# JCU_Logo_RGBCP1404/CP5632 2016 SP23/53 Assignment 1 – Reading List 1.0

Task:

You are to plan and then code a console-based program in Python 3, as described in the following information and sample output. This assignment will help you build skills using selection, repetition, file input/output, exceptions, lists, functions and string formatting. Do not to define any of your own classes. Assignment 2 will build on this program with more advanced code constructs including classes and a Graphical User Interface (GUI). Work incrementally on this task: complete small parts of it at a time rather than trying to get it all working at once. Some requirements have in-text references like [0] that refer to the resources list near the bottom. Everything you need to know to complete this assignment can be found in the subject materials.

Program Overview:

This program is a simple reading list that allows a user to track books they wish to read and books they have completed reading. The program maintains a list of books in a file, and each book has:

* title, author, number of pages, whether it is required or completed (r or c)

Users can choose to see the list of required books or completed books, including a total of the number of pages of the book list. The lists will be sorted by author then by number of pages (increasing). [1]

Users can add new books and mark books as completed.

They cannot change books from completed to required.

Program Functionality Details:

Ensure that your program has the following features, as demonstrated in the sample output below.

Your program should:

* display a welcome message with your name in it
* display a menu for the user to choose from [2]
* return to the menu after each action and loop until the user chooses to quit
* error-check user inputs as demonstrated in the sample output [3]
* load a CSV (Comma Separated Values) file of books (just once at the very start); a sample CSV file is provided for you and you must use this format [4]
* *when the user chooses* ***list required****:* display a neatly formatted (lined up) list of all the required books with their details and a total of the number of pages of these books [3]
* *when the user chooses* ***list completed****:* display a similarly formatted list of completed books
* *when the user chooses* ***add****:* prompt for the book’s title, author and number of pages,   
  error-checking each of these, then add the book to the list in memory (not to the file)
* *when the user chooses* ***mark completed****:* display the list of required books (same as for list), then allow the user to choose one book (error-checked), then change that book to completed
  + if no books are required, then a "No books" message should be displayed and the user returned to the menu (this is the same if there are no completed books)
* *when the user chooses* ***quit****:* save the books to the CSV file, overwriting the file contents

Planning:

Write up the algorithms in pseudocode for the two functions: **load books** and **complete a book**.

Do this in a docstring (comment) directly above each of these functions.

At the very top of your code file, add a docstring containing your **name, date, brief program details** and **a link to your project on GitHub**.  
Follow the guide to good pseudocode and examples presented in the subject to ensure this is done to a high standard. [5] You may show this part of the assignment to your tutor during practical time to get comments or suggestions.

You may do pseudocode for more than these two functions, but it is not required.

Coding Requirements and Suggestions:

* Make use of named constants where appropriate.
* Use functions appropriately for each significant part of the program: this is the divide-and-conquer problem-solving approach. Remember that functions should “do one thing”.   
  Look for situations where functions can be used to reduce code duplication (e.g. when you display the required lists and when you mark a book as completed – this is very similar).
* For efficiency, you should ***only load the books file once***. Store the results appropriately in a list of lists and pass that to any functions that need access to it. Note: this variable should not be global. The only global variables you may have are CONSTANTS.
* Note that the menu choice should handle uppercase and lowercase letters.
* Use exception handling where appropriate to deal with input errors (including entering numbers and selecting books). You should be able to use generic, customisable functions to perform input with error checking (e.g. getting the title and author can reuse the same function).
* Check the rubric carefully to see any other aspects of the coding that you will be assessed on.

Output Requirements:

Sample output from the program is provided below. Ensure that your program matches the sample output including spaces, spelling, and especially the formatting of the book lists. Think of this as helpful guidance as well as training you to pay attention to detail. The sample output is intended to show the full range of situations including user input error handling.

Submission:

The assignment 'starter' will be provided to you via GitHub Classroom (see below). This will include the books CSV file and a Python file called CP1404assignment1.py

Submit **this** Python 3 (.py) code file containing your comments and pseudocode by uploading it on LearnJCU under Assessment (click on the title of the assignment).

