=borrow_module.borrow(library,name,borrowed_database)
        library=bList[0]
        borrowed_database=bList[1]
    elif ask==3:
        name=input("Enter your name: ")
        rList=return_module.returns(library,name,borrowed_database,returned_data
        library=rList[0]
        returned_database=rList[1]
    elif ask==4:
        break
    else:
        print("The input number doesnot exist.")
```

Figure 7: Choices for User

Here, choices is provided to the user either to display or borrow or return or exit. If user chooses to display, display_database function is called, passing the dictionary as the parameter, which displays the available books. If the user chooses to borrow the book, firstly name is asked and then a function(borrow) is called from another

module(borrow_module) and the returned information is captured in a dictionary. The information is written in respective database. Similarly, if the user chooses to return, name is asked and a function returns is called from the module(return_module). The information is again stored in respective database. If the user chooses to exit, loop ends and is the user inputs other choices it displays a suitable output.

```
writetofile_module.write_to_txt("library.txt",library,"l")
writetofile_module.write_to_txt("borrowed.txt",borrowed_database,"b")
writetofile_module.write_to_txt("returned.txt",returned_database,"r")
```

Figure 8:Calling the Function to Rewrite

At last of the main module, whole changed information of the books and database of borrowed and returned are written in respective txt files.

Borrow module(borrow_module.py):

```
import datetime
def borrow(library,name,borrowed_databse):
    check=True
    while check==True:
        book_id=input("Please enter the book-id or n to exit: ")
        if (book_id=="n"):
            return [library,borrowed_databse]
            break
        if (book_id in library):
            if (int(library[book_id][2])>0):
                book_name=str(library[book_id][0])
                #Removing '$'sign for calculation
                book_price=float(library[book_id][3][1:])
                #taking borrowed date from the user
                b_date=input("Enter borrowed date (mm/dd/yyyy): ")
                #splitting the given date info by '\n'
                b_date=b_date.split("/")
                b_date=datetime.date(int(b_date[2]),int(b_date[0]),int(b_date[1]
                print("You have successfully boroowed the book")
                #checking the entered name in borrowed database
                #appending the info if it exists
                if (name in borrowed_databse):
                    borrowed_databse[name][0].append(book_name)
                    borrowed_databse[name][2].append(book_price)
                #if not a new entry is made
                else:
                    borrowed_databse[name]=[[book_name],b_date,[book_price]]
                #decreasing the quantity of borrowed book by 1
                library[book_id][2]=str(int(library[book_id][2])-1)
            else:
                print ("Book not in stock.")
        else:
            print("Book id doesnot exist in stock.")
```

Figure 9: Borrow Module

In borrow module, firstly a function is defined called borrow where library, entered name and an empty dictionary(borrowed_database) is passed as parameter. The program is dropped into a loop. The user is asked to either input the book ID or to exit. If user wants to exit the library and database is returned, else, the provided book Id is checked in the keys of the dictionary. If the book Id exists, again the quantity of the book is checked. If the quantity is not 0, user can borrow the book, else a suitable output is displayed. Borrowed date is asked with the user and whole information is added to the borrowed database. The quantity of book is decreased by 1 and the changed and stored value of library and borrowed database are returned.

Returned Module(return_module.py):

```
import datetime
def returns(library,name,borrowed_database,returned_database):
    if not(name in borrowed_database):
        print("This person has not borrowed books from us.")
        return [library,returned_database]
    sorted=True
    while sorted==True:
        try:
            book_id=input("Please enter the book-id or n to exit: ")
        except:
            print("error")
            continue
        if book_id=="n":
            print("The book has been returned successfully.")
            return [library,returned_database]
        else:
            b_date=borrowed_database[name][1]
            book_list=borrowed_database[name][0]
            r_date=datetime.date.today()
            diff=r_date-b_date
            borrowed_days=int(diff.days)
            if library[book_id][0] in book_list:
                book_name=library[book_id][0]
                book_price=float(library[book_id][3][1:])
                if borrowed_days<=10:
                    fine=0.0
                else:
                    fine=(borrowed_days-10.0)*0.1*book_price
                if name in returned_database:
                    returned_database[name][0].append(book_name)
                    returned_database[name][1].append(book_price)
                    returned_database[name][2].append(fine)
                else:
                    returned_database[name]=[[book_name],[book_price],[fine]]
                library[book_id][2]=str(int(library[book_id][2])+1)
            else:
                print ("The book id entered can't be found in persons record.")
```

Figure 10:Return Module

In return module, date is imported from the device. A function is defined i.e. returns where library, name, borrowed database and an empty dictionary(returned database) is passed as parameter. If the user wants to return the book, firstly, the name is checked in borrowed database, if the name is present the person can return the book, else, a suitable output is given. Again, after the user gives book Id, the book Id is checked in the user's database. If it is present, the information is stored in returned database, else,

again the book Id is asked. Lastly, library and returned database is returned from the function.

Writing Note Module(writetofile_module.py):

```
def write_to_txt(file_name,data_to_write,file_picker):
    file=open(file_name,"w+")
    for keys,values in data_to_write.items():
        if file_picker=="1":
            file.write(keys+","+values[0]+","+values[1]+","+values[2]+","+values[3]+"\n")
        elif file_picker=="b":
            file.write("*********************************************************************************\n")
            file.write("Borrow Note:\n\n")
            file.write("Name: "+keys+"\n")
            file.write("Book Borrowed\t\tDate\t\tPrice\n")
            sum=0.0
            for x in range(0,len(values[0])):
                file.write(str(values[0][x])+"\t\t"+str(values[1])+"\t\t"+"$"+str(values[2][x])+"\n")
                sum+=values[2][x]
            file.write("Grand Total: \t\t"+"$"+str(sum)+"\n")
            file.write("*********************************************************************************\n")
        elif file_picker=="r":
            file.write("*********************************************************************************\n")
            file.write(" Return Note:\n\n")
            file.write("Name: "+keys+"\n")
            file.write("Book Returned\t\tPrice\t\tFine\n")
            sum=0.0
            for x in range(0,len(values[0])):
                file.write(str(values[0][x])+"\t\t"+"$"+str(values[1][x])+"\t\t"+"$"+str(values[2][x])+"\n")
                sum+=values[2][x]
            file.write("Total Fine: \t\t"+"$"+str(sum)+"\n")
            file.write("*********************************************************************************\n")
    file.close()
```

Figure 11: Writing Note

This module is only for writing notes in the txt files. Here, file name, data to write and picker is passed in the parameter and the note is generated in a well-presented manner.

