islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC4002NI Information Systems

Assessment Weightage & Type

20% Individual Coursework

Year and Semester

2018-19 Autumn

Student Name: Sodip Bikram Thapa

College ID: NP01CP4A180059

Assignment Due Date: 18th January, 2019

Assignment Submission Date: 18th January, 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.

Table of Contents

Table of Figures

Table of Tables

1. Proposal

This proposal is written regarding the coursework assigned to us for the Information Systems module. It requires us to create a program in python that can be used to manage the database of a Library based on all the concepts that we have learned in this semester. The coursework was given to us in the 8th week and is required to be submitted before the end of 11th week.

- **Purpose**

The purpose of this coursework is to create a library data management which takes helps one to manage a library's database. The program takes instructions from the user and makes changes to the data stored in the main database and displays the requested information accordingly.

- **Problem Statement**

The problem that we are required to solve as a part of this coursework is to figure out a way to manipulate the data stored in a text file using all the concepts that we have learned so far in this semester which include data types, data structures, modules and functions.

- **Aims and Objectives**

The main aim of this coursework is to create a simple program to manage a library's database. To accomplish this,

This problem can be approach by creating a program that reads the information stored in a text file and stores it list in a 2D-list. The program will take instructions as input from the user and make changes to the 2D-list when someone borrows or returns books. The program will the print out a receipt after each transaction and overwrite the edited 2D-list to the main database.

- **Target Audience**

The target audience for this program will be public/private libraries and educational institutions like schools and colleges having their own libraries. It can also be used by publications and book stores to keep track of their products

- **Hardware Requirements**

This program doesn't require any kind of special hardware to run. Its only requirement is that python 3.7 must be installed on the computer system.

- **Activity Description and Timeline**

Below is a chart representing the overall progress of the coursework:-

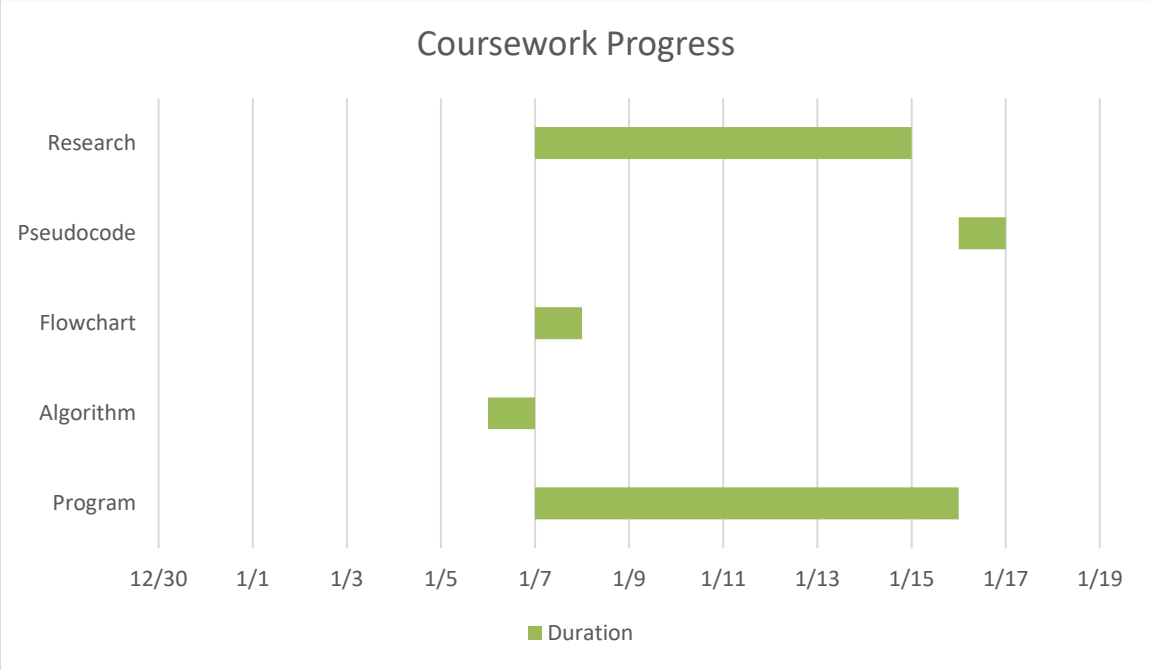


Figure 1: Progress chart

2. Introduction

This report along with the program is also a part of the coursework. It includes all the details related to the program and all the research done in order to create the program. It also contains the timeline of the coursework's progress over the span of four weeks.

Keeping track of all the books and transactions in a library is very important for the library to be able to function properly. This can be done more easily and quickly by using a computer with a data or file management system than using other manual or traditional ways. Using computers means the data will be more secure can also be backed up easily. For this coursework we were assigned to create a basic file management program which can be useful in management the database of a library.

The program is designed to manage the information related to the books in a library. It takes the library's data stored in a text file and changes it after each transaction according to the instructions provided to by user. The program is created in python 3 and requires python 3 to be installed for a computer to be able to run it. The program reads the data from text files and stores the data in a 2D-list. The program takes instructions from a user and makes changes accordingly to the 2D-list and overwrite the data main database with it. The program doesn't have graphical interface but is till easy use. Its main advantage is that it will run on any modern day computer as long as python 3.7 is installed on the system. Since the program is created using simple programming concepts, anyone with basic knowledge of python programming will be able to modify it as per their own need.

3. Discussion and Analysis

The main purpose of this coursework is to make the students familiar with the fundamentals of python programming based on everything taught to us so far. The main objective of this coursework is create a program for managing a library's database based on the concept of file handling and data structures and document the research and findings related to the creation of the program. The tools used for creating the programs are described below:-

Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. (Kumar, 2019)

The program was created in using python programming language since it is simple to use and understand. The python version during the creation of this program was 3.7 and in order for a system to run the program python must be installed beforehand.

MS Word

The report of the program was created and edited with MS Word.

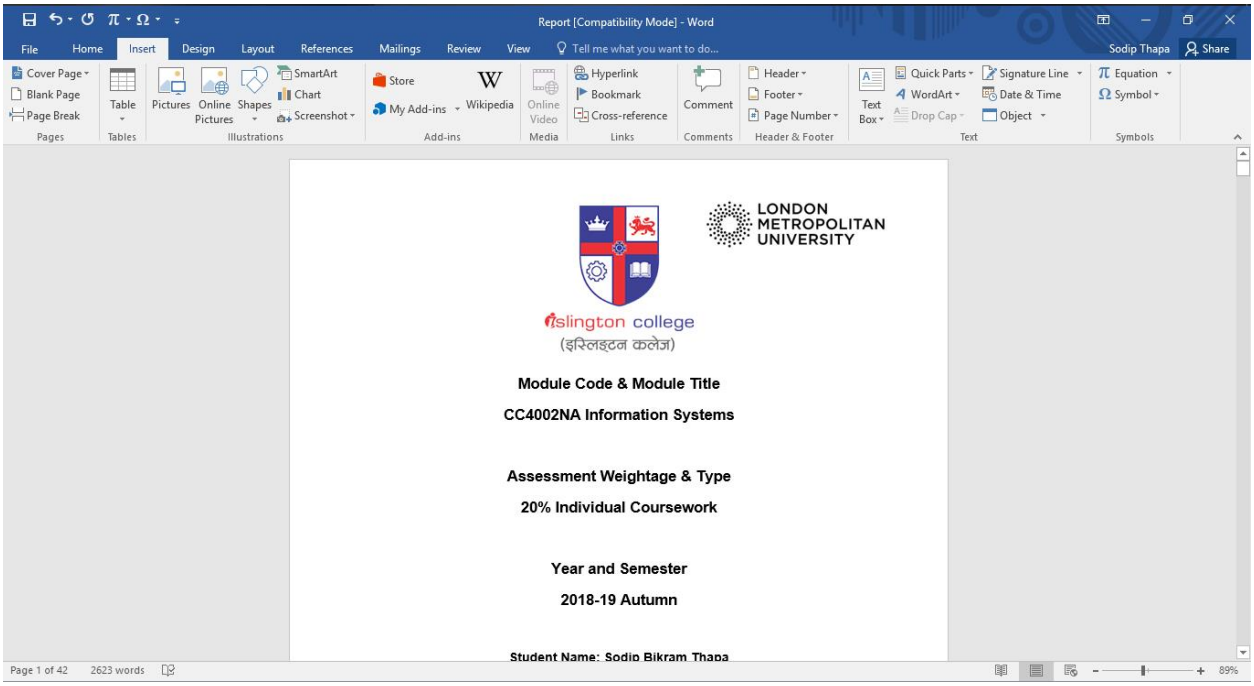


Figure 2: Report in MS Word

MS Excel

The progress of the coursework has been represented using a Gantt chart edited in MS Excel

