## Submission questionnaire

### Submitted by Shar Ahmad.

**How much time was spent developing it?**

This app took around 10-15 hours to develop. I started thinking about it on Friday and started experimenting with different frameworks and libraries. Then I decided to make a simple command line app on Saturday morning, wrote test cases to check my calculation methods and was done by complete program by Saturday night. I kept it simple however, used external JSON library for JSON parsing.

**How to use any features of the app that are not obvious?**

There are three simple unit tests that I wrote to test my methods calculations. Although they contain spoof data, that data can be manipulated and check if tests fail or pass. The login info and headers for the API call are stored in config.properties file. Upon unzipping make sure that folder structure is not changed.

**Ideas for improving your implementation to make the implementation “production**

**ready”, if any.**

There is always room for improvement. I think we live in microservice world and I’d like to make each method, each task a microservice so they are independent of each other. In addition to that the code can be refactored to be more efficient and use less space. Moreover, I’d convert methods to different classes so everything isn’t in one file.

I’d also probably use Spring or maven frameworks. Additionally, I’d like to use logger info and error so upon production release we can get more information if there are any errors. Currently the only logging I am doing is printing to the console. However, I believe it is production ready and scalable if used.

**Problems/Questions:**

The prompt asked for a summary of cloud coverage (text string) and further explains that This is the greatest amount of coverage listed if any. However, this data was not present. The prompt also didn’t specify if I had to manipulate the data and return a text string of the summary. Therefore, I am simply appending cloudCoverage to response.json.

In addition to that prompt also asked to put forecasted temperature in the forecast report, however this information was also not present anywhere in JSON responses, so this is emitted.