Testing:

Test 1:

```
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
1
Book id         Book name               Quantity                Price
B001            Harry Poter             26              $2
B002            Start With Why          0 <- Not in stock.$1.5
B003            Programming With Python         20              $1.5
B004            Summer Love             14              $3
B005            Romeo and Juliet                40              $4
B006            Famous Last Words               25              $2.5
B007            Great Expectations              30              $3
```

Figure 12. Display function

```
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
2
Enter your name: Ramesh Shrestha
Please enter the book-id or n to exit: B002
Book not in stock. <- Does not allow the user to borrow.
Please enter the book-id or n to exit: B001
Enter borrowed date (mm/dd/yyyy): 12/20/2018
You have successfully boroowed the book <- Allows to borrow.
Please enter the book-id or n to exit: n
```

Figure 13: Demonstration of Test 1

Test Number	**1**
Action	Input Book Id which is not in stock.
Expected Result	Not to let the user borrow the book.
Actual Result	Does not allow to borrow and gives a suitable output.
Test Result	Test Successful

Table 1:Test 1 Demonstration

Test 2:

```
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
2
Enter your name: Ramesh Shrestha
Please enter the book-id or n to exit: B003
Enter borrowed date (mm/dd/yyyy): 12/20/2018
You have successfully boroowed the book
Please enter the book-id or n to exit: n
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
4
>>>
```

Figure 14: Choice to Borrow

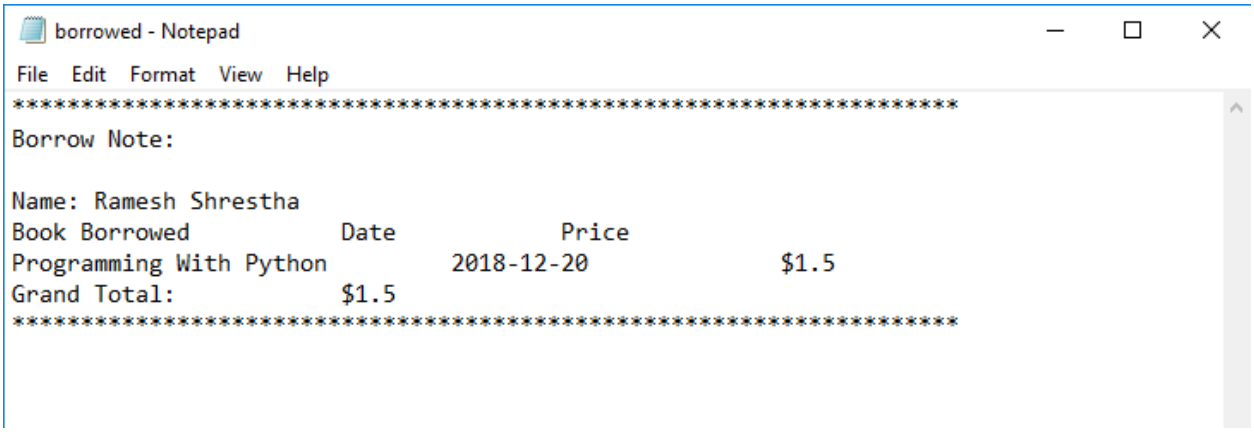


Figure 15: Generated borrow note

Test Number	**2**
Action	User was allowed to borrow a book.
Expected Result	To generate a note for borrower.
Actual Result	Generated the note with date and price.
Test Result	Test Successful.

Table 2: Test 2 Demonstration

Test 3:

```
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
2
Enter your name: Ramesh Shrestha
Please enter the book-id or n to exit: B003
Enter borrowed date (mm/dd/yyyy): 12/22/2018
You have successfully boroowed the book
Please enter the book-id or n to exit: n
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
3
Enter your name: Suraj Shrestha
This person has not borrowed books from us.
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
```

Does not allow the user to return the book.

Figure 16: Demonstration of Test 3

Table Number	**3**
Action	User who hasn't brought the book wants to return the book.
Expected Result	Should not let the user return the book.
Actual Result	Does not allow to return the book and displays suitable message.
Test Result	Test Successful.

Table 3: Test 3 Demonstration

Test 4:

```
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
2
Enter your name: Ramesh Shrestha
Please enter the book-id or n to exit: B001
Enter borrowed date (mm/dd/yyyy): 12/22/2018
You have successfully boroowed the book
Please enter the book-id or n to exit: n
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
3
Enter your name: Ramesh Shrestha
Please enter the book-id or n to exit: B001
Please enter the book-id or n to exit: n
The book has been returned successfully.
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
4
>>>
```

Book was returned.

Figure 17: Book Returned

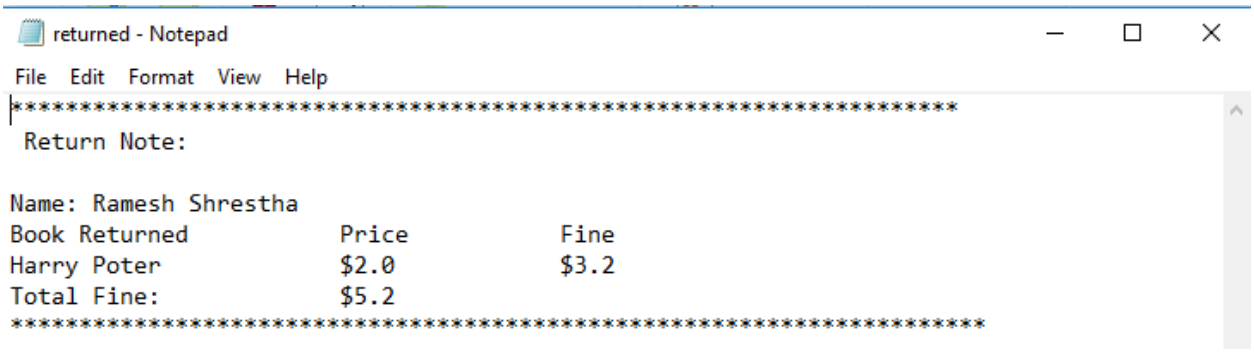


Figure 18:Generated Returned Note

Table Number	**4**
Action	Let the user return the book.
