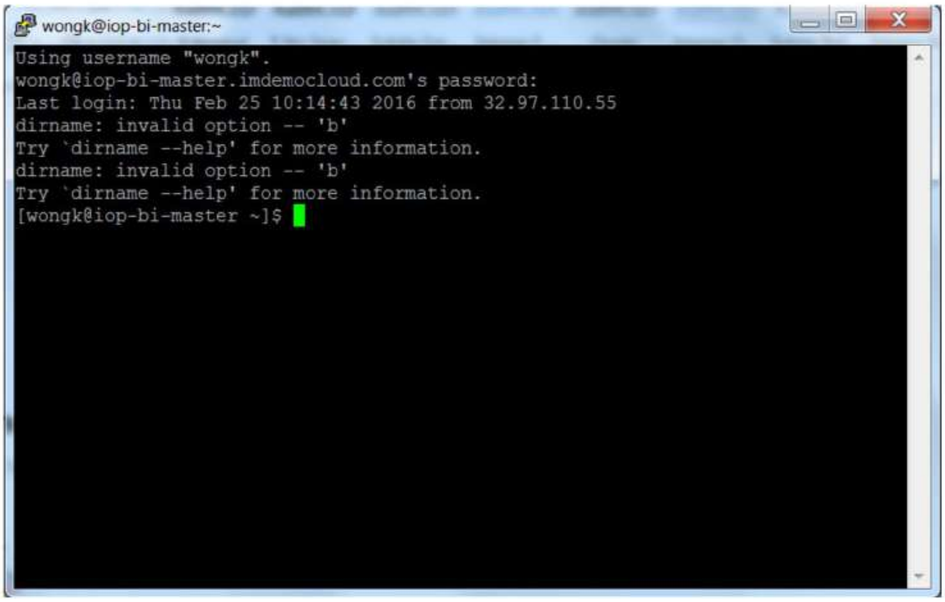
1.1 Load and examine the sample data

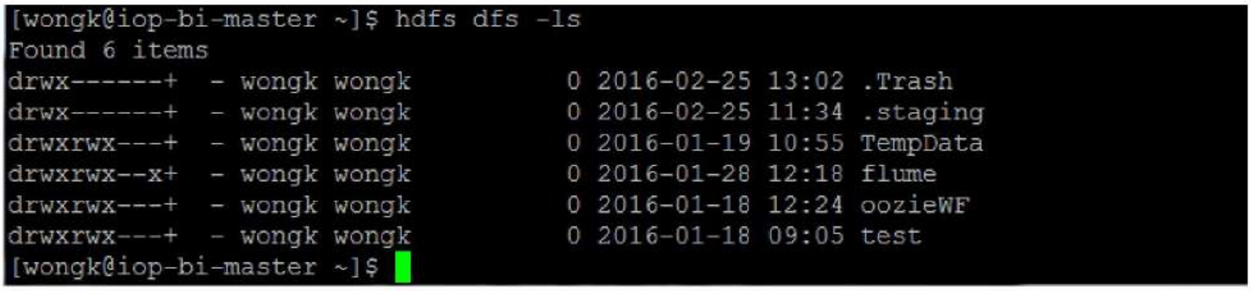
__1. You should start with a window like this after you log in



```
Using username "wongk".
wongk@iop-bi-master.imdemocloud.com's password:
Last login: Thu Feb 25 10:14:43 2016 from 32.97.110.55
dirname: invalid option -- 'b'
Try `dirname --help' for more information.
dirname: invalid option -- 'b'
Try `dirname --help' for more information.
[wongk@iop-bi-master ~]$
```

__2. In the PuTTY window, examine the directory structure of the hadoop file system of your directory. Type:

```
hdfs dfs -ls
```

