CP1404/CP5632 Assignment 1 – Books to Read 1.0

Task:

You are to write a Python (3) program, 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 define any of your own classes or use constructs that haven't been taught in this subject. Assignment 2 will build on this with more advanced constructs including dictionaries, classes and a Graphical User Interface (GUI).

Everything you need to complete this assignment can be found in the subject materials. You should find the programming guides helpful, and you are expected to follow the style guide: https://github.com/CP1404/Starter/wiki

Start your work by clicking this link to create a new repository in GitHub Classroom:

https://classroom.github.com/a/2GOH2f69

Do not use any other repo or a copy of this one... just use this repository!
This will give you a new repo containing starter files including a README for your project, all of which you must use. Do not add any other files in this project, and do not rename anything - just use this as your assignment repo. Do not "download" this repo, but rather *clone* this repo using PyCharm ("Get from VCS"). You will probably need to create and use a **token** to access your private repository. Do not make your repository public at any stage.

Program Overview:

This program is an interactive book list that allows a user to track books that they wish to read and books they have completed. The program reads and writes a list of books in a CSV file. Each book has the following data:

- title, author, number of pages, whether it is completed (c) or unread (u)

Users can display the list of books, which should be sorted by author then by title.
Users can add new books and mark books as completed.
Users cannot change books from completed to unread.

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 own name in it.
- Load a CSV text file of books (just once at the very start) into a list of lists - a sample CSV file is provided for you, and you must use this format.
- Display a menu for the user to choose from until they choose to quit.
- ***Display**:* display a neatly lined up list of all the books with their details. Unread books have * next to them) and a count of these books. You need to determine the size of the title and author fields to get them to line up correctly.
- ***Add**:* prompt for the Book's title, author and number of pages, error-checking each of these, then add the book to the book list in memory (not to the file); new books are always unread initially.
- ***Complete**:* prompt the user to choose one book by number, error-checking for a valid book number, then change that book's status to completed. If no books are unread, then display the message in the sample output below.
- When the user quits the menu, save the books to the CSV file, overwriting file contents.

Coding Requirements:

1. Work incrementally on this task: complete small parts of it at a time rather than trying to get it all working at once.
2. You are assessed on your use of version control including commits and commit messages, so please commit appropriately for each “small 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.
3. Edit the module docstring at the very top of your code file to contain your own details.
4. The menu should handle both uppercase and lowercase letters as valid choices.
5. Make use of named constants as appropriate (e.g., for the characters that represent the completion status).
6. Use functions appropriately for each significant part of the program: this is the divide-and-conquer problem-solving approach. Follow the single responsibility principle.
7. For efficiency, you should ***only load the books file once*** – when the program starts.
8. **Only save the books file once** – when the user quits.
9. **Store the book data in a list of lists** and pass this to any functions that need access to it. Do not store a book’s index – this is just its position in the list.
10. **Do not use any global variables.** The only globals you may have are CONSTANTS. If you use global variables, your functions will be poorly designed.
11. Use exception handling as appropriate to deal with input errors (including when entering numbers and selecting books). You should 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 understand how you will be assessed. There should be no surprises here – this is about following the patterns and practices we have taught in class.

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 or other work (e.g., code found online or created with AI) 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 (lectures, practicals, textbook and other guides provided in the subject) contain all the information you need for this assignment. Do not use online resources or GenAI (e.g., Stack Overflow, ChatGPT 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.

Submission:

Submit your assignment1.py and README.md files in a single attempt to LearnJCU. Your Python file module docstring must contain an accurate link to your assignment GitHub repository to get any marks for version control.

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:

Sample output from the program is provided below. **Ensure that your program matches this, including spaces, spelling, and 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 a large (but not exhaustive) range of situations including user input error handling.

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

```
Developing the Leader Within You,John Maxwell,225,u
The 360 Degree Leader,John Maxwell,369,c
In Search of Lost Time,Marcel Proust,93,u
Starting Out with Python,Tony Gaddis,744,c
```

