



**Islington college**  
(इस्लिंग्टन कलेज)

**Module Code & Module Title**  
**CC4002NA Information Systems**

**Assessment Weightage & Type**  
**20% Individual Coursework**

**Year and Semester**  
**2018-19 Autumn**

**Student Name: Krishna Budhathoki**

**London Met ID:**

**College ID: np01cp4a180172**

**Assignment Due Date:2019-01-18**

**Assignment Submission Date:2019-01-18**

*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:*

#### *Purpose:*

*The purpose of the coursework is to make a library management system by using text files in the python to store data. Python should be able to use the data from the text file and manipulate the quantity and price so that it can be used for borrowing and returning the books for the library management system.*

#### *Problem statement:*

*One of the major problem library maybe facing while daily transaction is to manipulate the value of quantity and price after the books are borrowed and returned. Also mentioning the date and time and fine for late returning of book is necessary. The quantity should be updated according to the borrowed and returned book by the user.*

#### *Aim and objective:*

*My main objective is to make a library management system which will be able to interact with the user until it wants to exit the program. my aim is to make a borrowing and returning books option where the user can borrow and return books by choosing the option and the quantity should be updated in display. Then also the price and date should be mentioned in the billing file .whenever the person returns the book, he/she should be able to return the book according to the time and if he/she does not return it in time then he/she would be fine accordingly.*

#### *Proposed approach:*

*For the above mentioned problem, at first I would use python to convert the text file into a data structure such that it would be easy to manipulate the quantity and price. Then I can use import datetime so that I can input date and time in billing text and also in returning file where if the person submits the book late then he/she would be fined accordingly.*

#### *Target audience:*

*This program can be used by a small library to maintain its data which can be efficient for the user. it can also be helpful for students as it teaches us how to manipulate data from the text file and also using the data structure more efficiently.*

#### *Hardware and software requirements:*

*Python idle version 3.7.1 is used as software.*

Processors: Intel Atom® processor or Intel® Core™ i3 processor

Disk space: 1 GB Operating systems: Windows\* 7 or later, macOS, and Linux



Figure 1 flowchart.....	9
Figure 2 main .....	18
Figure 3 borrow .....	19
Figure 4 return.....	20
Figure 5 bill.....	20
Figure 6 late bill.....	21
Figure 7 test 1 .....	21
Figure 8 test 2 .....	22
Figure 9 test 3 .....	22
Figure 10 test 4 .....	23
Figure 11 test 5 .....	24
Figure 12 .....	25
Figure 13 .....	26
Figure 14 .....	27
Figure 15 .....	28
Figure 16 .....	29
Figure 17 .....	30

### Introduction:

In this project, we should be able to read the files from the text so that it can be used by python to display the data from the text file. We should clearly understand about data structure. We should be able to use the Boolean conditions for applying the different scenarios. Dictionary is used for manipulating the value of quantity and price for borrowing and returning the value .at first the program asks for options in which the user would have three options borrow, return and exit. The program should be able to ask the user to input name, name of the book and quantity after displaying the information .then the program should show the name of book and price to be paid. After executing the program, a text file should be saved which consists of borrowed date, price of the book, name of user and name of book. We can use import datetime for writing the date and time. For returning the books, the quantity should be able to return to its original state after it has been returned and it should also track its returned date such that the program can fine for late submission of book.it should also have exiting option so that the user can exit the program to his/her own choice. There should be five modules which are used in this program. Main module which is used for displaying the options. Borrowing module which is used for borrowing the book with given current time, return module which use for returning the book with is given time. Bill module should be made for making bills in the text file and finally late bill module should be made for fining the late submission.

### Discussion and analysis

At first, five modules were made which was given name according to their tasks. Main.py is used for displaying the options for menu of library management system. Borrow.py is used for displaying the data from the text file and borrow the books from the display and update the quantity of the books. We used a function which was called in main.py option 1.we used dictionary data structure to manipulate the data from the text file. The borrow.py should also be able to store the date and time in which the user has borrowed the books which is done by importing datetime. The third module is return.py which is used for returning the books. The book should be returned in time

or the user would be fined according to time. This module also recovers the values of quantity which was reduced after borrowing the book. After the book is returned, the quantity of book should also be recovered. The module name is bill.py which stores bill for borrowing the book. For every person who borrowed the book, a random file would be saved containing the name, book name, price and borrowed date which would be used later in returning the book. The last module is latebill.py which is used for fining the users who have not returned the books after 10 days. They would be given a separate bills in text file which contains name, name of books and price of fine.

### Algorithm

step 1:start

step 2:input option(borrow,return,exit)

step 3: if option==1 then

    go to step 4

    else if option==2 then

        go to step 12

    else option==3 then

        go to step 20

Step 4: run borrow.py

Step 5: display book,quantity and price

Step 6: input user,userbook,quantity.

Step 7: if userbook==librarybook and quantity is less than quantity in library then

    If yes then go to step 9

    If no then go to step 8

Step 8:display error.

Step 9:display user , userbook , quantity, price and date

Step 10:generate bill on above information(step 9)

Step 11:return to step 1

Step 12input return.py

Step 13:display book quantity and price

Step 14:input user,userbook,quantity and date of purchase

Step 15:if user,userbook,quantity match then

Go to step 17:

Step 16:display error

Step 17:check whether the book was returned in time

If yes then

Go to step 19

else

Go to step 18

Step 18:according to the late submission ,fine

Step 19:issue the check bill as returned in time

Step 20:exit

Step 21:stop





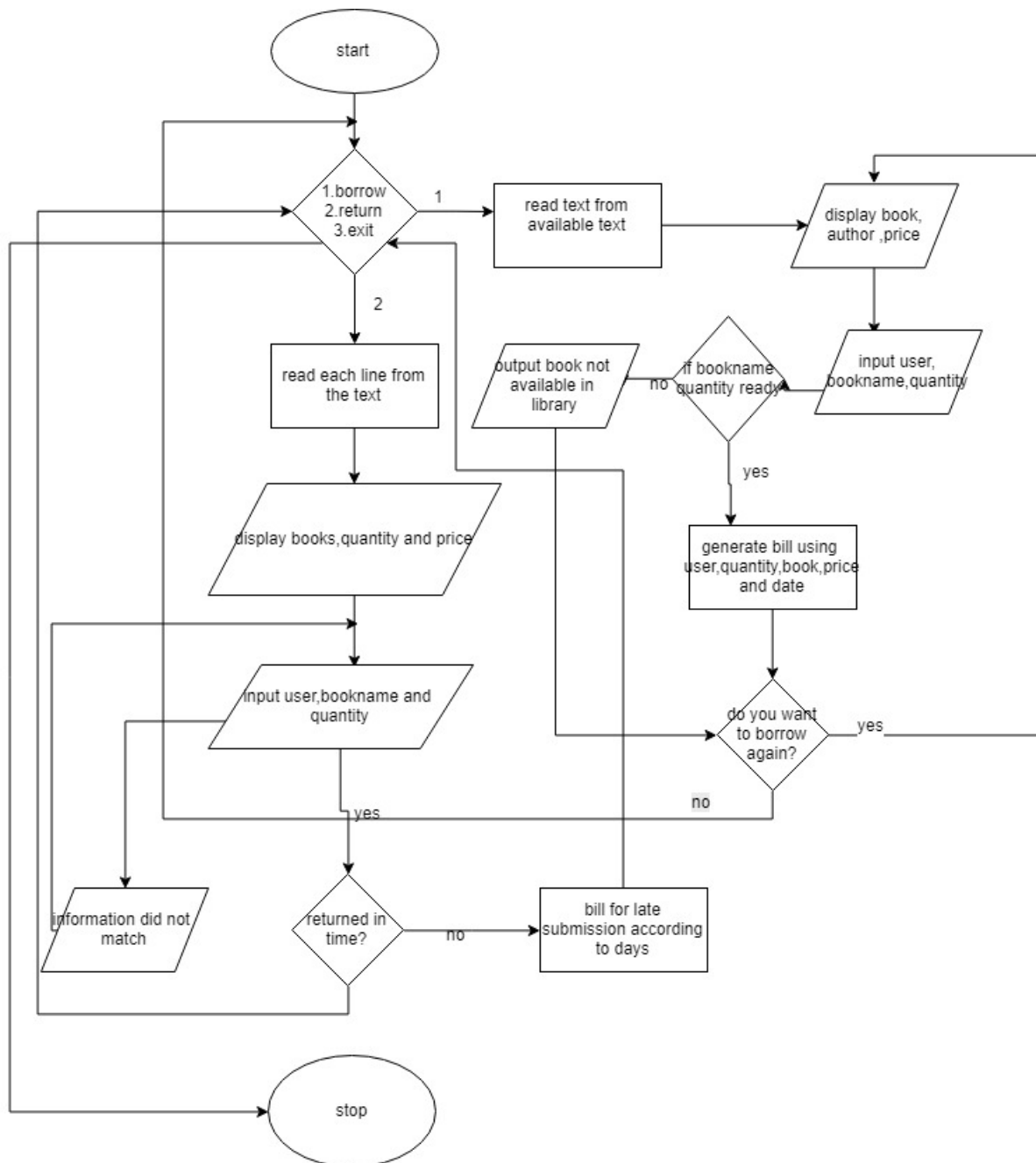
*Flowchart*

Figure 1 flowchart

*Pseudo code:*

*Pseudo code for main.py:*

*While true:*

- 1. output "borrow"*
- 2. output "return"*
- 3. output "exit"*

*End while*

*Option=" "*

*Success=false*

*While success==false:*

*Try :*

*option=integer(input enter a valid value)*

*except:*

*output "enter a valid value"*

*end while*

*if option==1 then*

*read.borrow.py*

*else if option==2 then*

*read.return.py*

*else*

*output "thank you for visting our library!"*

*end if*

*end*

*Pseudo code for borrow.py*

*Import datetime*

*Import random*

*Import bill*

*While true:*

*Open libraryfile*

*Read libraryfile*

*Output(bookid bookname author quantity price)*

*A=1*

*For line in libraryfile*

*Output("b00",a,"\\t\\t"+replace"," with "\\t\\t")*

*A=A+1*

*User=empty*

*Success=false*

*While success==false:*

*Try:*

*User=input(string)*

*Except:*

*Output(enter a valid value)*

*User\_book=empty*

*Success=false*

*While success==false:*

*Try:*

*User\_book=input(string)*

*Except:*

*Output(enter a valid value)*

*Open libraryfile*

*Read libraryfile*

*For line in libraryfile:*

*D=empty dictionary*

*S=1(assign the value)*

*Line Replace ("\\n" with " ")*

```

        Line Split(,)
    S=s+1
    X=0(define a variable such that only in some keys and value in dictionary)
End for
For key(i),value(j) in dictionary items:
    Price=quantity*float of j[3]
    If user_book==j[0] then
        X=1
        If quantity >0 and quantity<j[3] then
            b=int(j[2])
            b-=quantity
            d[i][2]=b

        bill.generate(user,user_book,quantity,price,DateofPurchase)
            record=
user,user_book,str(quantity),str(price),dateofpurchas
            open list_borrow
            write record
            close list_borrow
            output(user,str(quantity),user_book,price)
        else:
            output(string,quantity,user_book)

    if x==0 then
        output(string,user_book)
    open libraryfile
    write libraryfile
    for key(i),value(j) in libraryfile:
        for var(k) in value(j):
            write libraryfile(str(k))
        close libraryfile
    end for

```

```
again=input(string)
if again.lower()==y then
    continue
else:
    stop
```

*pseudo code for return:*

```
import datetime
import latebill
userlibrary=empty
success=false
while success==false:
    try:
        name = input(string)
        Success = True
    except:
        output(string)
user_book=empty
success=false
while success==false:
    try:
        user_book = input(string)
        Success = True
    except:
        output(string)
quantity=empty
success=false
while success==false:
    try:
        quantity = input(string)
```

```

        Success = True
    except:
        output(string)

usertime=0
success=false
while success==false:
    try:
        user_book = input(string)
        Success = True
    except:
        output(string)

open libraryfile
read libraryfile
d=empty dictionary
for each line in libraryfiles:
    s=1(assign the value)
    line replace("\n" with "")
    line split(",")
    s=s+1
    X=0(define a variable such that only in some keys and value in dictionary)
End for

for key(i),value(j) in libraryfile:
    latefine=$0.5
    if user_book == j[0] then
        x = 1
        if quantity > 0 and quantity < 100 then
            b= int(j[2])
            d[i][2] = b
            if User_Time < 10:
                output=Userlibrary, Userbook, str(UserQuantity), DateofReturn

```

```
        libraryfile open
        libraryfile output
        libraryfile close
    else:
        quantity == False
    end for
end if

If x == 0:
    User_book == False
    libraryfile open
    if User_Time > 10 then
        Billlate.generate(User,User_book,late_fine,DateofReturn)
    For key (i),value(j) in d.items:
        for k(variable) in value(j):
            file write (k , )
        close libraryfile
```

pseudo code for bill:

```
import random
import datetime
file name =pay,random number
open filename as bill_file
bill_file.write(Username: username)
bill_file.write(Book: book)
bill_file.write(quantity: str(quantity))
bill_file.write(Price: str(price))
bill_file.write(Date: datetime)
bill_file.close
```

Pseudo code for latebill:

Import random

Import datetime

file name =latebill,random number

open filename as latebill\_file

latebill\_file.write(Username: username)

latebill\_file.write(Book: book)

latebill\_file.write(quantity: str(quantity))

latebill\_file.write(Price: str(fine))

latebill\_file.write(Date: datetime)

latebill\_file.close



### *Data structure:*

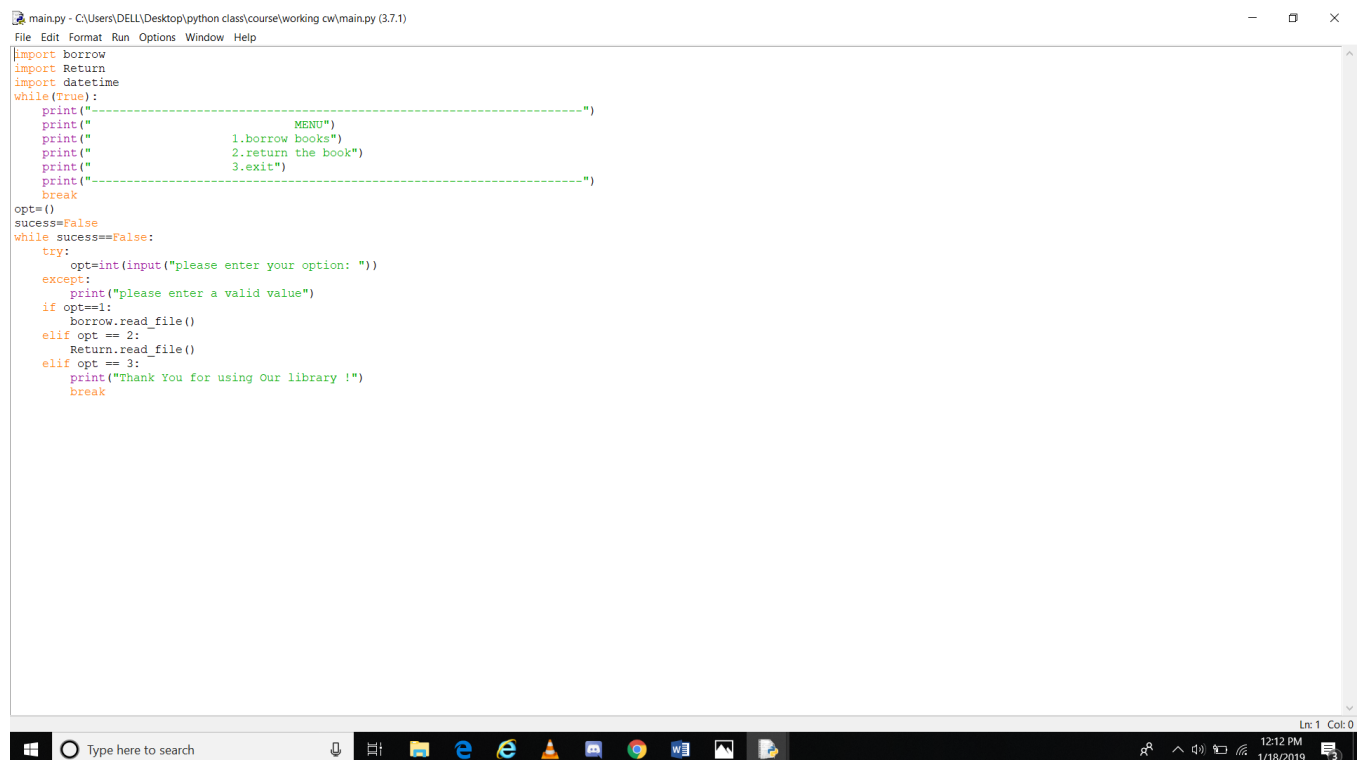
*Dictionary: dictionary is a collection of data which stores data using key:value which is inclosed by curly braces{} and separated by commas','.in this program, dictionary is used for manipulating the values of quantity and price. Python uses every line to iterate with text file such that it makes a definitive dictionary data type.then we can use those data by using values from the dictionary to manipulate the quantity so that it can be updated in the display.*