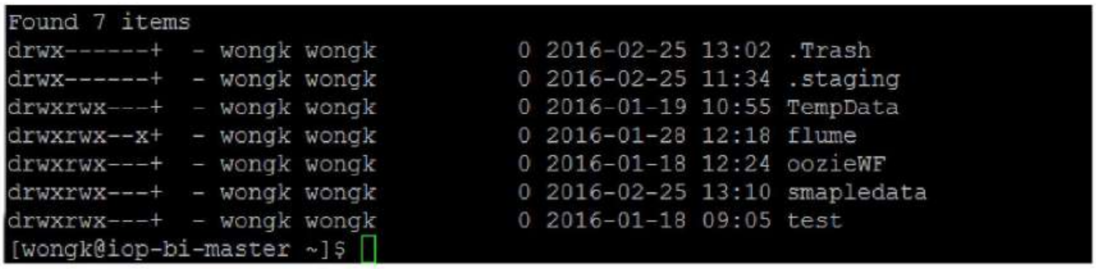


```
[wongk@iop-bi-master ~]$ hdfs dfs -ls
Found 6 items
drwx------+  - wongk wongk          0 2016-02-25 13:02 .Trash
drwx------+  - wongk wongk          0 2016-02-25 11:34 .staging
drwxrwx---+  - wongk wongk          0 2016-01-19 10:55 TempData
drwxrwx--x+  - wongk wongk          0 2016-01-28 12:18 flume
drwxrwx---+  - wongk wongk          0 2016-01-18 12:24 oozieWF
drwxrwx---+  - wongk wongk          0 2016-01-18 09:05 test
[wongk@iop-bi-master ~]$
```

You want to make a new directory to store the sample data files that you will use for this exercise.

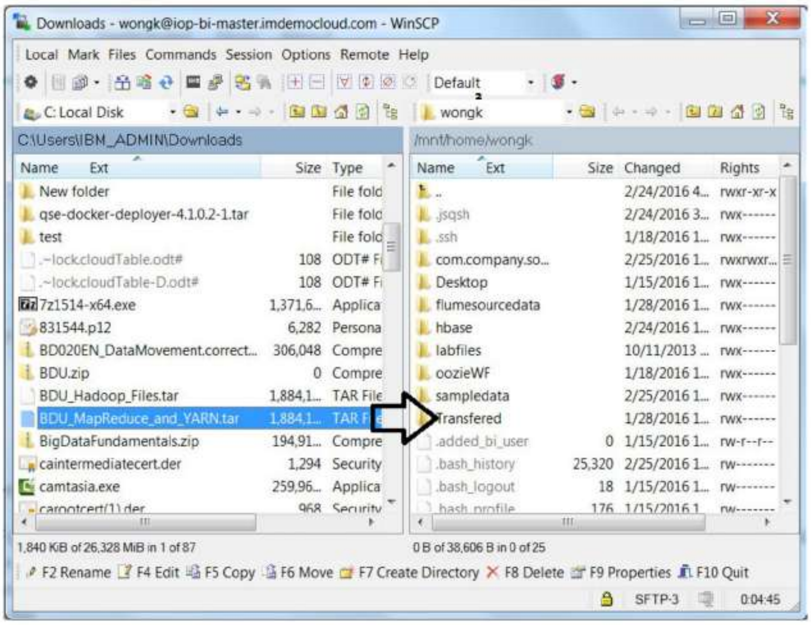
__3. Make a new directory called sampledata/, and verify that it was created. Type:

```
hdfs dfs -mkdir sampledata

hdfs dfs -ls

hdfs dfs -chmod 777 sampledata
```



```
Found 7 items
drwx------+  - wongk wongk          0 2016-02-25 13:02 .Trash
drwx------+  - wongk wongk          0 2016-02-25 11:34 .staging
drwxrwx---+  - wongk wongk          0 2016-01-19 10:55 TempData
drwxrwx--x+  - wongk wongk          0 2016-01-28 12:18 flume
drwxrwx---+  - wongk wongk          0 2016-01-18 12:24 oozieWF
drwxrwx---+  - wongk wongk          0 2016-02-25 13:10 smapledata
drwxrwx---+  - wongk wongk          0 2016-01-18 09:05 test
[wongk@iop-bi-master ~]$ 
```

__4. Now, to move **BDU_MapReduce_and_YARN.tar** file into cloud. Open up WinSCP and transfer the file to your home directory. Just drag the file from the left side to the home directory on the right side.


Downloads - wongk@iop-bi-master.imdemocloud.com - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
C: Local Disk
wongk

C:\Users\IBM_ADMIN\Downloads

Name Ext	Size	Type
New folder		File fold
qse-docker-deployer-4.1.0.2-1.tar		File fold
test		File fold
.~lock.cloudTable.odt#	108	ODT# Fi
.~lock.cloudTable-D.odt#	108	ODT# Fi
7z1514-x64.exe	1,371,6...	Applica
831544.p12	6,282	Persona
BD020EN_DataMovement.correct...	306,048	Compre
BDU.zip	0	Compre
BDU_Hadoop_Files.tar	1,884,1...	TAR File
BDU_MapReduce_and_YARN.tar	1,884,1...	TAR F
BigDataFundamentals.zip	194,91...	Compre
caintermediatecert.der	1,294	Security
camtasia.exe	259,96...	Applica
carootcert(1).der	968	Security