Expected Result	Return note is generated.
Actual Result	Return note was generated in suitable format.
Test Result	Test Successful

Table 4: Test 4 Demonstration

Test 5:

```
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
1
Book id         Book name               Quantity                Price
B001            Harry Poter             30 <-           $2
B002            Start With Why          0               $1.5
B003            Programming With Python         20              $1.5
B004            Summer Love             15              $3
B005            Romeo and Juliet                40              $4
B006            Famous Last Words               25              $2.5
B007            Great Expectations              30              $3


1) Display Books 2) Borrow Books 3) Return Book 4) Quit
2
Enter your name: Ramesh Shrestha
Please enter the book-id or n to exit: B001
Enter borrowed date (mm/dd/yyyy): 12/22/2018
You have successfully boroowed the book
Please enter the book-id or n to exit: n
1) Display Books 2) Borrow Books 3) Return Book 4) Quit
4
>>>
```

Figure 19: Borrowed Book

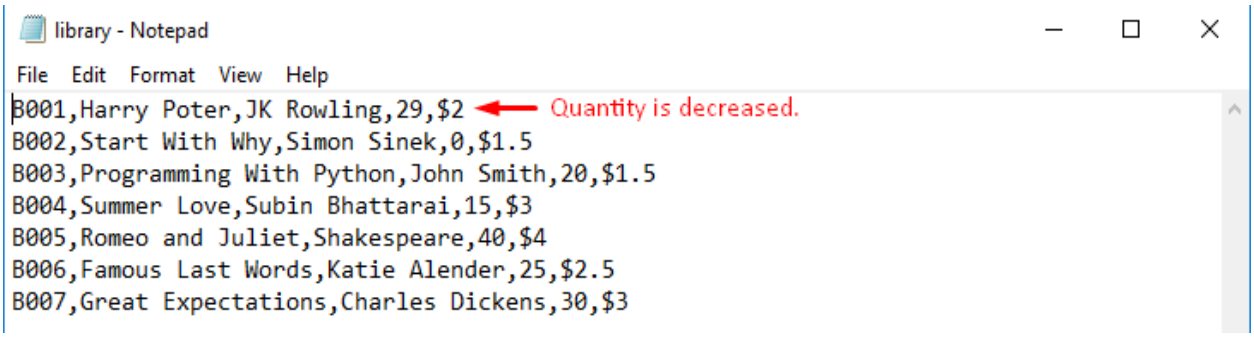


Figure 20: Demonstration of Test 5

Table Number	**5**
Action	Borrow a book.
Expected Result	Quantity of the book is decreased.
Actual Result	Quantity was decreased to 29 from 30.
Test Result	Test Successful.

Table 5: Test 5 Demonstration

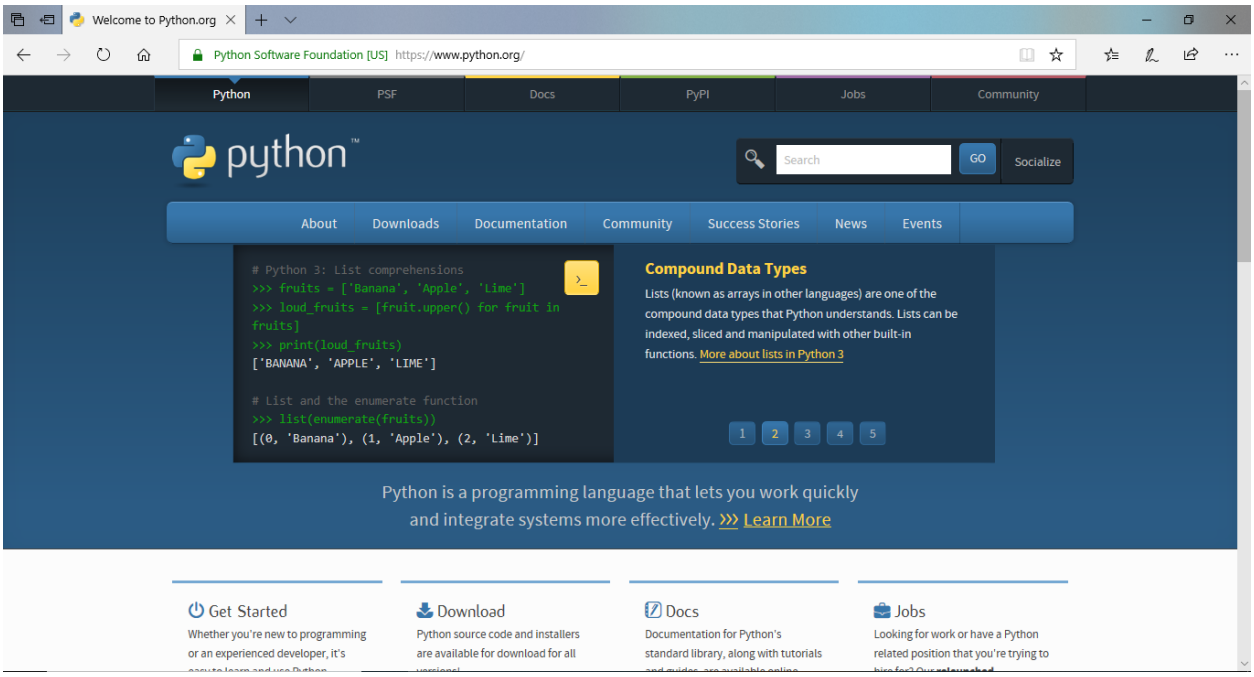
Research:

Constant effort was done along with many researches, in order to complete the given coursework within the given time. Many websites, journals and books were being accessed and research was carried out to bring up this final project. Self-effort and hard work had to be done with the enthusiasm of the work. After a lot of research and effort the final project is ready to carry out its function. In fact, without any research, this project is not possible because we need to know different functions of the python to make the program efficient. While debugging the program, the research from different websites, helped me to tackle them. By research, I could sketch out skeleton of the program and led it to success with my full concentration. These researches made my concept clear about modules and functions of python programming.

