Team Project - Project Plan and Teams
CS 420 - Fall 2013 - Ms. Katz

Project Plan Due
10am Tuesday September 10

Goals
- to get acquainted with your group
- to develop a project plan
- to get started on the team project

Overview
You're going to be working on a team project throughout this term. Team organization, collaboration, and communication will be major factors in your success. Your first task as a team is to decide on a project plan and especially a schedule.

You will use iterative enhancement to develop the project in at least three full iterations through the project life cycle. The first iteration will be somewhat informal as we will not have covered particular software development techniques. However, the later iterations will involve appropriate formalisms.

Project Plan Document
The project plan you turn in on September 10th should contain firm dates you plan to deliver specifications, high-level-design, low-level design, and the remainder of the deliverables for the first iteration. It should also contain proposed dates for some of those items for the second and third iterations. Because you are learning how to do such planning, you'll have an opportunity later in the term to revise this project plan document for those iterations.

You do not have to wait for my approval of your dates to proceed with specifications. You may start that now.

Below find a list of the items that should be included in the project plan. We will study some of these, such as the cost estimates, later. For now, include as many portions as you deem relevant. Use the creation of this document as a tool for guiding your discussion within your team of what you're developing and how.

- **purpose**
  - describe the problem to be solved

- **background information**
  - describe brief history of the project, references to related projects, any existing documents

- **processes to be used**
  - list overall development model, tools and techniques to be used

- **subsystems and planned releases**
  - show division of project into parts such as roughly what each iteration will include

- **risks and challenges**
  - describe any constraints (such as learning to use the language and development environment)

- **tasks**
  - list tasks that will be completed for each iteration

- **cost estimates**
  - give current cost estimates (not in initial plan)

- **team**
  - team name, list who is on team, skills team members have, skills team members need, who will do what, who will be the team leader during which halves of the iterations

- **schedule and milestones**
  - give deadlines for completion of tasks (using Gantt or PERT charts later)

This document, as all documents for this course, should be an editable document available to all team members. You may turn in paper or email me the documents. Be sure you keep copies for your team.

For all document in this course, I may ask you to revise them before I give them a final grade.
Other Project Documents
Remember that the documents should be helping you communicate with each other as well as with me. For each iteration, you will deliver at least the following documents.

specification
This defines what your software will do. For the early iterations, this should indicate which portion of the system you will develop for this iteration; use precise English.

high-level design
This graphical depiction of the system architecture shows which components will be developed and how they interact. In the first iteration, this should show the objects to be developed. Later iteration will have more detail and be in UML notation.

low-level design
This view of the design is more detailed and a combination of text and pictures. In the first iteration, for each class, give a short class description, list the member functions including the parameters, and list the instance variables. A diagram should show how the classes are related and especially any inheritance and dependencies. In later iterations, this will be more detailed and formal.

source code
The source code itself should be well-documented. This should not be printed as long as I can see it electronically. Each function should have a header describing its purpose, its inputs, its outputs, its primary author(s), and its modification history. The deliverable document is a short description of how to access your code and execute it. Be sure to backup your code to a different computer frequently.

test plan
This document describes what tests and testing techniques you used and what the results were. Try to break your program. In later iterations, you will have JUnit tests as well.

team project evaluation
This describes your group experiences, what you plan to change in the next iteration, and what you learned as a team. It should also include a summary of what each person was responsible for and did.

Some of these items will be delivered at the end of the iteration. The specification and designs should be delivered earlier. I'll try to grade them by the next class period. You will be graded on the quality of each document you deliver. If you revise that document within an appropriate period of time after seeing my evaluation, I will adjust the grade to reflect this additional effort. This does not mean that the new grade will replace the original. I may insist that you revise the document before I give it a final grade.

Teams
I've divided you into teams of five members. Each member is expected to contribute substantially to the team effort. See me as soon as problems arise. But realize that storming is normal; talk about it.

Each team member should be the team leader through at least half of one iteration. The team leader is responsible for setting meeting times, settling disputes, asking for clarification of requirements, and enforcing deadlines agreed upon by the team. Otherwise, the team should be a democracy (egoless teams) with each person's opinion considered. Let me know in the project plan who will be the leader of your team during each half-iteration. That person will be my contact for any team business. However, any of you can talk with me about your project. You do not need to do that as a group.

You should meet frequently as a group. I will set aside some classtime for this activity, but it won't be often. It will not be frequently enough to complete all your discussions and collaborative work. This project will take time. I've tried to keep the problem complexity manageable, but you will need to work outside class as a whole team, in pairs, and individually.

Although this is a team project, I will consider individual efforts (including facilitating team interactions, cooperation, irresponsibility, and violation of team agreements) in assigning grades. Doing the project yourself rather than as part of the team will lower your grade. Normally, everyone in the team will receive the same grade on a delivered portion of the project.

Remember that you are team members sharing a project. Treat it as a job. Be professional. Don't let your ego get caught up in the programming. Criticisms of your code are criticisms of the code, not you. Be open to new ways of viewing the problem. Respect other people's egos, opinions, and feelings. You have to work with each other.

Be responsible. Attend team meetings and be ready to work on the project. Pull your fair share and let others pull theirs. This project should take priority over elective activities; it is classwork. See me early if you have problems with team members not sharing the work.