*String is used for writing and reading the text files.the quantity and price is converted from string into integer and floats so that it can be used for manipulating the data.later we can write the data for the change in quantity.*

*Boolean is used for iterating the program such that it can run until its true and it is also used for checking the invalid values in username ,bookname and others respectively.*

## Program:

The program consists of five modules which are `main.py`, `borrow.py`, `return.py`, `bill.py` and `latebill.py`. `main.py` is used for calling the `borrow.py` and `return.py`. `borrow.py` is used for displaying the data from the textfile. After displaying the textfile, the data in textfile is converted into dictionary which is used to manipulating the value of quantity while borrowing. Proper name is given for user, name of book, quantity and price. Then loop is used for iterating the value of quantity from the dictionary which is used for updating the quantity in the display. `bill.py` generate is used for passing the parameters of user, quantity, bookname. Then a text file is used for writing list of user who borrowed the books. `datetime` is used for inputting the date and time.



```

main.py - C:\Users\DELL\Desktop\python class\course\working cw\main.py (3.7.1)
File Edit Format Run Options Window Help
import borrow
import Return
import datetime
while(True):
    print("-----MENU-----")
    print("1.borrow books")
    print("2.return the book")
    print("3.exit")
    print("-----")
    opt=int(input("please enter your option: "))
    try:
        opt=int(input("please enter your option: "))
    except:
        print("please enter a valid value")
    if opt==1:
        borrow.read_file()
    elif opt == 2:
        Return.read_file()
    elif opt == 3:
        print("Thank You for using Our Library !")
        break
    success=False
while success==False:
    try:
        opt=int(input("please enter your option: "))
    except:
        print("please enter a valid value")
    if opt==1:
        borrow.read_file()
    elif opt == 2:
        Return.read_file()
    elif opt == 3:
        print("Thank You for using Our Library !")
        break
    success=False

```

Figure 2 main

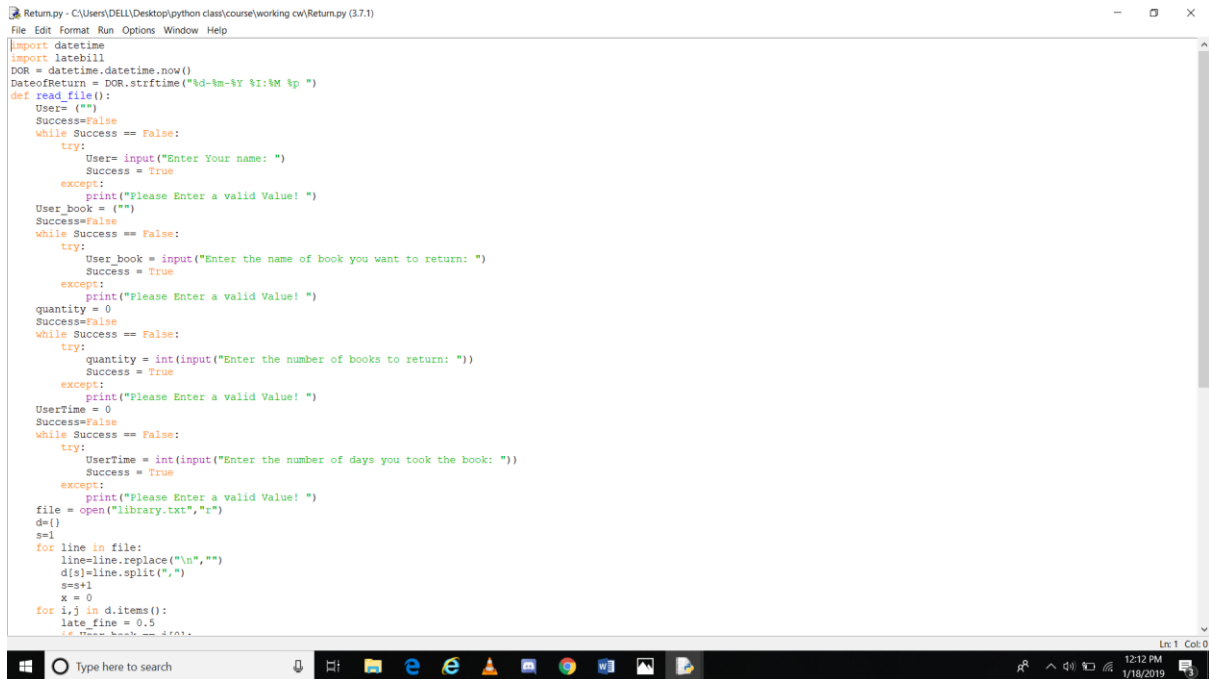
```

borrow.py - C:\Users\DELL\Desktop\python class\course\working cw\borrow.py (3.7.1)
File Edit Format Run Options Window Help
import bill
import datetime
import random
DOP = datetime.datetime.now()
DateofPurchase = DOP.strftime("%d-%m-%Y %I:%M %p ")
DateofReturn = DOP+datetime.timedelta(days = 10)
DateofReturn = DateofReturn.strftime("%d-%m-%Y %I:%M %p ")
def read_file():
    while True:
        file=open("library.txt","r")
        print("-----")
        print("Book ID          Book Name          Author          Quantity          Price($) ")
        print("-----")
        a=1
        for line in file:
            print(" b00",a,"\t\t\t"+line.replace(",","","\t\t\t"))
            a=a+1
        user=""
        success=False
        while success==False:
            try:
                user=input("enter your name: ")
                success=True
            except:
                print("please enter a valid value")
        user_book=""
        success=False
        while success==False:
            try:
                user_book=input("enter the name of the book you want: ")
                success=True
            except:
                print("enter a valid value")
        quantity=0
        success=False
        while success==False:
            try:
                quantity=int(input("enter the number of books you want: "))
                success=True
            except:
                print("enter a valid value")
        file=open("library.txt","r")
        d={}
        s=1
        for line in file:
            line=line.replace("\n","")
            d[s]=line.split(",")
            s=s+1

```

Figure 3 borrow

*Return.py module is similar to borrow.py because it uses user,userbook and quantity to input the value.then it is iterated through dictionary value for updating the quantity of books.we also have to mention the return date in which the user would be fined if he/she returns after 10 days accordingly.*