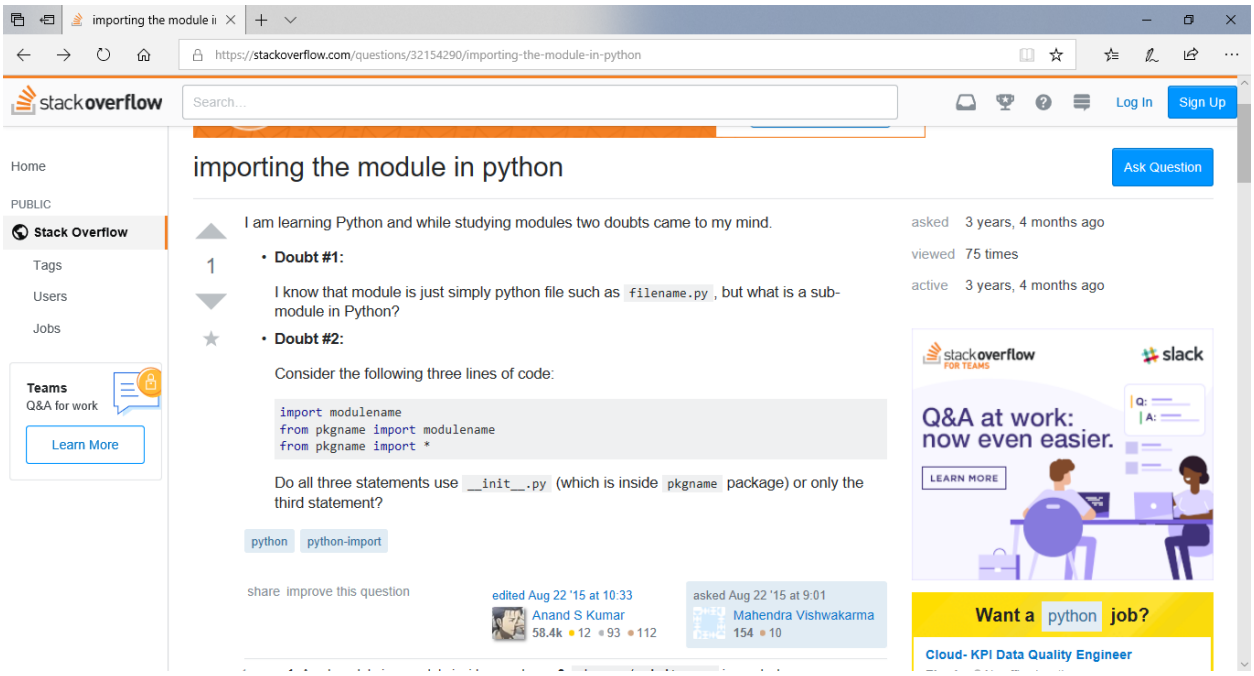
Some of the websites, books and journals, I researched from, are given as below:

- **Websites:**

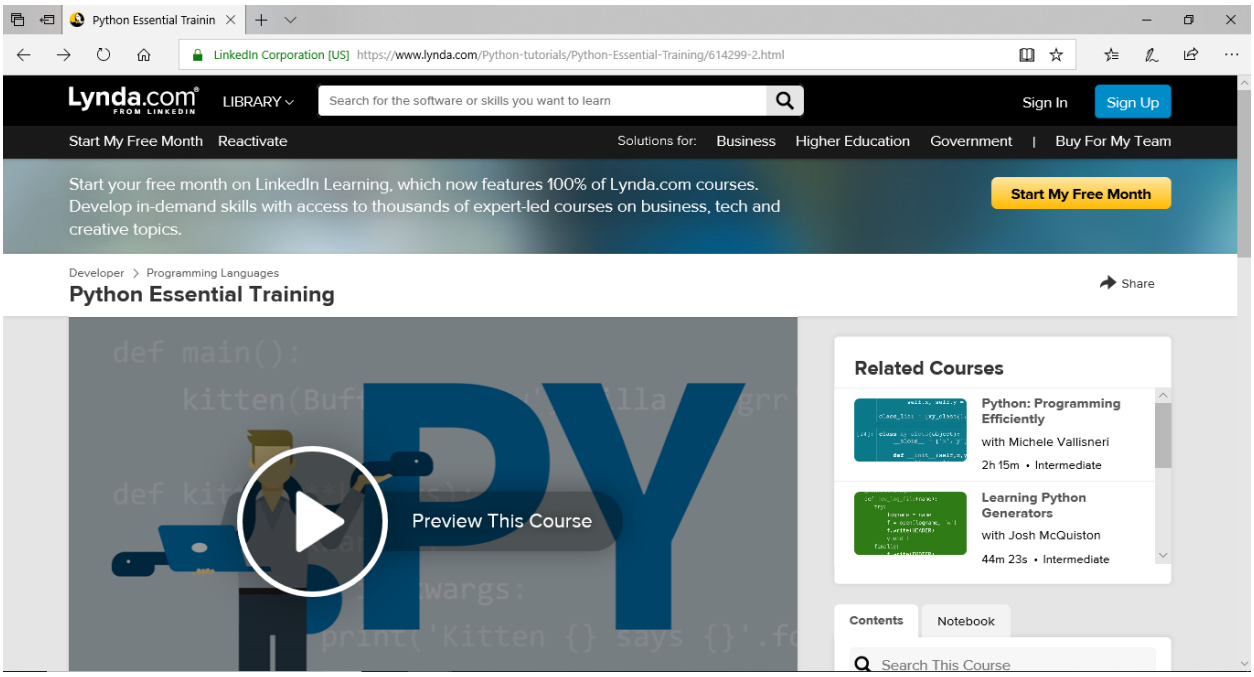
1. https://www.python.org/



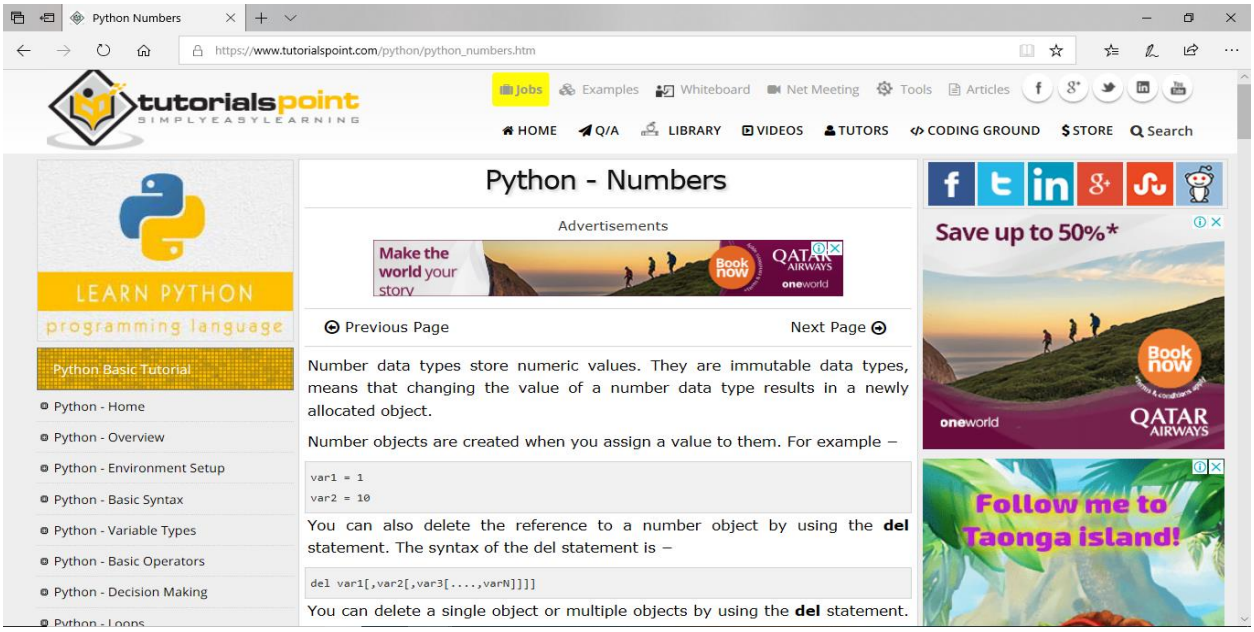
2. https://stackoverflow.com/



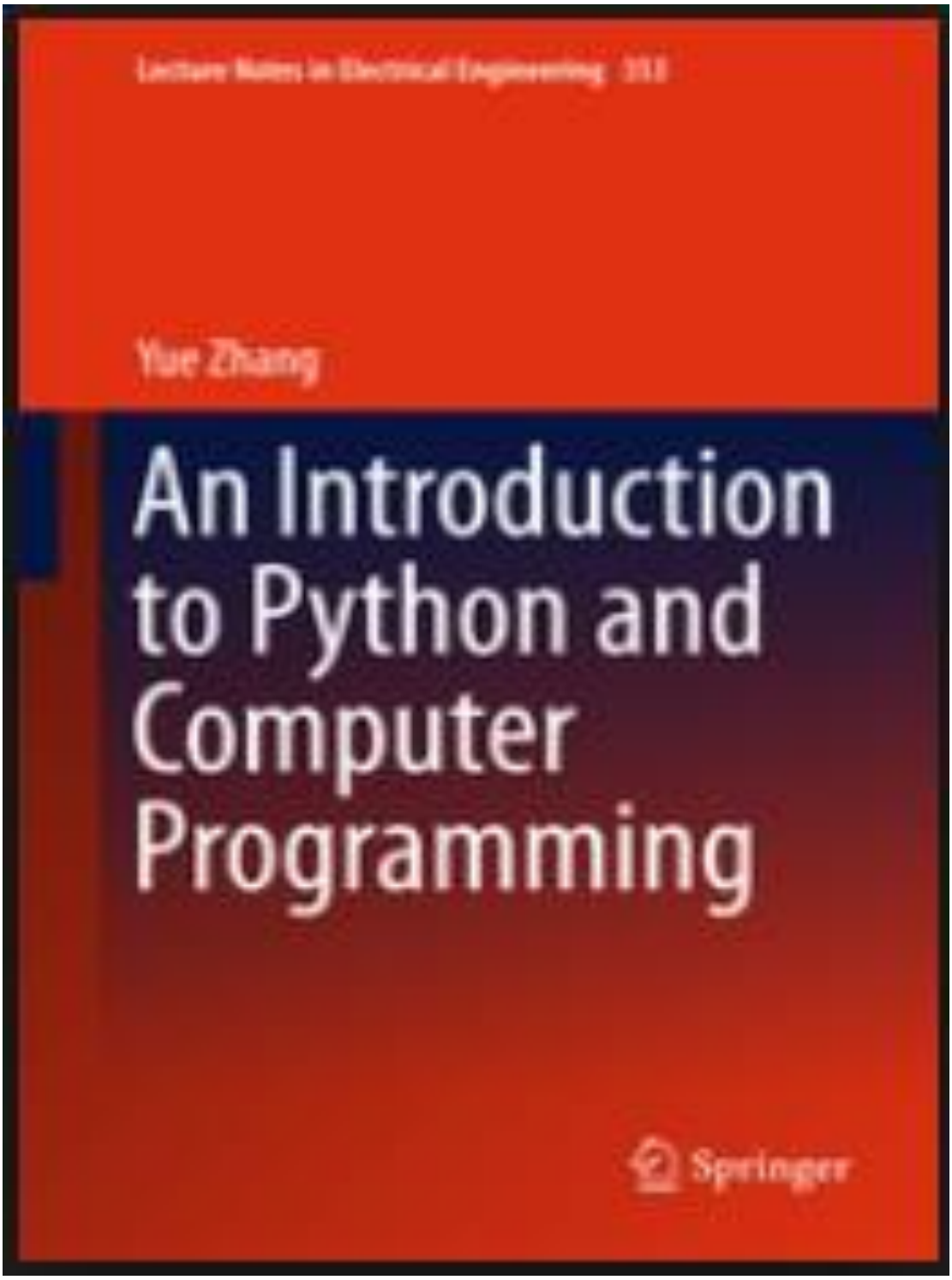
3. https://www.lynda.com/



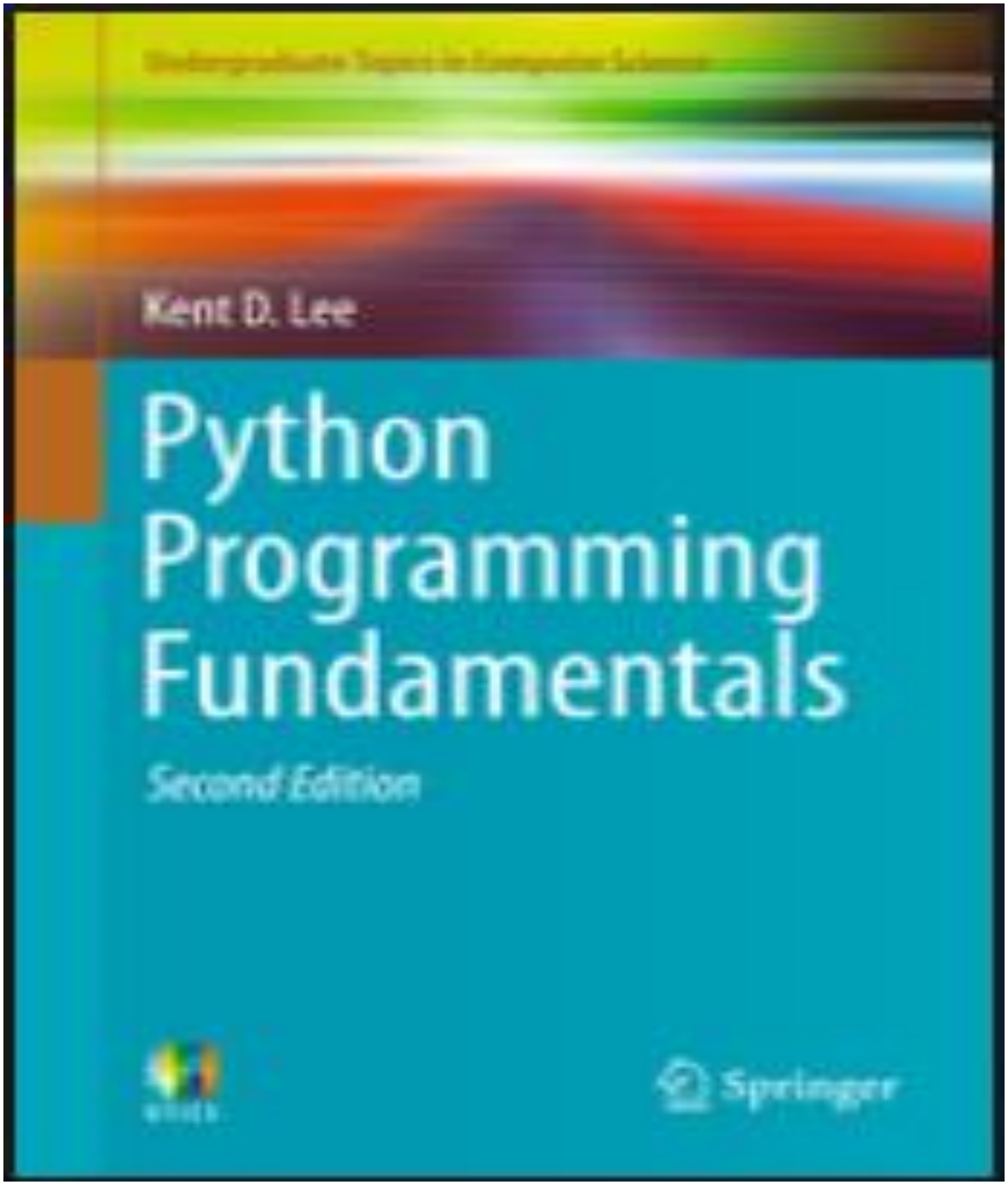
4. https://www.tutorialspoint.com/



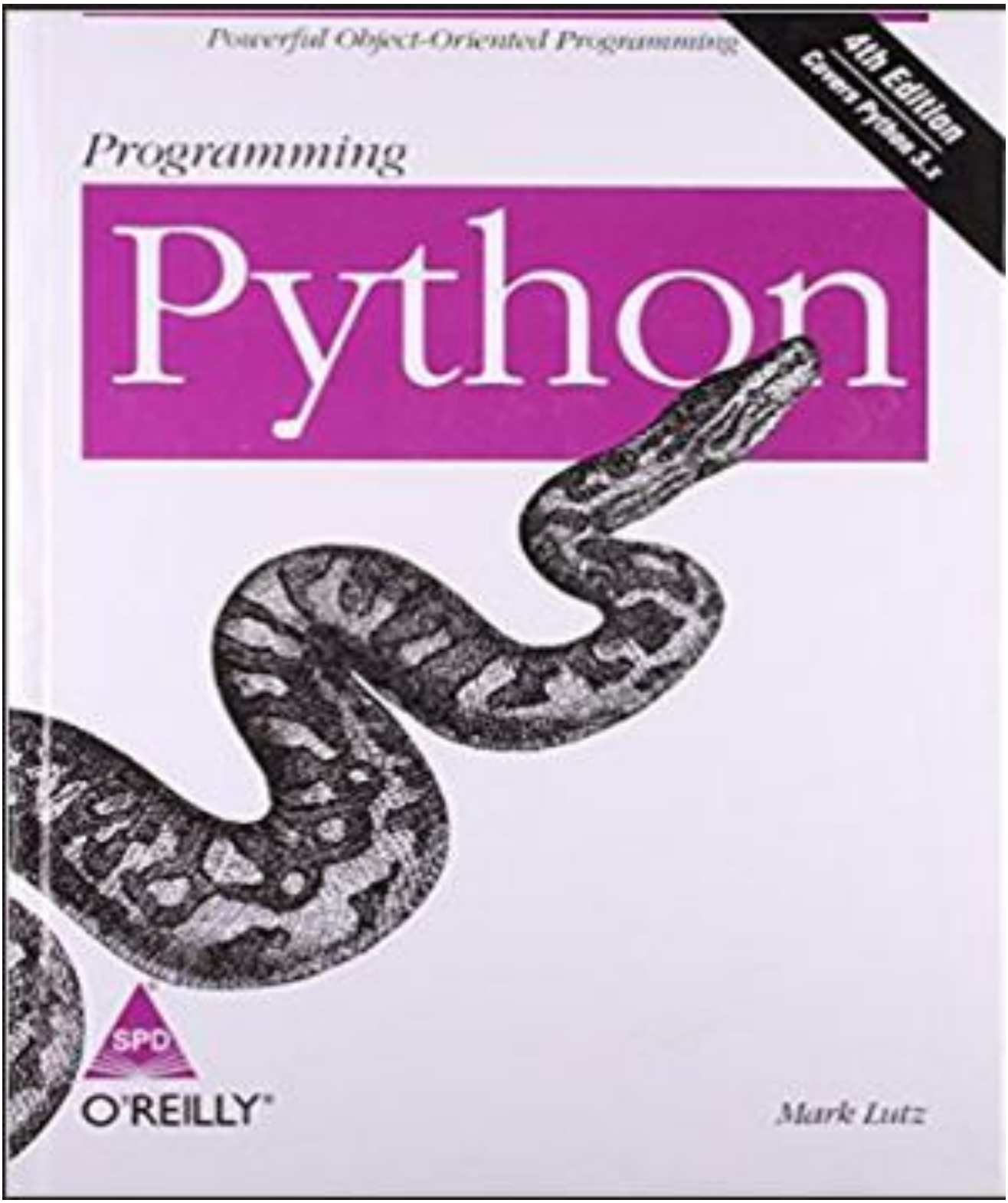
- **Books:**
  1) An Introduction to Python and Computer Programming



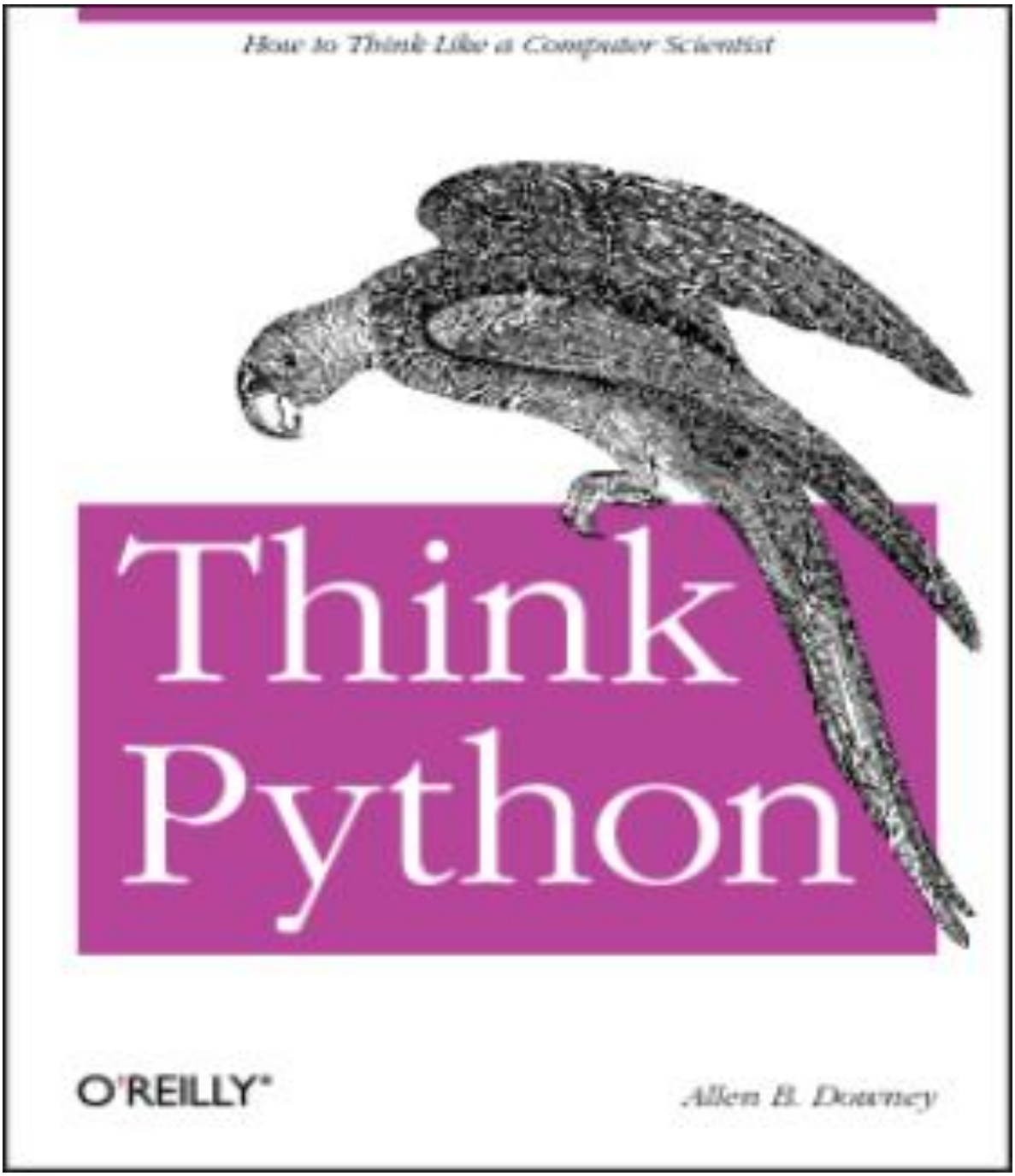
2) Python Programming Fundamentals



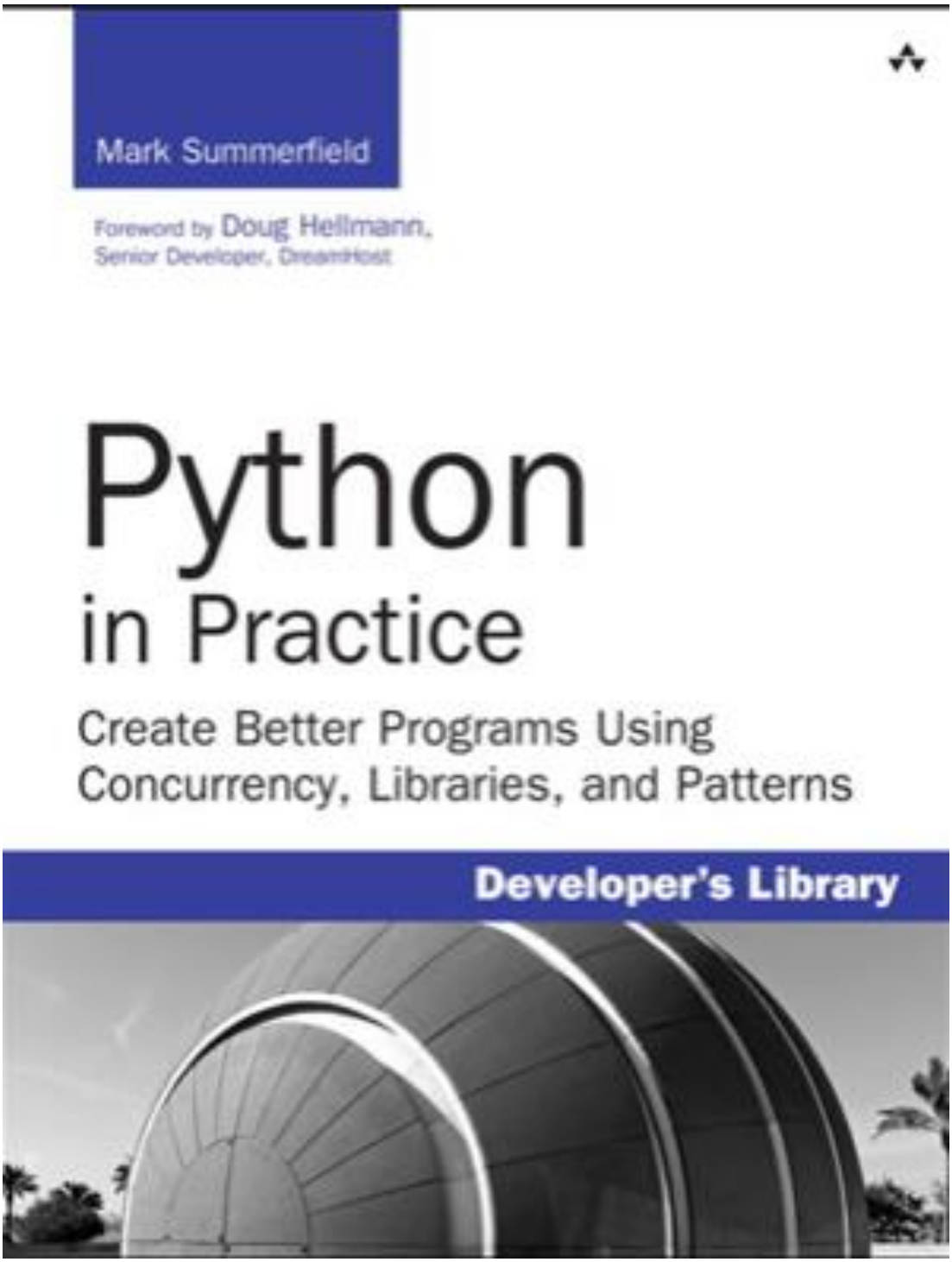