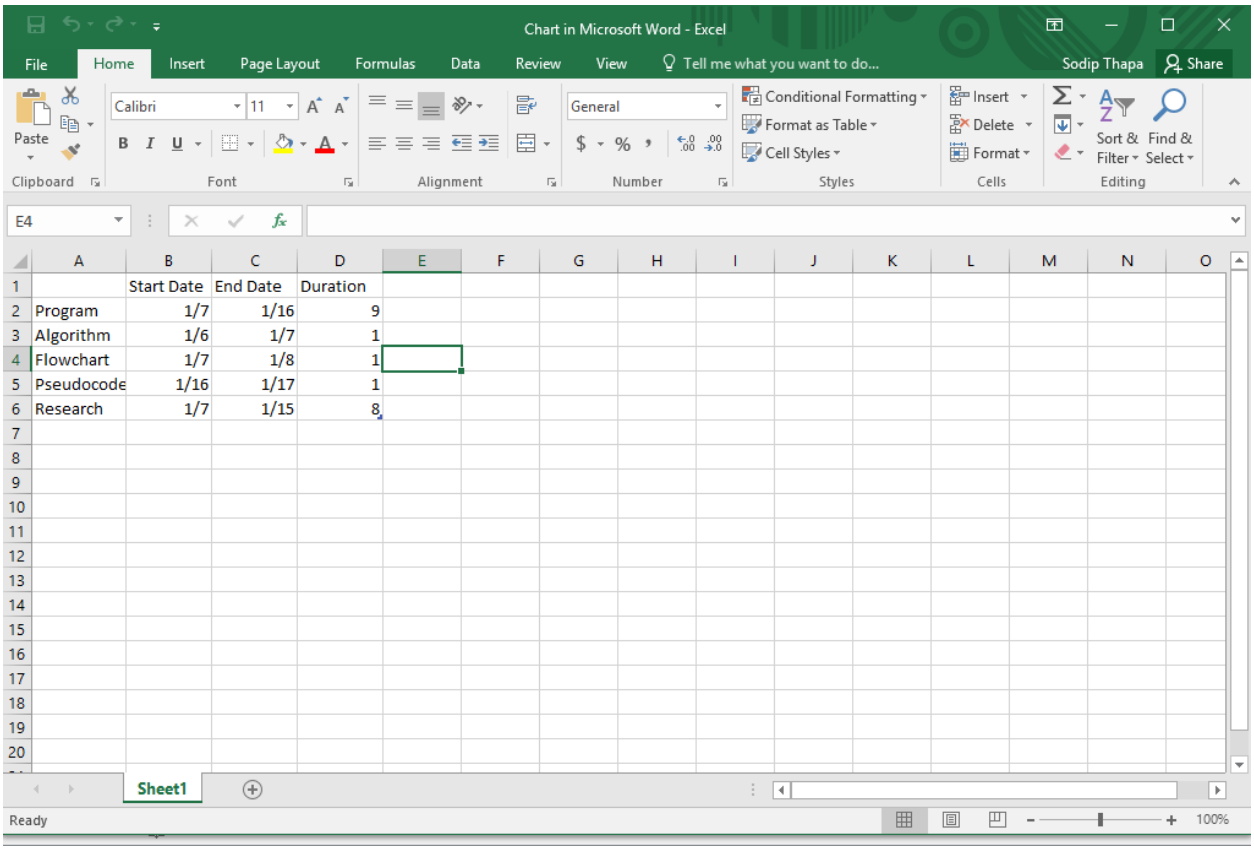


Figure 3: Gantt Chart in MS Excel

4. Algorithm

- **Stepwise Algorithm**

Step 1:- Start

Step 2:- Display Library Menu

Step 3:- Input user instruction

Step 4:-

Step 4.1:-If user instruction= 1 then, read database and display booklist then go to

Step 4.2:- Else if user_input= 2 then

Step 4.2.1:- Read database and display booklist,

Step 4.2.2:- Input student name

Step 4.2.3:- Input student ID no.

Step 4.2.4:- Input the number of books to be borrowed (1 or 2)

Step 4.2.5:- Input serial numbers of those books

Step 4.2.6:- Print the receipt of the transaction containing the student name and ID no. of the student, information of the books as well as the amount to be paid by the student, date of borrowing and deadline for returning the books.

Step 4.2.7:- Then write all the information into the file named after the student's id

Step 4.2.7:- Decrease stock of the books that have been borrowed, from the database.

Step 4.2.7:- Go to Step 2

Step 4.3:- Else if user input=3 then

Step 4.3.1:- Read database and display booklist

Step 4.3.2:- Input student ID no.

Step 4.3.3:- Read file named after the same student ID

Step 4.3.4:- Increase stock of the books that are stored in that file in the main database

Step 4.3.4:- Compare the deadline to return the books stored in the file with current date and display the amount of fine to be paid by the student if the current date exceeds the deadline

Step 4.3.4:- Erase all data stored in the file

Step 4.3.5:- Go back to Step 2

Step 5: Else if user input=4 then

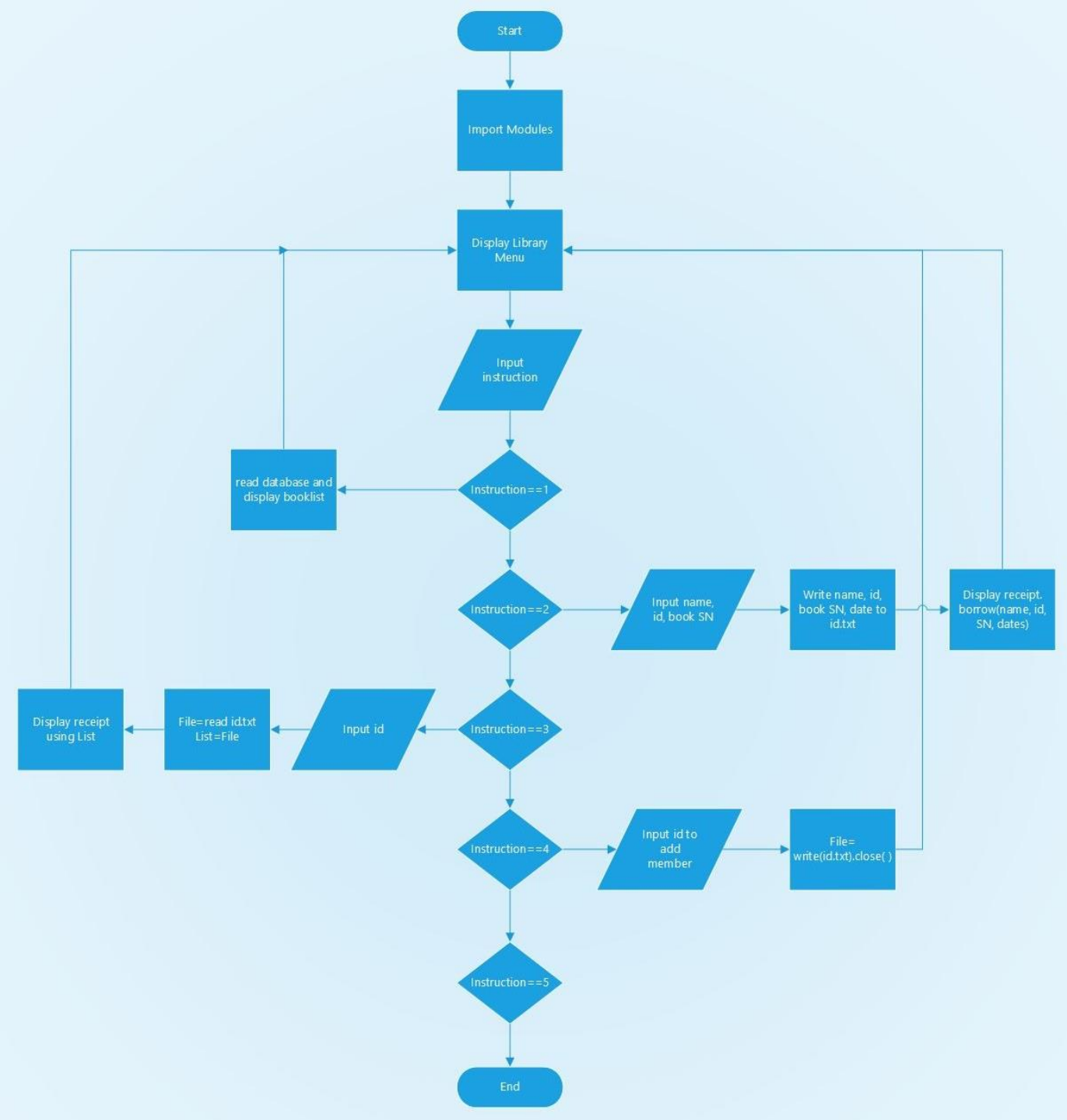
Step 5.1:- Input the student ID no. of the student to be added as a new member

Step 5.2:- Create new file with name same as the student ID no. taken as input Step 5.1

Step 5.3:- Go back to Step 2

Step 6:- Else if user input= 5 then end program

- **Flowchart**


Start
Import Modules
Display Library Menu
Input instruction
Instruction==1
read database and display booklist
Instruction==2
Input name, id, book SN
Write name, id, book SN, date to id.txt
Display receipt. borrow(name, id, SN, dates)
Instruction==3
Input id
File=read id.txt List=File
Display receipt using List
Instruction==4
Input id to add member
File= write(id.txt).close( )
Instruction==5
End

- **Pseudocode**

Main module:

Import **display** module

Import **receipt** module

Import **write** module

Import **read** module

Main function:

Print library menu

Input user_instruction

2D-List= read.booklist (“books.txt”)

If user_instruction==1:

display*.displaybooklist_function (“books.txt”)*

Call **Main function**

Else if user_instruction==2:

display*.displaybooklist_function (“books.txt”)*

Input student_name

Input student_id

Input no.of_books

If no.of_books>2:

Print “A student cannot borrow more than 2 books”

Else if no.of_books==1:

Input S.N. of book

receipt.*one_book_borrowed_function* (student_name, student_id, S.N.)