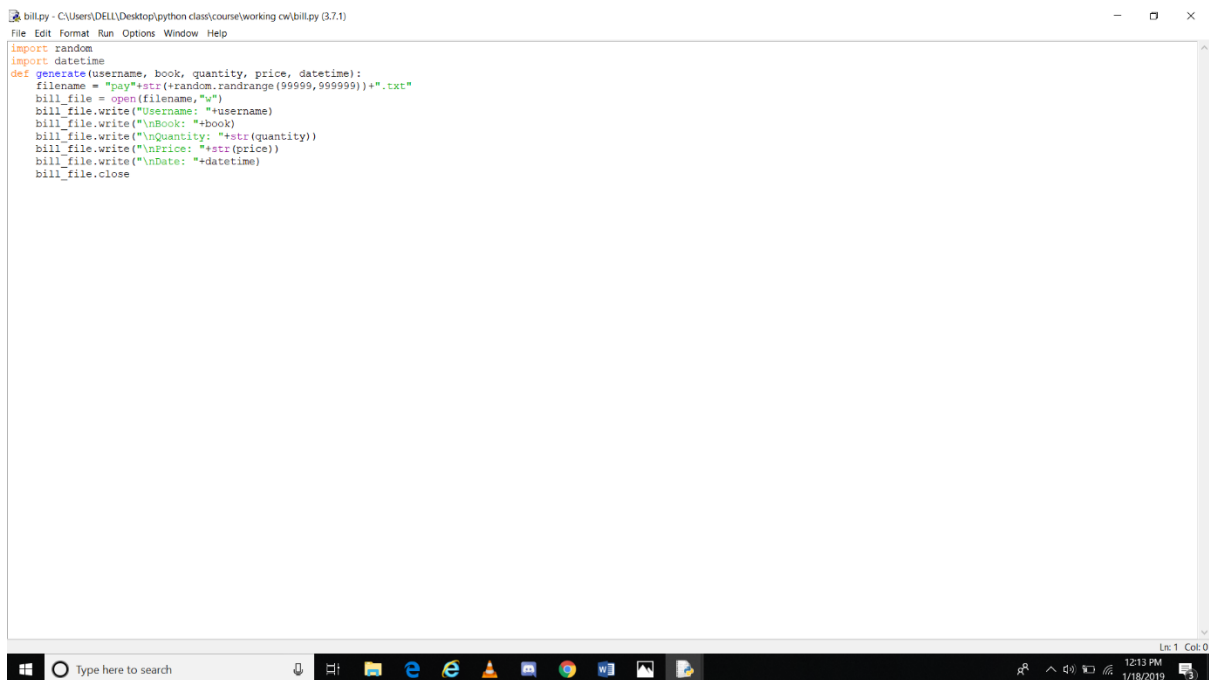
```

Return.py - C:\Users\DELL\Desktop\python class\course\working cw\Return.py (3.7.1)
File Edit Format Run Options Window Help
import datetime
import latebill
DOR = datetime.datetime.now()
DateofReturn = DOR.strftime("%d-%m-%Y %I:%M %p ")
def read_file():
    User= ("")
    Success=False
    while Success == False:
        try:
            User= input("Enter Your name: ")
            Success = True
        except:
            print("Please Enter a valid Value! ")
    User_book = ("")
    Success=False
    while Success == False:
        try:
            User_book = input("Enter the name of book you want to return: ")
            Success = True
        except:
            print("Please Enter a valid Value! ")
    quantity = 0
    Success=False
    while Success == False:
        try:
            quantity = int(input("Enter the number of books to return: "))
            Success = True
        except:
            print("Please Enter a valid Value! ")
    UserTime = 0
    Success=False
    while Success == False:
        try:
            UserTime = int(input("Enter the number of days you took the book: "))
            Success = True
        except:
            print("Please Enter a valid Value! ")
    file = open("library.txt","r")
    d=[]
    s=1
    for line in file:
        line=line.replace("\n","")
        d[s]=line.split(",")
        s=s+1
        x = 0
    for i,j in d.items():
        late_fine = 0.5
        if User_book == d[i][1]:

```

Figure 4 return

*Bill.py is used for storing the name of user who borrowed the books.it is used by using the parameters from the borrow.py. which is used to store data by writing in text file.also the data of purchase should be mentioned by improting datetime.*



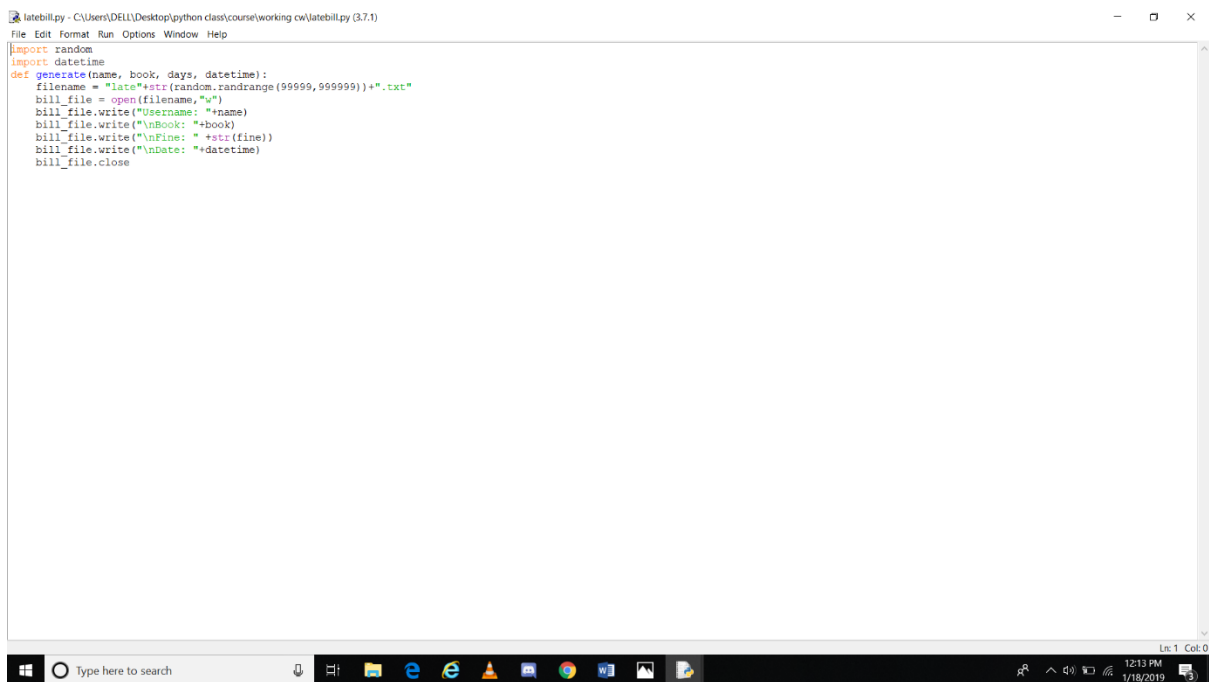
```

bill.py - C:\Users\DELL\Desktop\python class\course\working cw\bill.py (3.7.1)
File Edit Format Run Options Window Help
import random
import datetime
def generate(username, book, quantity, price, datetime):
    filename = "pay"+str(random.randrange(99999,999999))+".txt"
    bill_file = open(filename,"w")
    bill_file.write("Username: "+username)
    bill_file.write("\nBook: "+book)
    bill_file.write("\nQuantity: "+str(quantity))
    bill_file.write("\nPrice: "+str(price))
    bill_file.write("\nDate: "+datetime)
    bill_file.close

