

# Assignment 6: Managing Collections

This assignment is to be completed individually. **It is not a team project.**

You must document (using comments in your code) all resources (beyond our official textbook, Sakai, the class discussion forum, or the class website) that you used to help you complete this assignment. For example, if you referenced a code example from an online website such as Stackoverflow or github you could acknowledge it as follows:

```
// Next we print 'Hello, World' to the console.
// Based on help from: https://stackoverflow.com/questions/826948/syntax-error-on-print-with-python-3
print("Hello, world!")
```

You must provide an attribution to any resource you consulted to complete this assignment except for our official textbook, Sakai, class discussion forum, or the class website. Resources that require attribution include — but are not limited to — websites, books, notes from other students, tutors, or help from other people (friends, classmates, etc.). Under no circumstances are you to look at or copy any material related to another student's solution for this assignment.

To repeat, you must attribute **any resource** you consulted to complete this assignment other than our official textbook, Sakai, class discussion forum, or the class website. Failure to provide attribution is a violation of the honor code.

## Overview

For this assignment, you'll be creating a simple program to manage a collection of books and movies for a library. You will be provided with a basic program "skeleton" that loads the contents of the collection from two data files (one for books; one for movies). You'll then need to add the program code required to allow users to perform a variety of operations (e.g., checking in and out books, or adding new items to the collection). The full list of operations your program must support and other specific requirements are outlined below.

**Skeleton Code.** Unlike past assignments, you will not need to start from scratch on this assignment. Instead, you should start with the [skeleton code](#) provided here. To complete the assignment, add the required functionality to this skeleton.

Your very first order of business, after reading through the assignment, should be to download, rename, and run this skeleton program. First, make sure it runs successfully. Then, make sure you understand how it works. In particular, pay attention to how the data is represented once it is loaded into your program. You will need to understand the data structure very well in order to complete this assignment successfully.

**Input Data.** There are two input data files for this assignment, both in [CSV \(comma-separated values\) format](#). The first line of each file contains a list of variables known for each item. The remaining lines contain the actual collection data, with one line per item. Each line contains multiple values, separated (as the term CSV suggests) by commas. While the skeleton program provided to you will parse this file, I suggest that you open these files in a text editor (e.g., Notepad on Windows) or Excel to see what the data looks like before you start coding. You should also look to see how the program skeleton works. Understanding the data files and the code that parses them will help you understand the data structures used by the skeleton program to represent the collections.

The first file, [books.csv](#), contains the list of books in the collection. It contains 8 variables per book, and the variable names are defined on the first line. They include: Title, Author, Publisher, Pages, Year, Copies, Available, and ID. The "Copies" field represents the number of copies owned within the collection. The "Available" field represents the number of copies available for checkout at the current point in time. The ID field is a unique identifier for the item.

The second file, [movies.csv](#), contains the list of movies in the collection. It contains 8 variables per movie, and the variables are defined on the first line. They include: Title, Director, Length, Genre, Year, Copies, Available, and ID. The "Copies" field represents the number of copies owned within the collection. The "Available" field represents the number of copies available for checkout at the current point in time. The ID field is a unique identifier for the item.

## Basic Requirements

Satisfying all basic requirements perfectly, with no points deducted for any reason, would earn a maximum score of 8 out of 10 for this assignment.

Your program should perform the following:

- Load the books and movies data files into the program, storing the data in a data structure using lists, dictionaries, and/or sets. **This**

**functionality is provided to you in the skeleton code. You should use the skeleton code as a starting point.**

- Display a menu that allows the user to perform a variety of actions, including:
  - Query for a book by pressing 'qb' then typing enter. The program should then allow a user to enter a query string which will be used to search against multiple fields in the collection: title, author, and publisher. The search should perform **partial string matching and be case insensitive**. For example, entering the query string "ion" should match both of the following strings: "Ions are Science" and "Frustration in Packaging". Details FOR ALL MATCHING BOOKS should be displayed.
  - Query for a movie by pressing 'qm' then typing enter. The program should then allow a user to enter a query string which will be used to search against multiple fields in the collection: title, director, and genre. The search, just as with books, should perform **partial string matching and be case insensitive**. Details FOR ALL MATCHING MOVIES should be displayed.
  - Request to check out an item by pressing 'co' and pressing enter. The program should then prompt the user to enter an ID number. The program should check to see if the requested item is available. If so, the availability number should be decreased by one and the user told that he/she has checked out the item. If the item is not available, or if the ID is not valid, the user should be shown an appropriate error message informing them about what went wrong.
  - Request to check in an item by pressing 'ci' and pressing enter. The program should then prompt the user to enter an ID number. The program should check to see that the requested item has indeed been checked out (that the available number is less than the copies number for the item). If so, the availability number should be increased by one and the user told that he/she has checked in the item. If the item is not checked out, or if the ID is not valid, the user should be shown an appropriate error message informing them about what went wrong.
  - Exit the program by pressing 'x' then typing enter.

The console output produced by your program should be nearly identical to [the sample output provided below](#). While the specific books or movies displayed will vary because of differences in user input, the prompts and messages printed to the console when using your program should match those in the sample output to receive full credit.

## Advanced Requirements

Satisfying both the basic and advanced requirements perfectly, with no points deducted for any reason, will result in a full 10 out of 10 score for this assignment.

Expand on the basic requirements by extending your program as follows.

- Allow the user to request the display of the entire collection of either books (by typing "db") or movies (by typing "dm"), paging through the results 10 at a time. The user should be able to press <enter> to move on to the next group of ten items until the entire collection has been shown. The user should be able to press "m" (then <enter>) to quit the display process and return to the menu of options.
- Allow the user to add a new book (by typing "ab") or movie (by typing "am") to the collection.

Want an even greater challenge? You won't get extra credit, but test your skills by allowing the user to save the current collection to the input data files (by typing "s"). If done correctly, this will allow your checkin, checkout, and add commands to remain in place even when you restart your program!

## Sample Output

An example of the output produced by my solution to this assignment can be found [here](#).

## Grading Criteria

This assignment will be graded on a 10 point scale. Your grade for this assignment will be based on a combination of factors including:

- Correct functionality (e.g., Does your Python code do what it is supposed to do as outlined in the basic and/or advanced requirements?)
- Clarity of your solution (e.g., Did you solve the problem directly and efficiently? Or is your answer excessively complex and/or inefficient?)
- Coding style (e.g., Is your code readable with comments, meaningful variable names, and good whitespace/indentation?)
- Meets submission requirements (see below)

## Seeking Help

For general questions about Python, please use the class forum on Piazza to seek assistance. For questions that are personal in nature or that

would reveal a solution to the assignment, you ask for help by email or during office hours. However, please note that emailed questions will not receive an immediate response. It is likely that it will take 24-48 hours for me to respond.

---

## Submitting Your Solution

**Please Note:** You must name the main python file for your assignment "assignment<number>\_<onyen>.py". For example, for assignment 3 I would name my file `assignment3_gotz.py` because my onyen is gotz.

**Using the wrong name for your file will be cost you points on your assignment grade.** Please follow this requirement carefully!

Please submit your assignment via Sakai. You should submit a zip file containing your entire project folder. To create the zip file, follow the [submission instructions](#) that have been posted at the bottom of the "Other Information" page on our course website.

The due date for this assignment can be found on [the course schedule](#).