

**Module Code & Module Title**

**CC4002NI Information Systems**

**Assessment Weightage & Type**

**20% Individual Coursework**

**Year and Semester**

**2018-19 Autumn**

**Student Name: Ayush Amatya**

**London Met ID:**

**College ID: NP01CP4A180129**

**Assignment Due Date: January 18 2019**

**Assignment Submission Date: January 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 mark of zero will be awarded.*

**Proposal:**

This proposal is written to address the second coursework of Information System. This coursework is an individual assignment which was given to us in 8th week after the start of our module and is to be completed and submitted by 11th week. It accounts for 20% of our total module grade. This coursework is about developing an application that can be used in any bookstore which stores all the transactions like borrowing and returning that occurs in a bookstore.

* **Purpose:**

The purpose of completing this coursework is to be able to develop any simple application that does specific activities by using python. The application developed in this coursework will read the text file and display all the books available for borrowing. Then with each transaction (borrowing) a note is generated for the particular borrower and is written into a file. The stock of the books is also updated after each transaction.

* **Problem Statement:**

Many bookstores and library do not make use of computer system. They still use note copy and register books in order to store the data of transactions. This method can turn out to be very hard and ineffective if we need to access a very old transaction or to check records of individual person or to check to books that are remaining to be returned. Writing the records in this method is also very time consuming and hard labor. Also, if a customer borrows many books from the store, then additional time is needed to calculate the total price to be paid and create the bill. When the order is big, then there is a possibility to create some errors by humans.

The main problem or order given to us by this coursework is to develop a library management system. We had to develop a software in python that works like a library management system and write the documentation to explain the working mechanism of our program. A detailed report to present our work was to be written.

* **Aims and objectives:**

The main aim of this coursework is to develop a library management system that accounts and records every transaction performed in a library which helps in proper management of the records helps to access any desired record with less effort.

To accomplish the aim of this coursework I made some week wise objectives so that the assignment would be completed in the given time period. Some of my objectives are as follows:

* Approach the literature and theoretical issues related to the project.
* Develop a list of activities that a program should perform.
* Develop a sampling frame of how the programs should run.
* Develop a strategy and design for data collection and analysis.
* Develop a design for displaying details and accepting data from user.
* Check the effectiveness of the program.
* Deal with the problems in the program.
* **Proposed approach:**

To solve all the problems given in our coursework, I took many approaches. Some of them are as follows:

* Lots of research on relevant topics like algorithms, pseudocodes, database, modules, in-built functions of python, etc.
* Prepare a pseudocode to write a sample frame of the program.
* Write the algorithm of how a program achieve its goal.
* Separate the program in different parts and develop separate small modules to achieve the goal of each part.
* Connect all the modules and develop a complete software using different functions of python effectively.
* Test the program with every possible inputs and check for errors.
* Identify and solve errors and bugs.
* Write a documentation of the program explaining its methods and use.
* **Target audience:**

The program developed in this project can be used by any books store or library that has facilities to let the customer borrow a book for curtain time interval with some fix rates. It helps the librarian and the owner of the bookstore to maintain the record of people who borrows or return the book. It makes them easy to calculate total price of the book borrowed, calculate total fine if they are late to return the book, know whether the book ordered by the customer is in our stock or not, etc. with little modification of this project, it can also be used in managing big markets.

* **Hardware and software requirements:**

In order to run this program, one needs a computer system of at least 2 GB RAM, a python application with tabulate.py in its library and a text editor. Tabulate.py can easily be installed from any browser. Its is to be copied in lib folder inside python folder.

* **Activities description and timeline:**

Only around 3 to 4 weeks were given to us to complete this massive coursework. I divided my time to complete this coursework such a way so that I would complete the work in time. My time management for this coursework is shown in the graph below:

Gantt Chart

**Table of Contents**

* [**Introduction:** 1](#_Toc535546563)
* [**Discussion and analysis:** 2](#_Toc535546564)
* [**Python** 2](#_Toc535546565)
* [**Lucid chart:** 3](#_Toc535546566)
* [**Excel:** 4](#_Toc535546567)
* [**Black box testing:** 4](#_Toc535546568)
* [**Algorithm:** 5](#_Toc535546569)
* [**Flowchart** 6](#_Toc535546570)
* [**Pseudocode:** 7](#_Toc535546571)
* [Main.py: 7](#_Toc535546572)
* [Read.py: 7](#_Toc535546573)
* [Operations.py: 8](#_Toc535546574)
* [display.py 9](#_Toc535546575)
* [write.py: 9](#_Toc535546576)
* [**Data structure:** 10](#_Toc535546577)
* [**Integer:** 10](#_Toc535546578)
* [**Float:** 12](#_Toc535546579)
* [**String:** 12](#_Toc535546580)
* [**Dictionary:** 12](#_Toc535546581)
* [**Sets:** 13](#_Toc535546582)
* [**Program:** 15](#_Toc535546583)
* [**Testing:** 28](#_Toc535546584)
* [**Research:** 31](#_Toc535546585)
* [**Conclusion:** 32](#_Toc535546586)
* [**Reference:** 33](#_Toc535546587)

## **Table of figures:**

[Figure 1 (Dawson, 2017): Python 2](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571041)

[Figure 2 Lucid Chart 3](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571042)

[Figure 3 Excel 4](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571043)

[Figure 4 Flowchart 6](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571044)

[Figure 5 List index 11](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571045)

[Figure 6 ss of main.py 15](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571046)

[Figure 7 ss of operations.py 16](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571047)

[Figure 8 ss of main.py 16](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571048)

[Figure 9 ss of operations.py 17](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571049)

[Figure 10 ss of operations.py 17](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571050)

[Figure 11 ss of operations.py 18](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571051)

[Figure 12 ss of display.py 18](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571052)

[Figure 13 ss of display.py 19](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571053)

[Figure 14 ss of display.py 19](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571054)

[Figure 15 ss of read.py 20](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571055)

[Figure 16 ss of write.py 20](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571056)

[Figure 17 SS of program 21](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571057)

[Figure 18 SS of program 21](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571058)

[Figure 19 SS of program 22](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571059)

[Figure 20 SS of program 22](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571060)

[Figure 21 SS of program 23](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571061)

[Figure 22 SS of program 23](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571062)

[Figure 23 SS of program 24](#_Toc535571063)

[Figure 24 SS of program 25](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571064)

[Figure 25 SS of program 25](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571065)

[Figure 26 SS of program 26](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571066)

[Figure 27 SS of program 26](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571067)

[Figure 28 SS of program 27](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571068)

[Figure 29 SS of daily record 27](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571069)

[Figure 30 Black box test 1 28](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571070)

[Figure 31 Black box test 2 28](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571071)

[Figure 32 Black box test 3 29](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571072)

[Figure 33 Black box test 4 30](file:///F:\islington\Information%20System\CourseWork\CW%202\Documentation%20AYUSH.docx#_Toc535571073)

# **Introduction:**

This project report was done to develop a library management system. The main propose of this project is to make the librarian and bookstore owner easier to maintain and keep record of all the transactions easily.

Many bookstores and library do not make use of computer system. They still use note copy and register books in order to store the data of transactions. This method can turn out to be very hard and ineffective if we need to access a very old transaction or to check records of individual person or to check to books that are remaining to be returned. Writing the records in this method is also very time consuming and hard labor. Also, if a customer borrows many books from the store, then additional time is needed to calculate the total price to be paid and create the bill. When the order is big, then there is a possibility to create some errors by humans.

To overcome these problems, an application is developed in this coursework. This program is capable of many things. This program records every transaction in a library. It displays all the available books for borrowing to the user. If the customer borrows the book then, it generates a bill containing the name, date and time, name and details of the book borrowed and total price to be paid to borrow that book. The quantity of book in stock is updated after each book borrowed. If the customer has already borrowed a book previously, then this program first asks them whether they want to returned the borrowed book or they want to borrow another book. For returning, this program also displays all the borrowed book by that customer and updates the quantity in stock after each book returned. It also calculates the total fine applied to the customer if they are late to return the book. This program also stores the name of the book which are left to be returned along with the detail of the borrower. Additionally, this program records the daily transaction done in the library. It creates a new file with the date as the file name and store all the transactions of that day in the file.

# **Discussion and analysis:**

In order to complete this coursework many steps and methods we used. Firstly, I created some objectives to achieve in curtain time period. I managed the week wise objectives so that I would complete my assignment the given time duration. I researched many books and articles related to programming languages and database which may be helpful in creating this project report. I created a frame structure describing all the activities that my program should perform. I created the pseudocode for all the modules including all the process that should be carried out to write the program codes. I developed step-by-step algorithm for the main module, clearly showing all the process and direction of flow of the program. To explain the direction flow of the program and working mechanism of the program more broadly, I also drew the flow chart following the steps of the algorithm.

Some of the tools and technologies used to complete this coursework are as follows:

* **Python**:

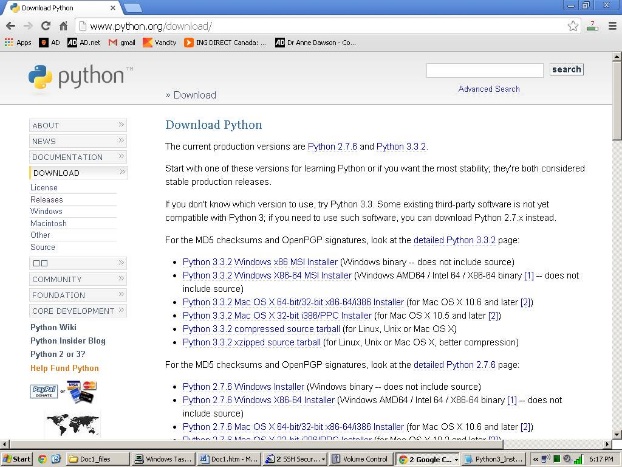
This program developed in this coursework was design in python. These types of program can also be created other programming languages. But I choose python as my programming language as it is easy to use and user friendly. For using python for writing program, at first, the latest version python was downloaded from the official python website and installed in the computer system. After the successful installation of python, IDLE (Integrated Development Environment) which is a part of python package was used as a text editor. Python is a high-level interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python makes a quicker development cycle as we just type our code and run it, without the intermediate compilation step. As python is interpreted and nor compiled into native machine instructions, codes written for one platform will work on any other platform that has the python interpreter installed. Hence python is portable. I used python to develop the program of this coursework as it is simple. As programming languages go, python is relatively uncluttered, and the developers have deliberately kept it that way*.* (Sturtz, 2018)

Figure 1 (Dawson, 2017): Python

## **Lucid chart:**

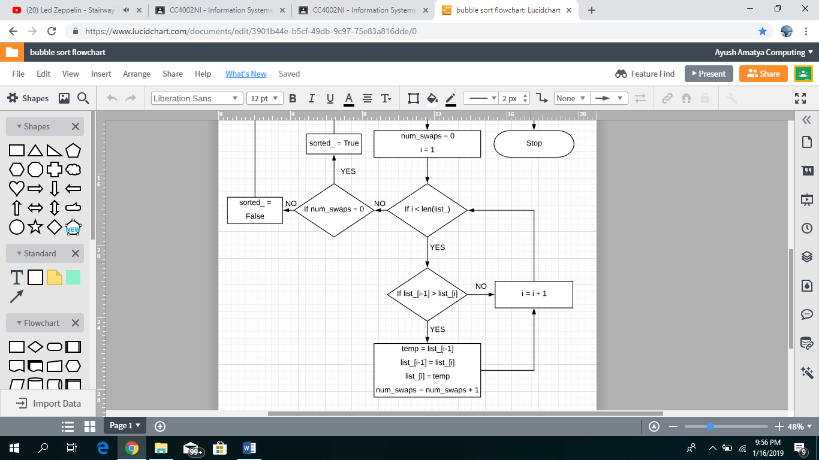
I drew the flowchart for this coursework with the help of online lucid chart. Lucid chart is clean and simple user interface. Lucid chart is a cloud-based service that allows user to create diagrams of various types. It is also known as flowchart software. It has possibility to create organized charts, prototypes, wireframes, etc. The current software of lucid chart was released in 2010 which is now the main competitor of Microsoft Visio. Being both a collaboration and flowchart creating tool, it has various other features like: being a cloud-based tool, collaboration, several options to save results, a diversity of shapes, customization, compatibility and integrations. This application guarantees the security of information by means of data encryption, robust backup and secure data centers. Many reviews refer lucid chart to the best flowchart software thanks to its intuitively comprehensive user interface. Since it is a web-based tool, we can access our account and files from any computer. However, we cannot access our charts without internet. It also has an excellent revision history which allows us to roll back edits or start a new document version(Fox, 2017)

Figure 2 Lucid Chart

## **Excel:**

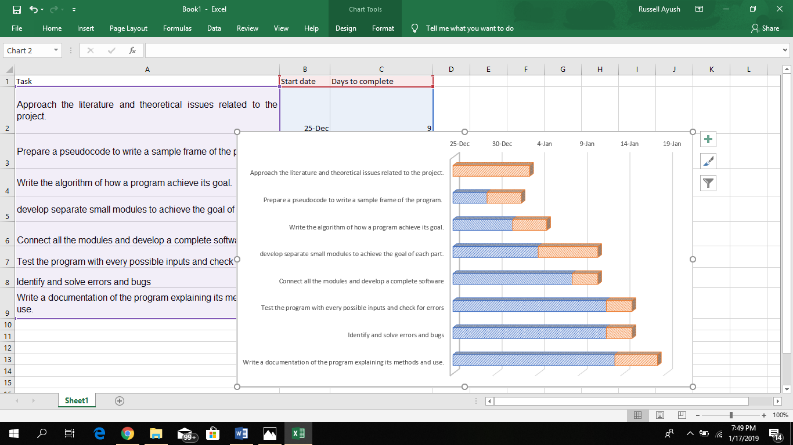
I made the Gantt Chart that shows all my time management and work division in a graph for this coursework using Excel. A Gantt chart shows all the activities of a project and their duration as a bar chart, with the timescale at the top or bottom. The activities are placed on the bar chart in sequence, starting in the top left-hand corner, and following their precedence rule. A Grant diagram in Excel represents tasks in the form of horizontal bar charts. It does not have a built-in template as an option, but we can create Gantt chart using the bar graph function of MS Excel.(Cheusheva, 2018)

Figure 3 Excel

## **Black box testing:**

Black box testing is used in python to test whether the program has any errors or not. Black box testing belongs to a testing called unit testing. In black box testing each function and module is tested separately. This type of testing only checks the input and output structure of the program. Black box testing finds errors such as:

1. Incorrect or missing function
2. Interface errors
3. Errors in data structures
4. Behavior or performance error
5. Errors such as logical errors

# **Algorithm:**

An algorithm is the step-by-step procedure or formula for solving a specific problem. In programming, an algorithm provides a step-by-step instruction about how a program should be coded and written. By the help of given above algorithm of the instruction to write a library management system we can easily get idea about how the program should be coded.

STEP 1: Start

STEP 2: Ask and store user name

STEP 3: Read file where the details of books is stored and store it in a 2D list.

STEP 4: Read file where the details of borrower is stored and store it in a 2D list

STEP 5: Ask does user wants to borrow book? (y/n). If yes go to 6 else go to 10

STEP 6: Display books available for borrow

STEP 7: Ask user which books he want to borrow

STEP 8: Display bill including his name, date and time and details of book borrowed

STEP 9: Decrease the stock of borrowed book in from the file

STEP 10: Ask user wants to return the book or not? If yes go to 11 else go to 16

STEP 11: Display the books he had borrowed from the 2D list

STEP 12: Ask user which book he wants to return

STEP 13: Display a note with his name, date and time and details of book returned

STEP 14: Increase the stock of returned book in 2D list

STEP 15: Check whether he exceeded 10 days or not. If yes go to 16 else go to 17

STEP 16: Apply fine for each day late and display total fine

STEP 17: Re-write 2D list of books in stock in the file

STEP 18: Write the detail of transaction performed in a new file

STEP 19: Re-write 2D list of details of borrower in the file

STEP 20: Ask user wants to do next transection or not? If yes go to 2 else go to 21

STEP 21: Stop

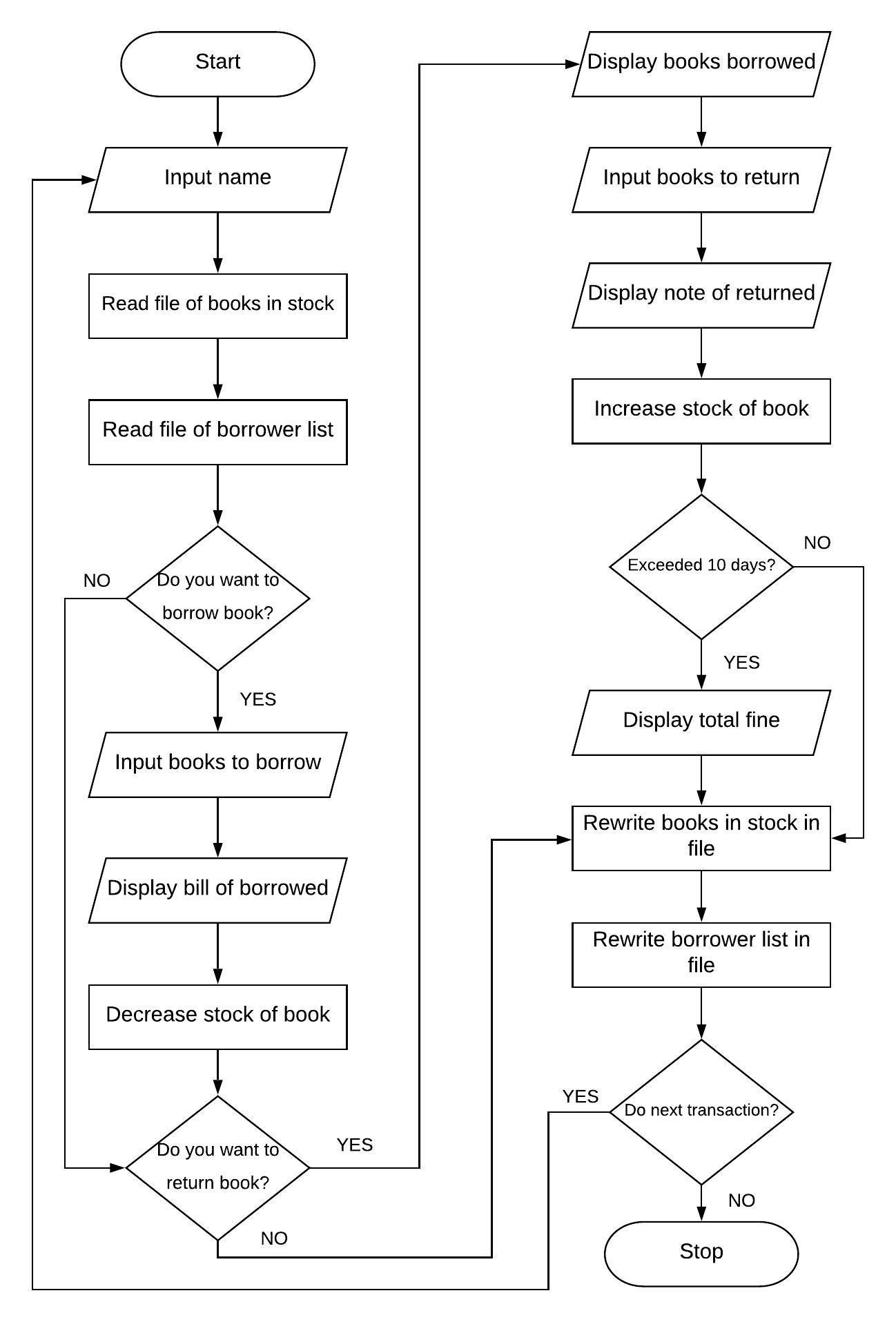
******Flowchart**: Flowchart design according to the above algorithm is given below:

Figure 4 Flowchart

## **Pseudocode:**

Pseudocode is a detailed description of what a computer program or algorithm must do, expressed in a formal high-level language rather than a programming language. (Rouse, 2016). I created one main module and many sub modules and functions to complete this project report. First, I wrote the pseudocode of all those module and functions before writing down its codes.