```

Figure 5 bill

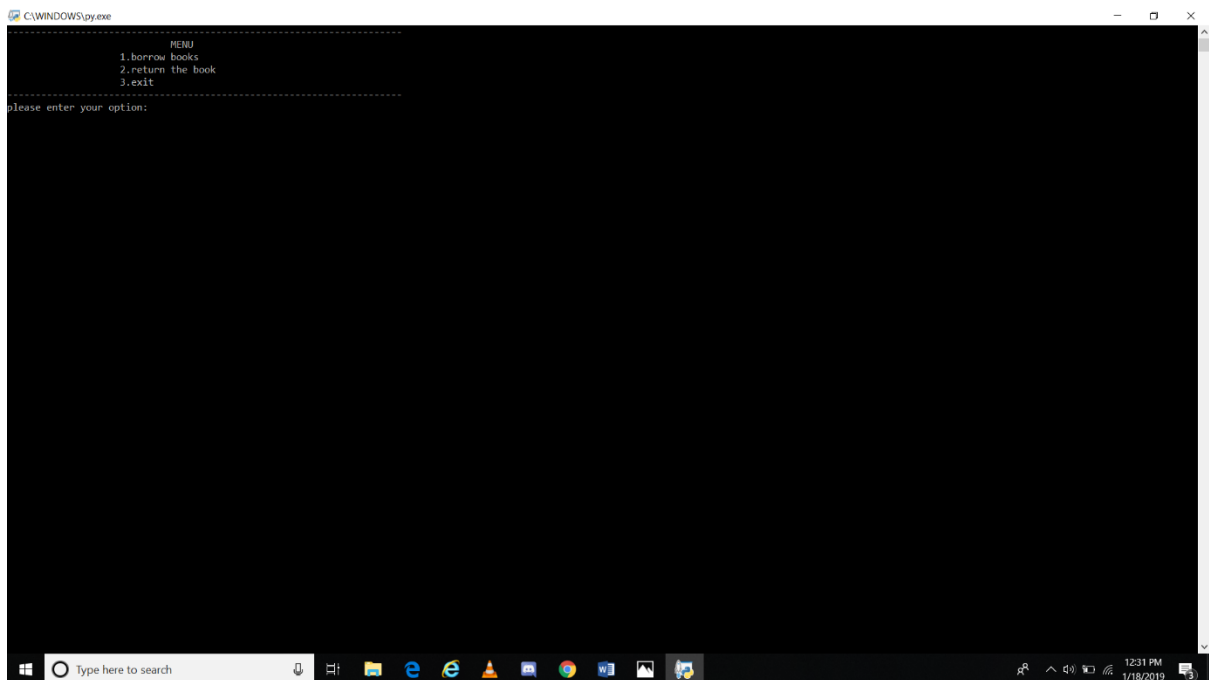
*Latebill.py is used for storing the data in text file when the user returns the book after days.the user gets fined accordingly.it is similar to bill.py where we store date of return and fine to be paid.*



```
latebill.py - C:\Users\DELL\Desktop\python class\course\working cw\latebill.py (3.7.1)
File Edit Format Run Options Window Help
import random
import datetime
def generate(name, book, days, datetime):
    filename = "late"+str(random.randrange(99999, 999999))+".txt"
    bill_file = open(filename, "w")
    bill_file.write("Username: "+name)
    bill_file.write("\nBook: "+book)
    bill_file.write("\nFine: " +str(fine))
    bill_file.write("\nDate: "+datetime)
    bill_file.close
```

Figure 6 late bill

## testing



```
C:\WINDOWS\py.exe
MENU
1.borrow books
2.return the book
3.exit
.....
please enter your option:
```

Figure 7 test 1

```

C:\WINDOWS\py.exe
MENU
1.borrow books
2.return the book
3.exit

please enter your option: 1

Book ID      Book Name      Author      Quantity      Price($)
-----
b00 1        Harry Potter   JK Rowling   19             2
b00 2        Start with Why Simon Sinek   8              1.5
b00 3        Programming    John Smith   18             1.5
b00 4        The Hobbit     R. R. Tolkien 21             2
b00 5        The Odyssey    Homer        22             1.6
b00 6        Life of Pi     Yann Martel  12             2

enter your name: 

```

Figure 8 test 2

```

C:\WINDOWS\py.exe
please enter your option: 1

Book ID      Book Name      Author      Quantity      Price($)
-----
b00 1        Harry Potter   JK Rowling   19             2
b00 2        Start with Why Simon Sinek   8              1.5
b00 3        Programming    John Smith   18             1.5
b00 4        The Hobbit     R. R. Tolkien 21             2
b00 5        The Odyssey    Homer        22             1.6
b00 6        Life of Pi     Yann Martel  12             2

enter your name: krishna
enter the name of the book you want: Harry Potter
enter the number of books you want: 2

krishna has borrowed 2 Harry Potter at $ 4.0 in 18-01-2019 12:31 PM
Do you want to borrow again?

Book ID      Book Name      Author      Quantity      Price($)
-----
b00 1        Harry Potter   JK Rowling   17             2
b00 2        Start with Why Simon Sinek   8              1.5
b00 3        Programming    John Smith   18             1.5
b00 4        The Hobbit     R. R. Tolkien 21             2
b00 5        The Odyssey    Homer        22             1.6
b00 6        Life of Pi     Yann Martel  12             2

enter your name: 