**Git/GitHub:**You must use Git version control and keep your project updated online in GitHub.

If you have not yet created a GitHub account, do this now as explained at <https://github.com/CP1404/Starter/wiki/Software-Setup#github>   
You must be easily identifiable from your GitHub username. Make sure you register for the student discount so you get free private repositories: <https://education.github.com/discount_requests/new>

You are assessed on your use of version control including commits and commit messages, so commit regularly (each logical chunk or milestone) and use meaningful commit messages in the imperative voice. Your commits should show steady work completed over reasonable time, not all in a short period.

Due:

Submit your assignment bythe date and time specified on LearnJCU.

Submissions received after this date will incur late penalties as described in the subject outline.

Integrity:

The work you submit for this assignment must be your own. You are allowed to discuss the assignment with other students and get assistance from your peers, but you may not do any part of anyone else’s work for them and you may not get anyone else to do any part of your work. Note that this means you should never give a copy of your work to anyone or accept a copy of anyone else’s work. Submissions that are detected to be too similar to that of another student will be dealt with according to the College procedures for handling plagiarism and may result in serious penalties.

If you require assistance with the assignment, please ask **general** questions on the discussion forum, or get **specific** assistance with your own work by talking with your lecturer or tutor.

Sample Output:

The following sample run was made using a CSV file that contained:

The Practice of Computing Using Python,Punch and Enbody,792,r  
The 360 Degree Leader,John Maxwell,369,r  
In Search of Lost Time,Marcel Proust,365,c  
Developing the Leader Within You,John Maxwell,225,r

**Bold green** text below shows user input for this sample.

Reading List 1.0 - by Lindsay Ward

4 books loaded from books.csv

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **r**

Required books:

0. Developing the Leader Within You by John Maxwell 225 pages

1. The 360 Degree Leader by John Maxwell 369 pages

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

Total pages for 3 books: 1386

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **c**

Completed books:

2. In Search of Lost Time by Marcel Proust 365 pages

Total pages for 1 books: 365

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **m**

Required books:

0. Developing the Leader Within You by John Maxwell 225 pages

1. The 360 Degree Leader by John Maxwell 369 pages

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

Total pages for 3 books: 1386

Enter the number of a book to mark as completed

>>> **one**

Invalid input; enter a valid number

>>> **2**

That book is already completed

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **m**

Required books:

0. Developing the Leader Within You by John Maxwell 225 pages

1. The 360 Degree Leader by John Maxwell 369 pages

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

Total pages for 3 books: 1386

Enter the number of a book to mark as completed

>>> **1**

The 360 Degree Leader by John Maxwell marked as completed

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **u**

Invalid menu choice

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> a

Title:

Input can not be blank

Title: **Code Complete**

Author:

Input can not be blank

Author: **Steve McConnell**

Pages: **some**

Invalid input; enter a valid number

Pages: **-1**

Number must be >= 0

Pages: **960**

Code Complete by Steve McConnell, (960 pages) added to reading list

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **r**

Required books:

0. Developing the Leader Within You by John Maxwell 225 pages

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

4. Code Complete by Steve McConnell 960 pages

Total pages for 3 books: 1977

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **m**

Required books:

0. Developing the Leader Within You by John Maxwell 225 pages

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

4. Code Complete by Steve McConnell 960 pages

Total pages for 3 books: 1977

Enter the number of a book to mark as completed

>>> **0**

Developing the Leader Within You by John Maxwell marked as completed

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **m**

Required books:

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

4. Code Complete by Steve McConnell 960 pages

Total pages for 2 books: 1752

Enter the number of a book to mark as completed

>>> **4**

Code Complete by Steve McConnell marked as completed

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **m**

Required books:

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

Total pages for 1 books: 792

Enter the number of a book to mark as completed

>>> **3**

The Practice of Computing Using Python by Punch and Enbody marked as completed

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **m**

No required books

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **r**

Required books:

No books

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **c**

Completed books:

0. Developing the Leader Within You by John Maxwell 225 pages

1. The 360 Degree Leader by John Maxwell 369 pages

2. In Search of Lost Time by Marcel Proust 365 pages

3. The Practice of Computing Using Python by Punch and Enbody 792 pages

4. Code Complete by Steve McConnell 960 pages

Total pages for 5 books: 2711

Menu:

R - List required books

C - List completed books

A - Add new book

M - Mark a book as completed

Q - Quit

>>> **q**

5 books saved to books.csv

Have a nice day :)

