Summary

Throughout the semester you have been learning more about the basics of computer science and Java programming by studying, in a hands-on fashion, topics such as data and expressions, the use and creation of Java classes, conditionals, loops and data containers such as ArrayList. The final project invites you to explore, in greater detail, a real-world application of computer science. All of your project materials will be contained inside a project repository, which you will create using GitHub classroom link provided to you via Slack.

Accessing the Laboratory Assignment on GitHub

For your final project, you can work individually or in groups of two or three. If you decide to work in a group, each member of the group will be evaluated separately based on his or her contributions to the project. This evaluation will be determined largely from the feedback of the group members. If you work in a group, select a team leader, who will accept the assignment in the GitHub classroom. Then, as you have done for a previous lab assignments, the team leader can add the other members to their repository for the project. If you work by yourself you will still need to create a team that will consist of one member, you.

You should find the announcement that provides a link for the assignment in the #labs Slack channel that will be posted on Wednesday, November 6th, and accept it to see that GitHub Classroom created a new GitHub repository to access the assignment’s minimal starting materials and to store the completed version of your assignment. Now you should clone the repository, ensuring that you have selected “Clone with SSH” option.

Assignment Specifications

For your project you will develop a solution in Java for a real-world application of your choice. While you may choose your own application, your solution and implementation must satisfy the requirements outlined below. This is an opportunity for you to showcase what you learned in this course, without following a specific assignment. Of course, you can not claim the work of anyone else or your own previous work as a solution for this project, but you can certainly extend your previous work, someone else’s work provided you reference it correctly, and use your knowledge and experience as a backbone for this project. You may also download an open source project, for example from GitHub, and extend it in some way. If you decide to extend something we or someone else has already done, your extension must be significant. In that case you must also ensure you give proper attribution to previous work through the comments in your programs and description in your README, please see “HonorCode.pdf” in your in-class repository for the explanation of honor code violations in computer science. In other words, the problem that you choose should not just be a copy of one of the lab or practical assignments, or the class exercises, or the programs in the book or online with slight modifications. Your project must be extensive enough to qualify as a project, but not too extensive so that you can not finish it by the due date. Pick something realistic and preferably useful and fun, and something that you are interested in! You may use anything and everything we have learned in class, and research additional topics if needed.
Timeline: Deliverables

1. **Proposal** Deadline: by midnight on November 8th:
   Technical leaders will be available during the lab session on Thursday, November 7th, to give you feedback on your preliminary ideas and help you identify the feasibility of each idea. The instructor will be available via email and Slack during this week to provide feedback. Once you have identified an idea for your project, write a three paragraph (minimum of 150 words each) description of what you propose to do in the project Markdown file found in the “writing” directory of your project repository. Your proposal should give a brief overview of the proposed project and provide motivation for it. It should also justify how completing this project is feasible within the scope of our class and the time you have to complete it. You should provide at least two citations related to your project idea (articles or Web Sites can be cited as well). You do not need to include any specifications on how exactly you will implement your proposed project at this point, however you should broadly describe the Java programming concepts that you are likely to use. This is also the time for you to start the implementation of your project. First, create an outline of your implementation via diagrams and decide on your input and output. Then, you can translate your diagrams into the programs.

2. **Code WalkThrough** November 21, during the lab session: During the lab session, technical leaders and an instructor will do a code walkthrough on each project. By this point, you should have made a significant effort in your implementation, but it does not need to be complete. Code walkthrough is a type of software peer review, where each team/individual will go through the current implementation and technical leader and an instructor will ask questions and make comments about possible errors, violations of implementation standards, and point out other problems. A specification for the code review process will be given closer to the due date for this portion of the project assignment. You will use `progress.md` file to store the textual information and the diagram that you will use during the walkthrough. To see an example of including an image in the Markdown document, please see “Mastering Markdown” GitHub guide found at https://guides.github.com/features/mastering-markdown/.

