# Weather Simulator

Table of Contents

[Weather Simulator 1](#_Toc463955194)

[Description: 2](#_Toc463955195)

[Installation: 2](#_Toc463955196)

[Running the Simulator: 2](#_Toc463955197)

[Command line : 2](#_Toc463955198)

[Eclipse : 2](#_Toc463955199)

[Logics : 2](#_Toc463955200)

[Assumptions : 2](#_Toc463955201)

[Important Class 2](#_Toc463955202)

[Weather Simulator 2](#_Toc463955203)

[WeatherPrediction 3](#_Toc463955204)

[PredictionHelper 3](#_Toc463955205)

[Utility Package 3](#_Toc463955206)

[References 3](#_Toc463955207)

## Description:

Weather Simulator assumes earth like conditions that will take in: 1) Individual Location and calculate the Geo details or 2) Takes a date and displays the Geo details of all the Locations supported by the Simulator.

## Installation:

1. Copy the WeatherSimulator.jar into your local machine.
2. Check if Java 7 or later is installed in your machine
3. Copy the org.json.jar into your class path.

## Running the Simulator:

### Command line :

Java WeatherSimulator <usage>

Eg :

java WeatherSimulator Sydney “1980-12-12 16:05:06 ”

(or)

java WeatherSimulator “1980-12-12 16:05:06 ”

### Eclipse :

1. Load the project weather Simulator project into the workspace
2. Run as -> Run Configurations
3. Set the args with the usage as above parameters
4. Run as Java Application

## Logics :

### Assumptions :

1. The Average High and Low’s are are computed based on the Zone wise
2. Zones are classified as – Equator, Tropic of Cancer, Tropic of Capricon, Arctic and Antarctic based on the latitudes
3. Temperatures averages are calculated based on Seasons. Depending on the Zones seasons also change

## Important Class

Weather Simulator – Main class that invokes the requestALL as well as specific Location details

Request is send as the String and the response is a Array of WeatherObject. Result is sorted based on the Latitude overriding the comparator.

WeatherPrediction – It is the controller class that interface WeatherSimulator with BasicPrediction class.

PredictionHelper – It is the service class that handles the business logic

1. Calculates the Time based on the Longitude (for every 15 degree a hour changes)
2. Based on the Month and time the Season, Weather and Temprature is predicted with Random temperature deviation from the average.
3. Pressure is calculated based on the relative sea pressure, along with temperature and altitude
4. Humidity is calculated based on the Dewpoint variance

Altitude is calculated using Google API, the URL constructed with location and the API key that emits JSON response. Altitude is taken based on it. But for program sake it is stored in Hash table to avoind reusing the services over the internet.

### Utility Package

This class help to generate supporting data for the WeatherSimulator

## References

<http://hyperphysics.phy-astr.gsu.edu/hbase/kinetic/relhum.html#c4>

<http://www-das.uwyo.edu/~geerts/cwx/notes/chap16/geo_clim.html>

<https://developers.google.com/maps/documentation/elevation/start>

<http://andrew.rsmas.miami.edu/bmcnoldy/Humidity.html>

<http://www.lat34north.com/cities/CitiesLatitude.cfm>