**README FILE**

In order to run the java files:

1. Store the java files in a directory. Eg: “C:Desktop\MR\Assignment 1\src”
2. Store the input file that you want to run on these java files in the same directory.
3. Open the command prompt window.
4. Navigate to the directory where you have the java files and the input directory.eg: cd “C:\Desktop\MR\Assignment 1\src”
5. Then run the java files by writing javac followed by the filename.java. Eg: javac NoLock.java
6. Then type java filename “(I/P directory)”. Eg: java NoLock “1892.csv”
7. The java program will run and it will output the average temperature of each stationId along with the recorded average, max and minimum timings.

In order to run the program on AWS, you need to go to Eclipse and make a new project. Inside the project make a new package and then put the java file along with the input inside the package. Then, go to File->Export->Java->JAR file->(Select the project) and then click finish.

Then in AWS we put the JAR file inside our bucket along with the input file in the input directory. Then we create cluster using AWS using the documentation of EMR in HW1 and then we get the output folder in the S3 folder.

To run the standalone program you create a maven project and inside the pom.xml file, we insert this:

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-shade-plugin</artifactId>

<version>2.4.3</version>

<executions>

<execution>

<phase>package</phase>

<goals>

<goal>shade</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.apache.hadoop</groupId>

<artifactId>hadoop-mapreduce</artifactId>

<version>2.7.3</version>

<type>pom</type>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.apache.hadoop</groupId>

<artifactId>hadoop-common</artifactId>

<version>2.7.3</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.apache.hadoop</groupId>

<artifactId>hadoop-mapreduce-client-core</artifactId>

<version>2.7.3</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.apache.hadoop</groupId>

<artifactId>hadoop-mapreduce-client-jobclient</artifactId>

<version>2.7.3</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>jdk.tools</groupId>

<artifactId>jdk.tools</artifactId>

<version>1.7.0\_05</version>

<scope>system</scope>

<systemPath>${JAVA\_HOME}/lib/tools.jar</systemPath>

</dependency>

</dependencies>

Inside the <project></project> tags. Then we do Maven clean and build on the project to sort out any dependencies.

We also give arguments to the Program if needed by going to properties->Java Application. In the arguments tab we write the argument that we need to pass it into the function.

Then we run the program.