3. **Presentation** 4-6 December, during the class, practical and the lab session: By the presentation session, you should have finished or be nearly finished with your implementation, and have run some tests. In the presentation, you should describe the overview and motivation of your project, discuss challenges, your approach to implementation, results and analysis of your implementation. Use diagrams and a few bullet points rather than long sentences and equations. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Prepare for 3–5 minute presentation. Design at least 5 slides, including a slide with the title of your project and group members’ names. Every member of the group needs to contribute to the presentation talk. At the end of the presentation give a demonstration of your project.

4. **Final Report, Final Programs and Output** Deadline: 9 December, 2019 by midnight: Your final report should highlight the key contributions of your work and consist of at least six high quality paragraphs with a minimum of 200 words in each. The report should include a description of why the chosen topic is important and discuss the implementation that you undertook. The written material should be precise, formal, appropriately formatted, grammatically correct, informative, and interesting. In summary, your report should include:
   - The motivation for your project. Why is the application you chose important/useful?
• Detailed description of the work you completed for this project. Without giving a snapshot of the code you wrote, provide technical description of what you implemented and how you implemented it. In particular include software requirements and software design for your project.

• UML diagrams or flowchart showing your project’s software design (e.g., how classes are interacting with each other). To see an example of including an image in the Markdown document, please see “Mastering Markdown” GitHub guide found at https://guides.github.com/features/mastering-markdown/.

• Description of your results. Make graphs, tables, snapshots of your output, or anything else that can help me understand your results.

• Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges and the biggest rewards.

• If you worked in a team, you should also include a paragraph that describes the team work and the contribution of each team member.

Your programs should be well documented, including typically required Javadoc comments and single line comments. All of our regularly used naming and styling conventions should be utilized.

Core Requirements

The solution for your application that you choose (your programs) must have the following:

1. Use at least three classes (three java programs) that interact with each other, that is your class(es) call methods in other class(es).

2. Have at least six methods in addition to the main method (that is, your classes combined should have at least six methods).

3. Use one or more conditional logic statements, in the form of if and/or switch statement(s).

4. Use one or more iteration constructs, expressed as while, do-while, and/or for loops.

5. Use ArrayList and/or array(s) data structures.

Submission Details

Please run your program five times with different input, unless your program produces the same output each time, and save your output as a Markdown file inside “output” directory. If your implementation is using any input files, they should be placed inside “input” directory.

Please submit all of your Java programs, any input files, an output file and all writing requirements (three ideas, proposal, progress and the report), stored as Markdown files in the writing directory, via a GitHub repository, which you created earlier.

Evaluation of your Project Assignment

Your grade for this assignment will be a function of the whether or not all parts of the assignment were submitted in a timely fashion and whether they satisfied all requirements. Please note that since this is an open ended assignment GatorGrader is not used to automatically check if your programs meet all requirements. GatorGrader and Travis are only set up to check that your writing documents meet the minimum requirements. To evaluate your programs, you should use
gradle build and gradle run commands to ensure your programs compile and run correctly and that they follow the Google styling conventions. Then, you should carefully review the core requirements outlined in this assignment sheet to make sure your programs are meeting the basic requirements.

Specifically, the points assigned to each project component are found below. In addition to studying the efficiency and effectiveness of your Java source code, the instructor will also evaluate the accuracy of both your technical writing and the comments in your source code. Instructor will also check the commit log to ensure that each team member is contributing to the GitHub project, the absence of regular commits will result in a grade reduction. Please see the instructor if you have questions about the evaluation of this laboratory assignment.

10 points Proposal

20 points Code Walk Through

20 points Presentation

50 points Final report, project programs and the output

In adherence to the Honor Code, students should complete this assignment while exclusively collaborating with the other member of their team. While it is appropriate for students in this class who are not in the same team to have high-level conversations about the assignment, it is necessary to distinguish carefully between the team that discusses the principles underlying a problem with another team and the team that produces an assignment that is identical to, or merely a variation on, the work of another team. Deliverables from one team that are nearly identical to the work of another team or a solution found online will be taken as evidence of violating Allegheny College’s Honor Code. Please be original!