Project 5

COMP 220 – Computer Programming II

- **Initial Design:** **Due Monday, November 14 by 11:55 pm**
- **Completed Project:** **Due Monday, December 5 by 11:55 pm**
- **Peer Review:** **Due Monday, December 5 by 11:55 pm**

Collaboration Policy

This is a group project. You should find a group of two or three people. (If you are looking for a group, we will have a chance to coordinate groups in class when we start talking about this project.)

Learning Objectives

Completing this project will give you experience designing a program to accomplish a task and using Java collections.

Project Overview

You get to decide what your program will do for this project! Your program should satisfy the following criteria:

- Use at least two of the following kinds of collections: Set, Map, and List [13 points]
- Correctly implement the desired functionality (i.e., do what it is supposed to do). The functionality should be of at least moderate complexity (details below). [50 points]
- The code should be easy to understand:
    - Logical structure, appropriate for the given application [12 points]
    - Commented with Javadoc for classes and methods, and as needed within methods [5 points]
- Version control: 5 points plus 2 possible extra credit points. One person on your team must demo your project and show me your github or bitbucket repository --- extra credit is given if every member of your team has at least 2 commits to your repository.

Initial design, demo are worth the other 15 points. Peer reviews will not be graded, however anyone who does not submit a peer review will receive a failing grade on the project.

Project Complexity and Ideas

One benefit of an open-ended project is that you can build a system that is interesting to you! Furthermore, you can tailor the complexity of your project based on your experience level. However, every project should meet a basic level of complexity. I will work with you during class to determine a project that is complex enough but still realistic for the time we have.

Beyond Basics

Of the points for complexity and correct implementation, 5 points are for going beyond the basic level. Thus, you can earn an 'A' for meeting the basic requirements, but full credit is reserved for the projects

that have something extra. Examples include projects that have a GUI or projects with a complicated algorithm implementation.

Project Ideas

Projects like the Cipher project or Game of Life are too simple for this assignment. Here are some ideas for potential projects:

- Web crawling and analysis
    - There are several interesting projects in this arena. "Crawling" is the process of following links from web page to web page, retrieving the web pages and extracting some information out of them. The jsoup library is useful for analyzing information from the web. Here are some specific ideas:
    - Analyze link patterns: which pages have the most incoming links, outgoing links, internal links, external links, etc.
    - Build a web search engine: you can use simple word matching (i.e., how many of the query words are in each page?), TF-IDF ranking, or a basic version of Google's PageRank algorithm. Any algorithm beyond simple matching would be beyond the basics.
- Games
    - Simulate a game like Checkers, Othello, or Ultimate Tic-Tac-Toe. If you provide a solid GUI, a complex game like Monopoly or Stratego, or a rudimentary AI, that qualifies as beyond the basics.
- Word processing
    - Provide a “cursor” that shows where you are in a block of text. Allow users to move around in the text, to type and insert text, and to undo changes.
    - Providing a good GUI (blinking cursor, etc.) or additional interesting features would be beyond the basics.

These are just some ideas that I would find interesting. Feel free to pick a topic that is interesting to you, since that makes the project much more enjoyable!

Initial Design

1. Group Members

List the names of the people in your group.

2. Feature Descriptions

Describe what your program should do, listing all of the major features. For example, if I am writing a text editor, my description might be...

> *We will write a text editor that does the following:*
>
> - *Has a GUI with a flashing cursor to indicate the current position*
> - *Allows the user to click on the GUI to move the cursor*
> - *Lets the user type to add text at the cursor location*
> - *Lets the user delete text at the cursor location using "delete" or "backspace" keys*
> - *Has a "Save" button that opens a "Save" dialog box and saves the file in text format*

For clarity, you can also list features that you are not including. For example, I could say...

Our program will not support the following features one might expect from a text editor:

- *Loading a file*
- *Selecting blocks of text with the cursor*
- *Copy and paste*

3. Stub Classes and Javadoc

The "stub classes" should have all of the methods that you anticipate for each class, but each method should be empty (for void methods) or return a simple value (0, false, null, etc.). Take time to think through the return type and types and numbers of method parameters for each method. This code should compile so use dummy return values to make this happen.

The Javadoc goes beyond the UML diagram by stating what each method does, which is important information for your team to agree on. Each class and each method should have Javadoc comments.

Handin Instructions

Put the non-code information—group members, description of features, into a single PDF or Word document. Zip that PDF with the "src" folder from your project, which should contain the stub classes with javadoc. One person from your group should submit that zip file to d2l..

Completed Project

Group Component:

For the completed project, you should submit your code and schedule a demo with me (not everyone in the group needs to attend the demo).

During the demo you should be prepared to discuss your code and your project including:

1. The kinds of collections (Set, List, Map) you used. You only need two of the three kinds.
2. Design changes. For this question, consider a "change" to be adding or removing a class, or adding or removing a method from a class. Pick one of the following to answer:
   a. If you changed your design from the one you initially submitted, list three or more changes you made. (If you only made one or two changes, just list those.) For each change, briefly explain why you made the change: what makes the new design better?
   b. If you kept your original design, list two design changes that you could have made. For each, state if you think your current version or the proposed change would be better and briefly explain why.

Zip your "src" folder and any input files needed to run your program.
One person from your group should submit this to d2l.

Individual Component/Peer Review:

Each Group member should fill out and submit the peer review form found in the peer review assignment on d2l. This must include a paragraph or two detailing your contribution to the project. It does not need to be more than a paragraph, but should contain specific contributions and references to

actual code you wrote (eg. All of the Game.java class, or the gameLoop() method of the game class and all of the GUI code).

Grading Note

Since this project is more substantial than others in this class, it will carry more weight in your project average. The final grade will be a combination of the group overall score and an individual modifier. The expectation is that not every member of the group will do an equal share of the work, but that each student has significant contributions, equal to the work required for the game of life project for example. If you find yourself in a situation where you are not contributing (this is not uncommon in group projects through no fault of your own) please see me early. I would like to help manage your groups in these situations, but you need to reach out to me as early as possible, there is not much I can do if you wait until the day before the project is due. Also if your group members are not contributing please see me as well. In group projects it everyone's responsibility to make efforts to balance the work, if you are unable to do so you need to reach out to me as EARLY AS POSSIBLE.