2D-list2= 2D-list [Index= S.N. of book] [Index containing stock]-1

write. userdata (student_name, student_id, SN, borrowed date)

write.database (2D-list2)

Call **Main function**

Else if no.of_books==2:

Input S.N. of book1

Input S.N. of book2

receipt.*two_books_borrowed_function* (student_name, student_id, S.N. of book1, S.N. of book2)

2D-list2= 2D-list [Index=S.N. of book1] [Index containing stock of book1]-1

2D-list2= 2D-list [Index=S.N. of book2] [Index containing stock of book2]-1

write. userdata (student_name, student_id, SN of book2, SN of book2, date)

write.database(2D-list2)

Call **Main function**

Else if user_instruction==3:

Input student_id

1D-List=open (“student_id.txt”,”r”)

receipt.*return_books_function (*1D-list*)*

File= open (“student_id.txt”,”w”). Close ()

Call **Main Function**

Else if user_instruction==4:

Input student_id

File= *Open (“*student_id.txt*”,”w”).close ( )*

Call **Main Function**

Else if user_instruction==5:

*Print “*Thank You!*”*

End *If*

Call **Main Function**

5. Data Structures

The following are the primitive data types and collection data types used in the program:-

- **Primitive Data Types**

  The following are the primitive data types used in the program:-

  1. Integer (int)

  An integer in python can hold whole numbers that are positive, negative or zero. Unlike in Python 2.7, there are no different types of integers in the current version of Python as of now. (Pandovski, 2018)

  In the program created for this coursework the int data type is used instead of the float data type when 2 whole numbers are compared with each and to perform calculations involving whole numbers. This is because there isn't any part of the program that requires float data type in order to carry out a task and all the input and output are given out as whole number which can be stored using the int data type. The float is only more useful than the int data type when decimal numbers are used which isn't the case for this program.

  2. String (str)

  String represents a sequence of characters (text) inside double or single quotes. In Python, strings are immutable so once it's declared the value can't be changed, instead a new object is created. (Pandovski, 2018)

  In the program, the str data type is used when no calculations are performed on the input given to the program by the user. It is also used to convert the information to be stored in a text file since text files can only store data in the form of strings.

  3. Boolean

Booleans are the primitive data types which can hold only one value out of two constant objects True and False. The built-in function for converting an object to Boolean is bool1 (). (Pandovski, 2018)

In the program Boolean data type is used for variables that are used in while loops. This is because since it's more convenient to use only two kinds values for while loops; one to start the loop and one to end the loop. It is confusing for to keep track of variables used in while loops if all of them are of different kind.

- **Collection Data Types**

  Lists:-

Lists are a type of collection data type in python. A List in python is a mutable collection of elements that may or may not be of the same type. The elements are placed in an ordered sequence, separated by commas (,) inside a list and are accessed through indices assigned to the elements. The indices of a list start from 0 if starting from the first element whereas indices start from -1 if the last element is taken first. An example of list is: -

L = [-2, 4, 6, 7.5, 'Hello']

Values in lists are accessed by using the square brackets to slice along with the index or indices to obtain the value available at that index. For example:-

Print (L [2]) prints the element that is assigned to index number 2 of list L. (Tutorials Point, 2019)

The program converts the data stored in the text into a 2D-list and 2D-list is changed after a book is borrowed and the database is overwritten with the list. The information related to the student who has borrowed that books is then stored in a text file. When the student returns the book data stored in that file is converted into a 1D-list and according to that data more changes are made to the original 2D-list and the database is again overwritten with the 2D-list.

6. Program

```
        LIBRARY MENU
        ------------
NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

Enter 1 to display book list
Enter 2 to borrow
Enter 3 to return
Enter 4 to Add a new member
Enter 5 to exit:
```

Figure 4: Program Menu

The program displays instructions to manage the database to the use as shown in the above screenshot of the program. User must follow those instructions to be able to use the program.

Below is source code of the main module of the program:-

```
import display as d
import dime as dt
import read as r
import receipt as R
import write as ww
def labmenu():
    List=r.readbooks()
    year=dt.year()
    month=dt.month()
    day=dt.day()
    ui=1
    ui2=2
    ui3=3
    print("\t\t\tLIBRARY MENU\n\t\t\t------------\n\t\tNOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY\n\n\t\tEnter 1 to display book list\n\t\tEnter 2 to b
    user_input=int(input("\t\tEnter 4 to Add a new member\n\t\tEnter 5 to exit: "))
    try:
        if user_input==1:
            d.displaybooklist(List)
            labmenu()
        elif user_input==2:
            d.displaybooklist(List)
            name=input("Enter full name of the Student: ")
            sid=input("Enter their Student ID: ")
            if len(sid)>14 or len(sid)<14:
                print("Invalid Student ID!")
                labmenu()
            else:
                try:
                    listy=r.readlist2(sid)
                    if sid==listy[0]:
                        print("Student hasn't returned the books they've borrowed previously.")
                        labmenu()
                    else:
                        book_b=int(input("Enter the S.N. of the book that you want to borrow: "))
                        op=input("Do you want to borrow one more book(y/n)? ")
                        while ui==1:
                            if op=="y":
                                book_b2=int(input("Enter the S.N. of the other book: "))
                                if book_b2==book_b:
                                    print("You can't borrow multiple copies of the same book!")
                                else:
                                    R.borrow2(name,sid,book_b,book_b2,day,month,year)
```

Ln: 25 Col: 44

Figure 5: Program Source code

In order for a student to use a library they must first be added as a member from the program. This creates a text file which stores the information of the student such as the books they have borrowed and the deadline to return the books. To add a student

as a member their unique Student ID assigned to them by the college must be entered. The file that stores their information is named after their Student ID. According to the input given by the user, the program performs tasks as shown in the screenshots below.

```
                LIBRARY MENU
                ------------
        NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

        Enter 1 to display book list
        Enter 2 to borrow
        Enter 3 to return
        Enter 4 to Add a new member
        Enter 5 to exit: 1
| S.N.  | | Book Name             | | Author          | | Quantity  | | Rate(Rs)  |
| 1     | | The Shining           | | Stephen King    | | 15        | | 50        |
| 2     | | Looking For Alaska    | | John Green      | | 23        | | 55        |
| 3     | | Slaughter House Five  | | Kurt Vonnegut   | | 18        | | 35        |
| 4     | | The Outsider          | | Albert Camus    | | 19        | | 45        |
| 5     | | Salem's Lot           | | Stephen King    | | 21        | | 50        |
```

Figure 6: Display Book List

```
                    LIBRARY MENU
                    ------------
               NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

               Enter 1 to display book list
               Enter 2 to borrow
               Enter 3 to return
               Enter 4 to Add a new member
               Enter 5 to exit: 2
| S.N.  | | Book Name             | | Author         | | Quantity  | | Rate(Rs)  |
| 1     | | The Shining           | | Stephen King   | | 15        | | 50        |
| 2     | | Looking For Alaska    | | John Green     | | 23        | | 55        |
| 3     | | Slaughter House Five  | | Kurt Vonnegut  | | 18        | | 35        |
| 4     | | The Outsider          | | Albert Camus   | | 19        | | 45        |
| 5     | | Salem's Lot           | | Stephen King   | | 21        | | 50        |
Enter full name of the Student: Adarsh Subedi
Enter their Student ID: NP01CP4A180054
Enter the S.N. of the book that you want to borrow: 2
Do you want to borrow one more book(y/n)? y
Enter the S.N. of the other book: 5
                    RECEIPT
                    -------
               Name:  Adarsh Subedi
               Student ID:  NP01CP4A180054
               Book Borrowed:  Looking For Alaska
               Author:  John Green
               Price:  Rs. 55
               Book Borrowed:  Salem's Lot
               Author:  Stephen King
               Price:  Rs. 50
               Total Price:  Rs. 55 + Rs. 50 = 105
               Date Borrowed:  17 / 1 / 2019
               Return Date:  27 / 1 / 2019
```

Figure 7: Borrowing Books

```
                    LIBRARY MENU
                    ------------
              NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

              Enter 1 to display book list
              Enter 2 to borrow
              Enter 3 to return
              Enter 4 to Add a new member
              Enter 5 to exit: 2
| S.N.  | | Book Name             | | Author         | | Quantity  | | Rate(Rs)  |
| 1     | | The Shining           | | Stephen King   | | 15        | | 50        |
| 2     | | Looking For Alaska    | | John Green     | | 23        | | 55        |
| 3     | | Slaughter House Five  | | Kurt Vonnegut  | | 18        | | 35        |
| 4     | | The Outsider          | | Albert Camus   | | 19        | | 45        |
| 5     | | Salem's Lot           | | Stephen King   | | 21        | | 50        |
Enter full name of the Student: Sodip Thapa
Enter their Student ID: NP01CP4A180059
Student hasn't returned the books they've borrowed previously.
```