### Main.py:

Import read as r

Import write as w

Import operations as o

Import display as d

New = ’y’

While new == ‘y’:

Input name

Books\_for\_borrow=r.read\_item(“books\_in\_store.txt”)

Borrower\_list=r.read\_item(“borrowed.txt”)

Borrowed\_already=o.books\_he\_took(name,borrower\_list)

R\_b = input return or borrow

If r\_b == borrow:

Borrow\_now=d.display\_available(books\_for\_borrow,borrowed\_already)

books\_for\_borrow=o.edit\_stock("REDUCE",books\_for\_borrow,borrow\_now)

w.write\_books\_list("books\_in\_store.txt",books\_for\_borrow)

total = o.bow\_bill(name, date, borrow\_now, time)

w.write\_borrower\_list(name, date, time, borrow\_now, total, borrower\_list)

else:

list\_to\_return=d.display\_books\_took(borrower\_list, name)

return\_now= d.books\_returned(list\_to\_return)

returned\_details = o.detailed\_returned(list\_to\_return, date, return\_now, name, borrower\_list)

books\_for\_borrow=o.edit\_stock("ADD",books\_for\_borrow,return\_now)

w.write\_books\_list("books\_in\_store.txt",books\_for\_borrow)

fine = o.ret\_bill(name,time, date, return\_now,returned\_details)

borrower\_list= o.list\_after\_ret(borrower\_list,name,return\_now)

w.write\_borrower\_list(name, date, time, [], 0, borrower\_list)

end if

new = input new customer? (y/n)

### Read.py:

* Read\_item(file\_name):

List2 = []

Open file\_name, r

For lines in file\_name:

List1 = []

List1.append lines.split “,”

List2.apped list1

File.close

Return list2

### Operations.py:

* Bow\_bill(name, date, borrow\_now, time):

Output name, date, time

i = 1

total=0.0

for books in borrow\_now:

output i, books

total = total+books[2]

output total

return total

* Edit\_stock(to\_do,books\_for\_borrow,borrow\_now):

If to\_do == “REDUCE”:

For book in borrow\_now:

For list1 in books\_for\_borrow:

If book[0] == list1[0]:

List1[2] = List1[2]-1

Else:

For book in borrow\_now:

For list1 in books\_for\_borrow:

If book[0] == list1[0]

List1[2] = list1[2]+1

Return books\_for\_borrow

* list\_after\_ret(borrower\_list,name,return\_now):

for book in return\_now:

for list1 in borrower\_list:

if list1[0] == book:

list1.remove book

return borrower\_list

* books\_he\_took(name,borrower\_list):

list1=[]

for line in borrower\_list:

if name == line[0]:

for index = 4 to line.length – 1

list1.append line[index]

return list1

### display.py

* display\_available(books\_for\_borrow,borrowed\_already):

i=1

borrow\_now=[]

for book in books\_for\_borrow:

output i, book

i=i+1

book\_choose= input book to borrow

borrow\_now.append book\_choose

return borrow\_now

### write.py:

* write\_book\_list(file\_name, items):

open file\_name, w

for i in items:

file.write i.join “,”

file.close

return

# **Data structure:**

Data structures are a way of storing data in such a way that the stored data could be accessed and used easily anytime and anywhere in a program. The data structure also enables the programmers to perform various operations related to the data.

Python provide us various types of data structures where we can store data collectively. Some of the data structure that I used to in this program are:

1. Integer
2. List
3. Float
4. String
5. Dictionary
6. Sets

Short description of the working and uses of given data structures are as follows:

## **Integer:**

Integer is one of the data types provided by python. Any numeric value can be stored in integer. It can also be overwritten. The syntax to store an integer value is:

Syntax: <variable name>=<integer value>

We can also convert numeric string and other decimal numeric values in integer.

Its syntax is: int (<numeric string value>)

e.g. int (“5439”)

We can do all of the basic operations with integers like addition and subtraction using the standard plus and minus symbols. Multiplication uses the asterisk, and division uses a forward slash. Exponents use two asterisks.

In this program I have used integer in many places. I used integer to access to the specific index of the list that stores various data. I also used integer during loops when I needed to start the loop from specific number. In this program I also made the use of integer to display the serial numbers of many lists like: list of books available to borrow, list of books the customer had borrowed and many other places.

1. List:

List is one of the heterogeneous data structures that is capable of storing data of different types like: integers, string, float and Boolean collectively. The data stored in a list can be changed and modified, meaning being, list is a mutable data type. The list is stored between square brackets. i.e. [ ] and the elements are separated by comas. i.e. “,”. For example: list\_= [1, 2, a, b]

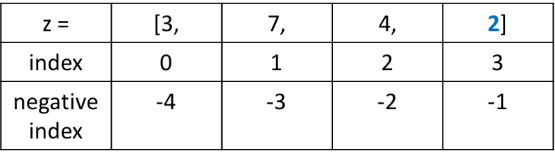


Figure 5 List index

* Indexing in list:

The unique position of where the element is stored in a list is known as its index number. The index number of a list always starts with 0 and ends at 1 less value of the total length of a list. List also supports negative indexing starting at -1 form the end. To operate in our desire element, we just need to know its index number. We can access or desired element by: <list\_name>[<index\_number\_of\_element>] for example: we can access ‘3’ in figure 4 by z [0] or z [-4].

* Iterating over a list:

The list can be iterated in many ways. Since it has index number that increases serially, for loop can be used to iterate over a list. Using for loop such that it stores the index number of the list, all the elements of the list can easily be accessed.

* In-built functions of list:

There are lots of in-built functions of list. Some of the common frequently used in-built functions of list are as follows:

List.append(x): add an item ‘x’ to the end of the list.

List.remove(x): removes the first item from the list whose value is ‘x’

List.pop (): removes and returns the last item in the list.

List.count(x): returns the number of times ‘x’ appears in the list.

Len(list): returns the total length of the list.

(Python Software Foundation, 2018)

* Use of list in this program:

In this program I used lists in many ways. To store the data of the file after reading a file I used 2D list. This made me very easier to access the required data anytime and anywhere. When I needed to update the quantities or any other data in the file, I updated them in the list and I re-wrote the list in the file. I used almost all of the in-built functions of lists like append, remove, and del to modify the data of the list in this program.

## **Float:**

Float is also a data types in python that stores the numeric value. It is similar to integer data type. But float data type can also store decimal numeric values whereas integer data type cannot.

In this project, I made use of float data type to store the total price of the book and the fine the customer had to pay if he is late to return the book. Since price of some of the books are decimal numeric values, I choose float datatype to store its value.

## **String:**

String is a data type in python which is used to store a textual data. Strings are contained by either single or double quotes. More precisely, string helps us to store a sequence of one or more characters (letters, numbers, symbols) that can be either a constant or a variable. We can also add strings in python by using the + symbol. The term "whitespace" refers to characters that the computer is aware of, but are invisible to readers. The most common whitespace characters are spaces, tabs” \t”, and newlines” \n”. We can iterate string similar as in list. Each letter of the string has its index values starting from 0. It also has many in-built functions like replace, split, etc.

## **Dictionary:**

Dictionary is a mutable data structure which are made up of key-value pair. Here ‘key’ is used to identify the item and ‘value’ is used to store the item. Here two items having a same ‘key’ name cannot be stored but the same value for two different key names can be stored. Dictionary are enclosed in curly braces. i.e. { } and separated by comma ‘,’. The data stored in dictionary can be changed and modified. For example:

Dic\_= {‘ayush’: 9818963833, ‘babin’: 9818322025, ‘nischal’: 971319488}

Here, ‘ayush’, ‘babin’ and ‘nischal’ are the key names and ‘9818963833’, ‘9818322025’ and ‘98741319488’ are the values stored in those key names respectively. To access the values stored in a dictionary, we must know the ‘key’ of that value. We can access them by: <dictionary\_name>[<key\_name>]. For example: in given above example, we can access the number of ayush by: Dic\_[‘ayush’]. There are many in-built functions of dictionary which makes us easier to work in dictionary. Dictionaries are exactly what you need if you want to implement something similar to a telephone book. None of the data structures that you have seen before are suitable for a telephone book.

In this project program, I used dictionary data type while user wants to return the books he borrowed. I stored “Name of book”, “Date borrowed”, “Date to return” and “customer name” as keys and stored list as its values. This made me easier to access data of particular person with their date of borrowed easier. It also made me easier to display the output in correct format using tabulate module of python where I used keys as the headers.

## **Sets:**

Sets are the unordered collection of distinct (unique) objects. It is also a mutable data structure where different data types can be stored. Since they are un-ordered, any specific order is not guaranteed. Meaning being, it has no concept of index. Sets are denoted by { }. For examples: a= {1, 2, 3, 4} and b= {3, 4, 5}

The set operations are similar to set operations in mathematics. Some of the commonly used set operations are:

a.union(b) # {1, 2, 3, 4, 5}

a.intersection(b) # {3, 4}

a.difference(b) # {1, 2}

b.difference(a) # {5}

len(a) # 4

In this project I used set while to remove all the common numbers when the program asks the user to input. If the user mistakenly enters multiple times the same numbers while borrowing or returning the books, I removed all the duplicate inputs using the facilities of the set as it only accepts unique elements.

# **Program:**

A library management program was developed in this coursework using python as the programming language. This program fulfills all the requirements that is needed in a well-managed library.

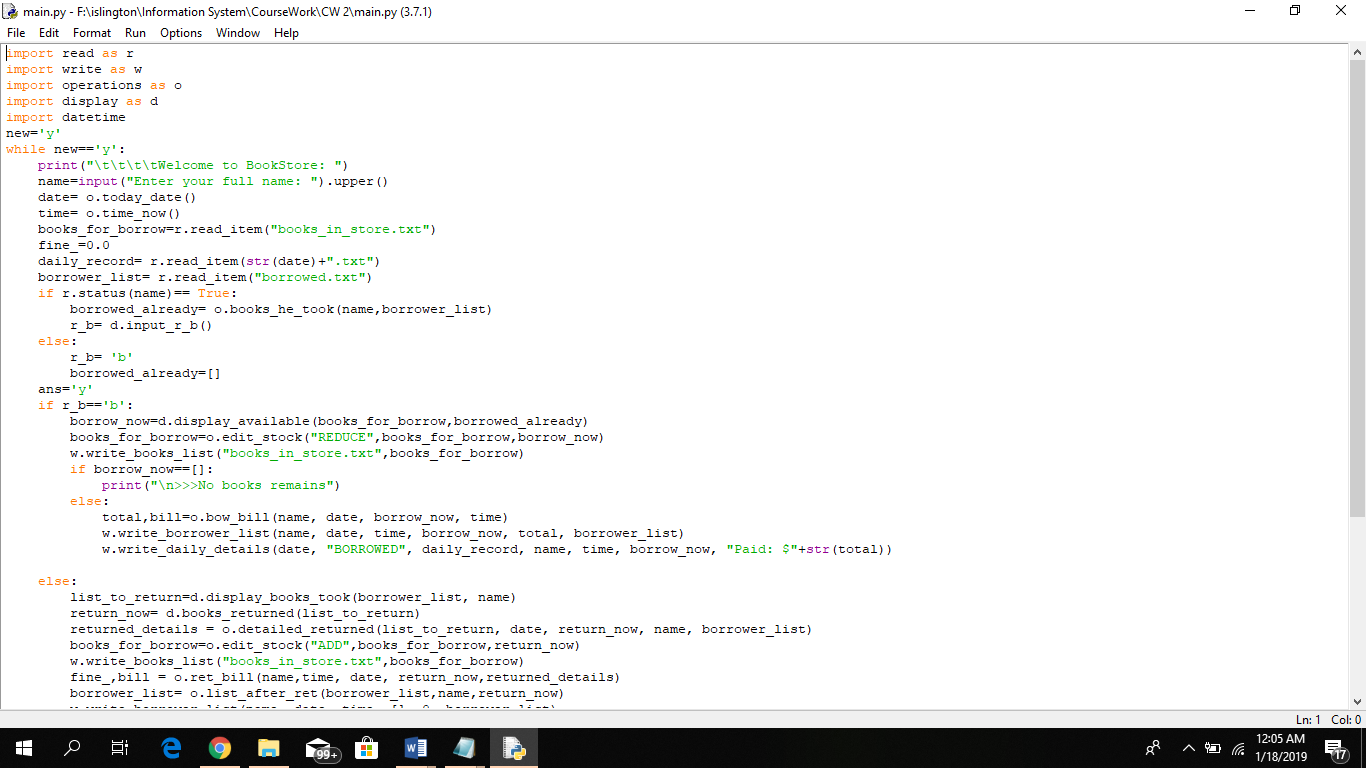
I have made one main module and 4 different sub modules of this program. All total I created 4 modules in this coursework names main.py, operations.py, display.py, read.py and write.py.

Figure 6 ss of main.py

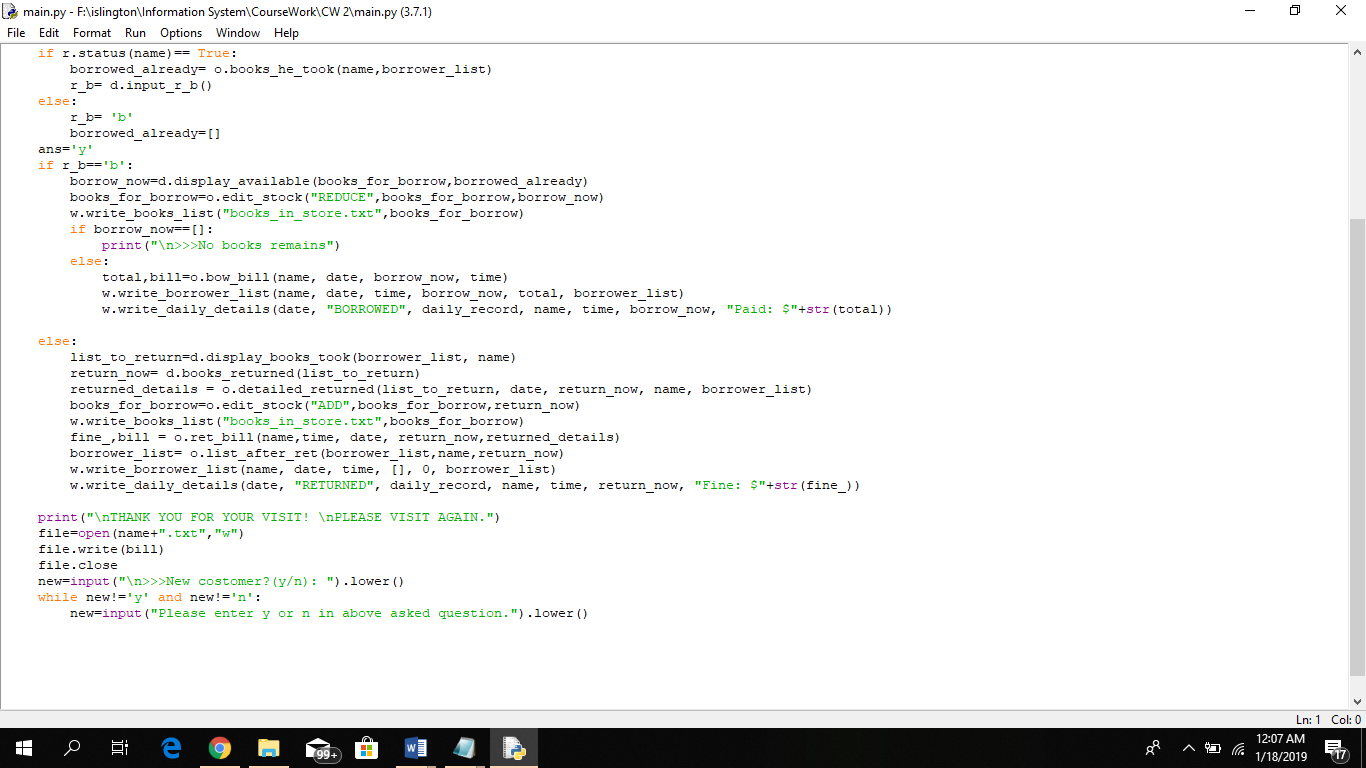
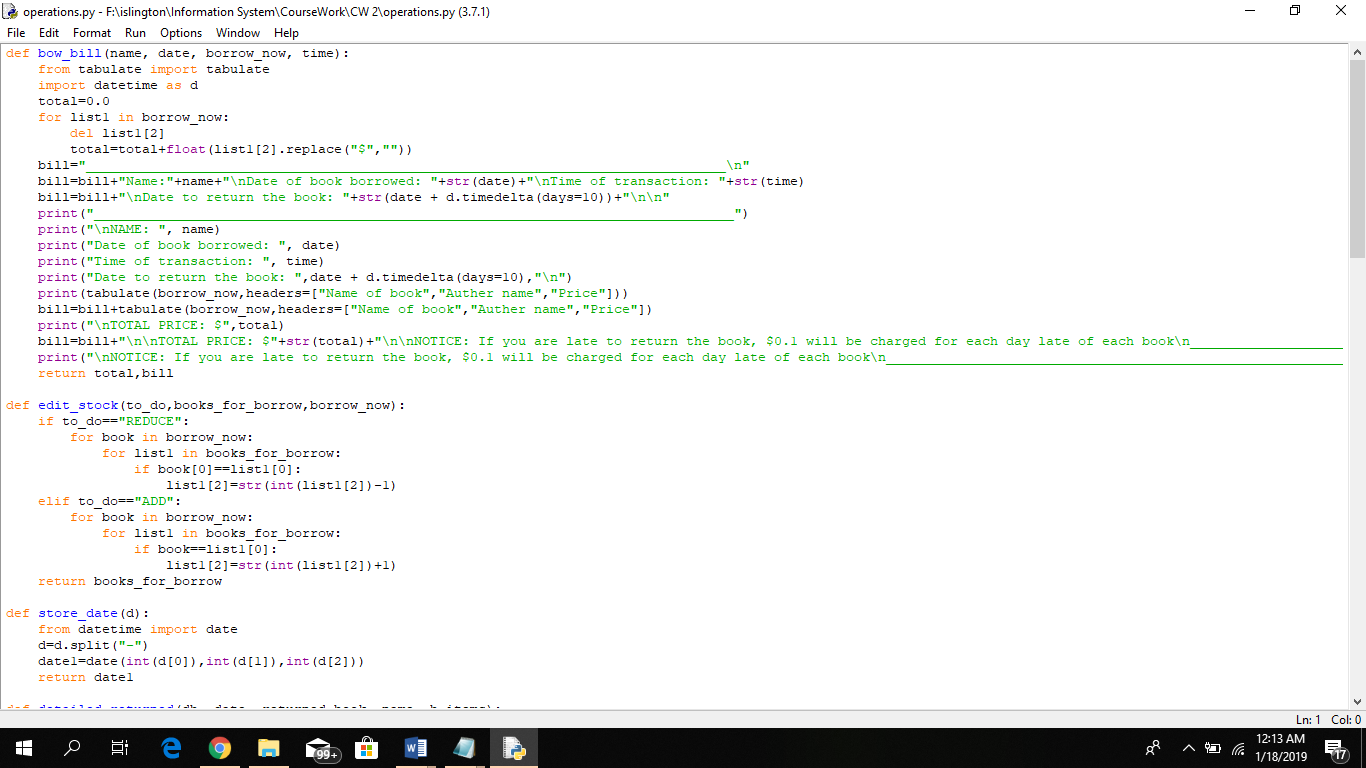


