Final Project

Description: The goal of this project is to the textbook you turned in for your mid-semester project. You should strive to design an accompanying program that incorporates most of the concepts you have learned across the semester. You should try to extend and improve on the mid-semester project as a starting point for this project. To complete this assignment, you will need to do the following:

- Turn in your complete textbook / how-to file
- Turn in a zip file of your program, including tests
- Turn in a reflection and documentation describing your program's intended uses
- Demonstrate your project to the class on our final exam day.

General Expectations:

- Your final submission should be neat, organized, and professional.
- Anywhere a graphic would help your explanation, feel free to use one.
- Each chapter should contain at least a page in length in text (e.g. 300 words).
- Avoid restating what you find online. Try to internalize what you are demonstrating and explain it in your own words. If using a personal metaphor or analogy helps you feel like the explanation is more your own, please use it.
- You may schedule office hours with Dr. Dym to review a chapter and receive a "sample grade" that shows what you would have earned for the work.
- Your program should be clean, organized, and run appropriately. It should be cohesive in its design, follow SOLID principles, and demonstrate some data management task.

Required Chapters

You may revise and resubmit chapters 1-6 for a higher grade if you lost points on them for your mid-semester project. Otherwise, you do not need to change them. Instead, you will need to add these new chapters, each 300 words minimum, and each with accompanying examples and supporting research.

Chapter 7 - Lists, Arrays, and Other Data Structures

Chapter 8 - Recursive Design

Chapter 9 - Equality and Basic Sorting Algorithms

Chapter 10 - Hierarchical Data Representations

Chapter 11 - Model, Controller, View Architecture

Chapter 12 - Object-Oriented Design Principles Revisited

For your final project, I will not tell you what to focus on in each chapter. It is up to you to review what was covered in each module and determine what you feel is most important to include in the minimum 300 word requirement for each chapter (and no chapter should exceed 600 words). This means you cannot cover everything. You will have to make choices about how you build a narrative around design. You will have to decide what principles of object-oriented design are most meaningful and helpful to you as a new programmer.

Required Program
The program that you turn in will, like your mid-semester project, demonstrate key concepts that the second half of your textbook chapters are discussing. However, for the final project you are required to turn in a cohesive program that accomplishes a goal. For the purposes of this project, that goal is *data management* of some kind.

What is data management? We have spent several weeks exploring ways of organizing, sorting, and accessing data in differently designed programs. Lists, arrays, maps–these are all methods for organizing data. Recursion, MVC, and SOLID design principles dictate how we should approach designing a program to manipulate that data and retrieve insights from it.

In general, your program should satisfy these requirements:

1. Work with a specifically identified data source of some type. This can be data you make up for your own project or a dataset that you pull from a source like kaggle.com. You should specify what data you are using in your code comments
2. Design a structure to handle that data: a list, an array, or a map.
3. Follow SOLID design principles for working with your data. There should be classes grouped in the program model, the program view, and the program controller.
4. Your program should generate insights about your data. Remember the example from the module where we could ask to see company employee statistics based on gender? Consider what kinds of insights you would want to draw from the data you are using.

Required Presentation
On the evening of August 14, you are required to present your program to the rest of the class (and any fellow Roux Institute faculty in attendance). You must either attend in-person or remotely. We will arrange a remote participation space for those who cannot be on campus from 6 to 9:20. You will be expected to answer questions from fellow classmates and other guests. You will be expected to explain and demonstrate your program running.

What You Need to Submit

1. A PDF file of your textbook/how-to project. If you use screenshots of your code, include a caption with a reference to what file the code is located in. When you cite external sources, include a citation to that source in a **References** section at the end of the book in APA style (https://apastyle.apa.org/style-grammar-guidelines/citations/basic-principles). If you use clip art, icons, or graphics that are not your own, then please reference where you retrieved the imagery from in either a caption or in the references section.

2. Your program in a zip file, containing all the related class files, drivers, and test files needed to demonstrate the practical programming concepts outlined in “Required Program.”
3. A separate reflection stored in a PDF labeled “FirstName_LastName_Reflection”. This reflection should be at least 200 words and address the following prompt:
    a. What was your understanding of good program design at the start of this class, and what is your understanding now?

Rubric

This project is worth 200 points, or 20% of your final grade.

1. The Textbook / How-to: 75 points
    a. Full points: You have a clean and well-formatted document. Each chapter contains insightful discussion on the requested topic. For all your research conducted, you have an appropriate APA-formatted citation. This implies that you have included at least a few references to external sources. You are not expected to know everything on your own. In fact, you need to demonstrate that you conducted research to articulate the information in this assignment.
        i. Each chapter has at least 300 words devoted to it. Including your already written chapters 1-6, chapters 7-12 should add another 2100 words minimum, meaning that your final document should be at least 4200 words. It is likely you will turn something in closer to 5000 words.
    b. Partial points: You will begin to lose points if you do not appropriately explore the topics designated in this assignment sheet. If you are unsure of what to write about, please contact Dr. Dym for a brainstorming session. You will receive partial credit if you do not include any references to external sources. You can lose points if you do not reference examples in your document. You are writing code files, so take advantage of them. Screenshot examples and paste them in as figures for reference. Include a diagram if it helps you illustrate your point. Draw and insert an explanatory cartoon if you want! In this section, the severity of points lost depends on how severe problems are across the entire document, not just in one chapter.
    c. What will not affect your grade: You will not lose points for
        i. English language syntax and grammar mistakes–unless these are so severe that they completely alter the meaning of what you wrote.
        ii. Ugly visual aids, diagrams, and graphics. We can’t all be artists.
2. The Program: 75 points
    a. Full points: You have turned in a zip file containing a program that demonstrates all the core concepts requested, including test files. It is cleanly formatted and contains comments throughout the code. You clearly identify what data is being managed in your program and how your program is designed to purposefully manipulate and draw insights from that data.

   b. Partial points: You will lose points for missing code that does not meaningfully engage with the programming prompt I have given you. You will also lose points significantly for ignoring or not adhering to the requested design principles.
   c. What will not affect your grade:
      i. English language syntax and grammar errors in your comments and other parts of the assignment, so long as the intended meaning is still clear.
      ii. Silly, strange, or boring examples for your programming files. They just need to work and be tied to a unifying theme.
3. The Presentation: 50
   a. Full points: You are present, either virtually or in-person, and you demonstrate your program successfully to at least two other individuals during the final class period on August 14. Dr. Dym will distribute a method of tracking who has presented to who at the time of the presentation day. Dr. Dym will also be asking to view people's demonstrations at this time.
   b. Partial points: You are not prepared to give a demonstration of your program, or your program does not work well. Points lost depend on how severe any errors are.
   c. No points: if you do not attend or participate in some meaningful way, you lose all points.