```

Figure 9 test 3

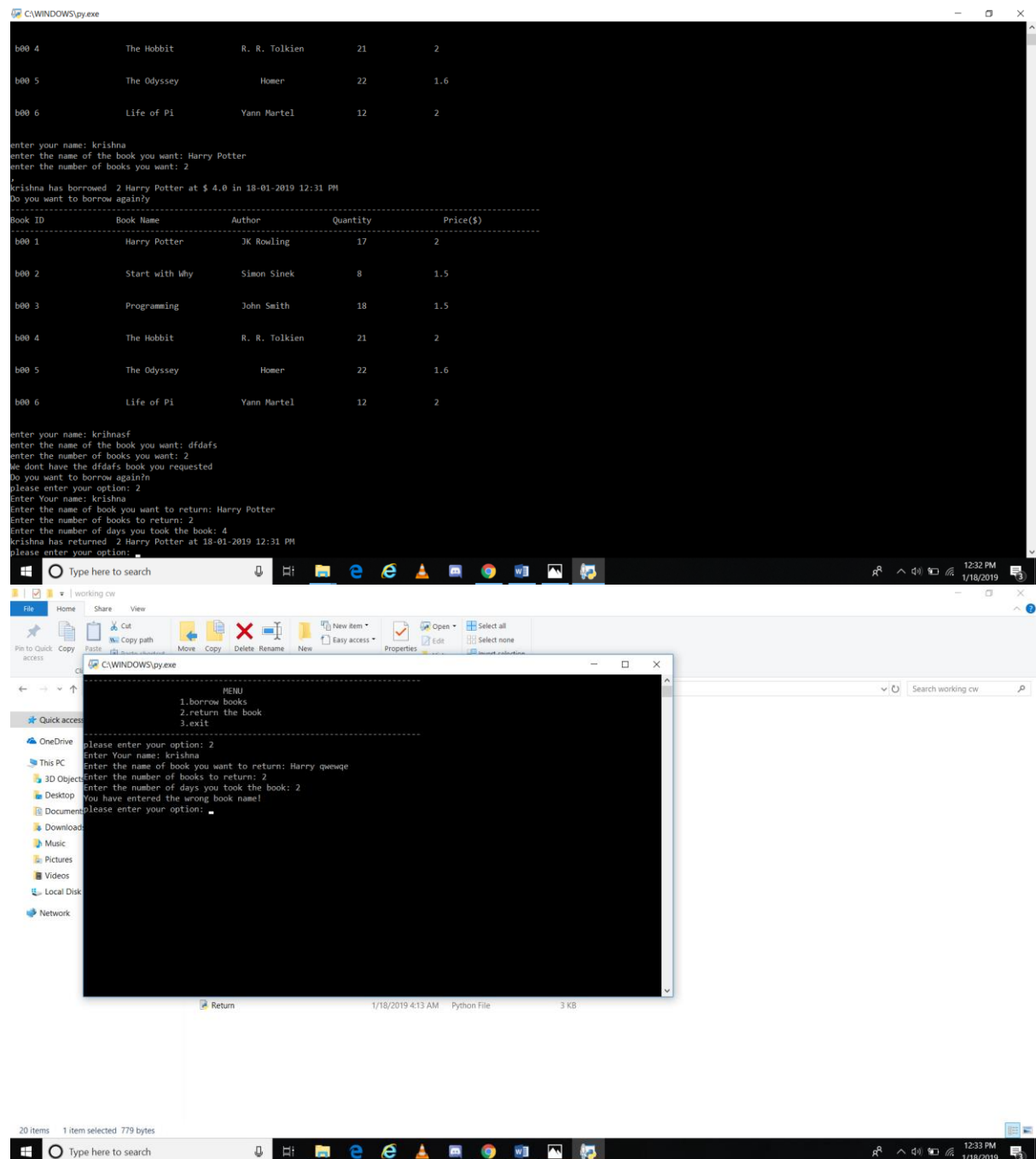


Figure 10 test 4

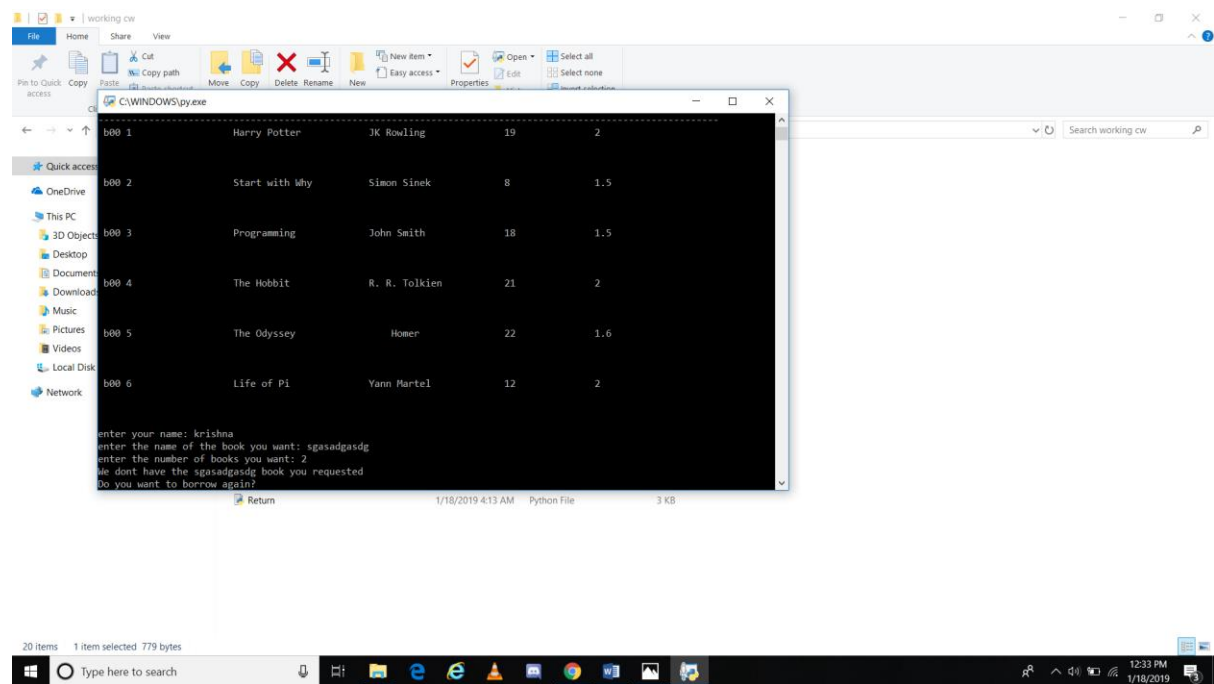


Figure 11 test 5

## Research

### Working with text files in python:

At first we need to be able to open the text file before using it.

The syntax to open a text file in the python is

`File_object=open ("filename","mode")` where `file_object` is the variable to add the file object.

Mode is an optional argument because its default value is "r" which means reading the file but mode can be used in different situations like reading the file, writing the file and appending the file. There is also "r+" which is special read and write mode which is used to handle both actions when working with a file. For writing a file we can use the following example:

`F=open ("workfile","w")`



The above example will be able to allow python to write in text file. If we don't have any text file with the following name then python will automatically create a text file so that it can store the written data. (Lutz, december,2010)

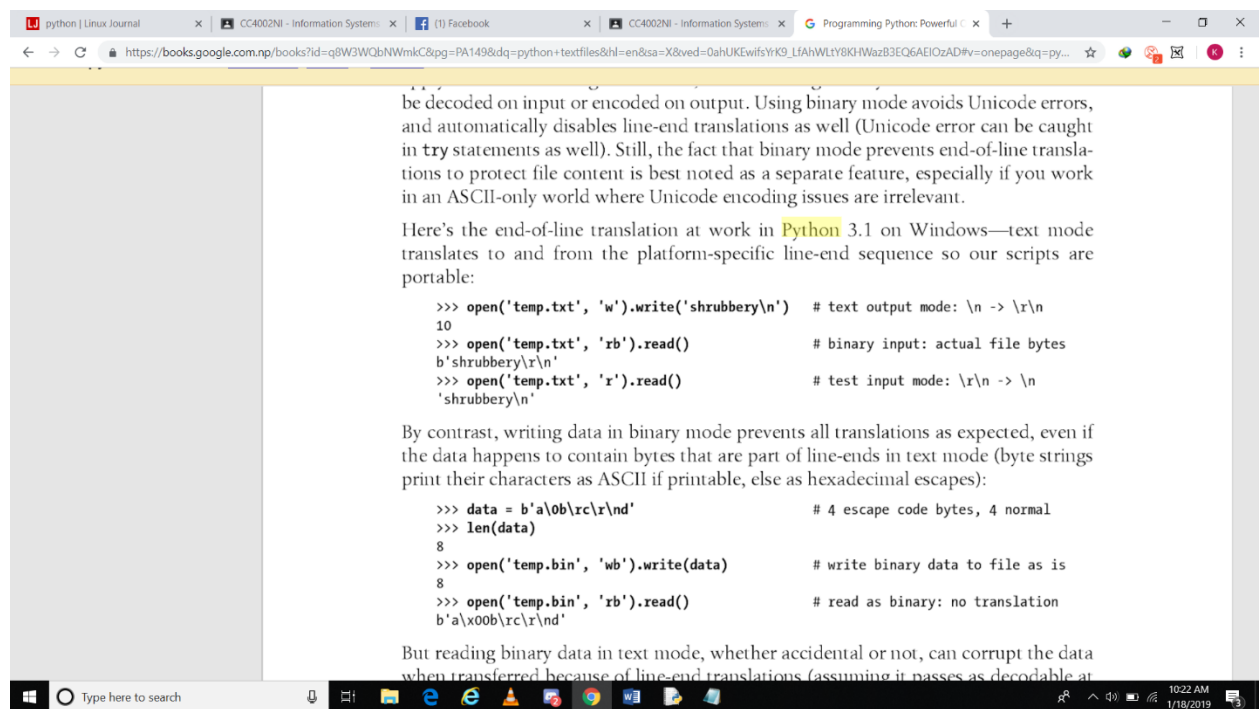


Figure 12

We can use loops over a file object by opening the file and using the read mode. we can use following example:

File=open("testfile.txt","r") where testfile.txt is a text file which is readable

For line in file:

Print (line)

This will return the every strings present inside the text file.

Whenever we keep the data in write mode and we write a specific thing then we should always close the file by using file.close().

For example:

File=open("testfile.txt","w")

File.write("this is a test")

File.close() (Dawson, 2010)