Figure 7 ss of operations.py

Figure 8 ss of main.py

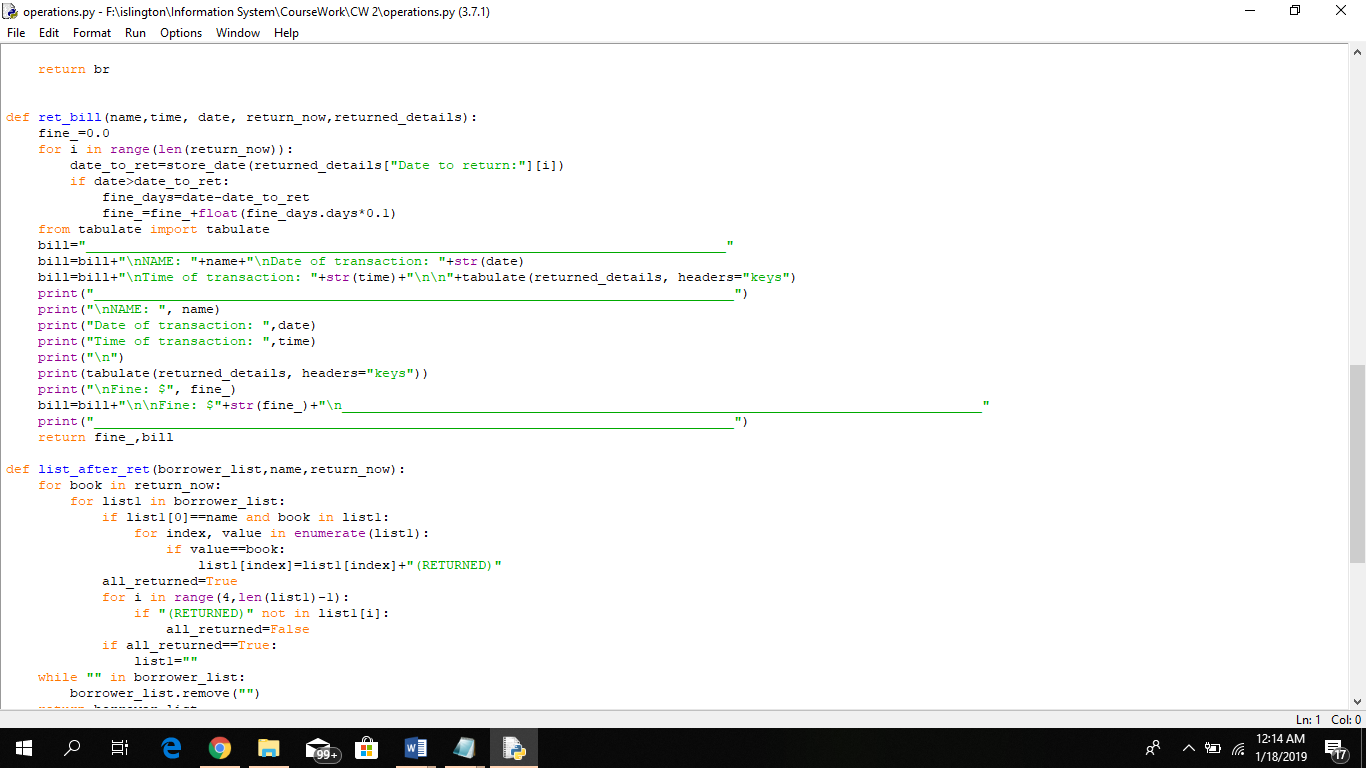
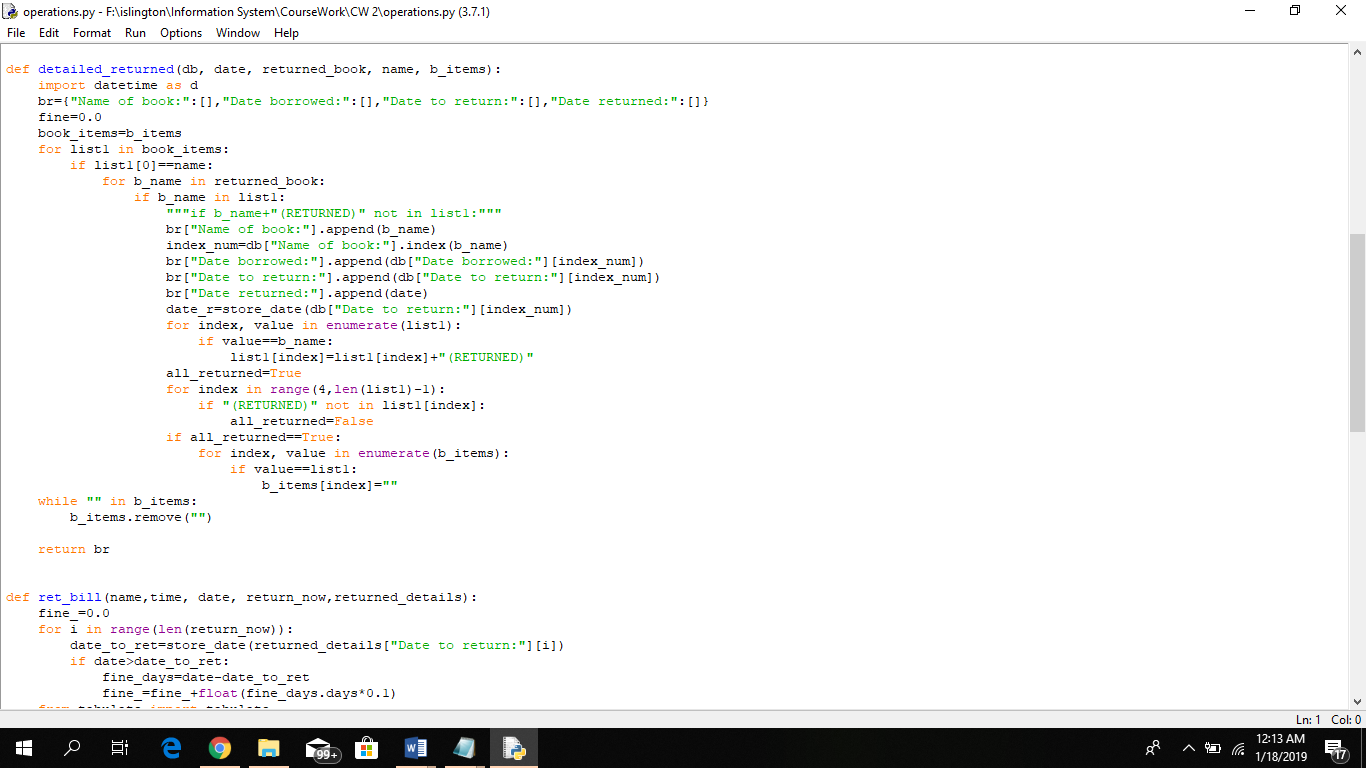


Figure 9 ss of operations.py

Figure 10 ss of operations.py

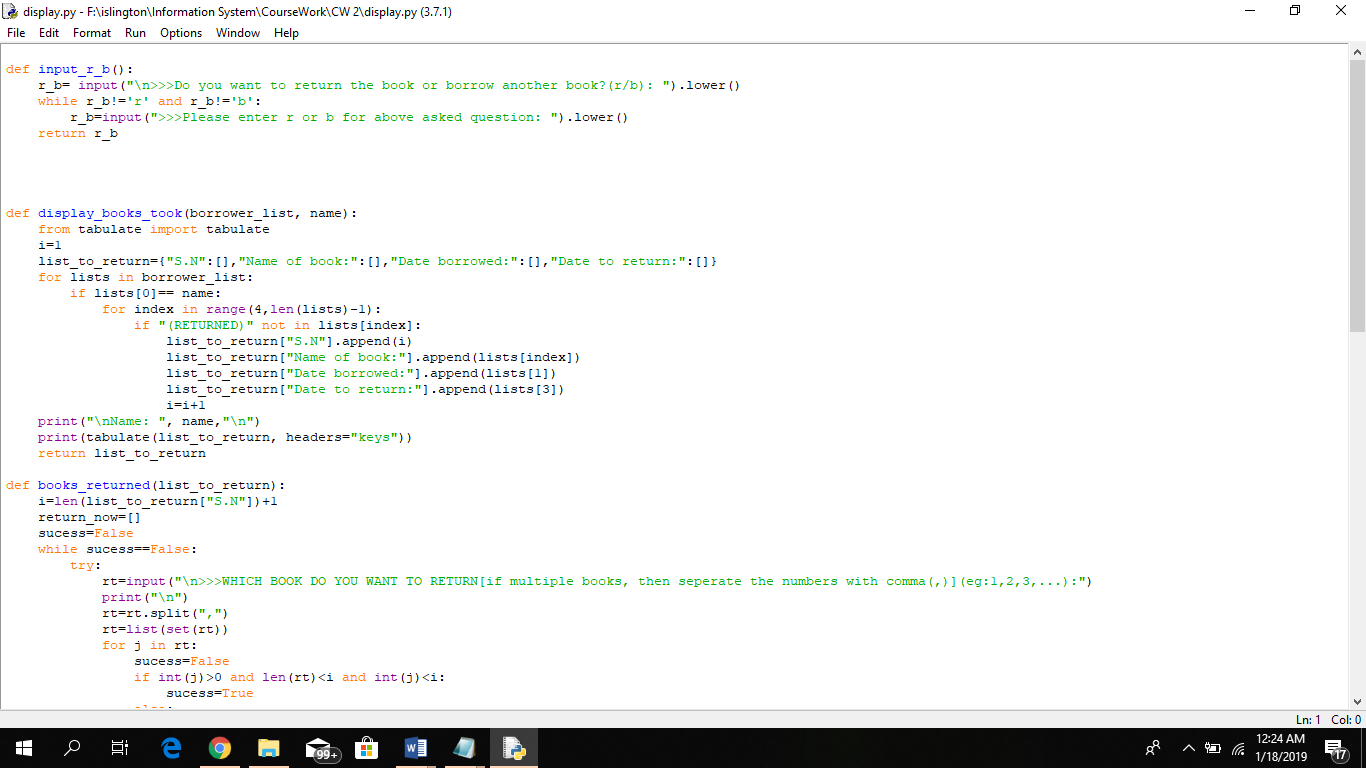
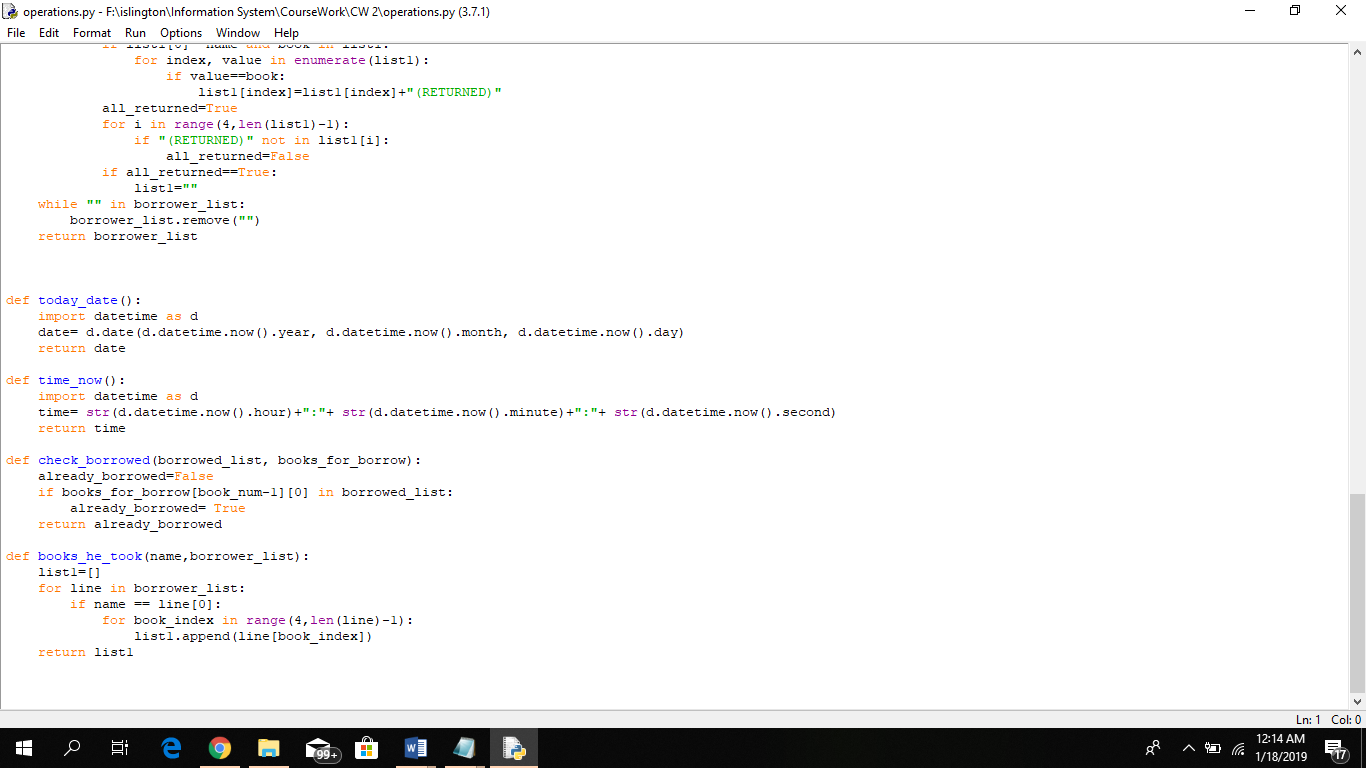


Figure 11 ss of operations.py

Figure 12 ss of display.py



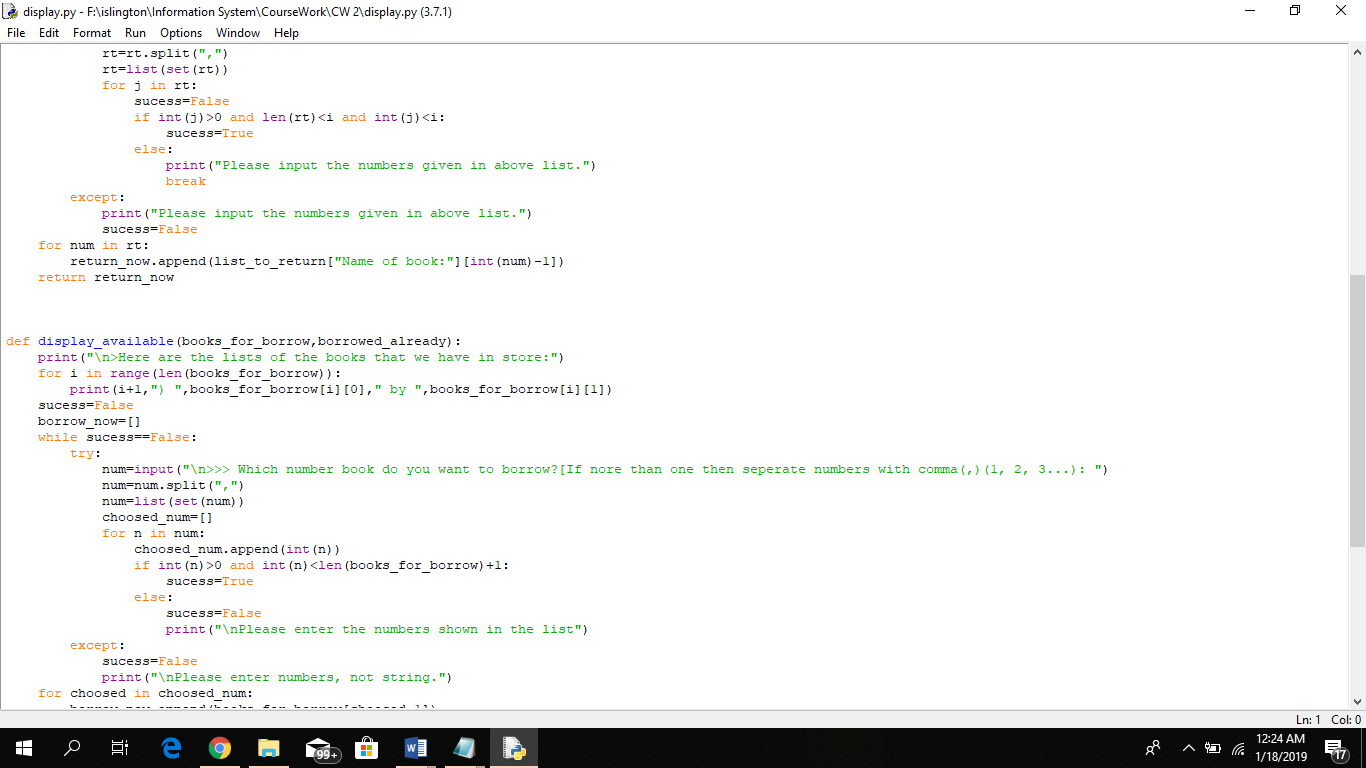
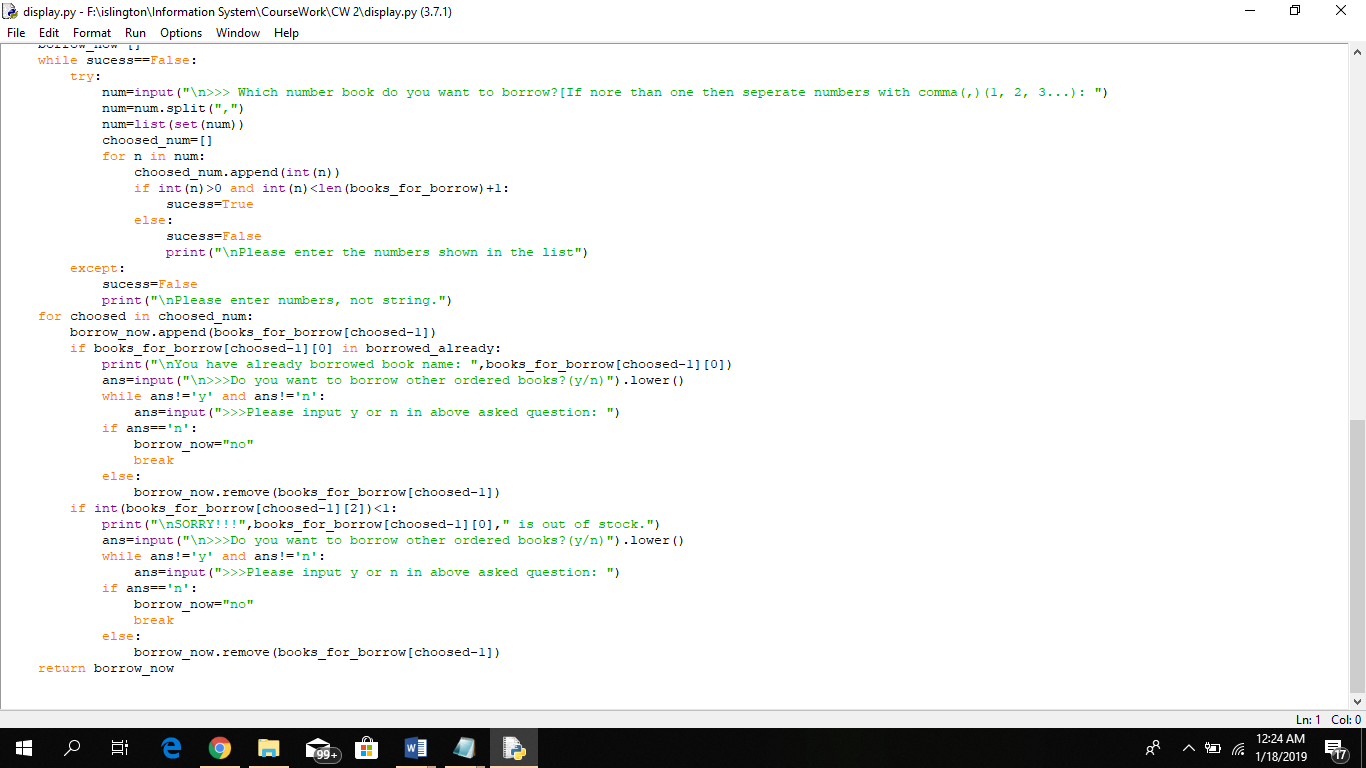