Figure 8: Not Returned

```
                    LIBRARY MENU
                    ------------
              NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

              Enter 1 to display book list
              Enter 2 to borrow
              Enter 3 to return
              Enter 4 to Add a new member
              Enter 5 to exit: 3
| S.N.  | | Book Name             | | Author         | | Quantity  | | Rate(Rs)  |
| 1     | | The Shining           | | Stephen King   | | 15        | | 50        |
| 2     | | Looking For Alaska    | | John Green     | | 22        | | 55        |
| 3     | | Slaughter House Five  | | Kurt Vonnegut  | | 18        | | 35        |
| 4     | | The Outsider          | | Albert Camus   | | 19        | | 45        |
| 5     | | Salem's Lot           | | Stephen King   | | 20        | | 50        |
Enter your Student ID: NP01CP4A180054
                    RECEIPT
                    -------
              Student Name:  Adarsh Subedi
              Student ID:  NP01CP4A180054
              No. of books borrowed:  2
              Books Returned:  Looking For Alaska , Salem's Lot
              Books Borrwed On:  17 / 1 / 2019
              Last Date to Return Borrowed Books:  27 / 1 / 2019
              Books Returned On:  17 / 1 / 2019
              Late Fine: None
```

Figure 9: Returning Borrowed Books

```
                    LIBRARY MENU
                    ------------
              NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

              Enter 1 to display book list
              Enter 2 to borrow
              Enter 3 to return
              Enter 4 to Add a new member
              Enter 5 to exit: 4
Enter the Student ID of the student you want to add as a member: NP01CP4A180077
Member successfully added!
```

Figure 10: Adding a New Library Member

```
                    LIBRARY MENU
                    ------------
              NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

              Enter 1 to display book list
              Enter 2 to borrow
              Enter 3 to return
              Enter 4 to Add a new member
              Enter 5 to exit: 5
Thank You!
>>>
```

Figure 11: Exiting the Program

7. Testing

Black box texting of the program was done in order to check if the program functioned properly or not.

- **Test 1**

Table 1: Test 1

Test No.	1
Action	Input 1 to display book list
Expected Output	Booklist will be displayed
Actual Output	Booklist is displayed
Test Result	Pass

```
                    LIBRARY MENU
                    ------------
               NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

               Enter 1 to display book list
               Enter 2 to borrow
               Enter 3 to return
               Enter 4 to Add a new member
               Enter 5 to exit: 1
| S.N.  | | Book Name             | | Author          | | Quantity  | | Rate(Rs)  |
| 1     | | The Shining           | | Stephen King    | | 15        | | 50        |
| 2     | | Looking For Alaska    | | John Green      | | 23        | | 55        |
| 3     | | Slaughter House Five  | | Kurt Vonnegut   | | 18        | | 35        |
| 4     | | The Outsider          | | Albert Camus    | | 19        | | 45        |
| 5     | | Salem's Lot           | | Stephen King    | | 21        | | 50        |
```

Figure 12: Test 1

- **Test 2**

Table 2: Test 2

Test No.	2
Action	Input any string
Expected Output	“Invalid Input!” will be printed
Actual Output	“Invalid Input!” is printed
Test Result	Pass

```
                    LIBRARY MENU
                    ------------
               NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

               Enter 1 to display book list
               Enter 2 to borrow
               Enter 3 to return
               Enter 4 to Add a new member
               Enter 5 to exit: a
Invalid Input!
```

Figure 13: Test 2

- **Test 3**

Table 3: Test 3

Test No.	3
Action	Input 4 to add new member
Action	Input any string that isn't a student id
Expected Output	"Invalid student id!" will be printed
Actual Output	"Invalid student id!" is printed
Test Result	Pass

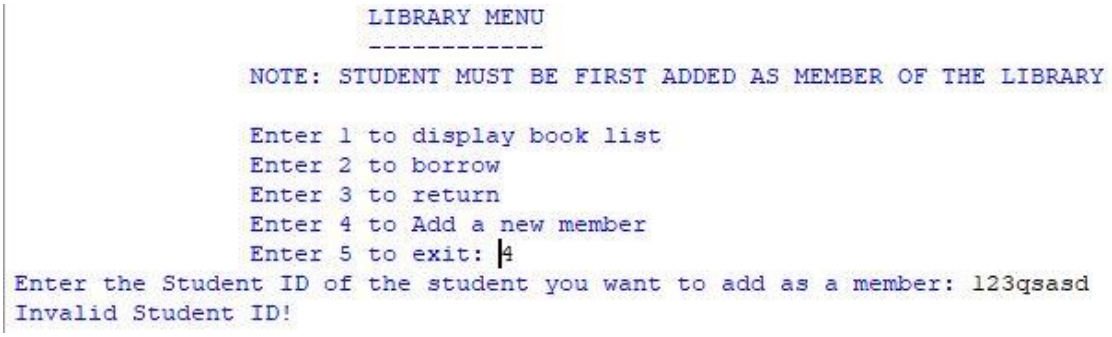


Figure 14: Test 3

- **Test 4**

Table 4: Test 4

Test No.	4
Action	Input 4 to add new member
Action	Input any student id to add new member
Expected Output	New member will be added and “Member successfully added!” will be printed
Actual Output	New member is added and “Member successfully added!” will be printed
Test Result	Pass

```
                    LIBRARY MENU
                    ------------
              NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

              Enter 1 to display book list
              Enter 2 to borrow
              Enter 3 to return
              Enter 4 to Add a new member
              Enter 5 to exit: 4
Enter the Student ID of the student you want to add as a member: NP01CP4A180051
Member successfully added!
```

Figure 15: Test 4

- **Test 5**

Table 5: Test 5

Test No.	5
Action	Input 2 to borrow books
Action	Input name, student id and S.N. of books
Expected Output	Receipt of the transaction will be printed
Actual Output	Receipt of the transaction is be printed
Test Result	Pass

```
                    LIBRARY MENU
                    ------------
               NOTE: STUDENT MUST BE FIRST ADDED AS MEMBER OF THE LIBRARY

               Enter 1 to display book list
               Enter 2 to borrow
               Enter 3 to return
               Enter 4 to Add a new member
               Enter 5 to exit: 2
| S.N.  | | Book Name             | | Author          | | Quantity  | | Rate(Rs)  |
| 1     | | The Shining           | | Stephen King    | | 15        | | 50        |
| 2     | | Looking For Alaska    | | John Green      | | 24        | | 55        |
| 3     | | Slaughter House Five  | | Kurt Vonnegut   | | 18        | | 35        |
| 4     | | The Outsider          | | Albert Camus    | | 19        | | 45        |
| 5     | | Salem's Lot           | | Stephen King    | | 22        | | 50        |
Enter full name of the Student: Meru Sangroula
Enter their Student ID: NP01CP4A180051
Enter the S.N. of the book that you want to borrow: 2
Do you want to borrow one more book(y/n)? n
                    RECEIPT
                    -------
               Student Name:  Meru Sangroula
               Student ID:  NP01CP4A180051
               Book Borrowed:  Looking For Alaska
               Author:  John Green
               Price:  Rs. 55
               Date Borrowed:  18 / 1 / 2019
               Return Date:  28 / 1 / 2019
```