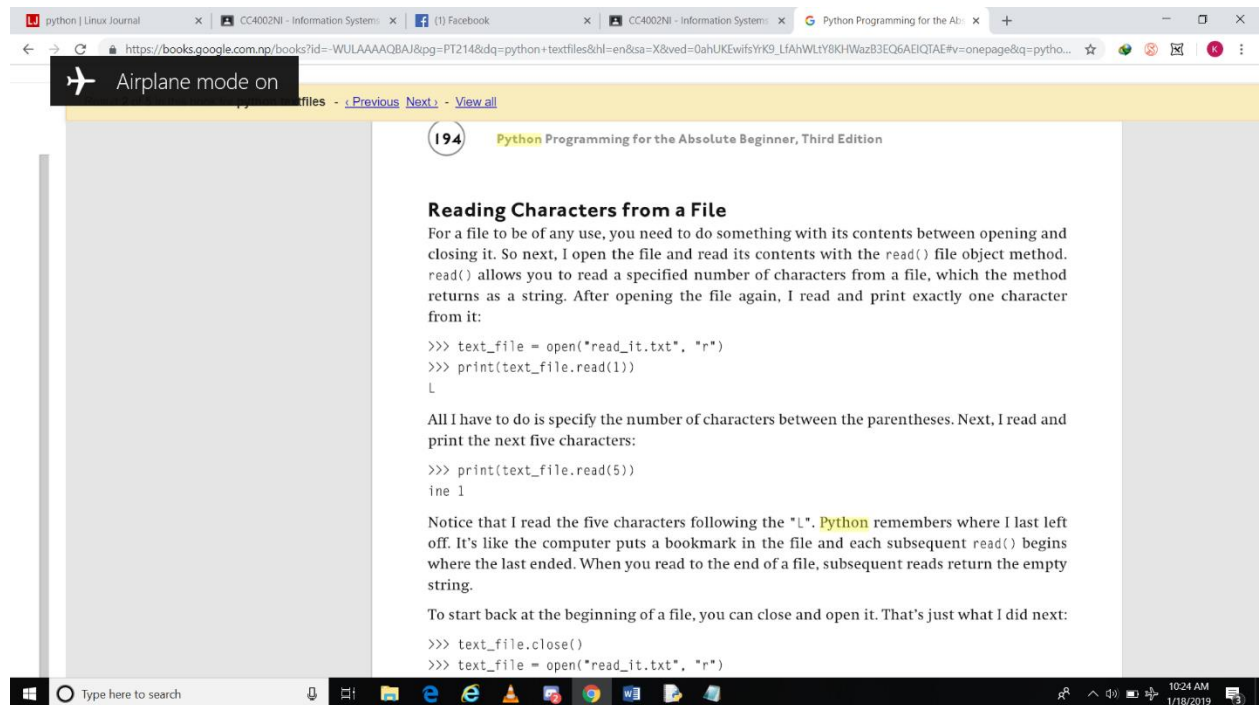


Figure 13

The text file can also be modified as making it open as a specific variable.

For example:

With `open(filename,mode)as fileobject`:

For line in fileobject:

#process line

Defining and using functions:

Several types of functions are available in python. we can define functions in python using `def` keyword. the form looks like this:

`Def functionName(parameter1,parameter2,...)`

Function block

Python functions can always return value. we can specify an explicit return value using the `return` keyword; otherwise, python returns `none` by default. (if we find unexpected `none` values appearing output, we should check that the function concerned has an

explicit return statement in it body.)we can also give default values to the parameters by following the name with an equal sign and the value. (Laura Cassell, 2015)

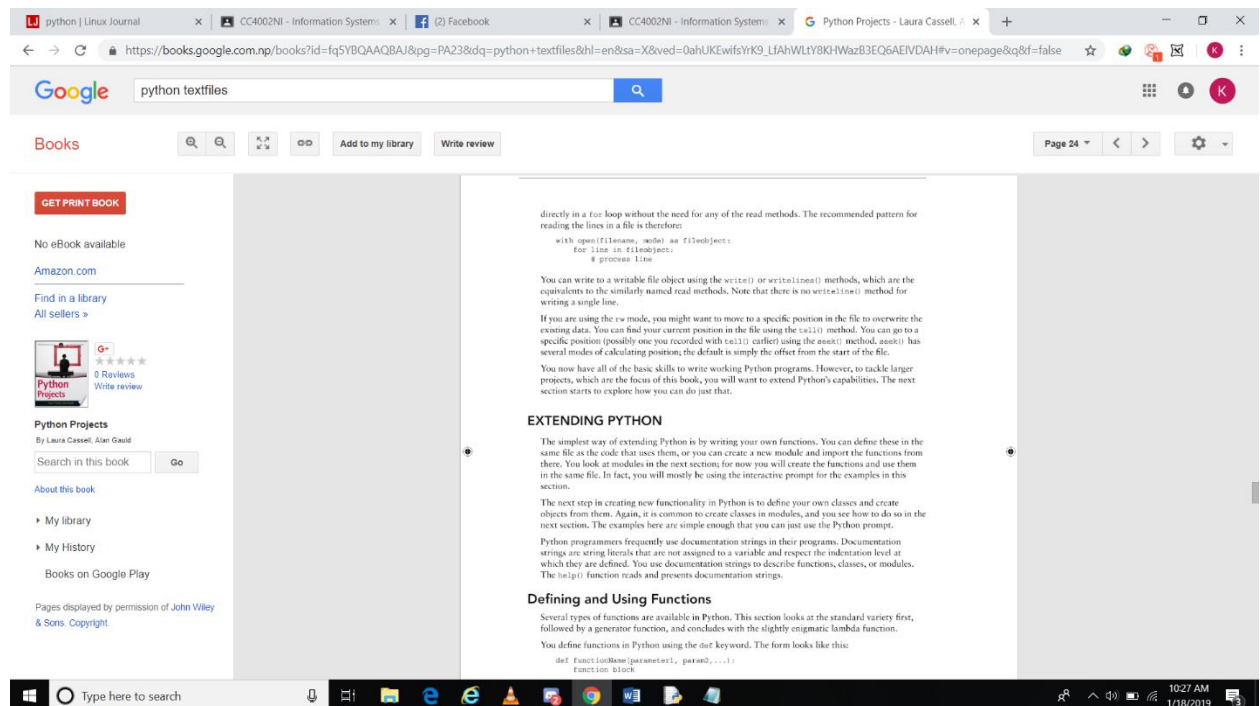


Figure 14

```
def straight_line(gradient,x,constant):
```

''' returns y coordinate of a straight line

>gradient \*x+constant '''

Return gradient\*x+constant

In the above code we have define the fucktion.

Now that we have defined the function,we can test it using some simple we can calculate in our head.we can try calling the function with gradient of 2, an x value of 4 and a constant of -3

#test with a single value first

```
Straight_line(2,4,-3)
```

~~>5 (Chun, 2001)

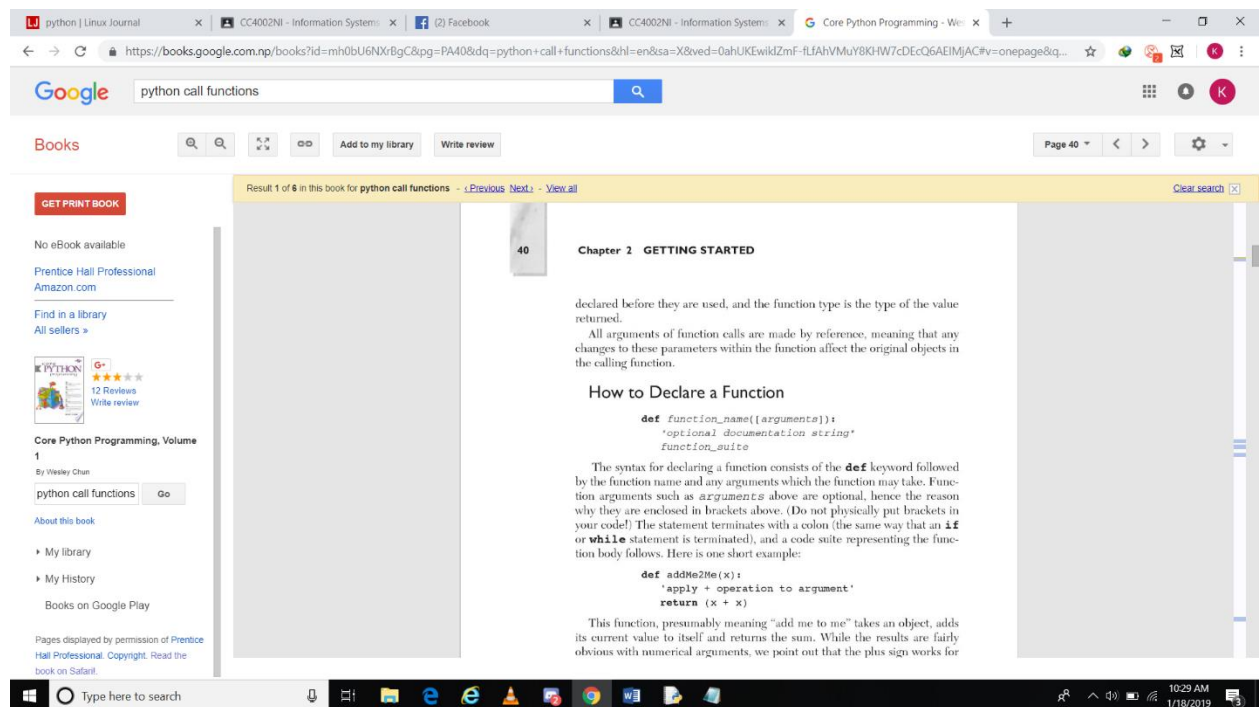


Figure 15

### Module creating, importing and sharing:

Actually creating a python module means opening a python file (filename.py).creating every module does not necessarily means that file has to import. Python can run the module with alone one module.while creating a module , the name of the module is used as variable names in python while importing a file.