At the end of this run, the saved CSV file contained:

Developing the Leader Within You,John Maxwell,225,c  
The 360 Degree Leader,John Maxwell,369,c  
In Search of Lost Time,Marcel Proust,365,c  
The Practice of Computing Using Python,Punch and Enbody,792,c  
Code Complete,Steve McConnell,960,c

References – Resources from Subject Materials:

Selected subject materials are referenced here to help you find guidance for specific parts of the assignment (e.g. sorting a list by multiple values is covered in [1] and you will find a template for writing menus for console programs in [2]). General references are not listed specifically but should be obvious (e.g. file input/output is covered in the lecture and practical on files).

1. itemgetter from Chapter 7 - Lists and Tuples.

2. Practical 01 - PyCharm, Control.

3. Practical 02 - Strings, Files, Exceptions.

4. Chapter 5 - Files and Exceptions 1.

5. Guide to Good Pseudocode.

Marking Scheme:

Ensure that you follow the processes and guidelines taught in class in order to produce high quality work. Do not just focus on getting the program working. This assessment rubric provides you with the characteristics of exemplary, good, competent, marginal and unacceptable work in relation to task criteria.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Exemplary (4)** | **Good (3)** | **Satisfactory (2)** | **Limited (1)** | **Very Limited (0)** |
| **Planning  Pseudocode for algorithms** | Clear, well-formatted, consistent and accurate pseudocode that completely and correctly solves the problem, for two functions. | Exhibits aspects of exemplary (left) and competent (right) | Some but not many problems (e.g. an incomplete solution, inconsistent use of terms, inaccurate formatting, not for two functions). | Exhibits aspects of competent (left) and unacceptable (right) | Very many problems or pseudocode not done. |
| **Program Execution**  **Correctness *Worth double*** | Program works correctly for all functionality required. | Program mostly works correctly for most functionality, but there is/are some required aspects missing or that have problems. | Program works incorrectly for all functionality required. |
| **Error checking** | Invalid inputs are handled well using exceptions and control logic as instructed, for all user inputs. | Invalid inputs are mostly handled correctly as instructed, but there is/are some problem(s), e.g. exceptions not well used. | Error checking is not done or is very poorly attempted. |
| **Similarity to sample output (including all formatting)** | All outputs match sample output perfectly, or only one minor difference, e.g. wording, spacing. | Multiple differences (e.g. typos, spacing, formatting) in program output compared to sample output. | No reasonable attempt made to match sample output. Very many differences. |
| **Quality of Code Identifier naming** | All function, variable and constant names are appropriate, meaningful and consistent. | Multiple function, variable or constant names are not appropriate, meaningful or consistent. | Many function, variable or constant names are not appropriate, meaningful or consistent. |
| **Use of code constructs** | Appropriate and efficient code use, including good logical choices for data structures and loops, good use of constants, etc. | Mostly appropriate code use but with definite problems, e.g. unnecessary code, poor choice of data structures or loops, no use of constants. | Many significant problems with code use. |
| **Use of functions** | Functions and parameters are appropriately used, functions reused to avoid code duplication. | Functions used but not well, e.g. incorrect/missing parameters or calls, unnecessary duplication or main code outside main function. | No functions used or functions used very poorly. |
| **Formatting** | All formatting is appropriate, including correct indentation, horizontal spacing and consistent vertical line spacing. PyCharm shows no formatting warnings. | Multiple problems with formatting reduces readability of code. PyCharm shows formatting warnings. | Readability is poor due to formatting problems. PyCharm shows many formatting warnings. |
| **Commenting** | Helpful block/inline comments and meaningful docstrings for all functions, top docstring contains all program details (name, date, basic description, GitHub URL). | Comments contain some noise (too many/unhelpful comments) or some missing program details in top docstring or some inappropriate or missing block/inline comments. | Commenting is very poor either through having too many comments (noise) or too few comments. |
| **Use of version control** | Git/GitHub used effectively and the repository contains a good number of commits with good messages that demonstrate incremental code development. | Aspects of the use of version control are poor, e.g. not many commits, or meaningless messages that don’t represent valuable incremental development. | Git/GitHub not used at all. |