Figure 16: Test 5

8. Research

- **Websites**

Fundamentals of Python Programming

https://pythonprogramming.net/introduction-to-python-programming/

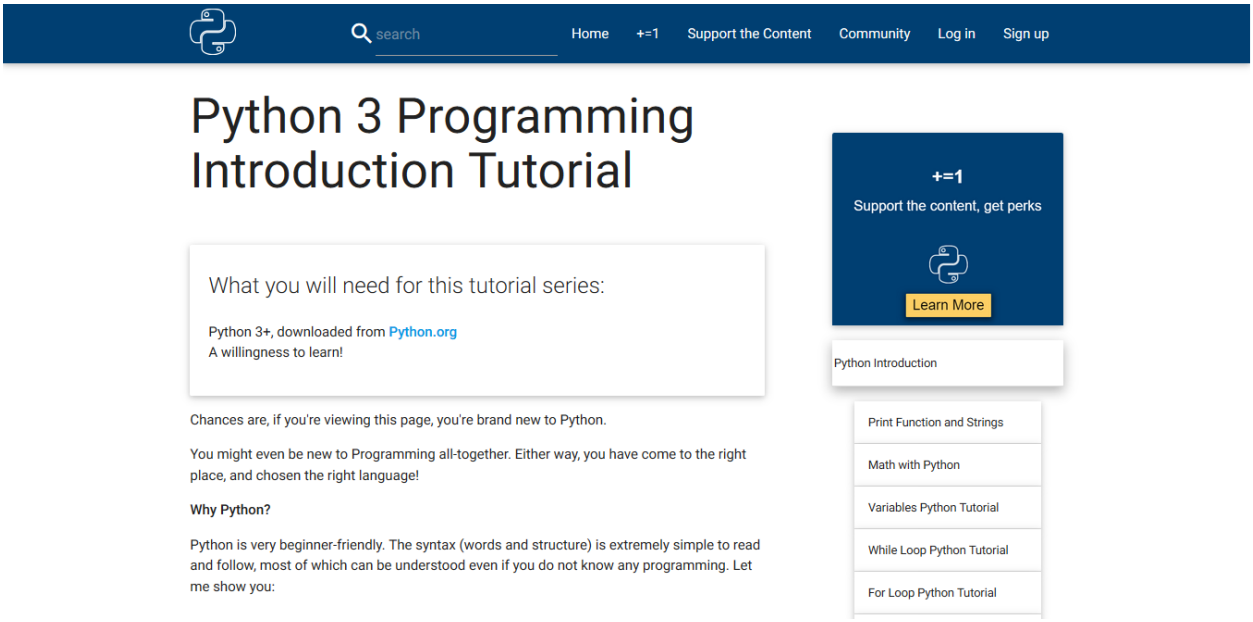


Figure 17: Website1

Basic/Primitive Python Data Types

https://realpython.com/python-data-types/

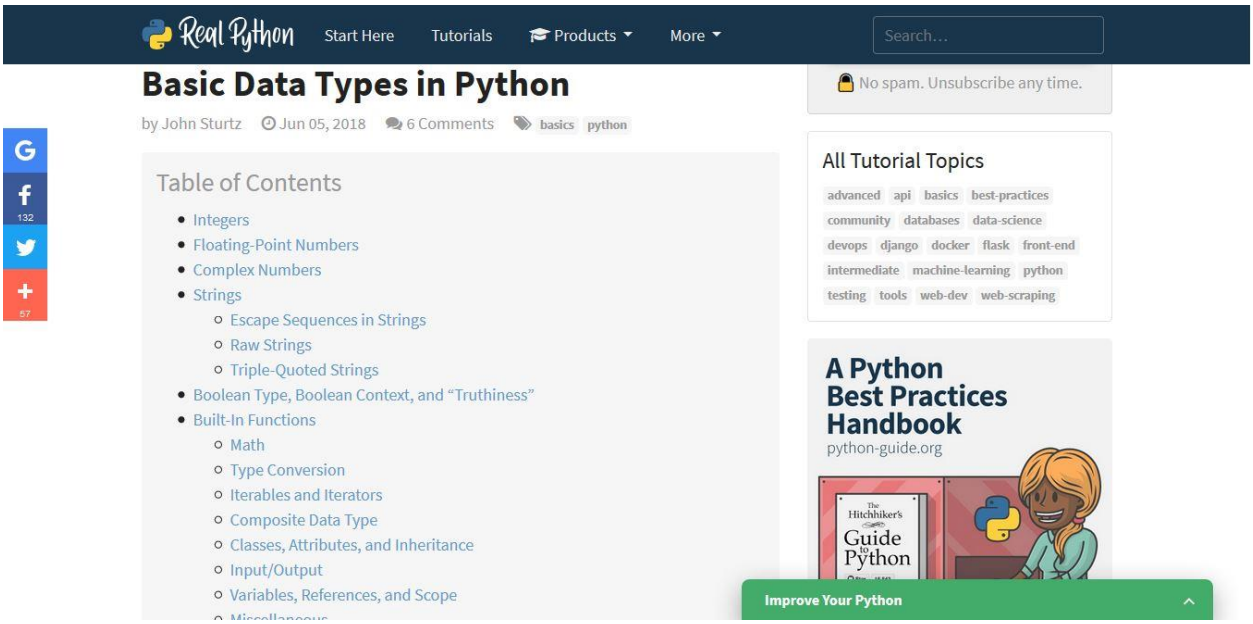


Figure 18: Website1

Lists in Python

https://www.programiz.com/python-programming/list