Figure 13 ss of display.py

Figure 14 ss of display.py

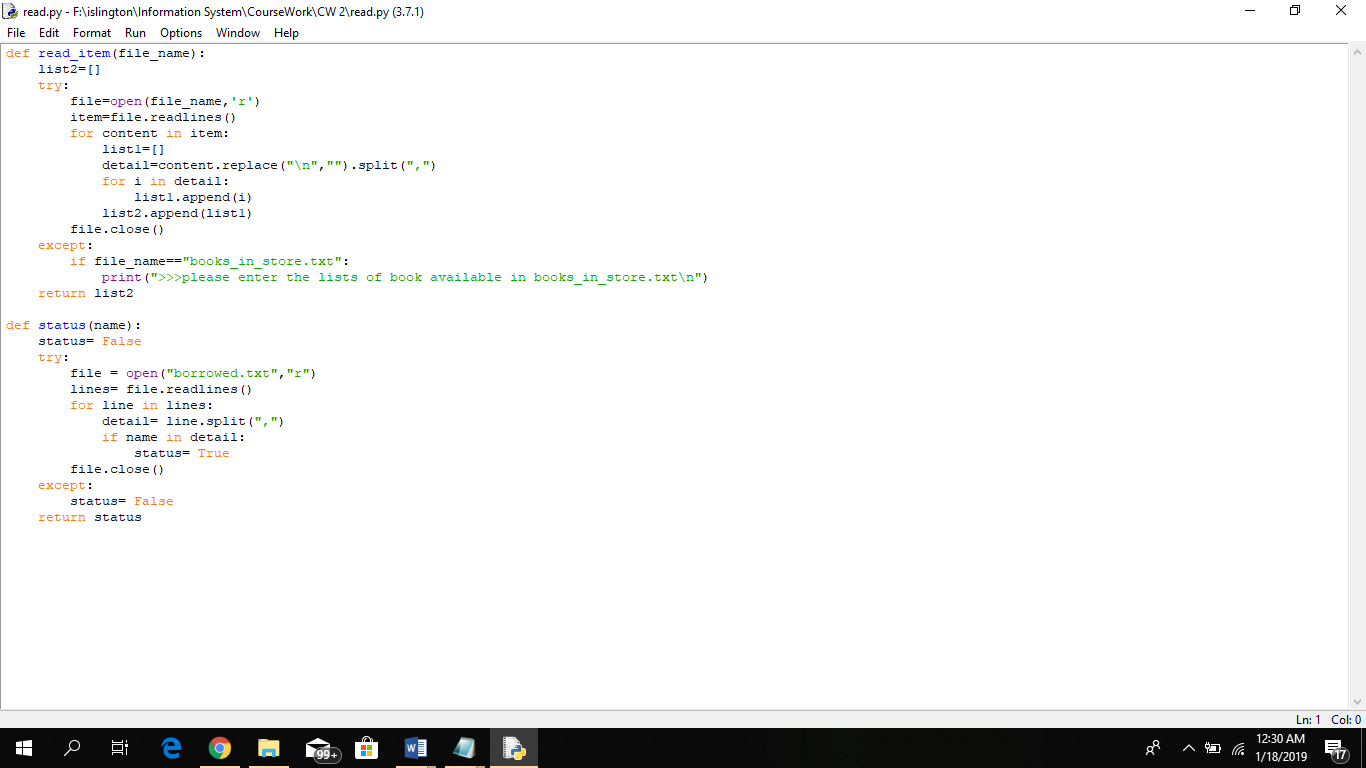
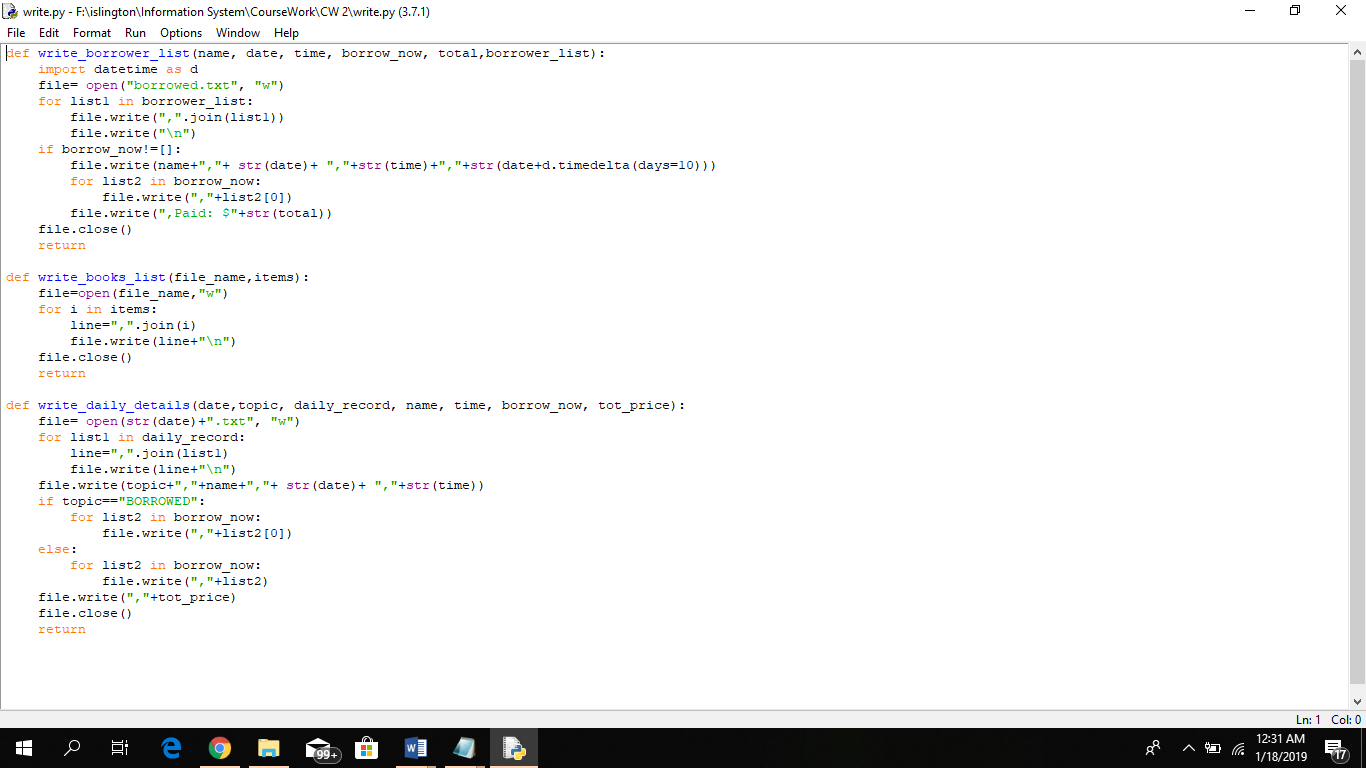


Figure 15 ss of read.py

Figure 16 ss of write.py

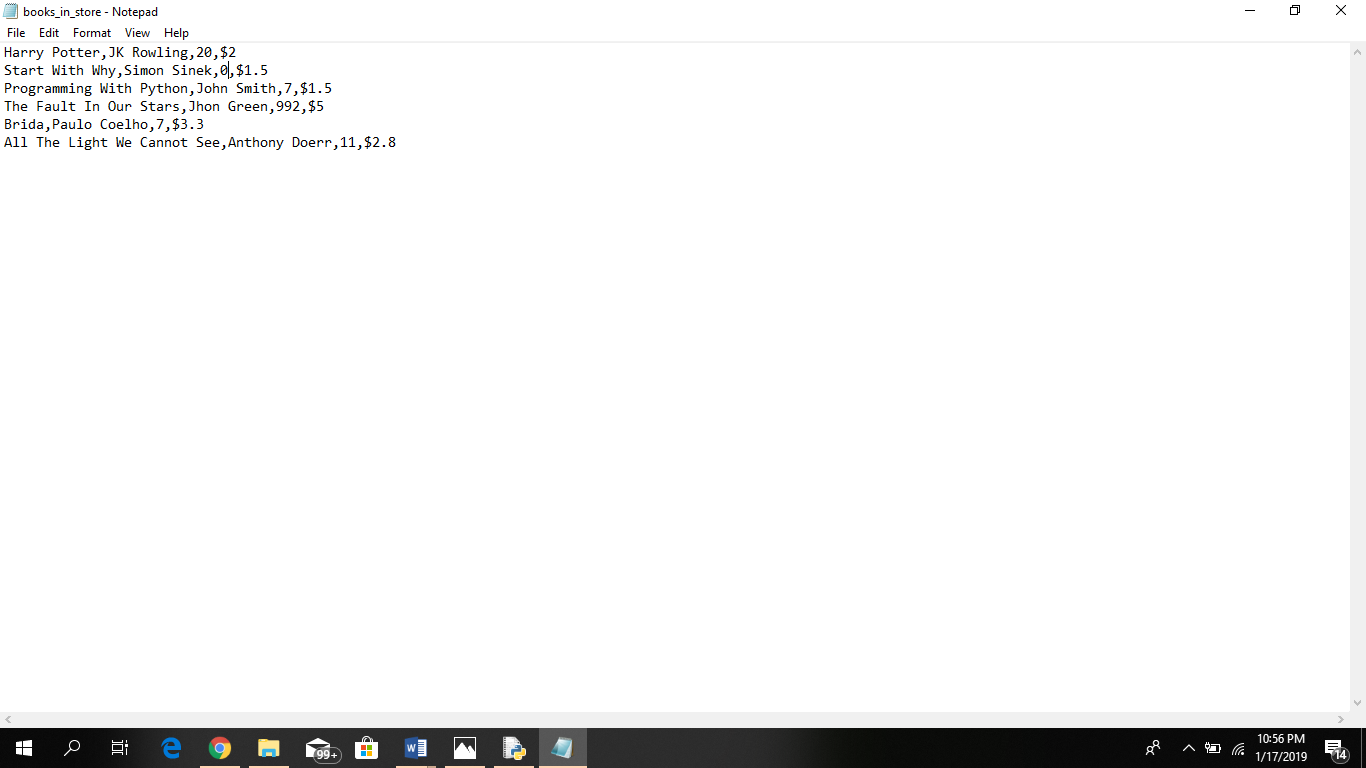
In the program that I developed in this project consist of one main module where all the other sub module is called. To run the whole program, user only need the access to this main module where every work is carried out. In this program, first text file is created where all the books available in the library for borrowing is stored including the writers name, quantity in the stock and its price.

Figure 17 SS of program

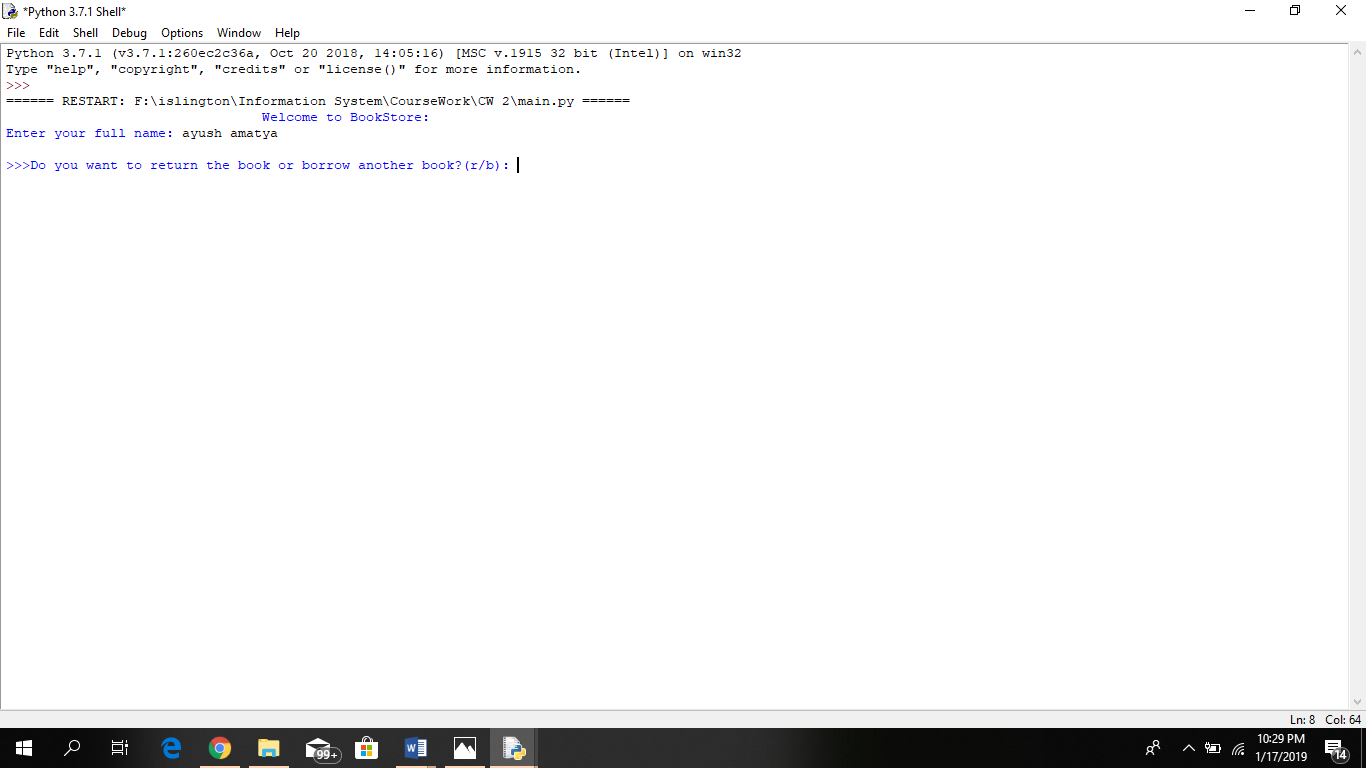
The program asks the user their names. It checks the name of the user in bower list where the details of the customer who borrowed the book previously is store. If the name is found in the borrower list, then the program asks the user whether he is here to return the previously borrowed book or is he here to borrow another book.

Figure 18 SS of program

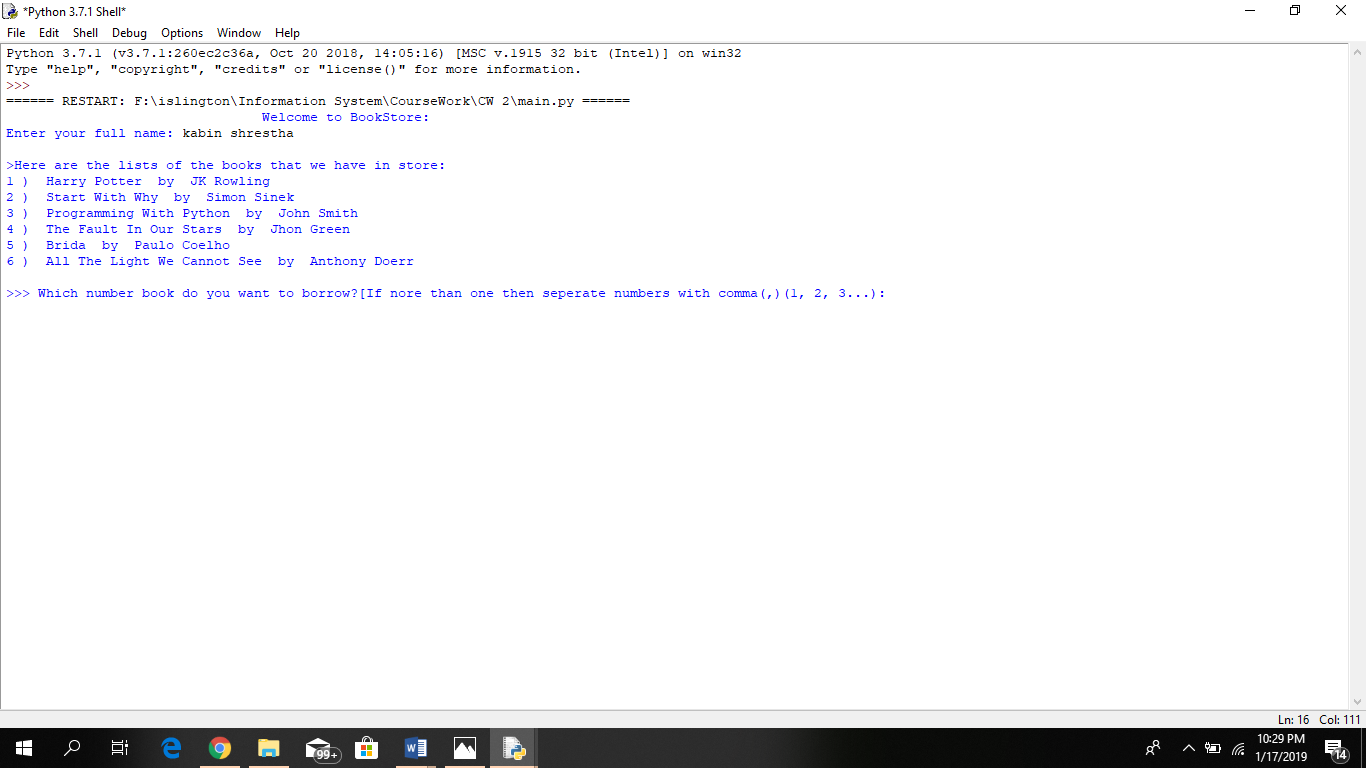
If the name is not found in the borrower list the programs runs the code for borrowing the book without asking them other question

Figure 19 SS of program

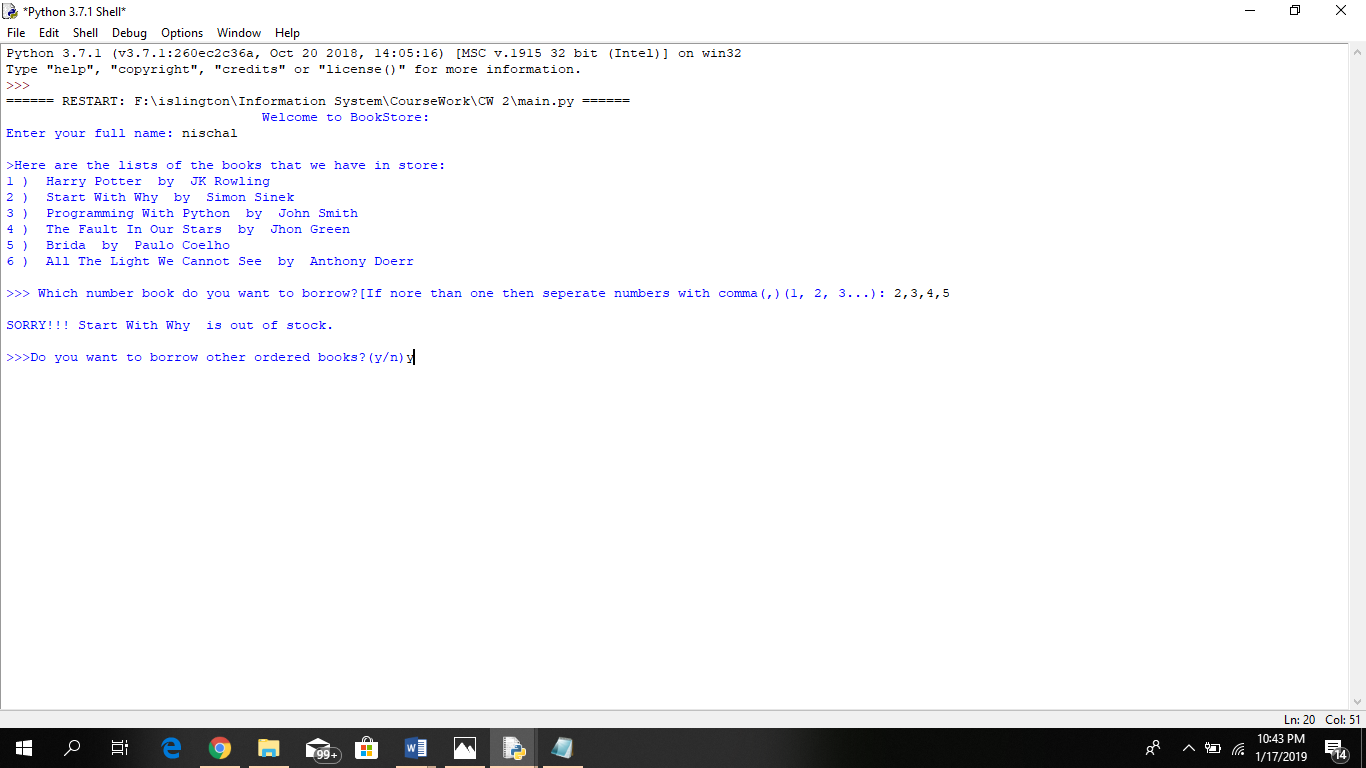
When the user comes to borrow the books, the program shows all the books in the library and asks the user which book he wants to borrow. Here the user can also borrow multiple books entering multiple numbers separated by comma e.g.1,2,3. If the book choosed by the user is out of the stock, the the program says the book is out of the stock and asks whether the user wants to borrow other ordered book or not.

