Project Overview and Rubric

Work Individually

You must develop the code for this assignment entirely on your own. If you encounter difficulties while working, you are encouraged to discuss general algorithms and debugging strategies with anyone you would like. If you feel that getting assistance will require someone else viewing your code, the course staff are the only people that you are allowed to share or show any part of your code with prior to the hard deadline for this assignment. You may not store or share your code in any way that other students can access.

Project Requirements/Expectations

For the project portion of this course we will be leveraging the data structures/algorithms we have implemented during the course as well as the developmental tools we covered over the course of this course. It is recommended you begin this project early relative to the remainder of the semester ahead of the ultimate due date. This section will outline what components we are expecting to be included in your project. The [Project Proposal](https://canvas.wisc.edu/courses/254888/assignments/1227060) will also be a good resource for details regarding this project.

**Flipgrid Demo Video**: A 2-3 minute video demonstrating the functionality of your project. Flipgrid is an online video communication platform an we will share a page for video submissions as the due date approaches.

**Makefile**: You will submit a Makefile with three required behaviors. The two targets we are looking for are a "clean" target and a "run" target. The "clean" target should remove the class files in your submitted folder. The "test" target should run your JUnit tests. Your application should also compile and run as the default target so calling "make" will run your work. You can include additional targets as you see necessary.

**Unit Tests**: You must implement 5 unit tests related to the functionality of your application. These tests should all pass upon submission.

**Application Front End**: The front end interface of your application will take in user prompts and perform necessary actions. You may have used user input in a previous course, but for reference [this (Links to an external site.)](https://www.w3schools.com/java/java_user_input.asp) may be a helpful read through. Your front end should offer appropriate and functional operations the user can take respective to your application.

**Application Back End**: The backend of your application will interact with your front end interface, load your data from your external file, and interact directly with your chosen data structure.

**Application Data**: You will submit a dataset either from an external source or created by you and an associated class to store the data in (e.g. a movie class to store data for an individual movie). You must submit your data dataset and extract that dataset from an external file (this might be helpful [https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html (Links to an external site.)](https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html)).

**Code Documentation**: Code should be clear and well documented similar to other coding assignments. Ensure that your final submission for this assignment is clearly organized, consistently styled, well documented with comments, and includes the following file header information at the top of each source file:

// --== CS400 File Header Information ==--

// Name: <your full name>

// Email: <your @wisc.edu email address>

// Notes to Grader: <optional extra notes>

Submit

During Week 7 we will provide a submission page where you can upload a zip file containing your work for this project as well as a Flipgrid page where you can submit your demo video. All of these components are due on 7/13 (late deadline on 7/15).

Grading Rubric

| **Component** | **Criteria** | **Points** | **Check** |
| --- | --- | --- | --- |
| Flipgrid Demo Video (2-3 minutes) | Walks through various implemented functionality | 10 |  |
| Makefile | Application runs on the "make" command without issue. | 10 | P |
| Includes a "clean" target to remove all class files | 10 | P |
| Includes a "test" target to run all tests | 10 | P |
| Unit Tests | 5 JUnit tests are provided and test application functionality | 20 | P |
| Tests all pass on submission | 10 | P |
| Application Front End | Clear instructions for user interaction | 10 | P |
| All user commands are functional | 25 | P |
| Application Back End | Interfaces with a chosen data structure/algorithm implementation. | 20 | P |
| Exposes functionality to the Front End through handler functions. | 15 | P |
| Application Data | Class implemented to model data with appropriate methods | 10 | P |
| Data extracted from file and read into class structure | 15 | P |
| Data stored in file and has all necessary features included | 10 | P |
| Code Documentation | Code is clear and well documented | 10 | P |
| **Total Points Possible** |  | **185** |  |