1,840 KiB of 26,328 MiB in 1 of 87

/mnt/home/wongk

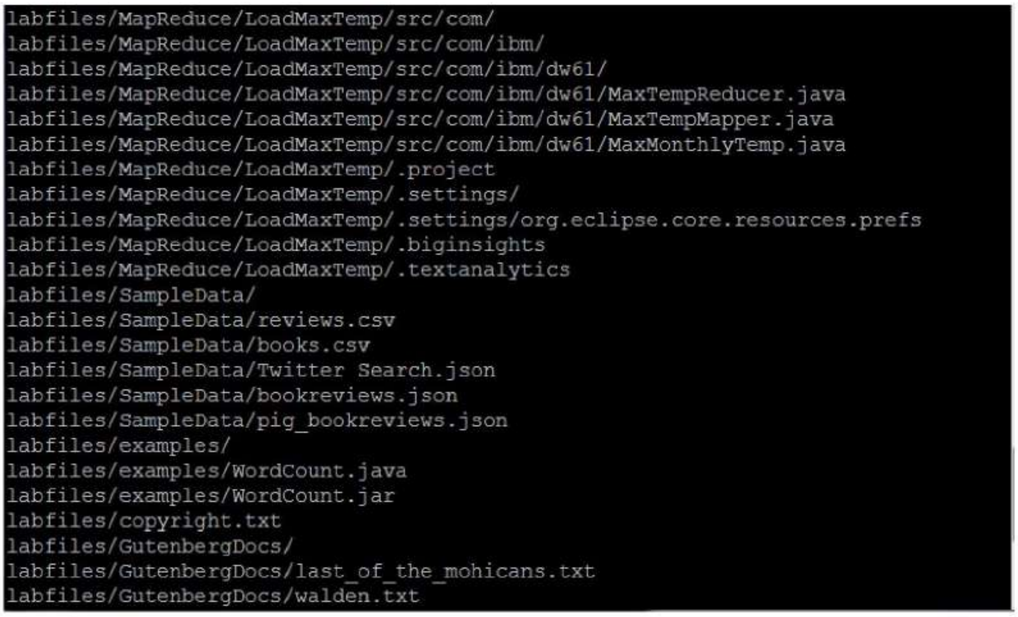
Name Ext	Size	Changed	Rights
..		2/24/2016 4...	rwxr-xr-x
.jsqsh		2/24/2016 3...	rwx------
.ssh		1/18/2016 1...	rwx------
com.company.so...		2/25/2016 1...	rwxrwxr...
Desktop		1/15/2016 1...	rwx------
flumesourcedata		1/28/2016 1...	rwx------
hbase		2/24/2016 1...	rwx------
labfiles		10/11/2013 ...	rwx------
oozieWF		1/18/2016 1...	rwx------
sampledata		2/25/2016 1...	rwx------
Transfered		1/28/2016 1...	rwx------
.added_bi_user	0	1/15/2016 1...	rw-r--r--
.bash_history	25,320	2/25/2016 1...	rw-------
.bash_logout	18	1/15/2016 1...	rw-------
.bash_profile	176	1/15/2016 1	rw-------

0 B of 38,606 B in 0 of 25

F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-3
0:04:45

__5. Then extract the file.

```
cd ~

tar -xvf BDU_Hadoop_Files.tar
```

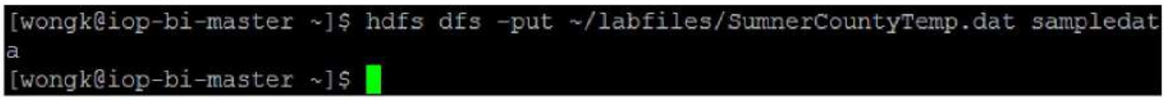


```
labfiles/MapReduce/LoadMaxTemp/src/com/
labfiles/MapReduce/LoadMaxTemp/src/com/ibm/
labfiles/MapReduce/LoadMaxTemp/src/com/ibm/dw61/
labfiles/MapReduce/LoadMaxTemp/src/com/ibm/dw61/MaxTempReducer.java
labfiles/MapReduce/LoadMaxTemp/src/com/ibm/dw61/MaxTempMapper.java
labfiles/MapReduce/LoadMaxTemp/src/com/ibm/dw61/MaxMonthlyTemp.java
labfiles/MapReduce/LoadMaxTemp/.project
labfiles/MapReduce/LoadMaxTemp/.settings/
labfiles/MapReduce/LoadMaxTemp/.settings/org.eclipse.core.resources.prefs
labfiles/MapReduce/LoadMaxTemp/.biginsights
labfiles/MapReduce/LoadMaxTemp/.textanalytics
labfiles/SampleData/
labfiles/SampleData/reviews.csv
labfiles/SampleData/books.csv
labfiles/SampleData/Twitter Search.json
labfiles/SampleData/bookreviews.json
labfiles/SampleData/pig_bookreviews.json
labfiles/examples/
labfiles/examples/WordCount.java
labfiles/examples/WordCount.jar
labfiles/copyright.txt
labfiles/GutenbergDocs/
labfiles/GutenbergDocs/last_of_the_mohicans.txt
labfiles/GutenbergDocs/walden.txt
```