Figure 20 SS of program

Also if the user had already borrowed the book he choosed then the program says that he has already borrowed that book and again asks the user whether he wants to borrow other ordered books or not.

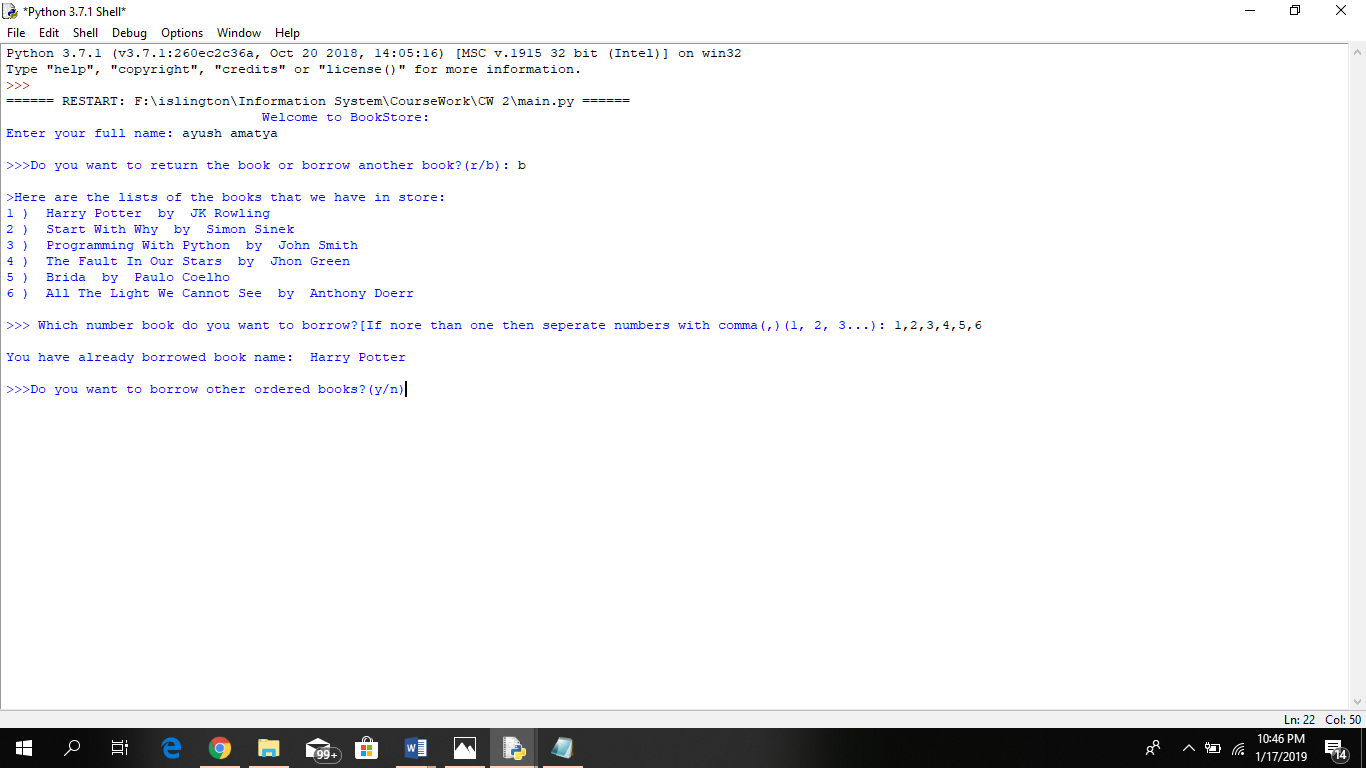


Figure 21 SS of program

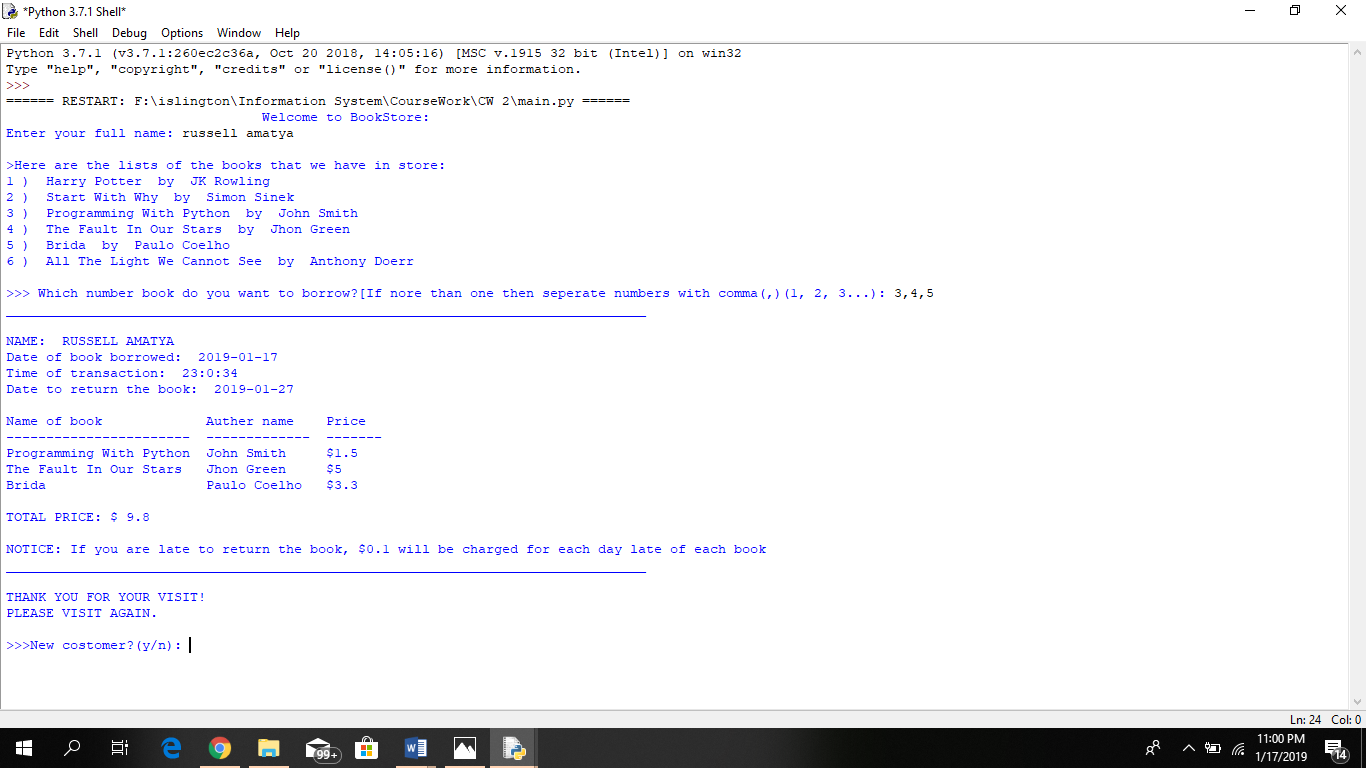
After the costomer finish borrowing the book, the program prints the bill including the name, date and time of transaction, details of books borrowed, and total price to be paid.

Figure 22 SS of program

If the customer is here to return the book, then the program displays the books he borrowed previously and asks the user to choose which books he wants to return. After he choose the books, he wants to return then the final note is displayed including his name, date of borrowed, date to be returned, date returned and details of the book returned. The stock of those book in text file is decreased by 1.

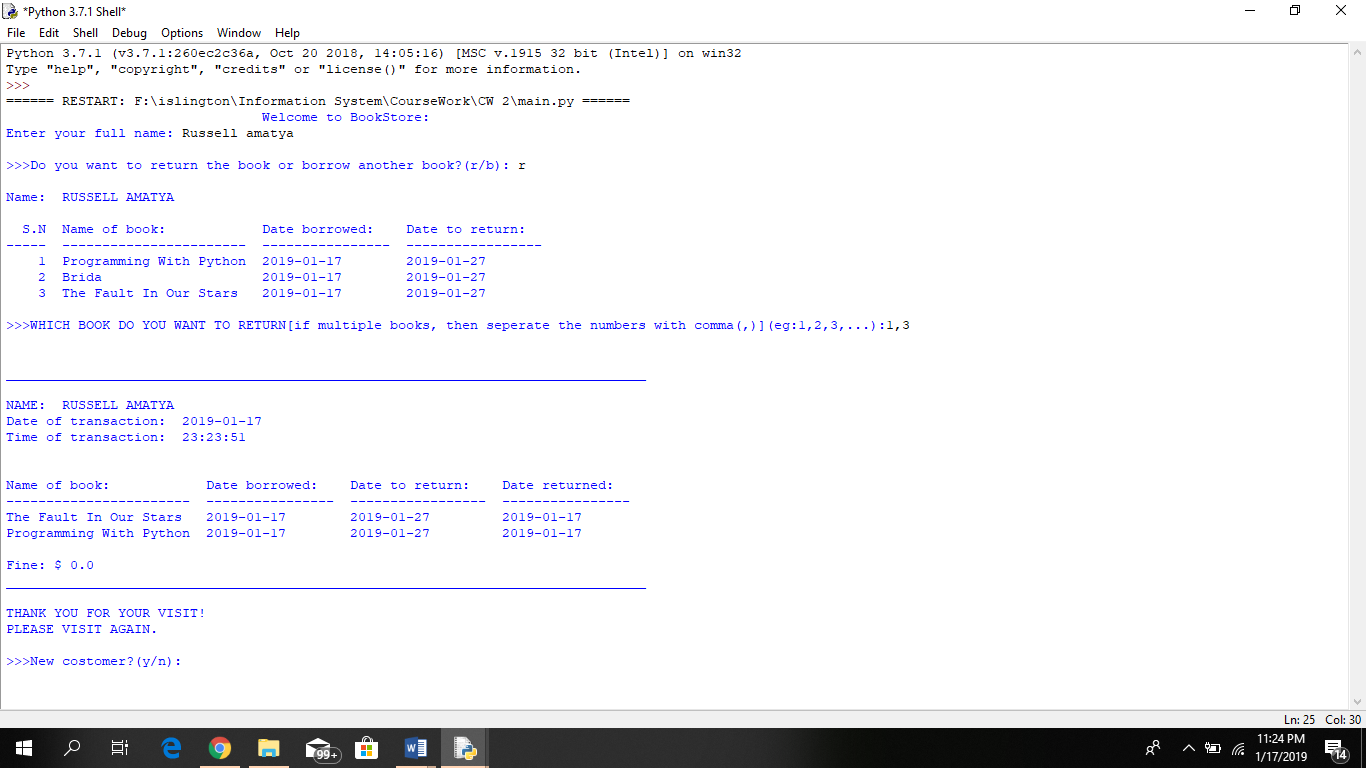
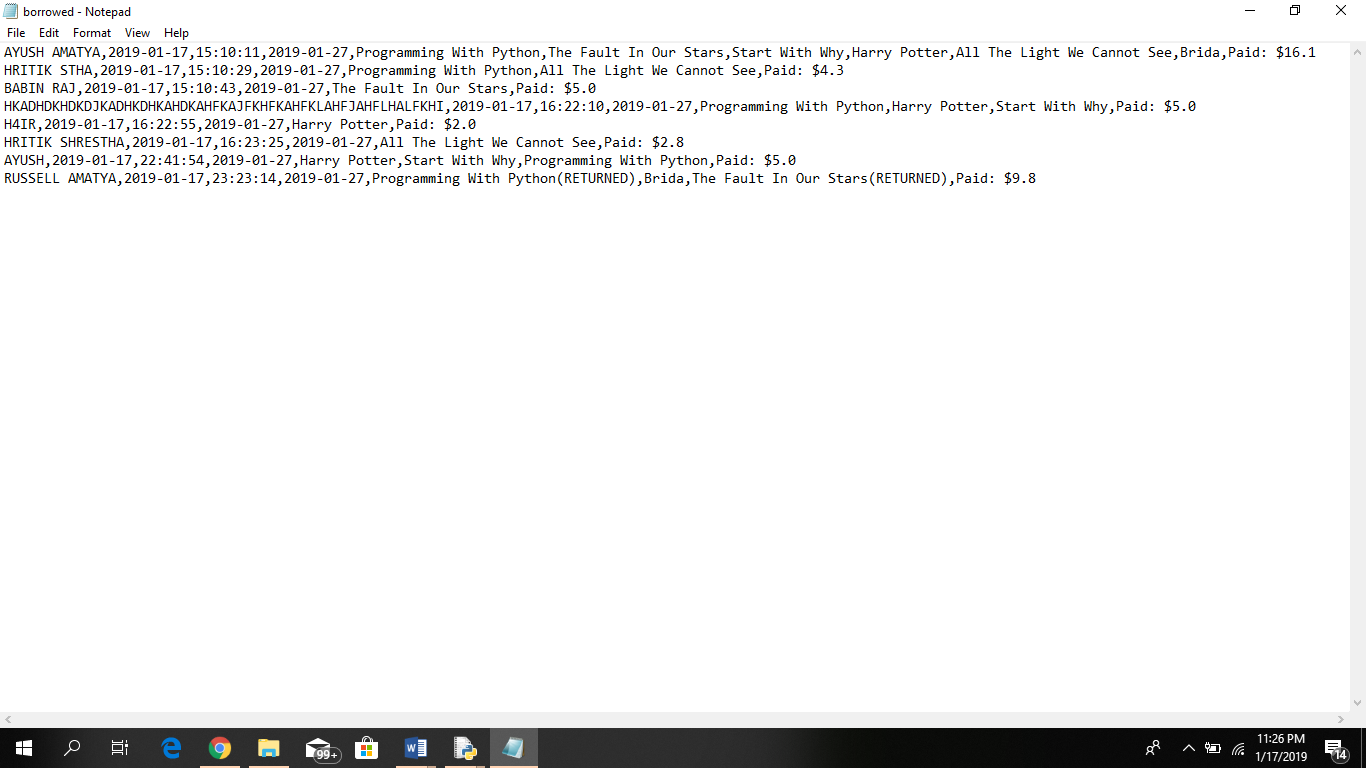


Figure 23 SS of program

If the customer does not return all the books he had borrowed, then only the books that he borrowed will have a tag of returned in the text file. And the stock of those book in text file is also increased by 1.

The record in the customer who returned all the books is deleted from the text file.

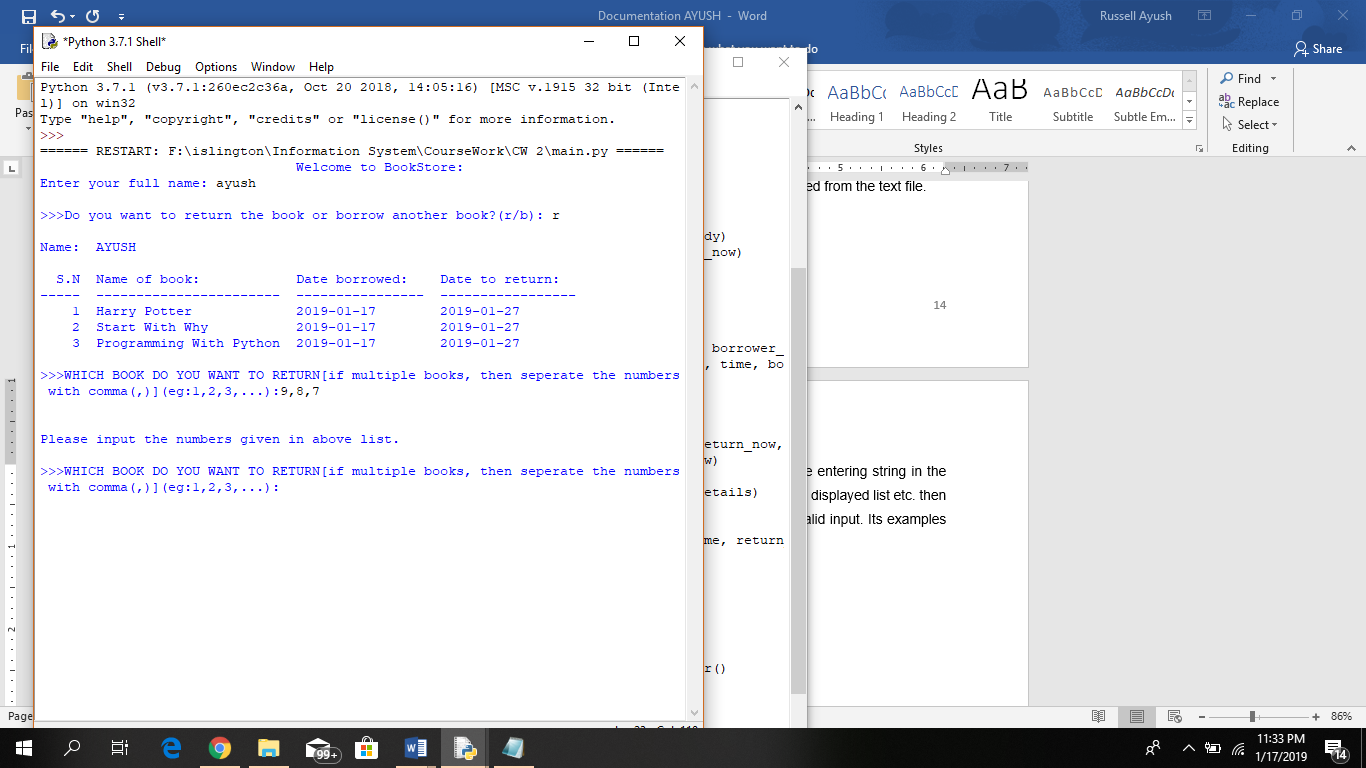
In the above program, if the user enters the invalid inputs like entering string in the place of numeric values, entering numbers which are not in the displayed list etc. then instead of terminating the program, it asks user to input the valid input. Its examples are shown in below figures:

