Software Design and Engineering

Lab Document

https://github.com/ocarpenter/Gradle-Weather-App
https://github.com/ocarpenter/Gradle-Weather-App.git

High Level Purpose Statement:	My goal for this lab is to transfer my Maven project from lab1 to a grade build. The Gradle project will use the dependencies for JavaFX and JSON just like the Maven project.
Experimental Design:	First, I need to transfer the application, weather, and resource files to the Gradle project setup. After refactoring the files, I will then add the JavaFX and JSON dependencies to the build.gradle. Also, inside of this file I will plugin the JavaFX library. Finally, the project should be properly setup for this lab, so we can then use (“./gradlew clean”, “./gradlew build”, and “./gradlew run”) to configure the build.
Resources Available:	References used using Gradle with Intellij JavaFX Gradle Doc JavaFX Gradle Setup Walkthrough JavaJSON Library
Time Estimate:	I think that I will spend about 4 hours on this project. For the first two hours I will transfer the application files to a Gradle project setup. Within these two hours I will also watch various videos on using Gradle for JavaFX compared to a Maven project. The next hour will be used to properly input the dependencies described in the Gradle JavaFX documentation. And the last hour will be used for writing this report, along with setting up the run config within Gradle and seeing that the project correctly runs.
Experiment Notes:	● Making a Gradle project is very easy and much like creating a Maven project. Key difference is that Gradle uses a build script and not a markup language. ● Transferring the main files and resources over to the Gradle project is pretty straightforward since it was properly set up from the previous lab. ● I am having some difficulties with the build.gradle file and following along with the documentation. It appears that the documentation and walkthrough video both use a different syntax. Hopefully if I actually download Gradle that this problem will be resolved. ● After reviewing this problem I had to switch from Kotlin to Groovy when creating the project.

Results:	The project now properly runs using a Gradle project on Intellij. I learned that Gradle is also a dependency manager and an automation tool for building and configuring projects. Furthermore, that means that Gradle and Maven could go hand and hand, the main difference I took away is that Maven uses a markup language instead of build script like Gradle. These tools make it very easy for users to download a github repo link and be able to easily run it without having to install and implement a build path to various libraries. In order to run this project, clone the repository into an Intellij. Next, open the terminal and write “./gradlew clean build” followed by “./gradlew run”.
Consequences for the Future:	In the future, I should make sure that I am using Groovy for the Gradle build instead of Kotlin in a Java project.