Project 2: Semester Project

Overview

The goal of the semester project is for students to be able to experience working on a project from concept to implementation. You'll be selecting a topic of your choice and developing a program about it. This will allow you to apply the material that has been covered in class as well as prepare you for the future as a software developer.

The timeline for the project is:

- **Proposals**: due via GitHub by 11:59 PM on Friday, February 17.
- **Milestone**: due via GitHub by 11:59 PM on Friday, April 7.
- **10-minute demo**: by appointment in the professor's office starting on Monday April 25 during office hours or via WebEx.
- **Final writeup**: due via GitHub by 11:59 PM on Monday, May 8.

Project Topics

You will need to select a topic for your project, which are typically one of the following:

1. Application project. This type is the most common type of project. Choose something that is in some form interesting to you and figure out ways where you can apply you Java skills to that topic. Examples of this type of projects include file compression system, file encryption, image filtering, etc.
2. Database project. If you have knowledge and are interested, this will give you a wider range of options to combine Java with a database. Examples here are a booking system, a personal health record, etc.
3. Networking project. This is probably the most difficult topic, but it provides the most interesting problems. You can use the existing java libraries to solve problems like an instant messaging application, P2P file sharing, etc.

It is usually best if you do the project in a topic that is of interest to you. This will make it into a more interesting exercise. If you need help deciding what to do, talk to your peers or talk to the Professor during office hours.

It is not simple getting started with a topic, but you can research about things that have been done in your area of interest or you can also try Google Scholar (https://scholar.google.com). Make sure that your writeup has a good flow.

The following is the criteria that will be used to evaluate your project:

1. The technical quality of the work.
    a. Good technical choices
    b. Reasonable approach
    c. Interesting and clever algorithms

   d. UML diagrams provided for the classes
   e. Coding guidelines have been followed
2. Significance.
   a. The problem chosen is interesting or a "real" problem
   b. The work performed will be useful
3. Documentation of the work
   a. Clarity of the writeup
   b. UML diagrams
   c. Good analysis and design
   d. Etc

The amount of code you'll have to write will also be a good measure of the effort that went into the project. So, keep that in mind when choosing your topic.

Project Submission

You will submit your work under different folders under the "prj/2" folder in GitHub.

Project Proposal

10 points

Your project proposal should be submitted as a PDF document (.pdf) that includes the title of the project as well as a 300-500 description of what you plan to do. Create a sub-folder called **proposal** under the **prj/2** directory.

Milestone

20 points

The milestone report will reflect what you have accomplished to date. It must include the UML diagram for you project, and a description of what is still left to be done. The milestone will be useful for you to stay on target. You should view the milestone as a draft for your final writeup so you can use what you write in the milestone as part of your final report. The assumption is that the intended audience is already familiar with Java.

The milestone report should not be more than 3 pages long. You'll submit it as a PDF file (.pdf) in a **milestone** subdirectory under the **prj/2** folder.

The following is a guideline of what your milestone and final report should include. This is typical of any research paper:

- **Abstract**. A brief description of the work.
- **Introduction**. Describes the motivation of the work and provides an outline of the paper.
- **Detailed System Description**. Describes what the system does and how specific users interact with it. It also describes how classes interact (in UML).
- **Requirements**. Describes what the specific details of the problem that the system is addressing.
- **Literature Survey**. Describes other work that has been done to address the same or similar problems.
- **User Manual**. Briefly describes how the system should be used.

- **Conclusion**. Summarizes the goals accomplished by the system.
- **References/Bibliography**. Provides all the references that were consulted during the project. This will probably be very brief.

10-Minute Demo

30 points

You will give a 10-minute demo of your project in your professor's office or via WebEx during the last two weeks of the semester. Ten minutes is not a lot of time, so you'll need to be concise. We'll need to stick to the 10-minute schedule so we can have everyone present. Your project must be functional by the time of the demo.

Final Writeup

40 points

The final writeup should not be longer than 8 pages. It should be submitted as a PDF file in GitHub under a folder called **writeup** under **prj/2**. The code should be submitted in a folder called **code** under prj/2. If your project uses any data, put it under a folder called **data**.

Your directory structure should look like this:

```
cmpt220lastname\
   hw\
      0\
      1\
      . . .
   prj\
      0\
      2\
        proposal\
           yourproposal.pdf
        milestone\
           yourmilestone.pdf
        writeup\
           yourwriteup.pdf
        code\
           file1.java
           file2.java
           . . .
        data\
           file1.txt (it depends if you have any data)
   labs\
      1\
      2\
      . . .
```

Expectations

This is a very challenging project that will require you to manage your time correctly. It is meant to be an individual project, so I will not accept team projects.