CECS174, Project 2

Media check-in / check-out
Points worth: 10% of your total score

Group work: with a group size of 5 people (7 Groups). List the names on the top of the code under comments section and **provide details** on what each member contributed to the code. **Grades will consider contribution to the project.**

What to submit: Source file –> **_Project2_Media.py_**

<mark>Due date/time: 12:30pm on Tuesday, May 7, 2019 **(No late submission!)**</mark>

Create a python program for managing media check-out/check-in with members (i.e. in a library)

There are two types of Media class - **Book** and **Video under Media class.**

- **Book** - title, author, publisher, number of pages.
- **Video** - title, author, publisher, running time.

There is a **Member** class:

- Member object has name and can check out and check in books and videos.
- If a book or a video is already checked out, then it cannot be checked out again until it is checked back in.
- Display a confirmation message during check-out/check-in by showing the name of the member and the information of the media.
- Store a list of checked out items for a member and create a method to display it.

To provide general information about the media and members:

- Keep track of how many books and videos are there and how many are checked out.
- Keep track of number of members

Coding Requirements:

- You are to create object-oriented code and keep all attributes within the objects.
- Your code is to use demonstrate the concept of OOP - **"encapsulation, inheritance, polymorphism"**
- You are to implement the print method so that you can display the contents of the object.
- Names used for attributes and methods are to be meaningful and easy to understand.
- A member shall only be allowed to check out 2 items at once.
- Create class methods for accessing class attributes.

Hints:

- **Media is a superclass** of Book and Video - There is no need to create a library class.
- All common attributes and methods from subclasses should be placed in the **superclass**.

- Anything specific to the subclasses should be stored and implemented in the **subclasses**.
- Consider **what** information should be stored **in instance** attributes and what **in class** attributes.
- Members is a standalone class for keeping information about the member.
  - → You need to implement methods to enable the member to interact with books and videos by **passing the instance** object as argument (refer to class example.)

- Follow good programming practices and make sure that your code includes plenty of comments.
- Refrain from reusing the same name for attributes/variables throughout the code for different purposes as this will get confusing and makes the code hard to read.

Include the Use Case of yours, you must:

- Create instances of book, video, and member classes.
- Call the **checkOut**() and **checkIn**() methods on each member.
- Call the **printCheckedOutItems()** method to display the items checked out by a member.
- Call the checkOut() method on a book/video that is already checked out by another member and verify that a message is displayed "The book /video is already checked out by member(name) "
- Create **a function displayStats()** to display information about the Media (its subclasses) and Member classes such as:
  - ○ Total number of books, and books checked out
  - ○ Total number of videos, and videos checked out
  - ○ Total number of members
  - ○ Display all items checked out

- **Have your test code enclosed within the special attribute**

   if __name__ == "__main__":

      ### Your test code goes here ###

- Below is an example for reference test code. **You shall create more test cases as needed.**

```
Book1= Book( ... )                      # creates an instance for a book
book2 = Book( ... )                     # creates an instance for a book
Joe = Member("Joe Smith")               # creates an instance for a member
Jim = Member("Jim Stuart")              # creates an instance for a member
Joe.checkOut(...)
video1 = Video( ... )                   # creates an instance for a video
Joe.checkOut( … )
Jim.checkOut( … )                       # should attempt to check out the same item as Joe's
Joe.checkIn( … )
Joe.printCheckedOutItems()
displayStats()                          # display the information
```

- Upload your original work onto Beachboard's Dropbox folder before the due date!
- Your submission should be free from errors, and run in Python version 3.x.x
- If your code does not run or terminates with errors, you will NOT receive any points.
- Include a comment section at the beginning of the code that lists the team members, as well as the contribution to the project. **Be specific to get the full credit**!

- Below an example of project contribution of members:

```
# Project2_Media

# Project Members: Jim xx, Tom xx, Sam xx, Eric xx

# Jim xx: (Project Manager)
# -Set up and organization
# -Contributed most of the two subclasses
# -Contributed to debugging entire program
# -Contributed to comments
#
# Tom xx:
# -All of the work regarding counters
# -Majority of the work in print formatting
# -Contributed to the subclasses' methods creation(checkOut/checkIn)
# -Contributed to debugging entire program
# -Contributed to comments
#
# Sam xx:
# -Heavily contributed to program framework
# -Heavily contributed to the Member class
#
# Eric xx:
# -Contributed to print formatting
# -Contributed to program framework
# -Contributed to Media class
# -Contributed to debugging check in/check out
```