

CP1404/CP1804 2018 SP2 Assignment 1 – Reading Tracker 1.0

Task:

You are to plan and 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 help references, like [0], that refer to the resources list near the bottom of this document. Check these references to find help on that topic. *Everything* you need to know can be found in the subject materials. In the past, some students have spent hours figuring out how to do something, only to find the solution was referred to right here.

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 reads and writes a list of books in a file, and each book has:

- title, author, number of pages, whether it is required (r) or completed (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]

Did you see that reference? Look below for [1] and you'll find out where we have taught you how to sort lists by multiple keys.

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***: display a neatly formatted (lined up) list of all the books with their details and a total of the pages and books left to read [3]. Note that the lining up is dynamic, based on the longest title and author [5]
- *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); new books are always required
- *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 message should be displayed and the user returned to the menu
- *when the user chooses **quit***: save the books to the CSV file, overwriting the file contents

Coding Requirements and Suggestions:

- 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**.
- Make use of named constants where appropriate. In general, you should use a named constant for any literals that you use more than once.
- 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 list and when you mark a book as completed... and getting the number of pages and the book number to complete are both situations where you need to get a positive integer)
- 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.
- Do not store a book's index – this is just its position in the list.
- 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.
- The output shows that the solution does not require correct plurals (e.g. "1 books"). You are welcome to leave yours this way, or you may add logic to print them correctly, but it is not required.

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.

Git & GitHub:

You must use Git version control and keep your project updated online in GitHub using the Classroom link we provide below. Do not create your own repository. You must follow these requirements:

Setup GitHub, as explained at <https://github.com/CP1404/Starter/wiki/Software-Setup#github>
You must be easily identifiable from your GitHub username and use your JCU email address as your primary GitHub email.

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

To start the assignment, log in to GitHub, then click the following link to get your own private repository hosted in the subject's GitHub Classroom organisation. This is **the** repository you must commit your work to, so clone it and push to it... do not use any other repo.

The assignment repo will begin with some 'starter' files for you, including a README you need to edit, the movies CSV file (and a temporary one as a backup) and a Python file for your assignment code. Use these files. Do not create any new files or remove these files.

<https://classroom.github.com/a/Ab-GQO45>

Integrity:

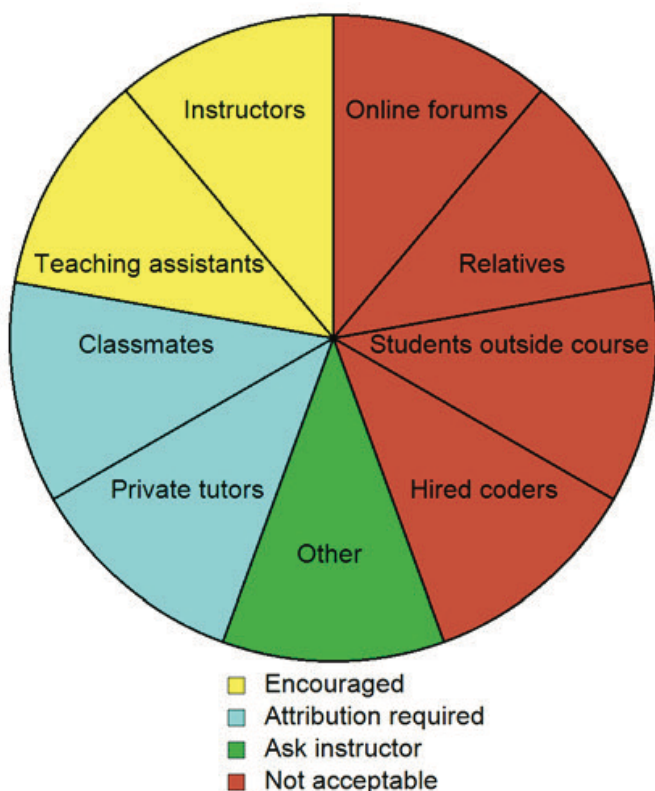
The work you submit for this assignment must be your own. Submissions that are detected to be too similar to that of another student will be dealt with according to the College procedures for handling plagiarism and may result in serious penalties.