Figure 24 SS of program

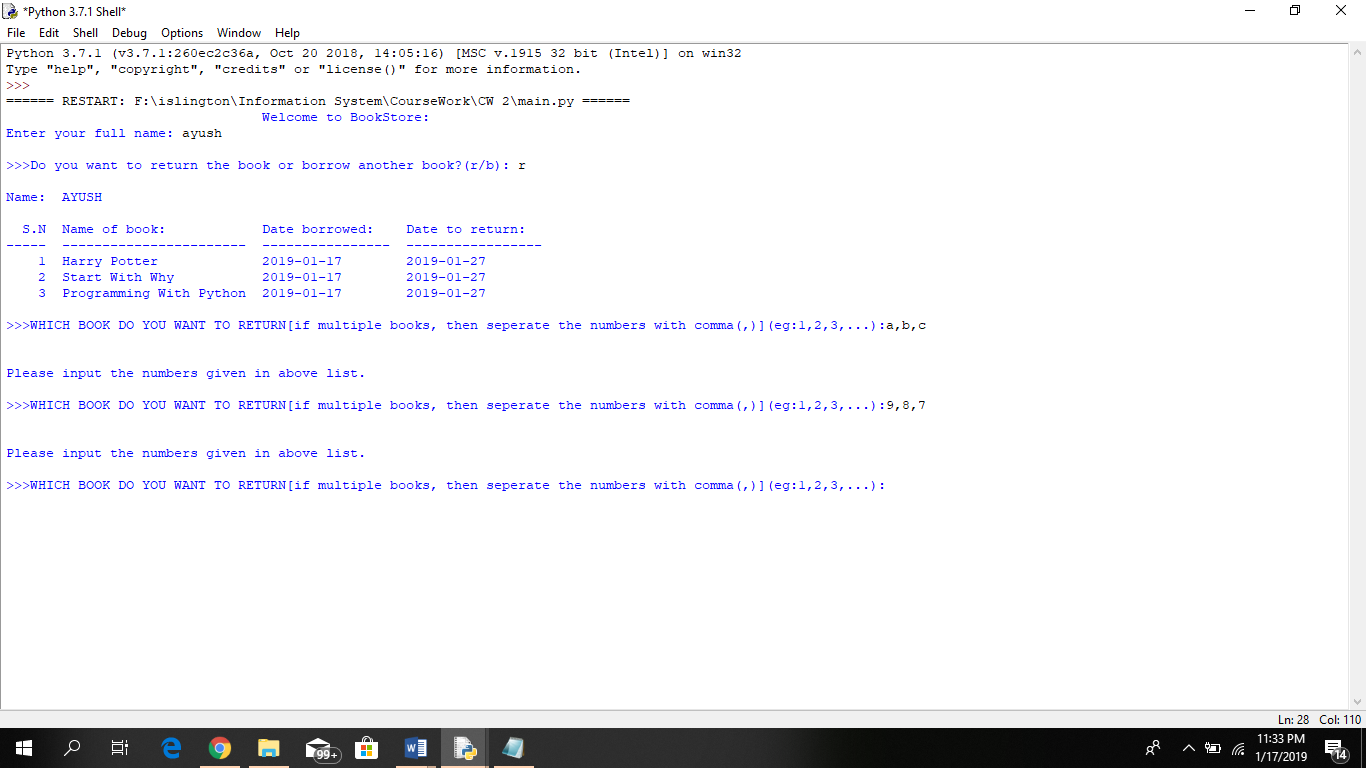


Figure 25 SS of program

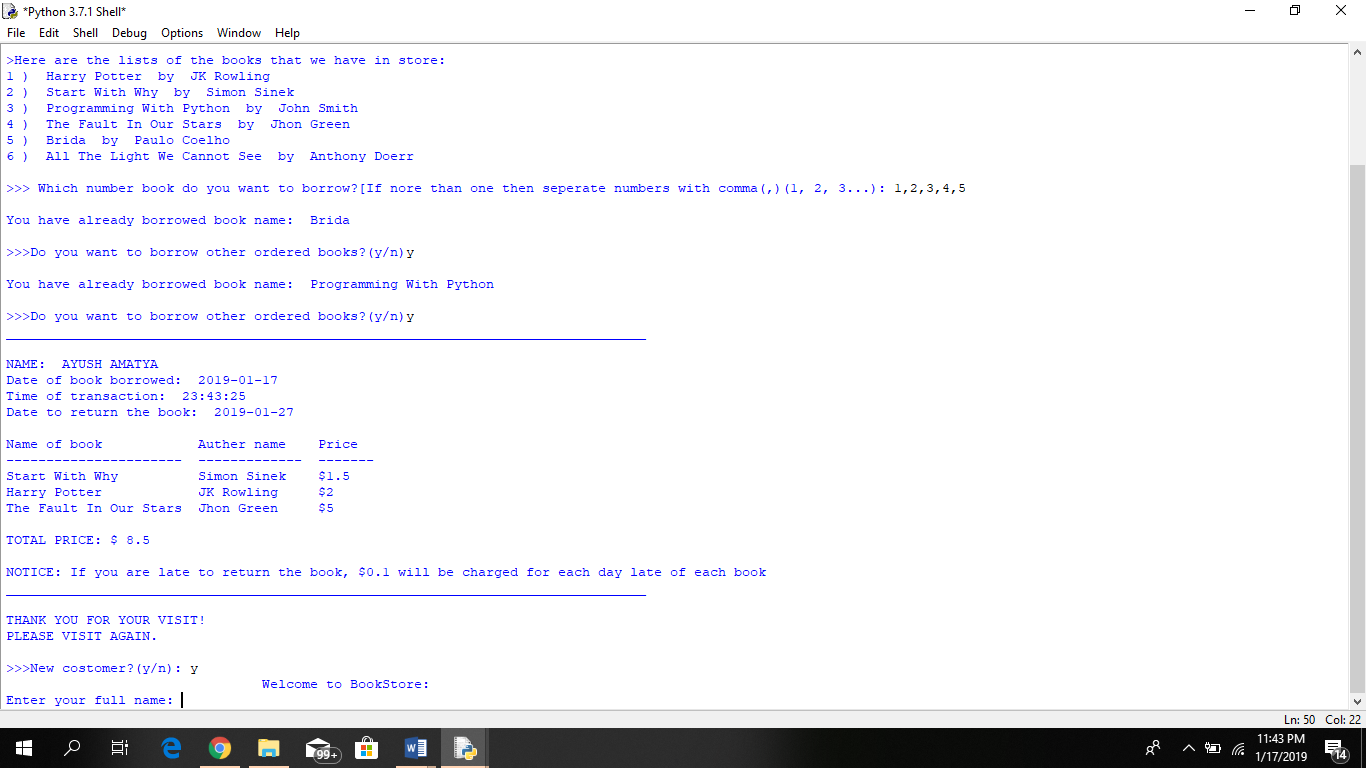
After the borrow and return process, the program asks whether a new customer wants to do the transaction or not. If yes then all the above code is replayed starting from entering the name of the customer.

Figure 26 SS of program

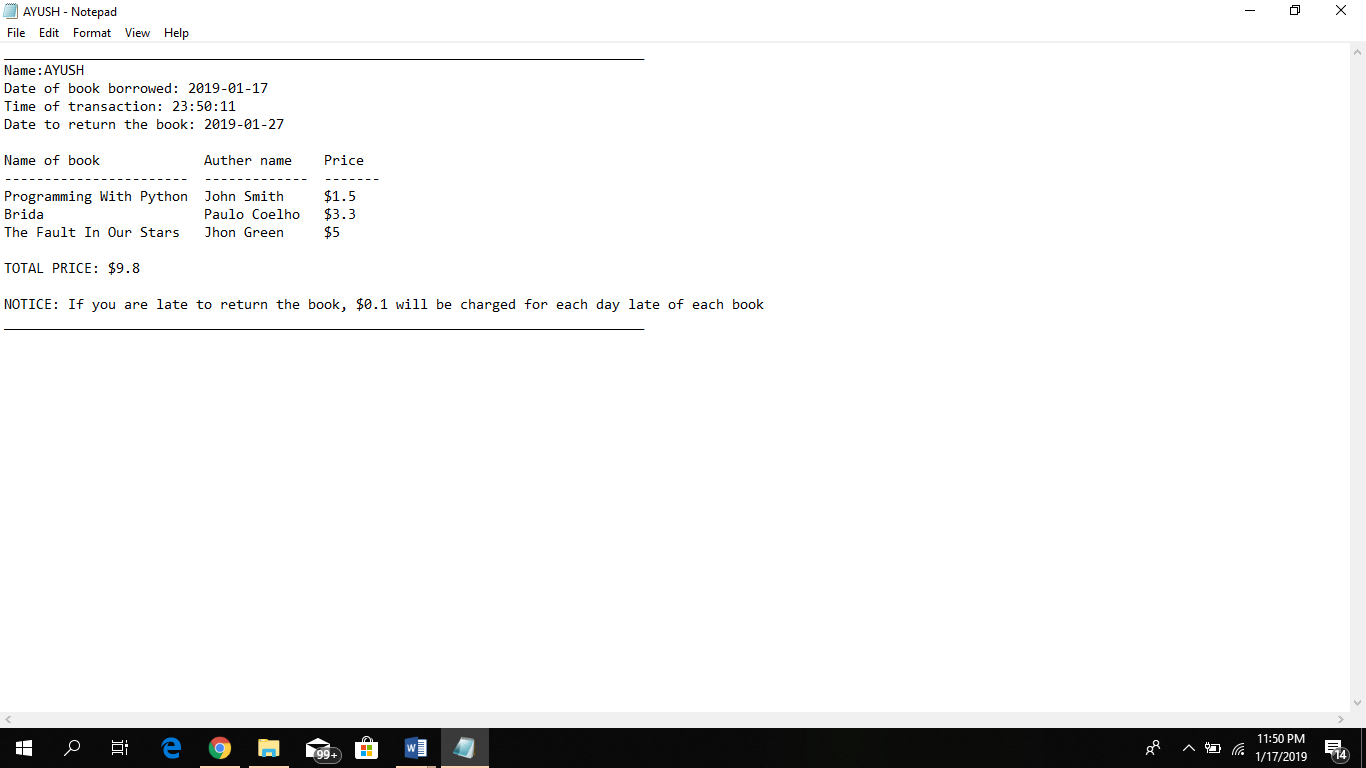
In the end, after all the the transaction is completed, then the program creates a new text file with the customer name as the file name where the lates note and bill of that costomer is stored. This note is to be printed and provide to the customer as a bill.

Figure 27 SS of program

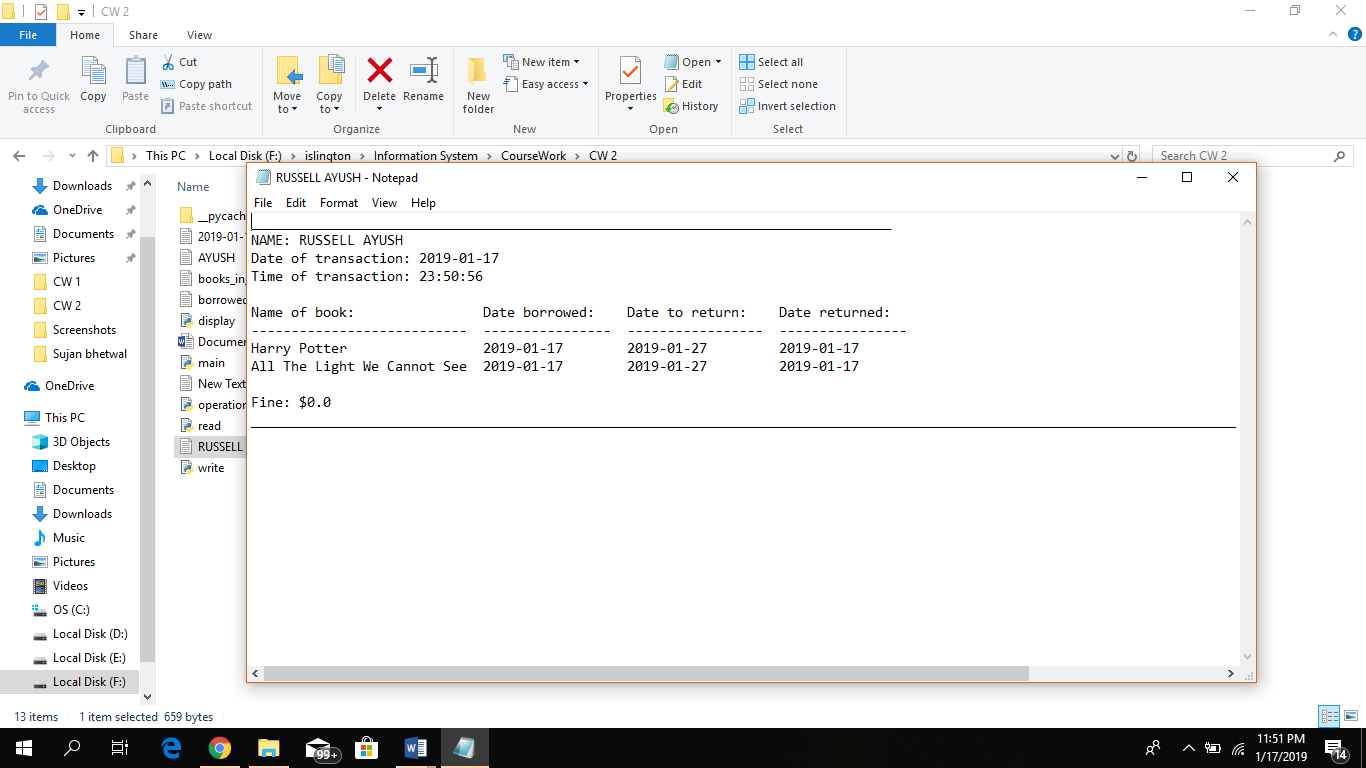


Figure 28 SS of program

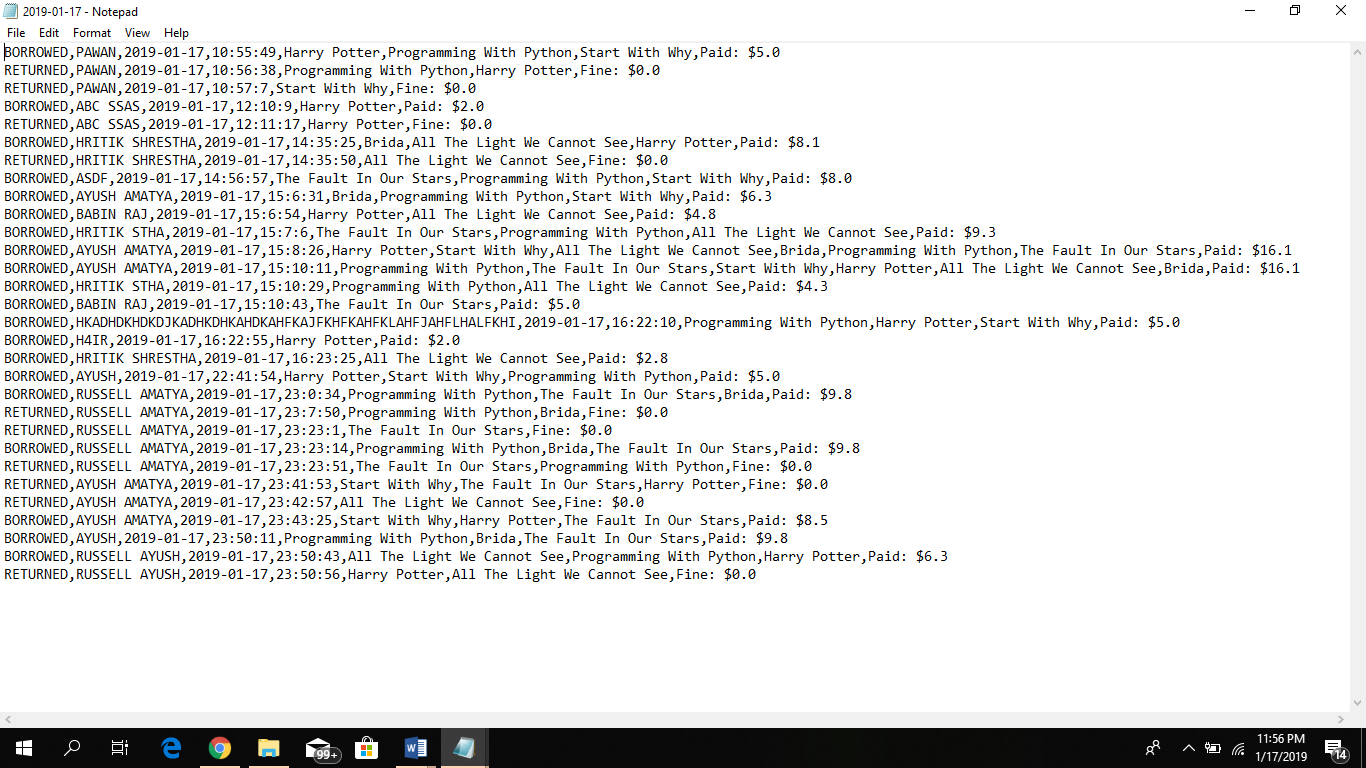
The program also creates a new text file regularly with the date as the file name and stores all the transaction done in that day in that file. It includes the type of the transaction (borrowed or returened), name of customer, date and time of transaction and detail of the books.

Figure 29 SS of daily record

# **Testing:**

The result for some of the black box testing done to check the correctness of my program is shown below:

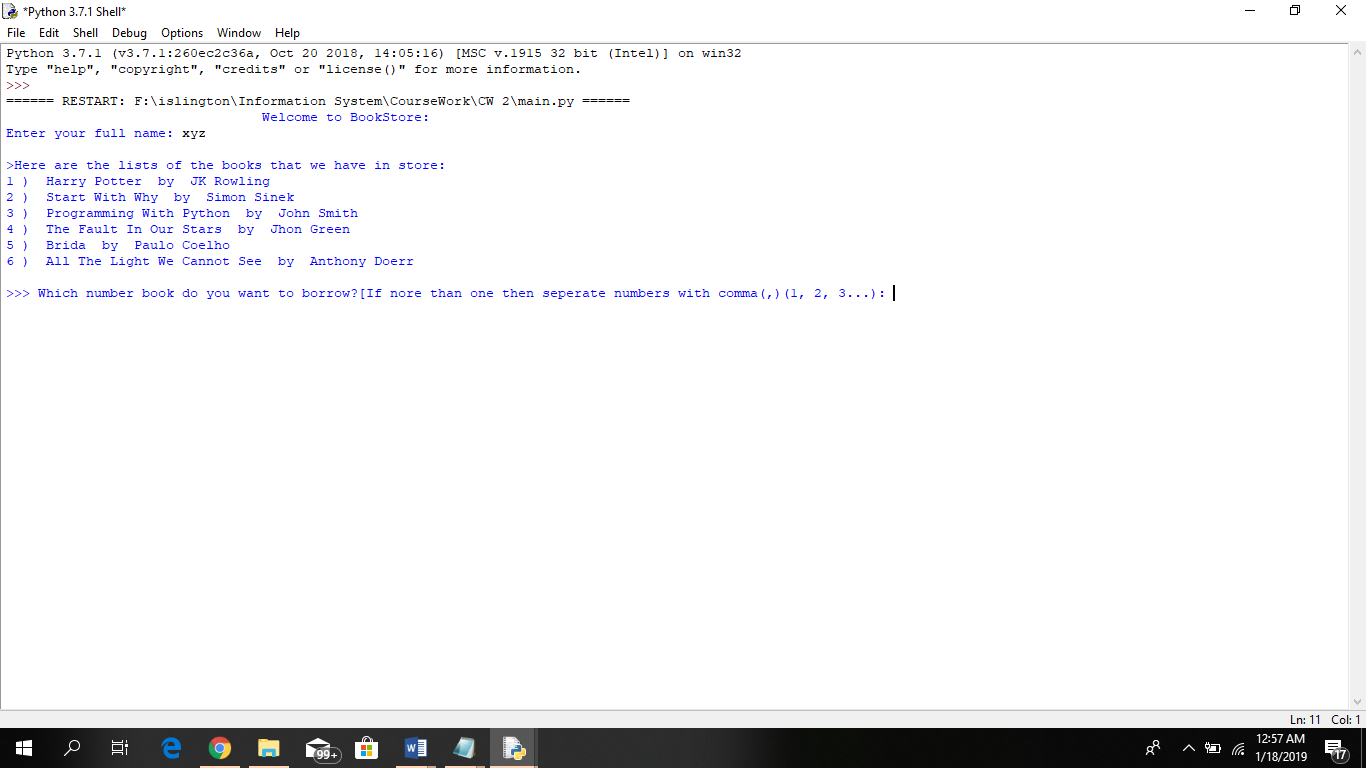
1. Test to check whether the program directly prints books lists to borrow without asking user to return or borrow book when new customer who has not borrowed book previously.

