Opt-Out Challenge 4

Reading Assignment: How to Program Java, 10th edition

- Chapter 10 – Object-Oriented Programming: Polymorphism
- Chapter 11 – Exception Handling: A Deeper Look
- Chapter 16 – Strings, Characters and Regular Expressions

Programming Challenges

Using Android Studio, build an application that connects to a server and periodically reports the status of phone parameters such as location, orientation, and Wi-Fi attributes. Program a Java server that can support this type of connection and displays the latest message it has received.

Coordination: At this point, several members have been successful at building server-client infrastructures. This time, students who attempt this exercise must coordinate with one another to make sure that all the phone clients can connect with servers. This can be achieved by jointly specifying an interface between the clients and servers. Options for the mode of interaction between the agents include JSON objects, `serializable` objects, repeated queries from the server, etc. This will be discussed in class to formalize and define the common interface.

Before the app attempts to connect to the network, it may be advisable to check whether a network connection is available, perhaps using `getActiveNetworkInfo()` and `isConnected()`. Note that, in some recent versions of Android, applications must perform network operations on separate threads. This is to prevent undue delays from causing a poor user experience.

Code

1. Implement the server in Java and the client as an Android application.
2. Using IntelliJ IDEA, Git, and GitHub, commit your code for the server as a project labeled `Challenge4server` under `Students/<GitHubID>/`.
3. Using Android Studio, Git, and GitHub, commit your application for the client as a project labeled `Challenge4android` under `Students/<GitHubID>/`.