CSE 131 Final Exam
Due: August 20th, at noon Central Time.

This is an open exam. You may consult any other sources that you wish, though you may not speak to other people about the content of this exam during the exam period. Any questions about the exam should be directed to the course instructor.

You can use code from your prior assignments. You can use resources that you find on the internet. The work that is submitted must be your own work, however. Submitting code from an online source or code that was written by anyone other than yourself will be considered a violation of the academic integrity policy and referred to the McKelvey School of Engineering for further action.

This exam will be graded based on the code that you commit and push to your code repository. Code that is dated after the end time of the exam your code repository will not be accepted, so make sure that you commit and push early and often.

Exam Mechanics:

You should create a repository by using this link:

All code for the exam should go in this repository. If your code is not in the repository by the end time of the exam, it will not be accepted. Please commit often and verify that your work got committed by checking your repository page on GitHub.

If you need help setting up, please ask, but note that you will not receive extra time on the exam due to technical issues.

Exam Overview

Your goal for this exam is to create a simple library application. This library will have two kinds of items: magazines and books. The library should be able to

Part 1 will have you complete a `Readable` interface that will be used for the items in the library.

Part 2 will have you implement the `Readable` interface on the two types of readable items: Magazines and Books.

Part 3 will have you implement the `Library` functionality.

Also note that a main method has been provided to you in the `Library` class. It contains examples of how to create and use the objects, so use this as a guide. You can and should add your own code to test your work more thoroughly: the provided tests are not comprehensive, and a more thorough set of tests will be run when your exam is graded.

Part 1: `Readable` interface (10 points)

An empty `Readable` interface has been created for you. Things that are readble should be able to:

- Return the title of a `Readable` item
- Return the total number of pages that a `Readable` item has
- Return whether this `Readable` item is a book. If a `Readable` item is not a book, you can assume that it is a magazine.

Part 2: `Book` and `Magazine` (30 points)

A Book has-a:

- Title
- Author
- Number of pages

You should create a constructor for your `Book` class. A `Book` should implement the `Readable` interface. In addition to the functionality provided by the `Readable` interface, `Book` should have a `toString`() method that displays the information it contains. Here is an example of what that might look like:

```
'1984' by George Orwell, 283 pages
```

A Magazine has-a:

- Title
- Number of pages
- Structure that maps articles to page number (using the provided HashMap)

You should create a constructor for your `Magazine` class. A `Magazine` should implement the `Readable` interface. In addition to the functionality provided by the `Readable` interface, a `Magazine` should be able to return the total number of articles. It should have a way to add an article to the list of articles. It should also have a `toString`() method that displays the information it contains. Here is an example of what that might look like:

```
Thrasher, March 2021, 87 pages
page 2, How to do a kickflip
page 57, How to explain skateboarding to your grandma
```

Part 3: Library

A Library has-a:

- Name
- List of `Readable` items contained in the library

You can assume that the `Library` will be empty when it is first created. You should create a constructor for your `Library` class.

A Library should be able to do the following:

- It should be able to take in a `Readable` item and add it to the list of items.
- It should be able to compute and return the total number of pages contained in all `Readable` items.
- It should be able to compute and return the total number of articles contained in `Magazine` items.
- It should be able to print out the information for all `Readable` items in the library. This output may look something like the following:

```
Items currently at Mid-County Branch
Magazine: Thrasher, March 2021, 87 pages
page 2, How to do a kickflip
page 57, How to explain skateboarding to your
grandma

======================
Book: '1984' by George Orwell, 283 pages
```

```
======================
Magazine: Highlights for Kids, June 2021, 21 pages
page 2, Color the Pony!
page 4, Connect the Dots!
page 7, Broccoli is Yummy!

======================
Book: 'House of Leaves' by Mark Danielewski, 709
pages
======================
```

Again, please note that code examples of these methods being used as well as their expected output has been provided to you. Please use this to help you and add more tests of your own as appropriate.

Submission

To submit your assignment, simply commit and push your work. Please double check that it got submitted by visiting the repository page on GitHub. If you can see your work on GitHub then it has been properly submitted. Work that is committed after the due date will not be accepted.