Homework 2 — Individual Project Proposal

Due Feb 23 at 11:59pm

(75 points)

Credit & Due Dates:

- Turn in: Part 1—`individual_project.pdf` on **Tuesday**, February 23rd
- Schedule a meeting with Sreesha on February 25 or Feb 26.
- Turn in: Part 2—a revised project proposal and a link to your github with your Part 3 code on **Sunday**, Feb 28

Instructions:

You have chosen to prepare to embark on a 9/10-week long individual project journey. Be aware that receiving a good grade on this assignment is **not equivalent to approval. You must obtain explicit instructor approval** to do an individual project for the rest of this class. This approval will be based on this homework, how appropriate your project is for the requirements and timespan, and your prior submissions and work in this course.

You may work with up to one partner on this project. If you choose to work with a partner, your project should be appropriately scaled up in scope versus if you were working alone.

Project Requirements:

Your project **MUST** incorporate the following:

1) Version control (+ continuous integration, *strongly* recommended)
2) A testing framework
3) Use of an object-oriented language or paradigm
    a) This mandates the use of classes and objects. So make sure to think of how your project can be implemented in this format. If your idea cannot be implemented this way, reach out to Sreesha to see if there is a way, or come up with an alternate idea.
    b) Make use of a class diagram if possible to show the different potential classes in your project.
4) Appropriate use and implementation of inheritance:
    a) If you are using a framework/library that uses inheritance, you should use this appropriately.
    b) Regardless, you should also implement an inheritance relationship between two objects of your own.
5) Design patterns — implement 2 of the 5 following design patterns: Singleton, Flyweight, Iterator, Factory, Prototype.
    a) You may request to substitute one of these design patterns with one not listed here—to do so, describe the chosen design pattern, why it is appropriate for your project, and how you will use it.
    b) Note: when you are writing this proposal, we will not have covered these in class—there are some resources available to you in the resources.md document on github, you are welcome to come to office hours to get further information, and you are strongly encouraged to do some of your own research. Your proposal doesn't have to be perfect—that's why it's a proposal.
6) An appropriate user interface — this can be a GUI or a text UI, but the interface must make sense in the context of the project
    a) GUIs are ***strongly*** recommended.

   b) We'll be using Qt and QtCreator in class, so if you don't have any experience with GUIs and it is appropriate for your project, we recommend using these tools.

7) A project presentation during the final week of class; details given after pause week. Expect to demo your project and give a presentation describing its architecture as well as technical hurdles that you encountered as you were developing it.

If you have a *very* strong reason that you should not have to include one of these, you may request an exemption. Any exemption requests that are "it will be easier" will be denied.

Projects that propose using Unity (or Unity-like environments) this semester must be *very* explicit about how you will be meeting the above requirements, what resources you'll be developing yourself, and what resources you'll be using from other sources. We do not recommend taking this path unless you already have experience using one or both of Unity and Qt.

Part 1: Turn in a document with sections addressing the following

Expect this document to be **at least 4 pages long**, single-spaced and well-formatted. Even if you need to change some details later, provide as much information as possible now. Number your answers to each question. Please DO NOT copy and paste the questions from here into your document. Use tables sensibly to format your answers and make them easy to read where appropriate. Include pictures or diagrams as helpful.

We expect you to do a significant amount of research and planning for this proposal.

1) What is your project?

2) What technologies are you planning on using? (Programming languages, libraries, etc. Be as comprehensive as possible. Rate each technology on a scale from "not familiar" to "completely familiar")

3) What are the essential parts of the project (complete the sentence "My project won't work if it doesn't have __________" as many times as necessary). Why are these essential and do you have a backup plan if they don't work?

4) What outside resources do you require? (for example, will you need a huge amount of genome data? Do you need art for your graphics? etc.) Where will the outside resources come from? Have you already located them and know that you have access to them? If you need outside data, be sure to consider any data cleaning and other tasks in your plan.

5) Make a proposal for your architecture. If you will have a front end and a back end, how will they interact, what will your database be? What will your data model be? How will your front end work? How will you deal with your user interface? How will it integrate with your object models? What classes/objects do you plan on having? What is each in charge of and how will they interact? How will the different technologies that you are planning on using work together?
   a) This should be a comprehensive initial plan of the objects you plan on having, what they will be in charge of, and how you will transfer data/information between the different components of your application.
      i) Make a proposal for how you will use inheritance with these objects.

ii) Make a proposal for which design patterns you will use (we won't have covered them all in class yet, so do some research—we'll talk about your proposal during your meeting in the week that this document is due).

b) If you will have a database, make a proposal for how your database will be laid out (see the note about proposing a data model above).

6) Make a proposal for your GUI's user interface. How do you plan to have it laid out? Where will the user access the different features? Include all the pages/UI you assume will be required for your application. (Use pictures to show us this).

7) Detailed plan. What do you plan to accomplish each week between now and the final week of the semester? Include **ALL** of the following for each week:

   a) What will you turn in for each class due date? This should be a detailed account and should include what a user can expect to interact with/see/be able to do at that point.
   This plan can change over the course of the semester, but should not change wildly.

      i) You will be required, at each due date, to turn in a detailed plan of what you are planning to accomplish for the next due date.
      ii) The plan that you propose now for the homework 3 due date will be your plan for that due date.
      iii) Note that the amount of time between the different due dates is not the same and should be scaled appropriately.

   b) What knowledge do you think that you will need to accomplish the goals for the week?
   c) Of that knowledge, what will you need to learn?
   d) Here are the major due dates for the rest of the semester:

      i) Homework 2, Part 2&3 - Feb 28th (you should scale this work so that it is planned to take you about 2 - 3 hours to accomplish).
      ii) Homework 3 - March 12th
      iii) Homework 4 - April 6th
      iv) Homework 5, checkpoint - April 29th
      v) Homework 5, final due date - May 2nd

8) You must explicitly address how you plan to stay engaged in this course, both in terms of lecture activities and in terms of working on the regular programming exercises, in particular if you are not using c++, Qt, or other frameworks that we'll be working with in class.

Part 2: Meet with the instructors and revise your proposal

Schedule a meeting on Canvas with Sreesha on February 25th or 26th. (Times to be posted). Based on the discussion from that meeting, revise your proposal accordingly.

Part 3: Start your project

Included in this part of your assignment is getting your code set up and ready to go. You should complete all of the following:

1) Set up your github with an appropriate README, a checkpoints folder (put your revised proposal in this folder **and** turn it in on Canvas), and an appropriate .gitignore for your project. Your README should include instructions for how to install any dependencies and run your project.

2) Set up the "barebones" of your project. This might mean getting one or more libraries integrated, this might mean making some skeleton classes, etc. This should be in accordance with what you proposed in your revised proposal for Part 2.
3) You should be able to run your project and see some output (this output does not need to be complex).