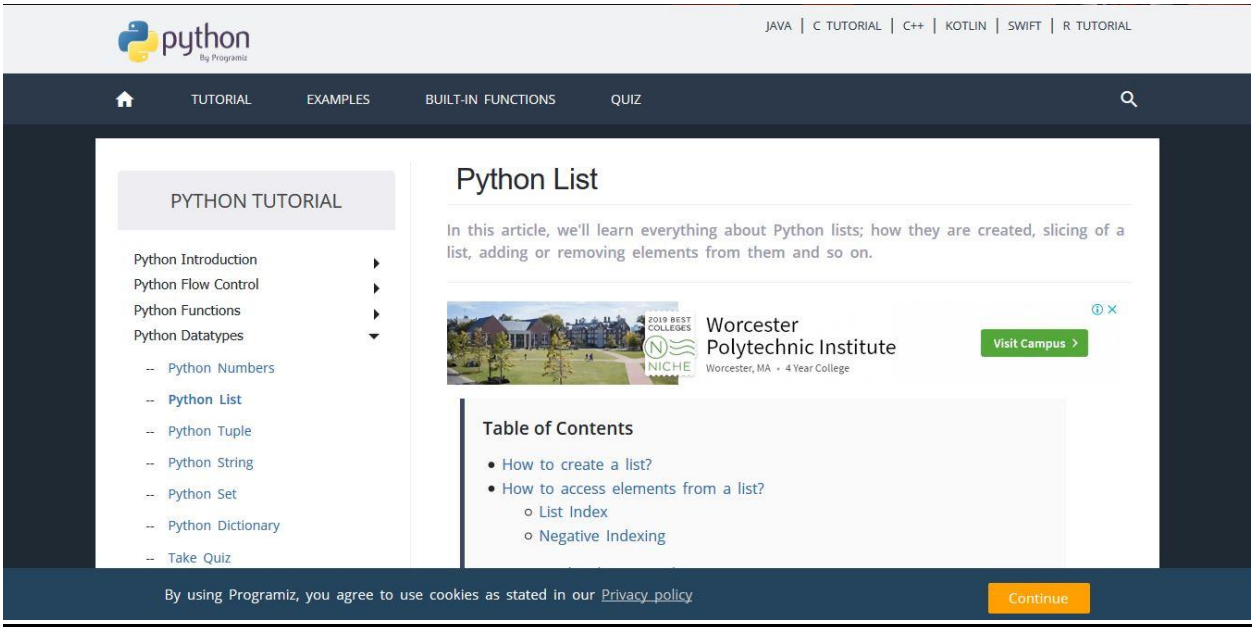


Figure 19: Website2

2D-Lists in Python

https://snakify.org/en/lessons/two_dimensional_lists_arrays/

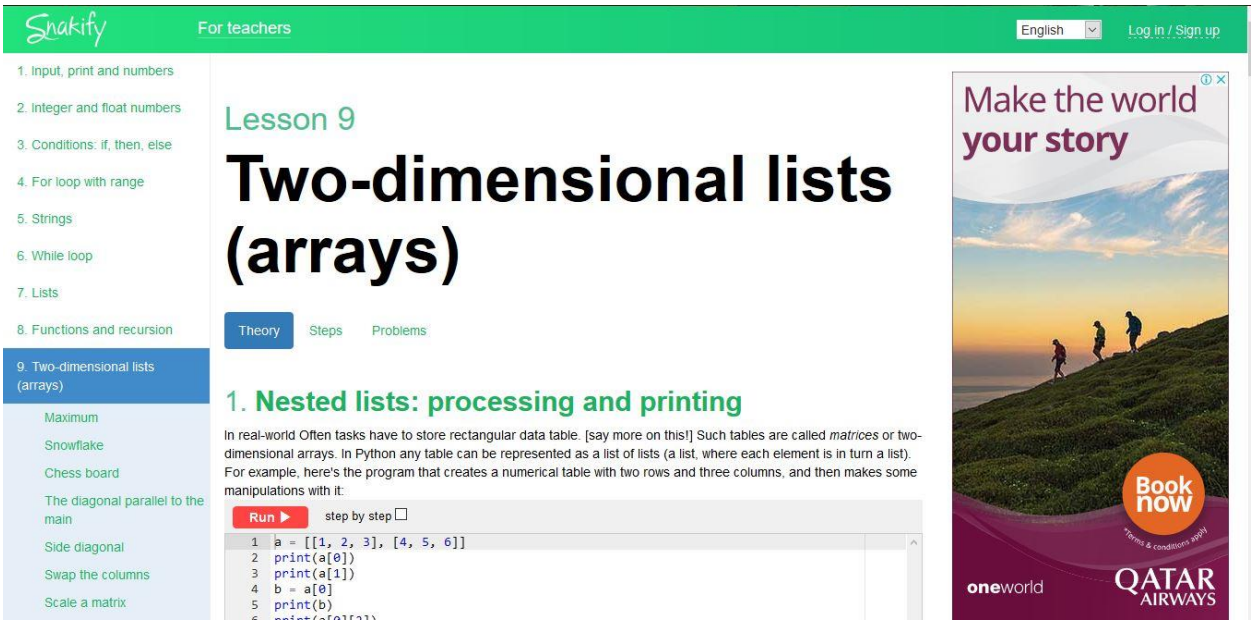


Figure 20: Website 3

File Handling in Python

https://www.w3schools.com/python/python_file_handling.asp

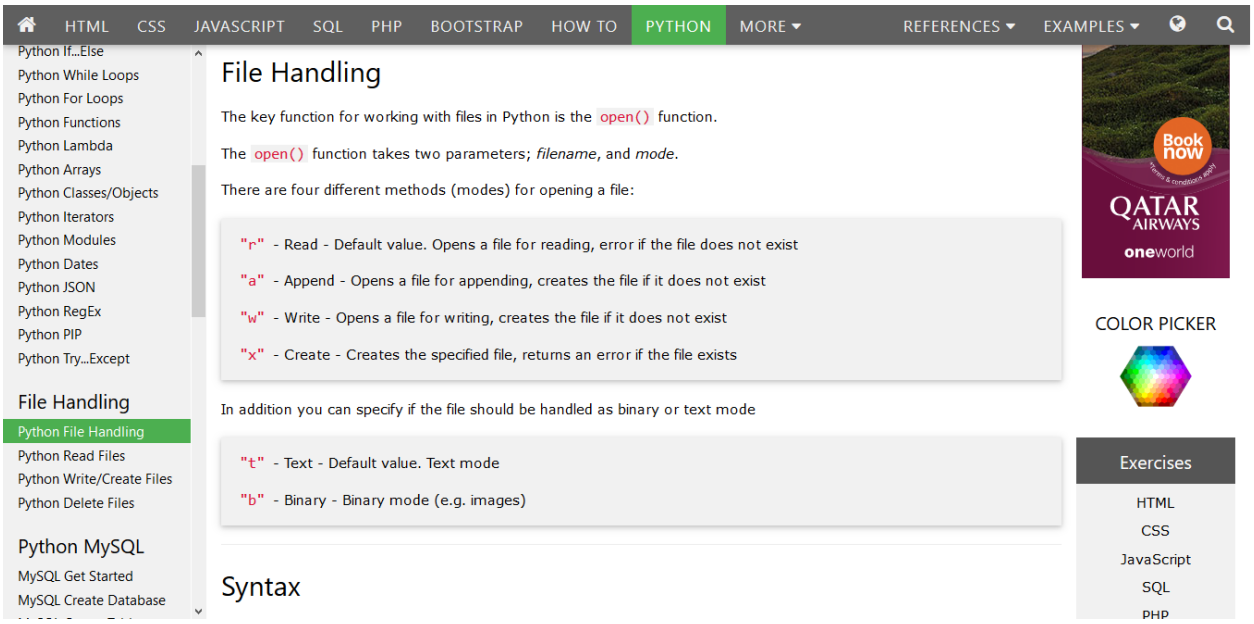


Figure 21: Website 5

- **Books**

Programming in Python 3

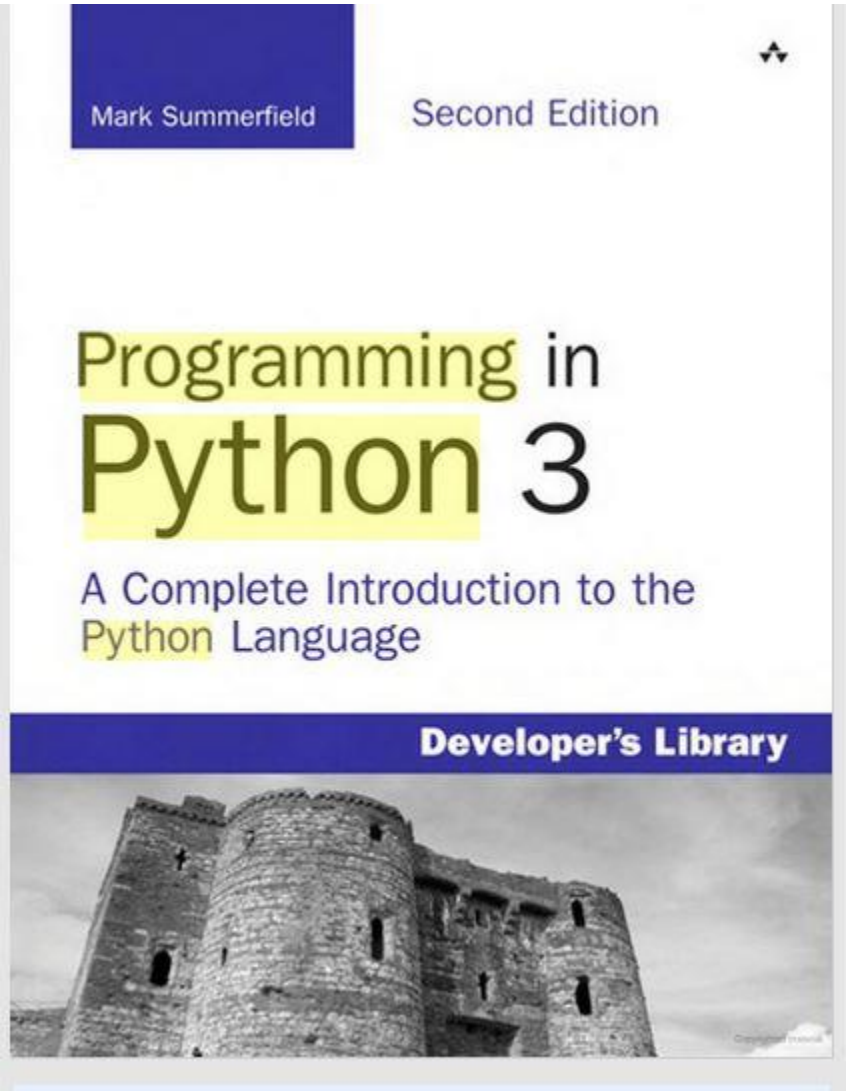


Figure 22: Book 1