Now your window should show something similar when it's done extracting the files.

__6. Now that the files are extracted, we will upload the temperature data from the local file system to the HDFS using the following commands:

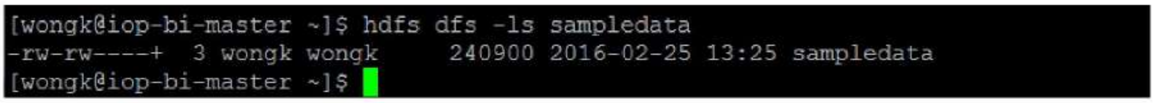
```
hdfs dfs -put ~/labfiles/SumnerCountyTemp.dat sampledata
```



```
[wongk@iop-bi-master ~]$ hdfs dfs -put ~/labfiles/SumnerCountyTemp.dat sampledat
a
[wongk@iop-bi-master ~]$ 
```

__7. Test to see that the file was uploaded correctly by typing the following command:

```
hdfs dfs -ls sampledata
```

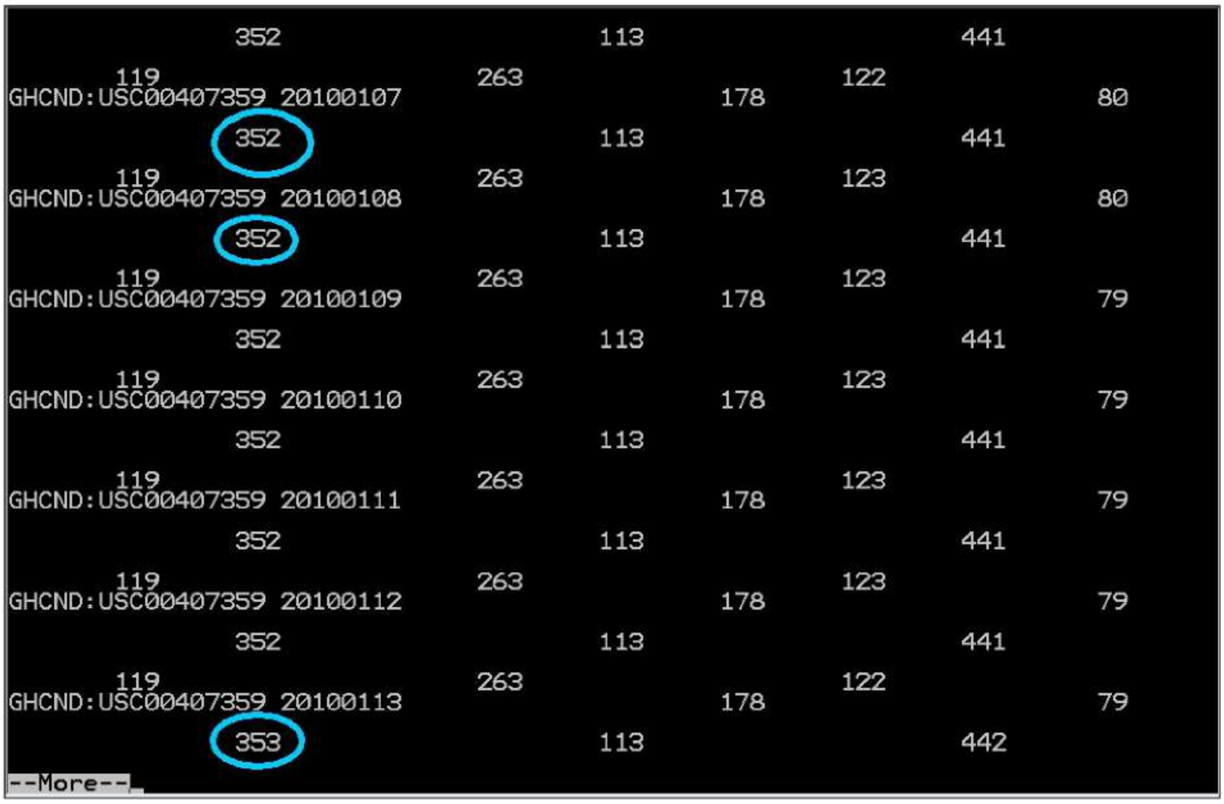


```
[wongk@iop-bi-master ~]$ hdfs dfs -ls sampledata
-rw-rw----+  3 wongk wongk     240900 2016-02-25 13:25 sampledata
[wongk@iop-bi-master ~]$ 
```