Figure 30 Black box test 1

Table 1 Black box test 1

|  |  |
| --- | --- |
| Test number | 1 |
| Action | Pass customer name xyz who has not borrowed the book previously |
| Expected result | Program should display book list to borrow directly without asking user wants to borrow or return |
| Actual result | Program directly displays book list |
| Test result | Pass |

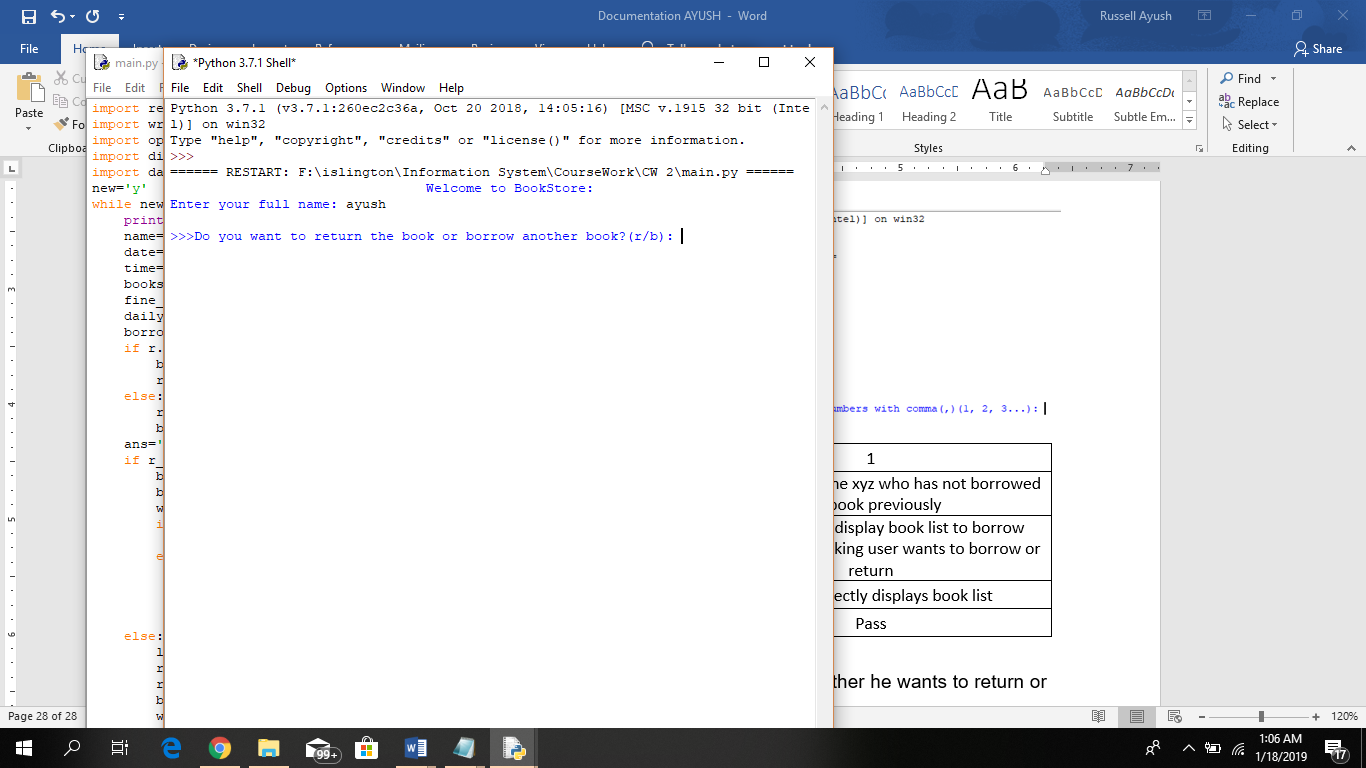
1. Test to check whether the program ask the user whether he wants to return or borrow the book if he has already borrowed the book previously.

Figure 31 Black box test 2

Table 2 Black box test 2

|  |  |
| --- | --- |
| Test number | 2 |
| Action | Pass customer name ayush who has borrowed the book previously |
| Expected result | Program should display ask user wants to borrow or return |
| Actual result | Program asked user to r or b |
| Test result | Pass |

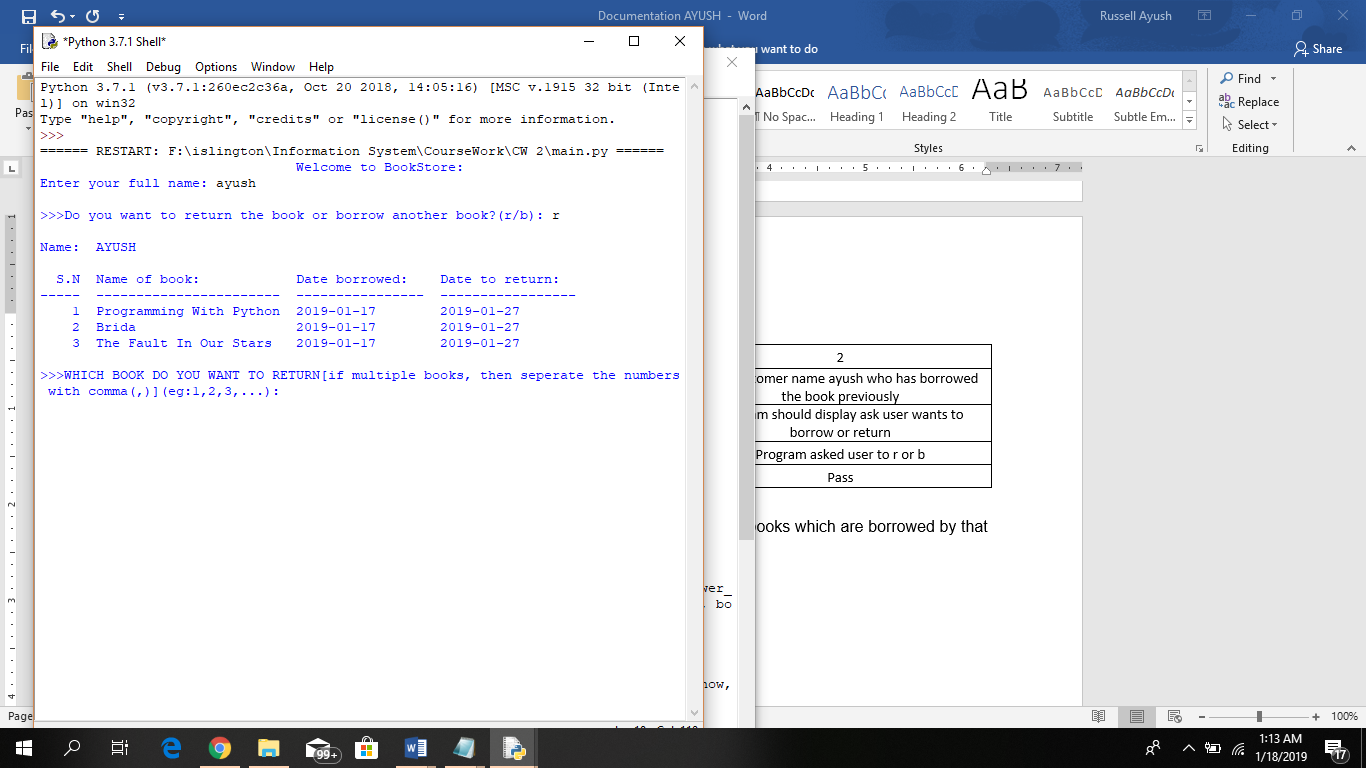
1. To check if the program displays only those books which are borrowed by that customers if he comes to return the book.

Figure 32 Black box test 3

Table 3 Black box test 3

|  |  |
| --- | --- |
| Test number | 3 |
| Action | Pass ‘r’ when program asks ‘r’ or ‘b’ |
| Expected result | Program should display books that are borrowed by that user only |
| Actual result | Program displayed only those book that are borrowed by customer |
| Test result | Pass |

1. To check if the program says to enter numeric values if string is entered in the place where numeric value was asked.

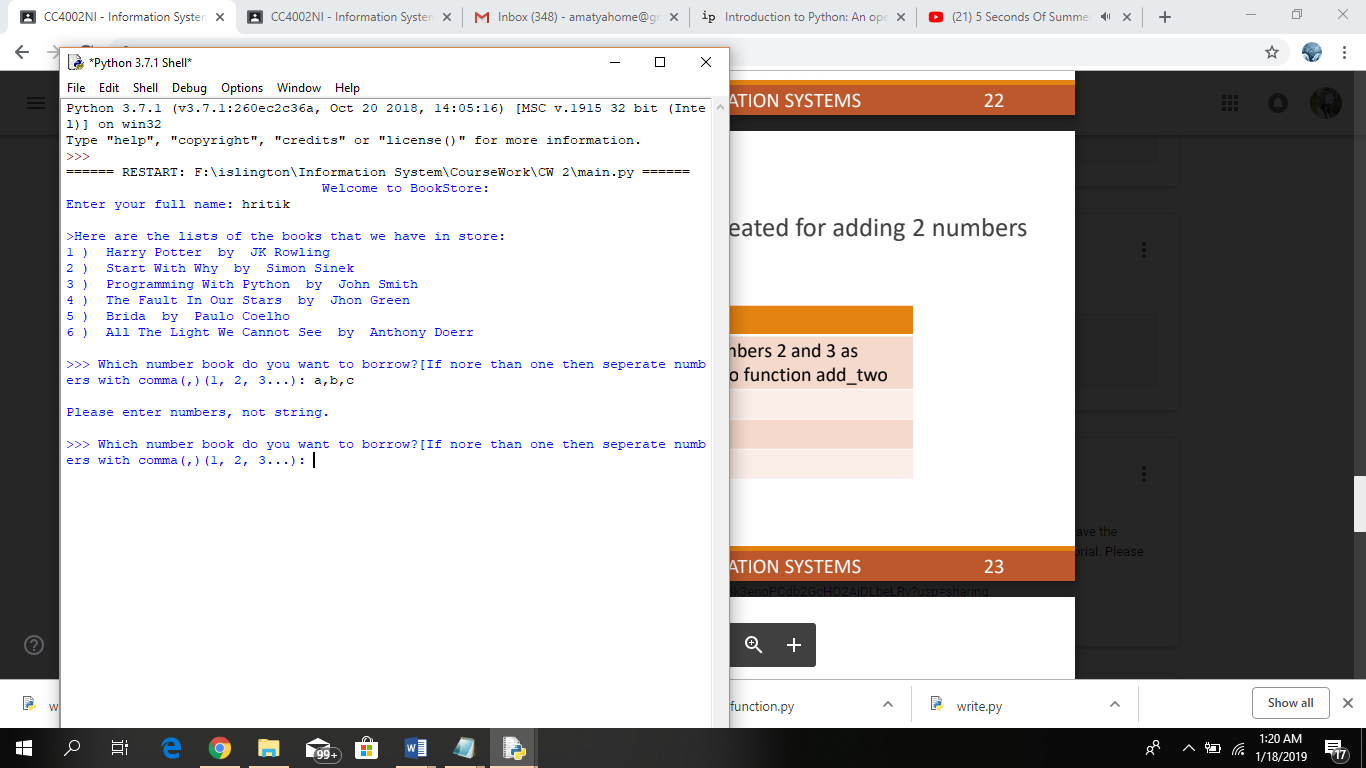


Figure 33 Black box test 4

Table 4 Black box test 4

|  |  |
| --- | --- |
| Test number | 4 |
| Action | Pass a, b, c where numeric value is asked |
| Expected result | Program should say to enter number |
| Actual result | Program said to enter number |
| Test result | Pass |

1. To check if the program edits the stock or not.

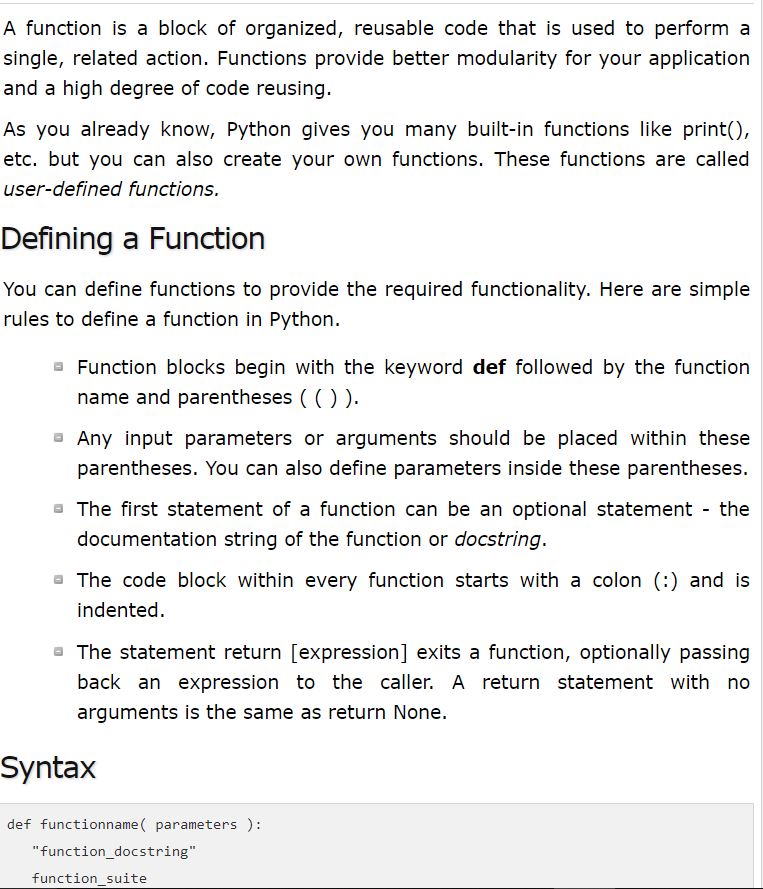
Table 5 Black box test 5

|  |  |
| --- | --- |
| Test number | 5 |
| Action | Borrow any book |
| Expected result | Program should decrease that book’s quantity in books stock text file |
| Actual result | Program decreased the book’s quantity |
| Test result | Pass |

# **Research:**

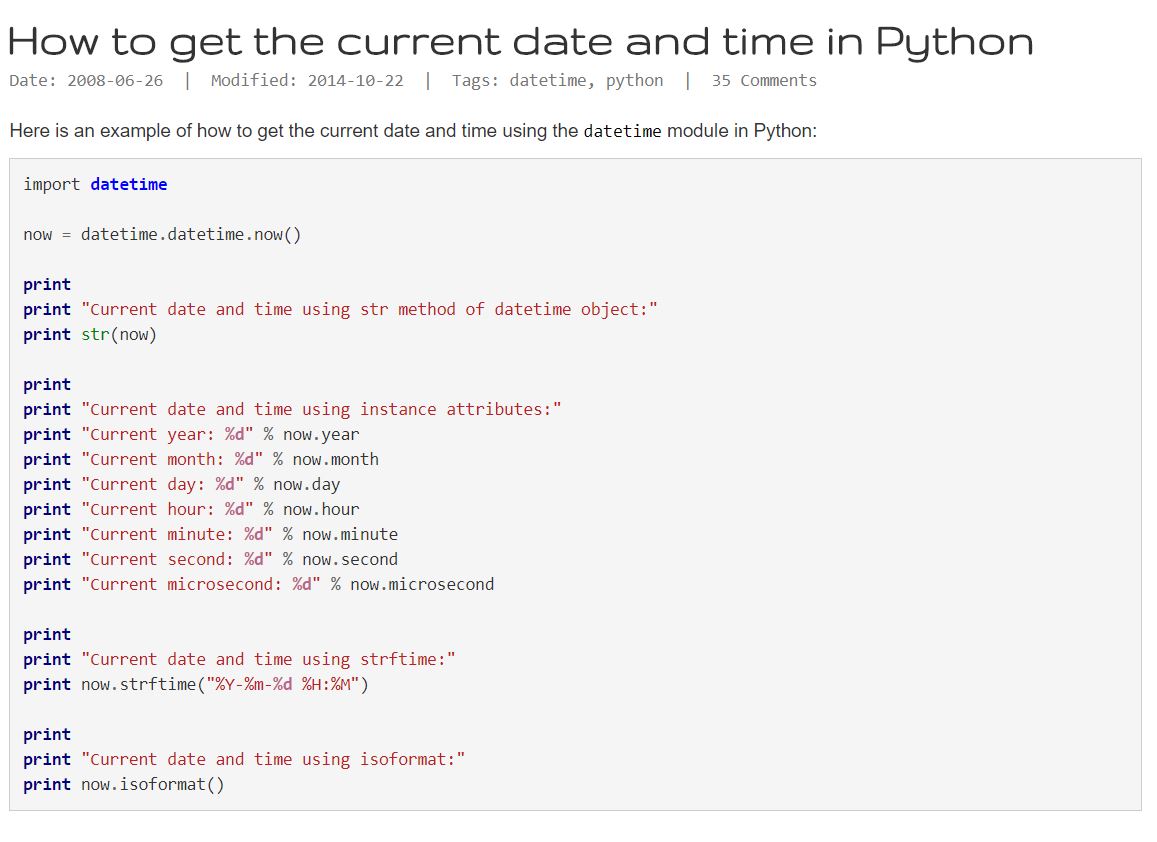
## **Website**

1. <https://www.tutorialspoint.com/python/python_functions.htm>



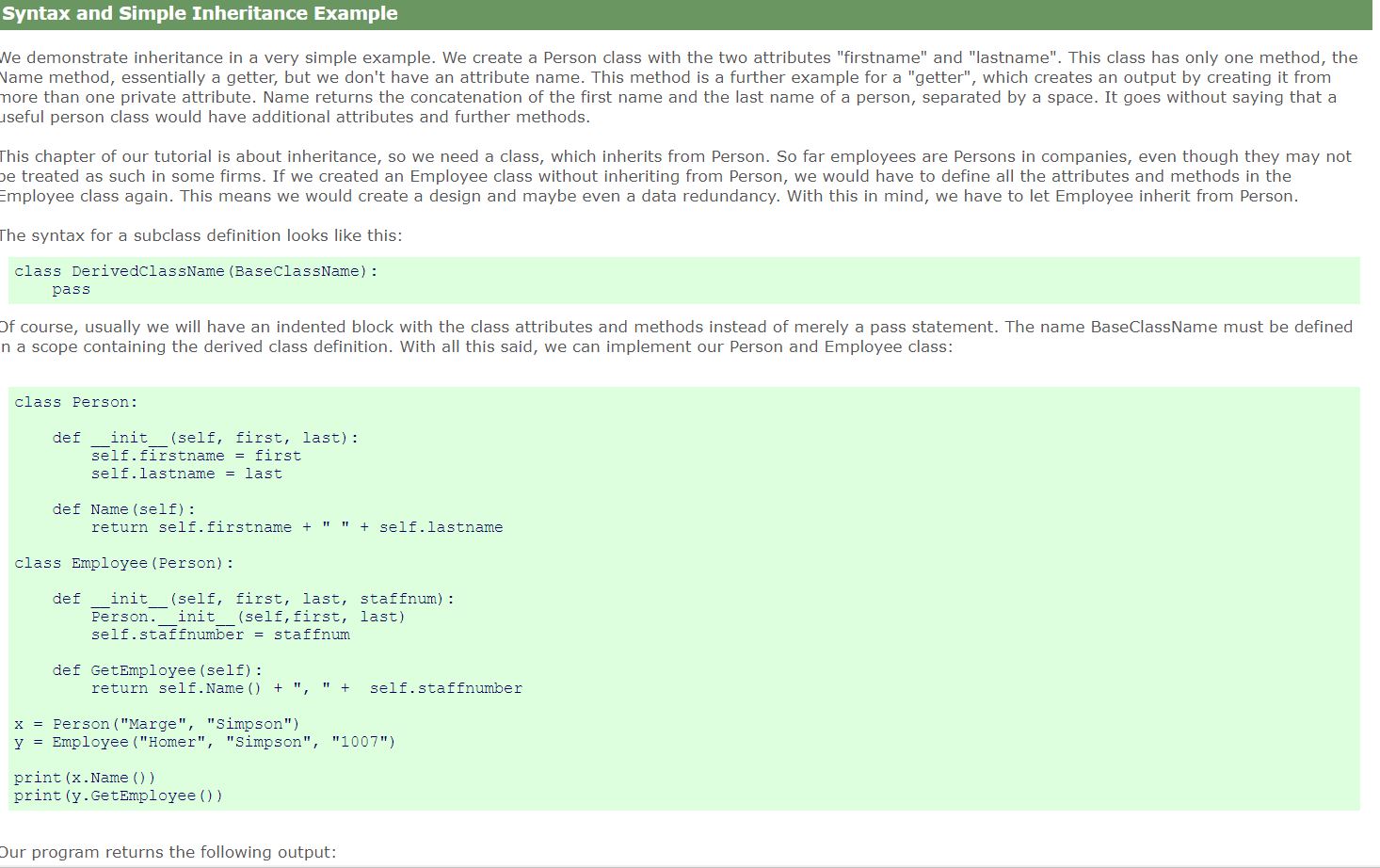
From this website I was able to learn about on how to define a function and the syntax on writing a function. I was able define what a function is in my report and to write my program using the functions.