The goals of this assignment include helping you gain understanding of fundamental programming concepts and skills, and future subjects will build on this learning. Therefore, it is important that you develop these skills to a high level by completing the work and gaining the understanding yourself. You may 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, including looking at another student's work or having a classmate look at your work. 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.

The subject materials (lecture notes, practicals, textbook and other guides provided in the subject) contain all of the information you need for this particular assignment. You should not use online resources (e.g. Stack Overflow or other forums) to find resources or assistance as this would limit your learning and would mean that you would not achieve the goals of the assignment - mastering fundamental programming concepts and skills.

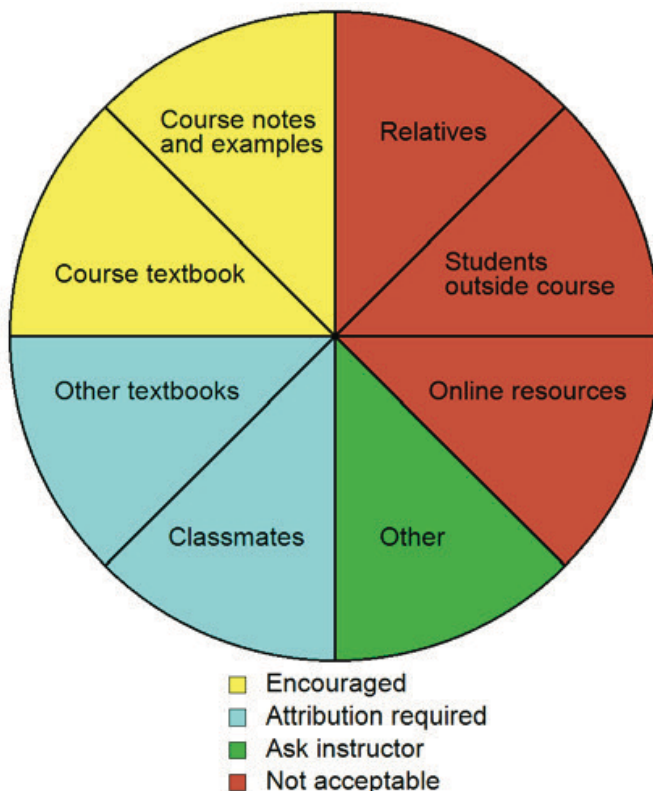
Assistance: Who can you get help from?

Use this diagram to determine from whom you may seek help with your programs. [6]



Resources: Where can you get code from?

Use this diagram to determine where you may find code to use in your programs.



Submission:

Zip up your project/repo directory, then submit this zip file (which will contain your .py code file, books.csv, your project reflection README.md, PyCharm project files and the .git directory). Ensure that you do NOT include a venv. Name the file like:

FirstnameLastnameA1.zip. Upload your single zip file on LearnJCU under Assessment (click on the title of the assignment).

Due:

Submit your assignment by the date and time specified on LearnJCU. Submissions received after this date will incur late penalties as described in the subject outline.

Sample Output:

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

```
Developing the Leader Within You,John Maxwell,225,r
The 360 Degree Leader,John Maxwell,369,r
In Search of Lost Time,Marcel Proust,13,c
The Practice of Computing Using Python,Punch and Enbody,792,r
```