Notice that your SumnerCountyTemp.dat files was uploaded correctly.

__8. You can view this data by executing the following command:

```
hdfs dfs -cat sampledata/SumnerCountyTemp.dat | more
```



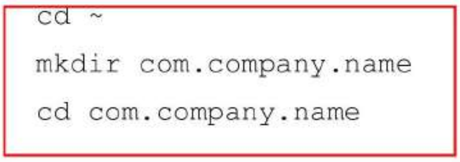
```
               352                     113                     441
       119                     263                     122
GHCND:USC00407359 20100107                     178                      80
               352                     113                     441
       119                     263                     123
GHCND:USC00407359 20100108                     178                      80
               352                     113                     441
       119                     263                     123
GHCND:USC00407359 20100109                     178                      79
               352                     113                     441
       119                     263                     123
GHCND:USC00407359 20100110                     178                      79
               352                     113                     441
       119                     263                     123
GHCND:USC00407359 20100111                     178                      79
               352                     113                     441
       119                     263                     123
GHCND:USC00407359 20100112                     178                      79
               352                     113                     441
       119                     263                     122
GHCND:USC00407359 20100113                     178                      79
               353                     113                     442
--More--
```

The values in the 95th column (354, 353, 353,353, 352, ...) are the average daily temperatures. They are the result of multiplying the actual average temperature value times 10. (Incidentally, that way you don’t have to worry about working with decimal points.)

__9. Press the spacebar a few times to scroll through the data and observe the temperature patterns by date. When you are satisfied, press **Ctrl+c** to break out of the piped output.

1.2 Start your Java project

__1. Create a directory to hold the three Java files that you will be making and make it accessible. The directory will be used to hold program artifacts and to separate it from the other things in the file system.



```
cd ~
mkdir com.company.name
cd com.company.name
```

1.3 Create the Java file for the mapper class

__1. Create a new Java file, **MaxTempMapper.java**:

```
vi MaxTempMapper.java
```

There is a standard set of imports that will also be used for the other two Java files that you create.

The data type for the input key to the mapper will be *LongWritable.* The data itself will be of type *Text.* The output key from the mapper will be of type *Text.* And the data from the mapper (the temperature) will be of type *IntWritable.*

You need a public class with name **MaxTempMapper**. For this class,

___a. You will need to import *java.io.IOException.*

___b. Extend **Mapper<LongWritable, Text, Text, IntWritable>**

___c. Define a public class called map.

___d. Your code should look like the following:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTempMapper extends
     Mapper<LongWritable, Text, Text, IntWritable> {

   @Override
   public void map(LongWritable key, Text value, Context context)
           throws IOException, InterruptedException {

   }
}
```

In the next section you will define the map method.

Note: *You can also create the .java file in notepad and transfer it via WinSCP*

1.4 Complete the mapper

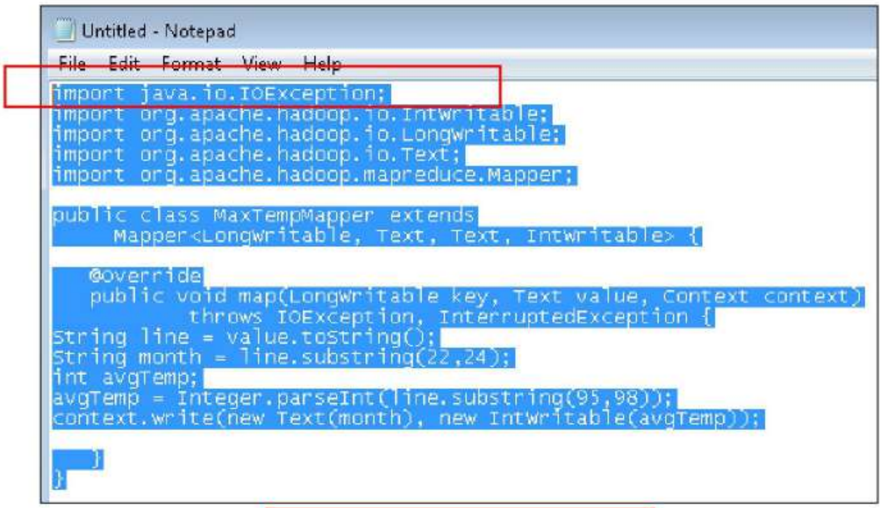
Your program will read in a line of data as a string so that you can do string manipulation. You will want to extract the month and average temperature for each record.

The month begins at the 22th character of the record (zero offset) and the average temperature begins at the 95th character. (Remember that the average temperature value is three digits, with implied one decimal place).

__1. In the *map* method, add the following code (or whatever code you think is required):