1. <https://www.saltycrane.com/blog/2008/06/how-to-get-current-date-and-time-in/>



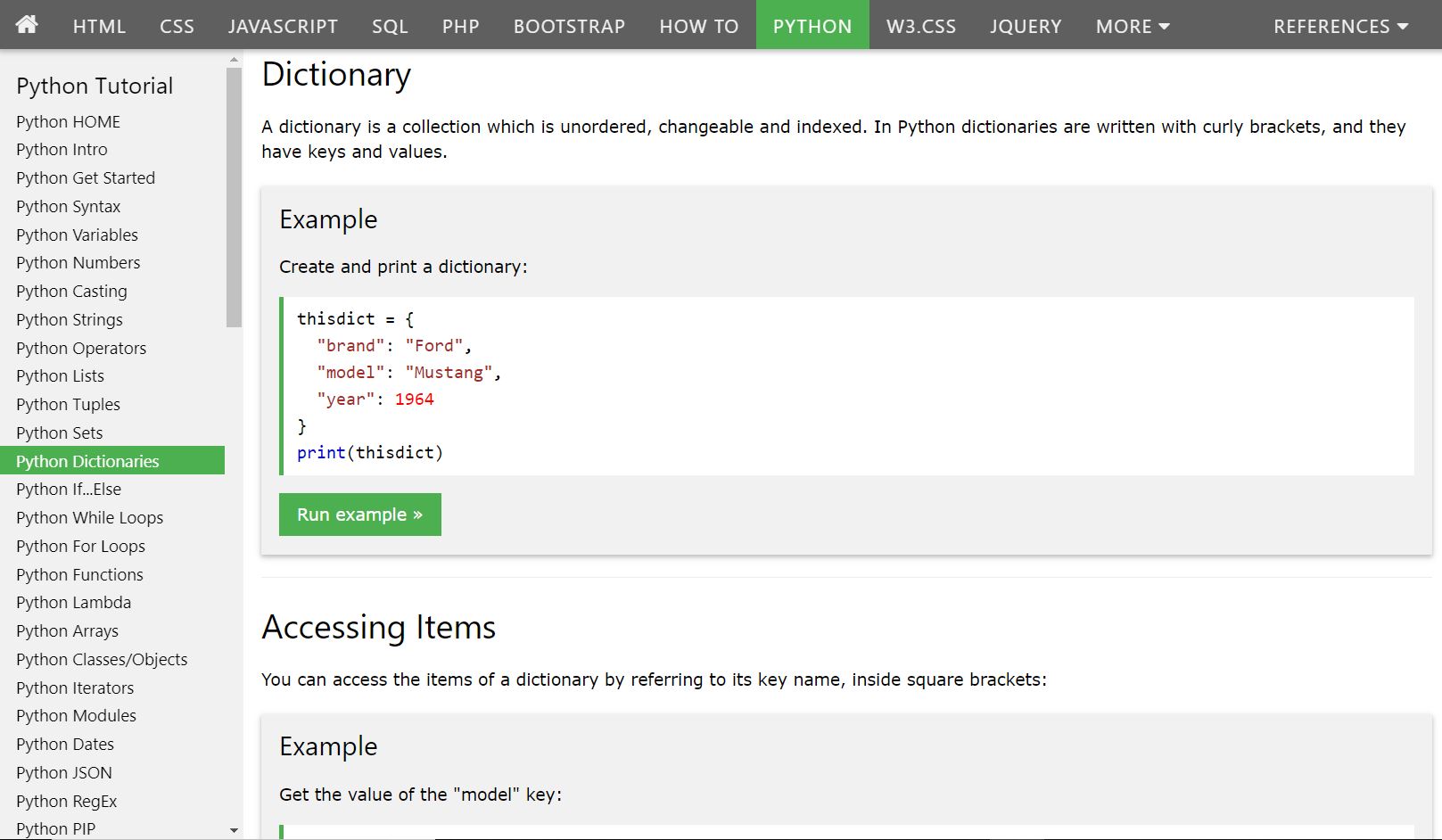
I was able to learn about how to import current date and time in my program. This helped me in generating the fine as well.

1. <https://www.python-course.eu/python3_inheritance.php>



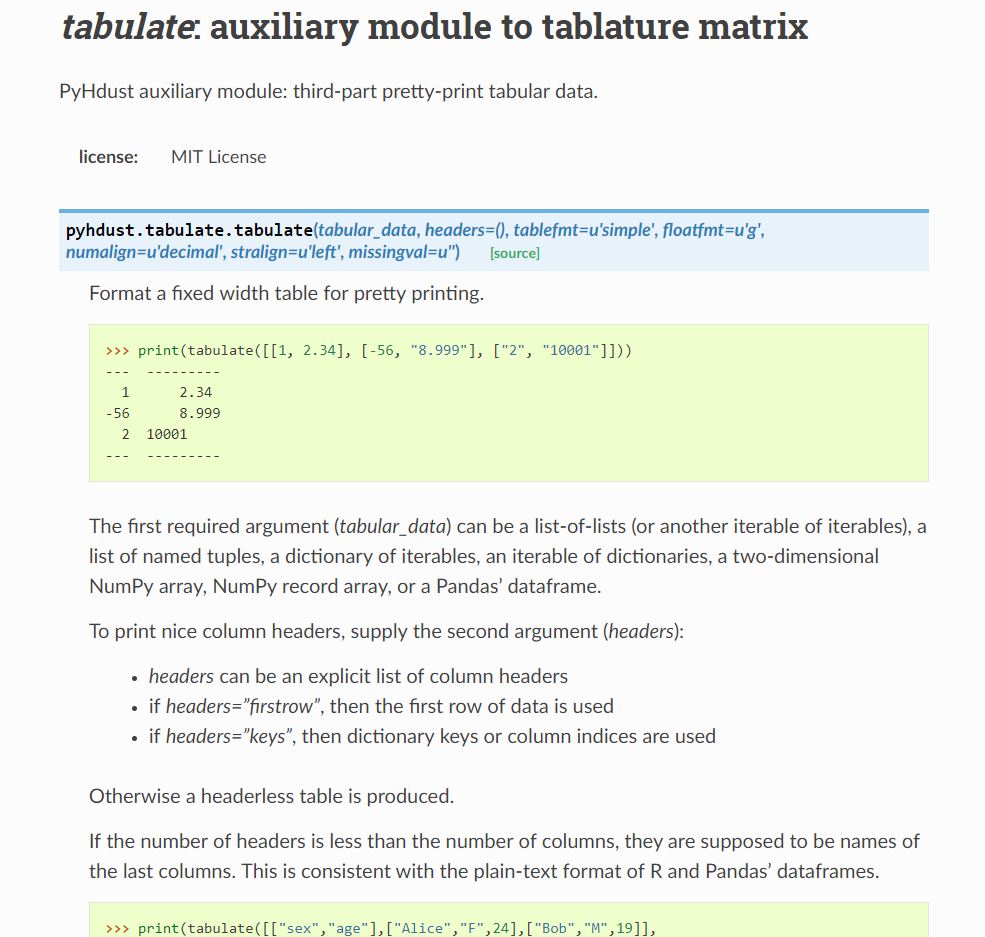
Inheritance is one of most important part on writing this program. This website helped me understand inheritance more and use it in my program.

1. <https://www.w3schools.com/python/python_dictionaries.asp>



Dictionary is one of a data types in python whuch are used for storing data in the form of keys and values. This site was refrence to me for learnig more about the data type use in in my report and program.

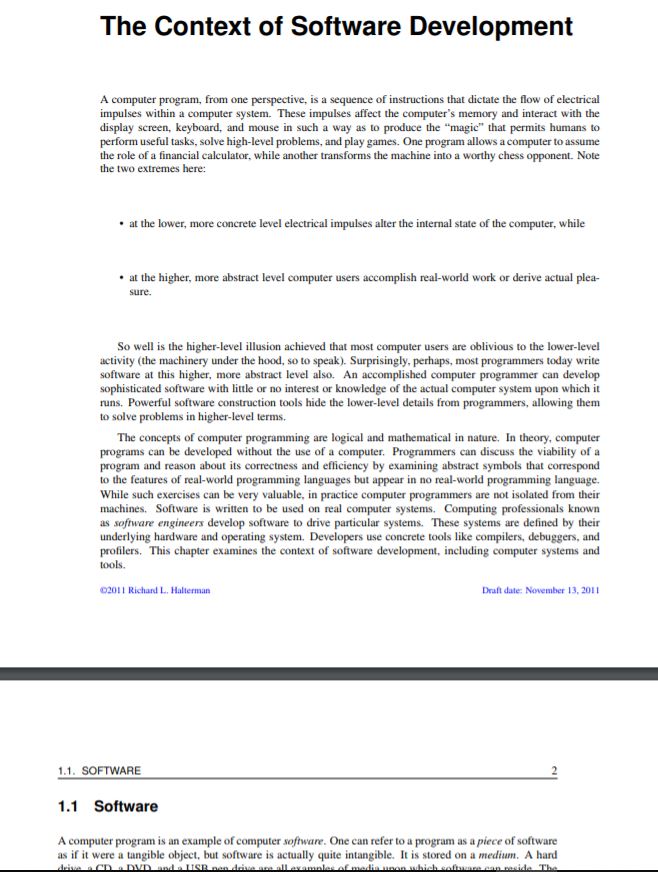
1. <https://pyhdust.readthedocs.io/en/latest/tabulate.html>



By using tablulate in my python code I was able to display my program in a clean and in a understanding way. Tabulate is just like giving tab but givivng tab after each code must be hard so I have used python in bullt module tabulate to give tab spacing.

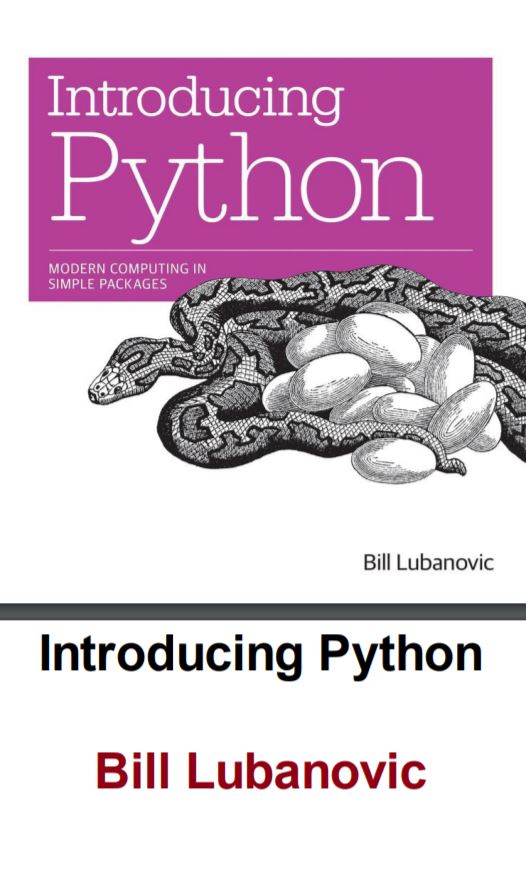
## **Book**

1. Learning to program with python:



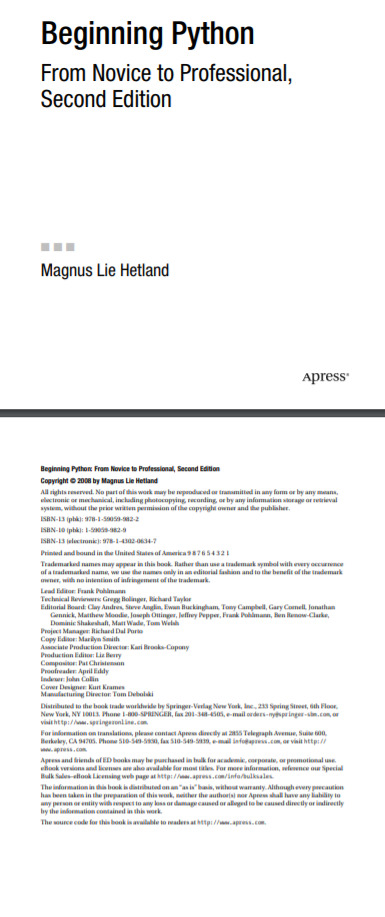
(Halterman, n.d.)

1. Introducing python:



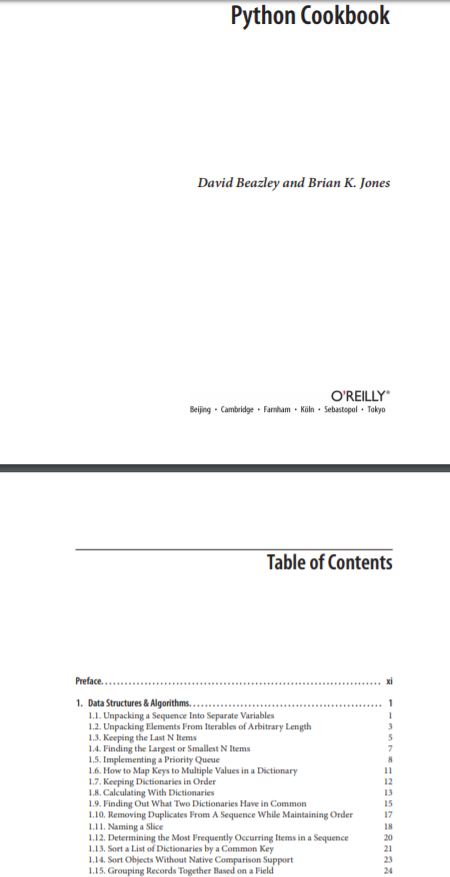
(Lubanovic, n.d.)

1. Beginning Python:



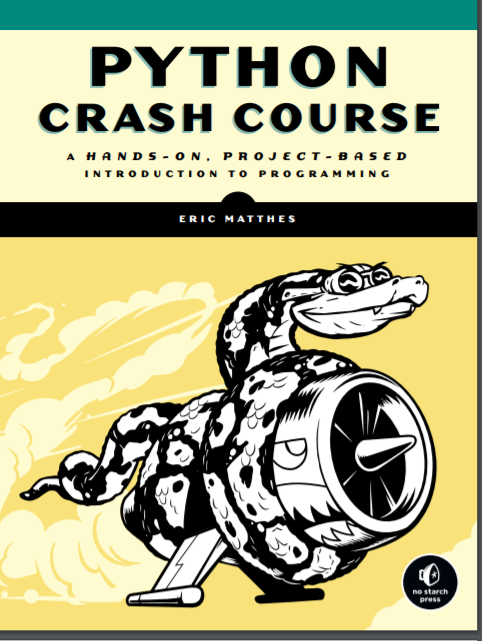
(Hetland, n.d.)

1. Python Cookbook:



(Jones, n.d.)

1. Python Crash Course:



(MATTHES, n.d.)

# **Conclusion:**

All the tasks assigned in the coursework was finally completed through much trial and errors. Conclusion: Finally, after complition of this project I have learn many new things. The tasks assigned in the coursework of Library management system weren't that easy. Alot of reasearch and study was done about Library management system using python. After the completion of this project we learn many different things about python and its programming. And lastly, the written and flowchart was drawn. Their was no bugs and errors in final check.the submission was done in time.

This coursework help to improve ourself and get better in programming.

It increase our skills and capacity of doing project in vast way. The most commonly used loops in it was for loop and while loop, if/else conditions and modularization was obtained. And learning would be alot helpful in pursue of development of career as a good programmer.

It requires a nights of hardwork and must be determinant for better improvement. Even it was tough it was fun doing it. This project was sucessfully completed, but their are more thing we must learn about programming in very vast ways. It feels great if our codes would be correct and program would run without bug and error.

# **Reference:**

* Cheusheva, S. (2018) *Ablebits* [Online]. Available from: <https://www.ablebits.com/office-addins-blog/2014/05/23/make-gantt-chart-excel/> [Accessed 17 January 2019].
* Dawson, A. (2017) *annedawson* [Online]. Available from: <http://www.annedawson.net/Python3_Install_Run.htm> [Accessed 16 January 2019].
* Fox, B. (2017) *A Medium Corporation* [Online]. Available from: <https://medium.com/@braydenfox/lucidchart-one-of-the-best-diagramming-wireframing-apps-noplag-review-dc234f16c85a> [Accessed 16 January 2019].
* Halterman, R.L. (n.d.) Learning to program with python.
* Hetland, M.L. (n.d.) *Beginning Python*.
* Jones, D.B.a.B.K. (n.d.) *Python Cookbook*.
* Lubanovic, B. (n.d.) *Introducing Python*.
* MATTHES, E. (n.d.) *Python Crash Course*.
* Python Software Foundation. (2018) *Python Software Foundation* [Online]. Available from: <https://docs.python.org/3/tutorial/datastructures.html> [Accessed 20 December 2018].
* Rouse, M. (2016) *WhatIs* [Online]. Available from: <https://whatis.techtarget.com/definition/pseudocode> [Accessed 18 December 2018].
* Sturtz, J. (2018) *Real Python* [Online]. Available from: <https://realpython.com/python-introduction/> [Accessed 16 January 2019].