```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import java.io.IOException;

public class MaxTemperature {

    // Mapper Class
    public static class TemperatureMapper extends Mapper<Object, Text, Text, IntWritable> {
        private Text year = new Text();
        private IntWritable temperature = new IntWritable();

        public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
            String[] line = value.toString().split(" "); // Split line by space
            if (line.length == 2) {  // Ensure valid format
                year.set(line[0]);  // First value is Year
                temperature.set(Integer.parseInt(line[1]));  // Second value is Temperature
                context.write(year, temperature);  // Emit (year, temperature)
            }
        }
    }

    // Reducer Class
    public static class TemperatureReducer extends Reducer<Text, IntWritable, Text, IntWritable>
{
        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {
            int maxTemp = Integer.MIN_VALUE;  // Start with lowest possible value

            for (IntWritable val : values) {  // Loop through all temperatures for this year
                maxTemp = Math.max(maxTemp, val.get());  // Find max temperature
            }

            context.write(key, new IntWritable(maxTemp));  // Emit (year, max temperature)
        }
    }
    // Driver Class
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Max Temperature");

        job.setJarByClass(MaxTemperature.class);
        job.setMapperClass(TemperatureMapper.class);
        job.setReducerClass(TemperatureReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));  // Input folder
        FileOutputFormat.setOutputPath(job, new Path(args[1])); // Output folder

        System.exit(job.waitForCompletion(true) ? 0 : 1);  // Run the job
    }
}
```

```
Execution Steps 1. mkdir usn_prog2
2. cd usn_prog2
 3. gedit MaxTemperature.java
 4. start-all.sh
 5. Jps
 6. export HADOOP_CLASSPATH=$(hadoop classpath)
7. mkdir Input
 8. cd Input
 9. gedit temperature_data.txt 2001 32 2001 35 2002 30 2002 28 2003 33 2003 36
 10. cd . .
 11. hadoop fs -mkdir /maxtemp_usn
 12. hadoop fs -mkdir /maxtemp_usn/Input
 13. hadoop fs -put ./Input/temperature_data.txt/ /maxtemp_usn/Input
14. export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
15. export PATH=$JAVA_HOME/bin:$PATH
16. javac -classpath $(hadoop classpath) -d . MaxTemperature.java
 17. jar -cvf maxtemp.jar -C . .
 18. hadoop jar maxtemp.jar MaxTemperature /maxtemp_usn/Input /maxtemp_usn/Input/output
19. hadoop fs -cat /maxtemp_usn/Input/output/part-r-00000
```