You should be able to figure out what parts of the sample output below are user input.

```
Reading Tracker 1.0 - by Lindsay Ward
4 books loaded from books.csv
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> f
Invalid menu choice
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> L
*1. Developing the Leader Within You      by John Maxwell      225 pages
*2. The 360 Degree Leader                  by John Maxwell      369 pages
   3. In Search of Lost Time                by Marcel Proust      13 pages
*4. The Practice of Computing Using Python by Punch and Enbody 792 pages
4 books.
You need to read 1386 pages in 3 books.
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> m
*1. Developing the Leader Within You      by John Maxwell      225 pages
*2. The 360 Degree Leader                  by John Maxwell      369 pages
   3. In Search of Lost Time                by Marcel Proust      13 pages
*4. The Practice of Computing Using Python by Punch and Enbody 792 pages
4 books.
You need to read 1386 pages in 3 books.
Enter the number of a book to mark as completed
>>> why?
Invalid input; enter a valid number
>>> 0
Number must be > 0
>>> 5
Invalid book number
>>> 4
The Practice of Computing Using Python by Punch and Enbody completed!
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> M
*1. Developing the Leader Within You      by John Maxwell      225 pages
*2. The 360 Degree Leader                  by John Maxwell      369 pages
   3. In Search of Lost Time                by Marcel Proust      13 pages
   4. The Practice of Computing Using Python by Punch and Enbody 792 pages
4 books.
```

```

You need to read 594 pages in 2 books.
Enter the number of a book to mark as completed
>>> 1
Developing the Leader Within You by John Maxwell completed!
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> l
  1. Developing the Leader Within You          by John Maxwell          225 pages
*2. The 360 Degree Leader                      by John Maxwell          369 pages
  3. In Search of Lost Time                    by Marcel Proust          13 pages
  4. The Practice of Computing Using Python by Punch and Enbody 792 pages
4 books.
You need to read 369 pages in 1 books.
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> m
  1. Developing the Leader Within You          by John Maxwell          225 pages
*2. The 360 Degree Leader                      by John Maxwell          369 pages
  3. In Search of Lost Time                    by Marcel Proust          13 pages
  4. The Practice of Computing Using Python by Punch and Enbody 792 pages
4 books.
You need to read 369 pages in 1 books.
Enter the number of a book to mark as completed
>>> 2
The 360 Degree Leader by John Maxwell completed!
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> m
No required books
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> L
  1. Developing the Leader Within You          by John Maxwell          225 pages
  2. The 360 Degree Leader                      by John Maxwell          369 pages
  3. In Search of Lost Time                    by Marcel Proust          13 pages
  4. The Practice of Computing Using Python by Punch and Enbody 792 pages
4 books.
No books left to read. Why not add a new book?
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> A
Title:
Input can not be blank
Title: Harry Potter and the Prisoner of James Cook University
Author:
Input can not be blank
Author:
Input can not be blank
Author: A Very Famous Author
Pages: just one
Invalid input; enter a valid number
Pages: 0
Number must be > 0
Pages: 1
Harry Potter and the Prisoner of James Cook University by A Very Famous Author, (1 pages) added to
Reading Tracker
Menu:
L - List all books
A - Add new book
M - Mark a book as completed
Q - Quit
>>> L
*1. Harry Potter and the Prisoner of James Cook University by A Very Famous Author 1 pages
  2. Developing the Leader Within You          by John Maxwell          225 pages
  3. The 360 Degree Leader                      by John Maxwell          369 pages
  4. In Search of Lost Time                    by Marcel Proust          13 pages
  5. The Practice of Computing Using Python by Punch and Enbody 792 pages
5 books.
You need to read 1 pages in 1 books.

```

```
Menu:
L - List all 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,13,c
The Practice of Computing Using Python,Punch and Enbody,792,c
Harry Potter and the Prisoner of James Cook University,A Very Famous Author,1,r
```

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.
https://github.com/CP1404/Practicals/tree/master/prac_01
3. Practical 02 - Strings, Files, Exceptions.
https://github.com/CP1404/Practicals/tree/master/prac_02
4. Chapter 5 - Files and Exceptions 1.
5. Practical 05 - Dictionaries, Code Reviews with PRs.
https://github.com/CP1404/Practicals/tree/master/prac_05
6. Negotiating the Maze of Academic Integrity in Computing Education.
<https://dl.acm.org/citation.cfm?doid=3024906.3024910>

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 (9, 10)	Good (7, 8)	Satisfactory (5, 6)	Limited (2, 3, 4)	Very Limited (0, 1)
Program Execution Correctness <i>Worth double</i>	Program works correctly for all functionality required.	Exhibits aspects of exemplary (left) and satisfactory (right)	Program mostly works correctly for most functionality, but there is/are some required aspects missing or that have problems.	Exhibits aspects of satisfactory (left) and very limited (right)	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 are well 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, meaningless messages that don't represent valuable incremental development.		Git/GitHub not used at all.