Assignment 0: Introduction and Setup

The first set of assignments are meant to expose you to the different aspects of the central team project that you will be completing for the rest of the semester, but on a smaller scale. This document contains important preliminary information about not only the project itself but the course as a whole.

Academic Honesty Policy:

The full academic honesty policy is available on courseworks and it is mandatory that you read and understand the policies as they are written there you will be required to submit in the first assignment an academic honesty pledge and you will be required to maintain this file's existence in your repository throughout the duration of assignment span. Failure to include this file will result in an immediate 0 on your assignment. Some key highlights that are relevant to the individual assignments:

1. AI services of **ANY KIND** (ChatGPT, Copilot, Claudia, Gemini etc) are prohibited to be used on these assignments

2. If for whatever reason you consult other sources such as stackoverflow, you must document the instance and how they affected your submission in a file called "citations.txt"

3. Under no circumstances are you allowed to use any source as described in (2) to acquire 'cheat code' refresh yourself of this definition from Professor Jae's policies: (Academic Honesty Policies (columbia.edu))

4. There is no collaboration between yourself and other students allowed for these individual assignments, group work will come later.

Course Prerequisites:

COMS 3157: Advanced Programming (or equivalent) is a very much enforced prerequisite for this course. Most of the following should also apply to you, from experience it is okay if you are iffy on some of this but the ones in bold are must haves:

1. **Read code, even if not write, in both Java and C - most code examples used in class will be in Java or C.**
2. Write code fluently in at least one of Java, C/C++ or Javascript/Node.js, at the full-stack or backend level, not just frontend. The team project will be a server or service (NOT an app) implemented in your team's choice of Java, C++ or Javascript/Node.js. The course will not teach these or any other languages, you are required to already know Java, C/C++ or Javascript/Node.js. **Python is not allowed and students who only know Python should not take this course.**

3. Maintain your own personal codebase(s) on github or similar version control repository.

4. Search for and read (or watch) documentation and tutorials for software development tools, frameworks, APIs, libraries, etc. on your own, without instruction or assistance from the teaching staff.

5. You have read chapter 1 of the "Software Engineering at Google" book at https://abseil.io/resources/swe-book/html/ch01.html and the "coping with hitchhikers and couch potatoes on teams" article at https://ieeexplore-ieee-org.ezproxy.cul.columbia.edu/stamp/stamp.jsp?tp=&arnumber=9625655 and still want to take this class.

If you have any questions about these prerequisites please contact the Head TA at the following address: gcn2106+4156@columbia.edu

System Setup For Individual Project:

This project was developed using Java and Maven so you will need to have those installed along with an IDE, we recommend IntelliJ (if you took 3134 with Dr. B you probably already have IntelliJ installed already)

Specifically Java 17 and Maven 3.9.5 were used so be sure you have these installed prior to attempting to fork the repository.

Definition of a ‘Substantial Commit’

Throughout the individual assignments you will be required to make a minimum number of what are called ‘substantial commits’ to your repositories with each assignment. Here we will clarify what we mean by a ‘substantial commit’ and examples of commits which would count as substantial or not.

A ***substantial commit*** is a commit that you make to your github repository that indicates non-negligible progress towards completion of the assignment objectives.

For example, if an assignment asks you to implement two features then completion of each feature would represent a ‘substantial commit’. We are doing this for multiple reasons and we have to define substantial commit as some may be tempted to artificially inflate their commit count by roundabout means - adding and removing a library in two separate commits for example - the teaching staff will be checking your commit history for these kinds of anomalies and your submission will be flagged if it is determined that you attempted to inflate your commit count. Here are some examples of substantial commits and not substantial commits:

“I implemented a brand new feature in the system that accomplishes x and makes progress towards y on the assignment” - **This is a substantial commit**

“I wrote the skeleton of the code and wrote the imports” - **This is not a substantial commit** writing all of this is expected by considering how this doesn’t have to be its own commit and the progress made is quite negligible this does not qualify.

“I wrote a README for how to use my system” - **This may be a substantial commit, for the individual project’s scale no but on a larger system then yes**

The minimum number of substantial commits required by each assignment was determined by those who tested the assignment beforehand and gave their evaluation of it.