**Application Description:**

The simple weather application is designed to ask a user for a city and use that input to provide the user with a five-day weather forecast for that city. The weather data is gathered from an API that gets the data from openweathermap.org. The weather forecast that is returned is in a JSON file format and must be parsed/de-serialized in order for the weather app to consume the data. Once the data has been de-serialized, my program provides the user with the temperature and wind speed for the next five days.

**Design:**

The design of this program was a heavily modified version of the Web Crawler from Program 1. Many functions like sendGet(), getResponseCode() and getWebsiteString() were used for this project. Although functions were used from the last project, there were many functions that were added including a converting mechanism from Kelvin to Fahrenheit and converting the JSON data from the API into a java class object. The use of JSON was the biggest addition to this program as Libraries like GSON and Lombok were used in order to de-serialize and convert the JSON data to a Java Object to store the data. Once this was accomplished, the API could be used more effectively to look up any city the provider had.

**Usage:**

This application was designed to be used by any user that can work on a computer and run a jar executable from a terminal. Inputs provided to the program are URL encoded in order to ensure proper link formatting and to prevent errors. For example, a response of Las Vegas will not break the URL link (example below) but be converted to the link below:

*Las Vegas: link is broken up without URL encoding:*

[http://api.openweathermap.org/data/2.5/forecast?q=las vegas&APPID=b351e40979955f1a541e74a6c4da057a](http://api.openweathermap.org/data/2.5/forecast?q=las%20vegas&APPID=b351e40979955f1a541e74a6c4da057a)

*Las Vegas: link is encoded preventing errors and returning correct results:*

<http://api.openweathermap.org/data/2.5/forecast?q=las%20vegas&APPID=b351e40979955f1a541e74a6c4da057a>

Additional safeguards exist for the user to prevent the program to crash from bad input. For example, if the user inputs “ exit”, the program will search for the city but will not be able to find a city that matches that input. A 404 will be returned by the openweathermap API and the program handles this by alerting the user and continuing with the loop. At this point, the user will be prompted to provide another city or to “exit” the program. Finally, the program continues to run until the user provides the exact word “exit” to ensure the user meant to exit the program.