Python Programming: An Introduction to Computer Science

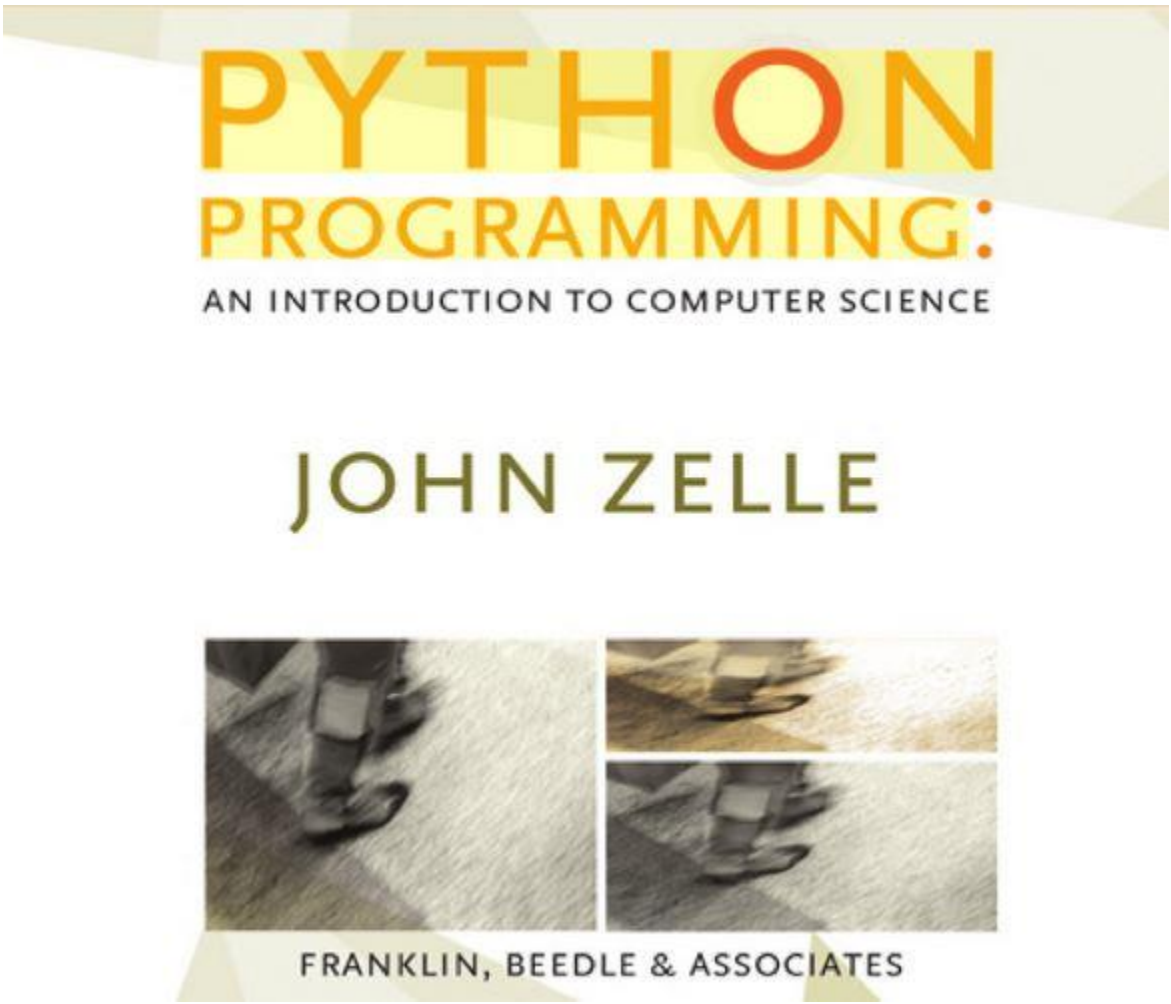


Figure 23: Book 1

Python Data Structures and Algorithms

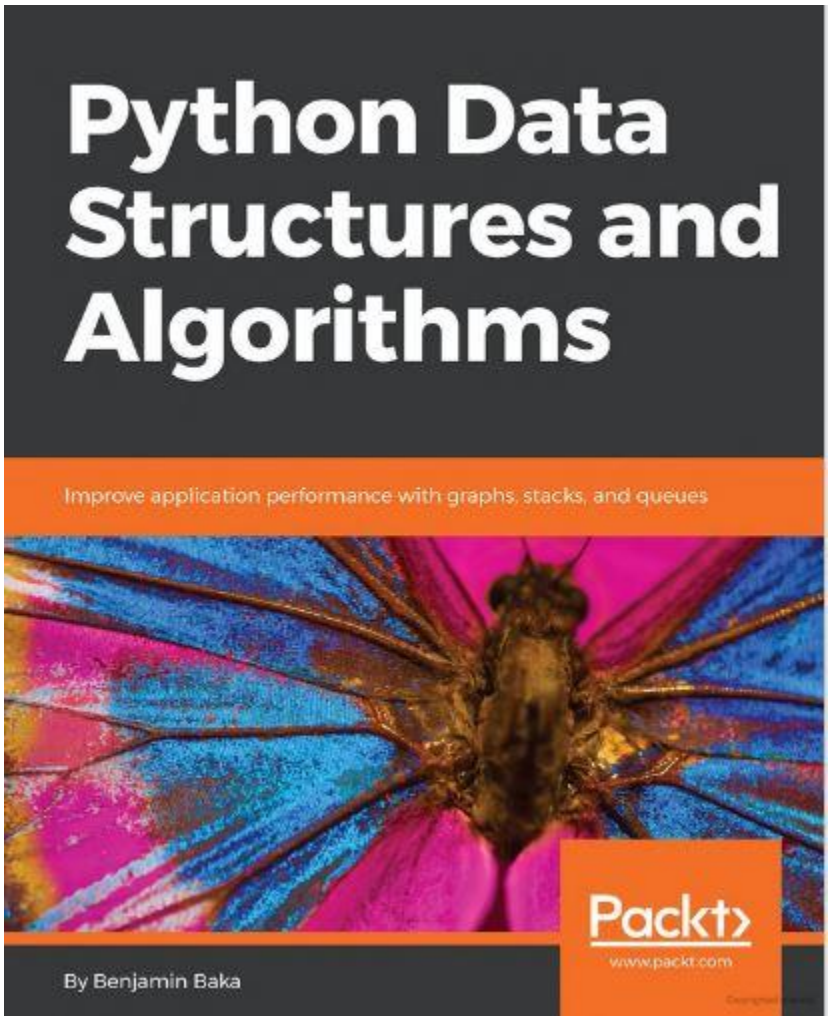


Figure 24: Book 2

Python Standard Library

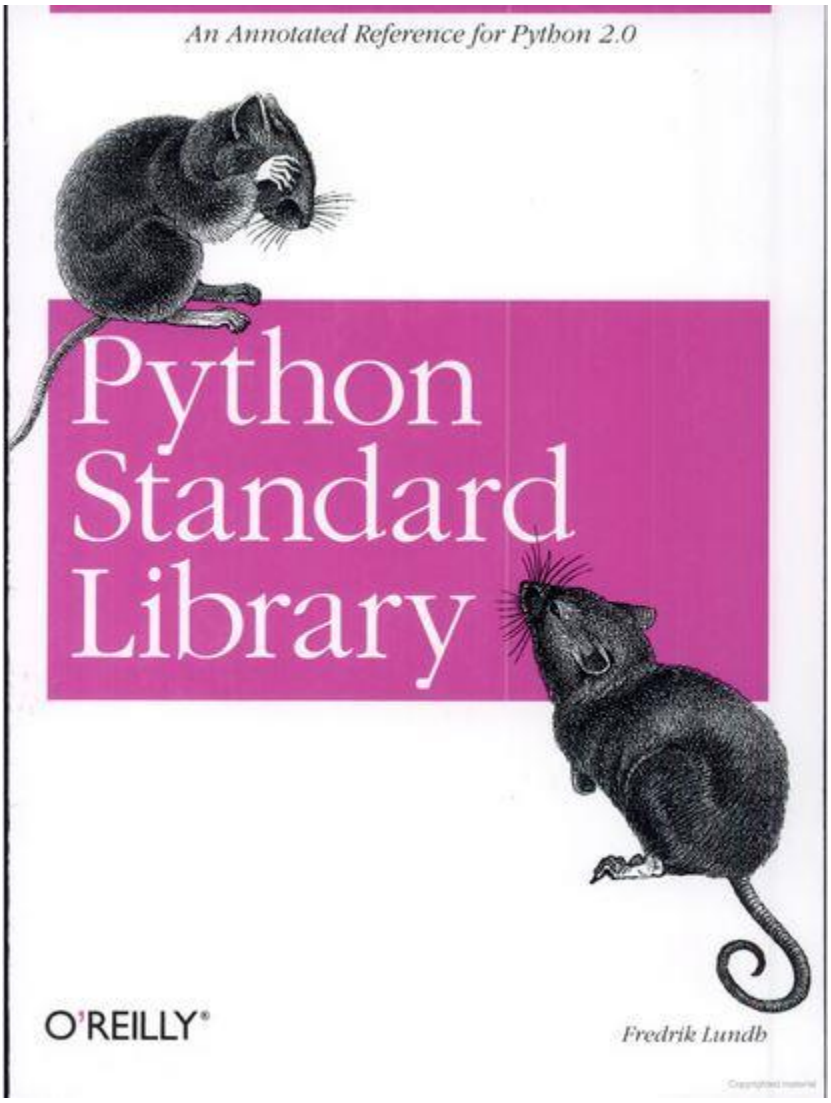


Figure 25: Book 3

Advanced Topics in Exception Handling Techniques

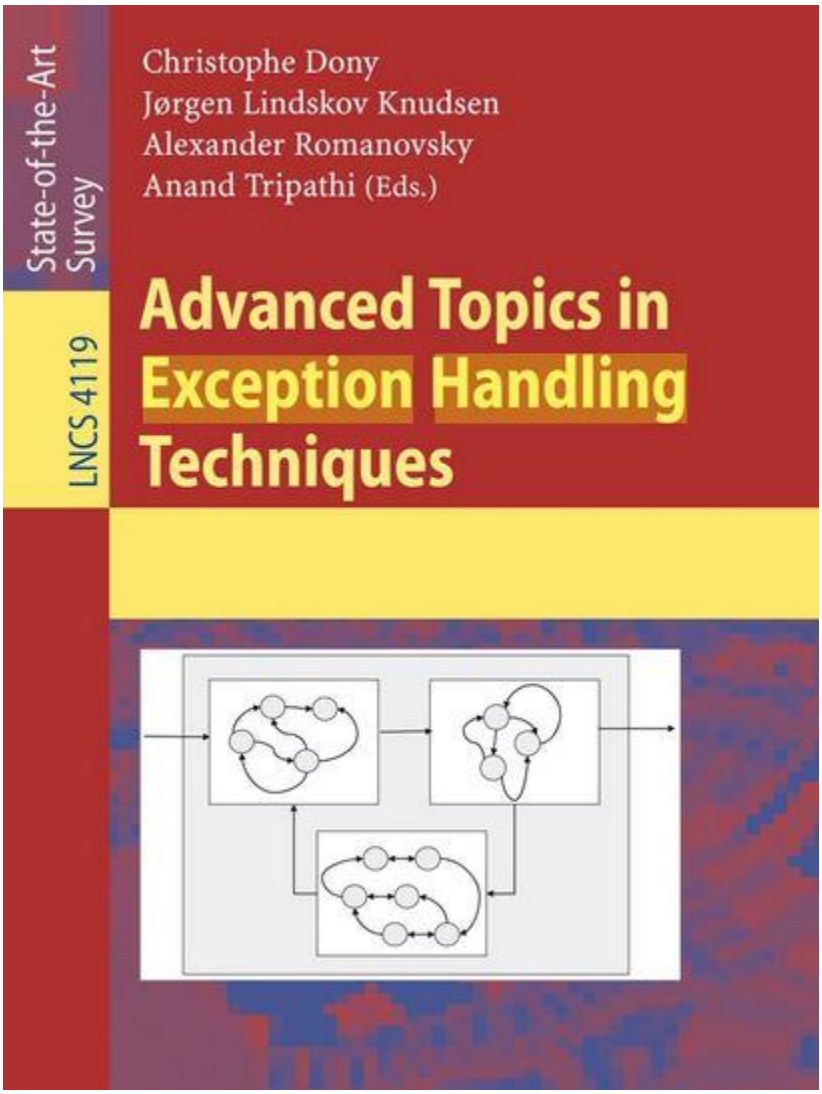
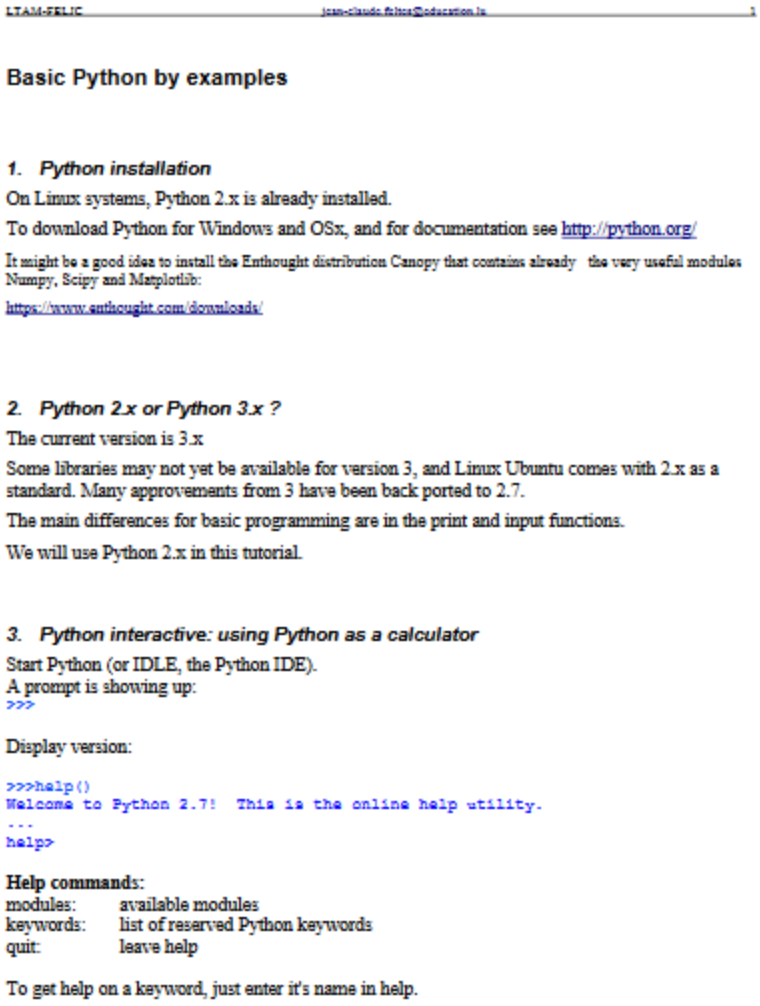