It should be clear what parts of the sample are user input for this sample run.

```
Books to Read 1.0 by Lindsay Ward
4 books loaded.
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> L
Invalid menu choice
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> d
*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   93 pages
 4. Starting Out with Python         by Tony Gaddis    744 pages
You still need to read 318 pages in 2 books.
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> a
Title:
Input can not be blank
Title:
Input can not be blank
Title: Why not?
Author:
Input can not be blank
Author: Barry Nalebuff
Number of Pages: I don't know
Invalid input - please enter a valid number
Number of Pages:
Invalid input - please enter a valid number
Number of Pages: -1
Number must be > 0
Number of Pages: 0
Number must be > 0
Number of Pages: 256
Why not? by Barry Nalebuff (256 pages) added.
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> c
*1. Why not?                         by Barry Nalebuff  256 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    93 pages
 5. Starting Out with Python         by Tony Gaddis     744 pages
You still need to read 574 pages in 3 books.
Enter the number of a book to mark as completed
>>> 
Invalid input - please enter a valid number
>>> oh, here too?
Invalid input - please enter a valid number
>>> 6
Invalid book number
>>> 0
Number must be > 0
>>> -1
Number must be > 0
>>> 2
Developing the Leader Within You by John Maxwell completed!
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> c
*1. Why not?                         by Barry Nalebuff  256 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    93 pages
 5. Starting Out with Python         by Tony Gaddis     744 pages
You still need to read 349 pages in 2 books.
Enter the number of a book to mark as completed
>>> 2
That book is already completed
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> c
*1. Why not?                         by Barry Nalebuff  256 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    93 pages
 5. Starting Out with Python         by Tony Gaddis     744 pages
You still need to read 349 pages in 2 books.
Enter the number of a book to mark as completed
>>> 4
In Search of Lost Time by Marcel Proust completed!
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> c
*1. Why not?                         by Barry Nalebuff  256 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    93 pages
 5. Starting Out with Python         by Tony Gaddis     744 pages
You still need to read 256 pages in 1 books.
Enter the number of a book to mark as completed
>>> 1
Why not? by Barry Nalebuff completed!
Menu:
D - Display books
```

```
A - Add a new book
C - Complete a book
Q - Quit
>>> c
No unread books - well done!
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> D
 1. Why not?                         by Barry Nalebuff  256 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    93 pages
 5. Starting Out with Python         by Tony Gaddis     744 pages
No books left to read. Why not add a new book?
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> a
Title: 100%
Author: Lindsay Ward
Number of Pages: 1
100% by Lindsay Ward (1 pages) added.
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> d
 1. Why not?                         by Barry Nalebuff  256 pages
 2. Developing the Leader Within You by John Maxwell    225 pages
 3. The 360 Degree Leader            by John Maxwell    369 pages
*4. 100%                             by Lindsay Ward      1 pages
 5. In Search of Lost Time           by Marcel Proust    93 pages
 6. Starting Out with Python         by Tony Gaddis     744 pages
You still need to read 1 pages in 1 books.
Menu:
D - Display books
A - Add a new book
C - Complete a book
Q - Quit
>>> q
6 books saved to books.csv
"So many books, so little time. Frank Zappa"
```

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

```
Why not?,Barry Nalebuff,256,c
Developing the Leader Within You,John Maxwell,225,c
The 360 Degree Leader,John Maxwell,369,c
100%,Lindsay Ward,1,u
In Search of Lost Time,Marcel Proust,93,c
Starting Out with Python,Tony Gaddis,744,c
```

Marking Scheme:

Ensure that you follow the processes and guidelines taught in class to produce high quality work. Do not just focus on getting the program working.
This assessment rubric provides you with the characteristics of exemplary down to very limited work in relation to task criteria covering the outcomes:

- SLO1 - develop and utilise best-practice coding techniques to develop solutions
- SLO2 - select and apply appropriate and efficient data structures
- SLO3 - manage software projects using version control

Criteria	Exemplary (9, 10)	Good (7, 8)	Satisfactory (5, 6)	Limited (2, 3, 4)	Very Limited (0, 1)
Correctness SLO1 20%	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 SLO1 10%	Invalid inputs are handled well using exceptions and control logic as taught, for all user inputs.		Invalid inputs are mostly handled correctly as taught, 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 SLO1 10%	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.
Identifier naming SLO1 10%	All function, variable and constant names are appropriate, meaningful and consistent, following style guide.		Multiple function, variable or constant names are not appropriate, meaningful or consistent, not following style guide.		Many function, variable or constant names are not appropriate, meaningful or consistent, not following style guide.
Use of code constructs SLO1, 2 15%	Appropriate and efficient code use, including good logical choices for data structures, decision and repetition structures, good use of constants, as taught in the subject.		Mostly appropriate code use but with definite problems, e.g., unnecessary code, poor choice of data structures, decision and repetition structures, limited use of constants.		Many significant problems with code use.
Use of functions SLO1 10%	Functions and parameters are appropriately used, functions are well reused to avoid code duplication, functions follow SRP well.		Reasonable but not good use of functions, e.g., poor parameter choices, function choices don't follow SRP well, too much low-level detail in main.		No functions used or functions used very poorly. ***Any use of global variables.***
Formatting SLO1 5%	All code formatting is appropriate, including correct indentation, horizontal spacing and consistent vertical line spacing. PyCharm shows no formatting warnings.		Multiple problems with code formatting reduce readability of code. PyCharm shows formatting warnings.		Readability is poor due to code formatting problems. PyCharm shows many formatting warnings.
Commenting SLO1 10%	Meaningful docstrings for all functions, appropriate block/inline comments, top docstring and README are complete.		Missing function docstrings, noise comments or missing block/inline comments, details missing from top docstring or README.		Commenting is very poor either through having too many comments (noise) or too few comments.
Use of version control SLO3 10%	Git used well (local first, not on GitHub directly), good number of commits with good messages that demonstrate incremental code development including appropriate "small milestone" commits.		Aspects of the use of version control are poor, e.g., not many commits, meaningless or repeated messages, evidence of direct GitHub website edits or uploads.		Git/GitHub not used at all.