For example:

```
def hello_printer():
```

```
    print "Hello world!"
```

```
Name = "John"
```

Import my\_module #name of the above file

```
My_module.hello_printer()
```

```
Print("creator:",my_module.name)
```

The (my\_module.py) is designed as module whose code is imported and reused in other python files.

Result:

Hello world !

Creator:john (Pavloski, november28,2017)

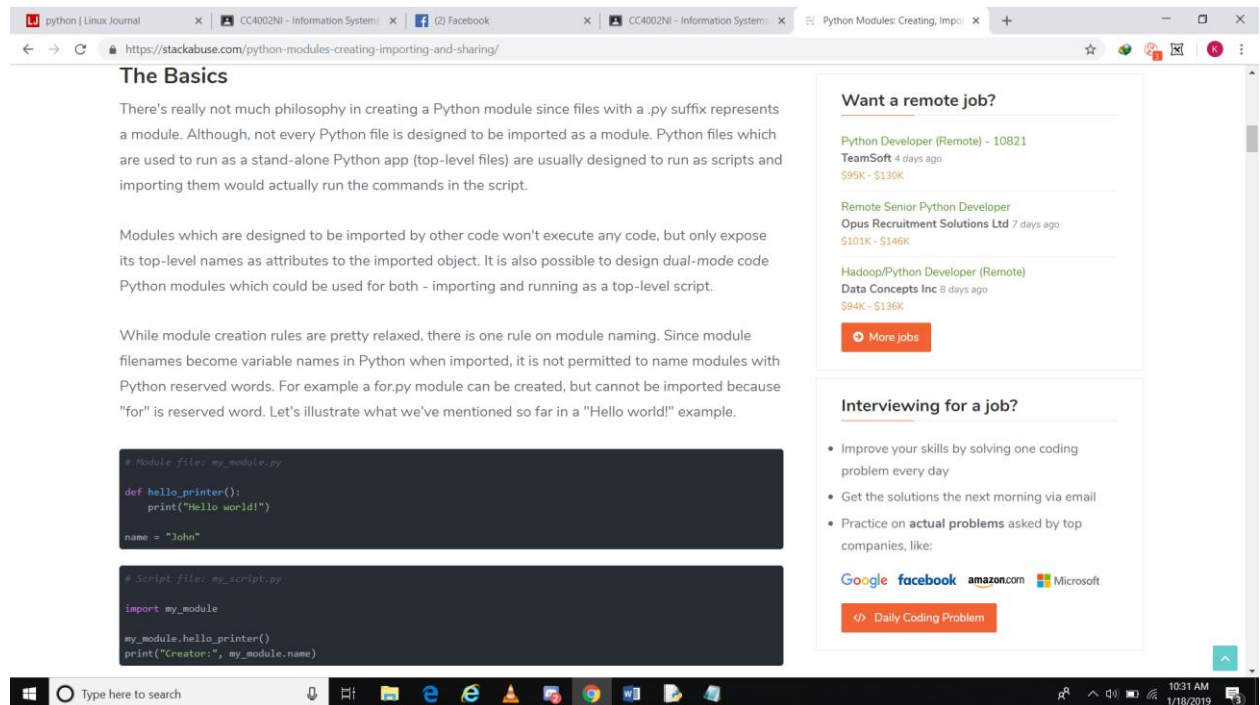


Figure 16

We can also rename the name of imported module.

For example:

Person1= { #the program is saved as mymodule.py

"Name": "john",

"Age":36,

"Country": "Norway"

)

Import mymodule as mx

A=mx.person1["Age"]

Print(A)

Gives result:36

Importing date and time:

A date in Python is not a data type of its own, but we can import a module named `datetime` to work with dates as date objects.

For example:

Import datetime

x=datetime.datetime.now()

print(x)

the result is current time during the program runned.

We can also use `strftime()` which can be used for taking certain parameter to specify the format of date and time.

For example:

Import datetime

```
x=datetime.datetime.(2018,6,1)
```

```
print(x.strftime("%B"))
```

result : june #because %B represents month name in full version

(IPSTech, 2016)

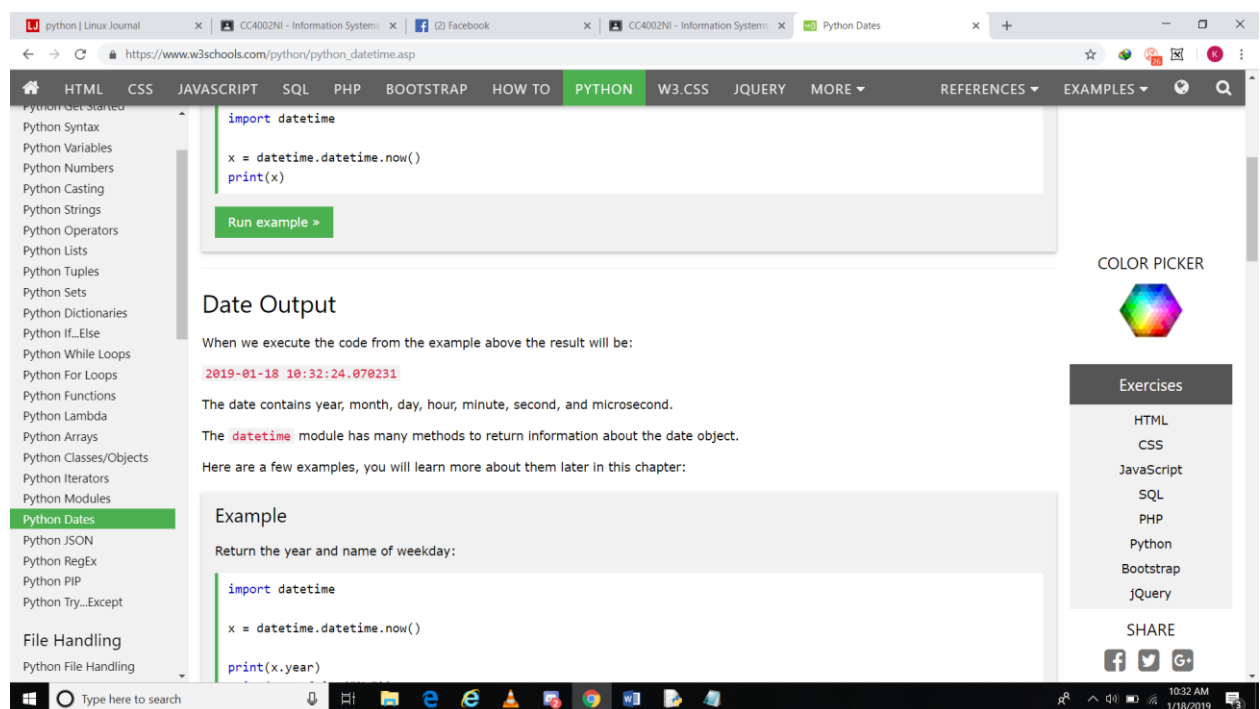


Figure 17

Working with dictionary:

Dictionaries, also known as associative arrays or hashes in Perl and other common languages, consist of key:value pairs enclosed in braces. They are particularly useful data structures because, unlike lists and tuples, the values are not restricted to being indexed solely by the integers corresponding to sequential position in the data series. Rather, the keys in a dictionary serve as the index, and they can be of any immutable data type (strings, numbers, or tuples of immutable data). A simple example, indexing on three-letter abbreviations for amino acids and including molar masses, would be `aminoAcids = {'ala':('a','alanine', 89.1),'cys':('c','cysteine', 121.2)}`. (Berk Ekmekci , Charles E. McAnany , Cameron Mura , june 7,2016)

**Conclusion:**

at the end of the coursework I want to conclude that we can make library management system using python from the text file which very helpful for us in future.it helped us in understanding the data structure more briefly and it helped us to manipulate the data from the text file.