Figure 26: Book 5

- **Journals**

Basic Python y examples by Jean Claude Feltes



LTAM-FELIC jean-claude.feltes@education.lu 1

Basic Python by examples

1. Python installation

On Linux systems, Python 2.x is already installed.

To download Python for Windows and OSx, and for documentation see http://python.org/

It might be a good idea to install the Enthought distribution Canopy that contains already the very useful modules Numpy, Scipy and Matplotlib:

https://www.enthought.com/downloads/

2. Python 2.x or Python 3.x ?

The current version is 3.x

Some libraries may not yet be available for version 3, and Linux Ubuntu comes with 2.x as a standard. Many approvements from 3 have been back ported to 2.7.

The main differences for basic programming are in the print and input functions.

We will use Python 2.x in this tutorial.

3. Python interactive: using Python as a calculator

Start Python (or IDLE, the Python IDE).
A prompt is showing up:

```
>>>
```

Display version:

```
>>>help()
Welcome to Python 2.7!  This is the online help utility.
...
help>
```

Help commands:

modules:	available modules
keywords:	list of reserved Python keywords
quit:	leave help

To get help on a keyword, just enter it's name in help.

Figure 27: Journal 1

Primitive Data Types by Margaret Reid-Miller

Common Primitive Types

Type	Description	Example of Literals
`int`	integers (whole numbers)	`42, 60634, -8, 0`
`double`	real numbers	`0.039, -10.2, 4.2E+72`
`char`	single characters	`'a', 'B', '&', '6'`
`boolean`	logical values	`true, false`

Figure 28: Journal 2

File Handling in Python by Imtiaz Abedin

Python Read File, Write File, Open File, Delete File, Copy File

IMTIAZ ABEDIN — LEAVE A COMMENT

In this tutorial we are going to learn about Python File Operations such as python read file, python write file, open file, delete file and copy file. Our previous tutorial was on Python Dictionary. You can find that in this link.

Table of Contents [hide]

Python File

In the previous tutorial we used console to take input. Now, we will be taking input using file. That means, we will read from and write into files. To do so, we need to maintain some steps. Those are

Figure 29: Journal 3

Python to learn Programming by A. Bogdanchikov, M. Zhaparov, R. Suiliyev

SciTech 2013 IOP Publishing
Journal of Physics: Conference Series 423 (2013) 012027 doi:10.1088/1742-6596/423/1/012027

Python to learn programming

A Bogdanchikov[1], M Zhaparov[1] and R Suliyev[1]

[1] Suleyman Demirel University, 1/1 Abylaikhan St., Kaskelen, Almaty, Kazakhstan

E-mail: meirambek.zhaparov@gmail.com

Abstract. Today we have a lot of programming languages that can realize our needs, but the most important question is how to teach programming to beginner students. In this paper we suggest using Python for this purpose, because it is a programming language that has neatly organized syntax and powerful tools to solve any task. Moreover it is very close to simple math thinking. Python is chosen as a primary programming language for freshmen in most of leading universities. Writing code in python is easy. In this paper we give some examples of program codes written in Java, C++ and Python language, and we make a comparison between them. Firstly, this paper proposes advantages of Python language in relation to C++ and JAVA. Then it shows the results of a comparison of short program codes written in three different languages, followed by a discussion on how students understand programming. Finally experimental results of students' success in programming courses are shown.

1. Introduction

Python programming language is most suitable as a first language to learn for newbie programmers, because it has powerful tools that reflect the way people think and the way they implement the code. [1] Moreover it minimizes extra keywords that are necessary to write syntactically correct program. This approach seems to be more productive than teaching C++ or Java languages, which have a lot of terms and elements that are related to the specifics of a language rather than to an algorithm realization. In addition, instructors in over a dozen universities, such as MIT, UC Berkeley, UC Davis, Sonoma State University, the University of Washington, the University of Waterloo, Luther College, and Swarthmore College, have used it for teaching the introductory programming course to the students of computer science department. [2]

Today, for a computer scientist, it is important to learn at least one programming language, because all innovations and technologies are based on thorough understanding of computers, operating system, software API or some hardware peripherals. All of which are created by programmers that use specific way of thinking. And to gain that way of thinking, one has to get used to one of the programming languages and become qualified in software development. [6]

For any person, who starts to learn programming, it is important to concentrate on programming concepts rather than on language specifics, because they may be different for various programming languages. But Python provides the highest level of programming. So the student does not have to think about memory management, which is unavoidable in C++, or class hierarchy, that is unavoidable in Java, or variable types and declarations, that exist in almost each programming language.

Figure 30: Journal 4

Program 2D Array with Python by Aymeric Rousseau

2D Array Construction

Preamble

Within this section, several references will be made to **x-axis** and **y-axis**. To avoid repetitions, we will represent them by the letters **x** and **y** in the rest of this article. Also, **Python** does not allow (not without the installation of additional modules) to create two-dimensional arrays, also called matrices: it will therefore necessarily use nested lists to achieve this. We will see that this is not a problem in practice, and that it is quite possible to manipulate data and / or coordinates in such structures.

First practical example

Turning now to practice, with several examples of programs that build two-dimensional arrays.

Below a first portion of simple code:

```
# -*- coding: utf-8 -*-
# Declarations
grid = [] # empty grid
row = 10 # number of rows
col = 10 # number of columns
value = 0 # value iterated
# Grid build
for line in range(row):
grid.append([])
for column in range(col): # 10x loop
```

Figure 31: Journal 5

9. Conclusion

All the objectives of this coursework have been met and the completion of program was done with many trials and errors. All the obvious and known errors have been handled properly. A lot of research has been done for this coursework, information from various books, journals as well as websites was used for creating the program. All the research and timeline of the progress of this coursework has been documented as per the instructions given to us by the teachers.

After completing this coursework my confidence towards python programming increased drastically. I became used to all the fundamental concepts and syntaxes of python programming. I also learned the importance of planning out everything before starting any project as well as the importance of documentation of your work. Before starting this coursework I wasn't used to writing comments in the source code but while working on the program of this coursework I realized the importance of writing comments for future review of your code and got used to writing comments in the source code.

I also learned about the importance of research while working on any kind of project as the project will be better if more time is put into research. I also realized how programming can be implemented in various fields and how computers can make various difficult task much easier. I also learned the concept of file handling in python programming language and also about the fundamentals of database management. I also learned how to debug a program and the concept of exception handling.

Working on this coursework also acted as a brief experience of what it takes to create all the programs and software that we use in our daily lives. In conclusion, this coursework was an opportunity for me become more efficient in programming and learn what it takes to become an efficient programmer.

10. References

Kumar, A. (2019) *Developer Insider* [Online]. Available from: https://developerinsider.co/the-introduction-to-python-programming-language/ [Accessed 18 January 2019].

Pandovski, Z. (2018) *Able* [Online]. Available from: https://able.bio/ZoranPandovski/primitive-data-types-in-python-3--57tqcfp [Accessed 10 January 2019].

Tutorials Point. (2019) *Tutorials Point* [Online]. Available from: https://www.tutorialspoint.com/python/python_lists.htm [Accessed 15 January 2019].