3) Programming Python



4) Think Python:



5) Python in Practice:



- **Journals:**

I. Programming with Python (Lindstrom, 2005)
II. The First Language - A Case for Python? (Jenkins, 2015)
III. On the performance of the Python programming language for serial and parallel scientific computations (Langtange, n.d.)
IV. An Introduction to Programming for Bioscientists: A Python-Based Prime (Charles E. McAnany, 2016)

Conclusion:

After the success of the project, the concept of module became clear. Many problems were faced while making the program but with research and hard work, the program could be debugged. With the help of many medias such as journals, websites and book research, the final project was possible. The aim of the project was not only to create a python program, but to acknowledge different library functions. The main motto of this program is to develop a habit of programming.

While starting the research, basic concept was only implemented but as soon as the research was done, many more concepts came in and programming became easier. Firstly, small programs were made, which was quite easy, but when the concept of module division came in, the program faced a lot of errors. The errors could be debugged with the help of researches. The main problem was to fix errors on program. It took almost 2 weeks to achieve the final program but writing the documentation was a bit easy. By understanding the program, documentation becomes easier.

Many websites and journals were accessed to overcome the obstacles of the program. Discussion with the lecturers became useful for the completion of the project. Also, vital role was played by the lecture slides provided by our teachers. Visiting the websites again and again, made the doubt clear. Hence, with many researches and hard-work, the program was completed along with the documentation.

Works Cited

Anon., 2019. *Python Dictionaries.* [Online]
Available at: https://www.w3schools.com/python/python_dictionaries.asp

Anon., 2019. *Python-Lists.* [Online]
Available at: https://www.tutorialspoint.com/python/python_lists.htm

Charles E. McAnany, 2016. An Introduction to Programming for Bioscientists: A Python-Based Primer.

Jenkins, T., 2015. The First Language - A Case for Python?. *Innovation in Teaching and Learning in Information and Computer Sciences ,* pp. 1-9.

Langtange, H. P., n.d. On the performance of the Python programming language for serial and parallel scientific computations. *Scientific Programming,* pp. 31-56.

Lindstrom, G., 2005. Programming With Python. 7(5), pp. 10-16.