```
String line = value.toString();
String month = line.substring(22,24);
int avgTemp;
avgTemp = Integer.parseInt(line.substring(95,98));
context.write(new Text(month), new IntWritable(avgTemp));
```

__2. From this document, you may wish to copy the entire content of the file into Windows Notepad where you can insert the code for the map method. Then transfer the java file (for ex: *MaxTempMapper.java*) to your com.company.name folder.



```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTempMapper extends
    Mapper<LongWritable, Text, Text, IntWritable> {

   @Override
   public void map(LongWritable key, Text value, Context context)
           throws IOException, InterruptedException {
String line = value.toString();
String month = line.substring(22,24);
int avgTemp;
avgTemp = Integer.parseInt(line.substring(95,98));
context.write(new Text(month), new IntWritable(avgTemp));

     }
}
```

If you are using vi, press **Esc**, and then type **:wq** to write and exit the vi editor and write your file

or

Press Esc then hit Shift + z twice.

1.5 Create the reducer class

__1. Create a new Java file, **MaxTempMapper.java**:

```
vi MaxTempReducer.java
```

You need a public class with name **MaxTempReducer** and the data type for the input key to the reducer will be *Text.* The data itself will be of type *IntWritable.* The output key from the reducer will be of type *Text.* And the data from the reducer will be of type *IntWritable.*

For your class,

___a. You will need to import **java.io.IOException**

___b. Extend **Reducer<Text, LongWritable, Text, IntWritable>**

___c. Define a public class called ***reduce.***

___d. Your code should look like the following:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MaxTempReducer extends
     Reducer<Text, IntWritable, Text, IntWritable> {
   @Override
   public void reduce(Text key, Iterable<IntWritable> values, Context
context)
              throws IOException, InterruptedException {
   }
}
```

1.6 Complete the reducer

__1. For the reducer, you want to iterate through all values for a given key. For each value found, check to see if it is higher than any of the other values.

Add the following code (or your variation) to the *reduce* method.

```
int maxTemp = Integer.MIN_VALUE;
for (IntWritable value: values) {
maxTemp = Math.max(maxTemp, value.get());
}
context.write(key, new IntWritable(maxTemp));
```

__2. Assemble your file in the vi editor, notepad or any way you choose, and remember to save your work.

1.7 Create the driver

__1. Create a new Java file, **MaxMonthTemp.java**:

```
vi MaxMonthTemp.java
```

You need a public class with name **MaxMonthTemp** and the standard set of import files.

The *GenericOptionsParser()* will extract any input parameters that are not system parameters and place them in an array. In your case, two parameters will be passed to your application. The first parameter is the input file. The second parameter is the output directory. (This directory must not exist or your MapReduce application will fail.)

Your code should look like this:

```
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.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class MaxMonthTemp {

   public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();

      String[] programArgs =
           new GenericOptionsParser(conf, args).getRemainingArgs();
      if (programArgs.length != 2) {
         System.err.println("Usage: MaxTemp <in> <out>");
         System.exit(2);
      }
      Job job = Job.getInstance(conf, "Monthly Max Temp");
      job.setJarByClass(MaxMonthTemp.class);
      job.setMapperClass(MaxTempMapper.class);
      job.setReducerClass(MaxTempReducer.class);

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

      FileInputFormat.addInputPath(job, new Path(programArgs[0]));

      FileOutputFormat.setOutputPath(job, new Path(programArgs[1]));

      // Submit the job and wait for it to finish.
      System.exit(job.waitForCompletion(true) ? 0 : 1);

   }
}
```

"

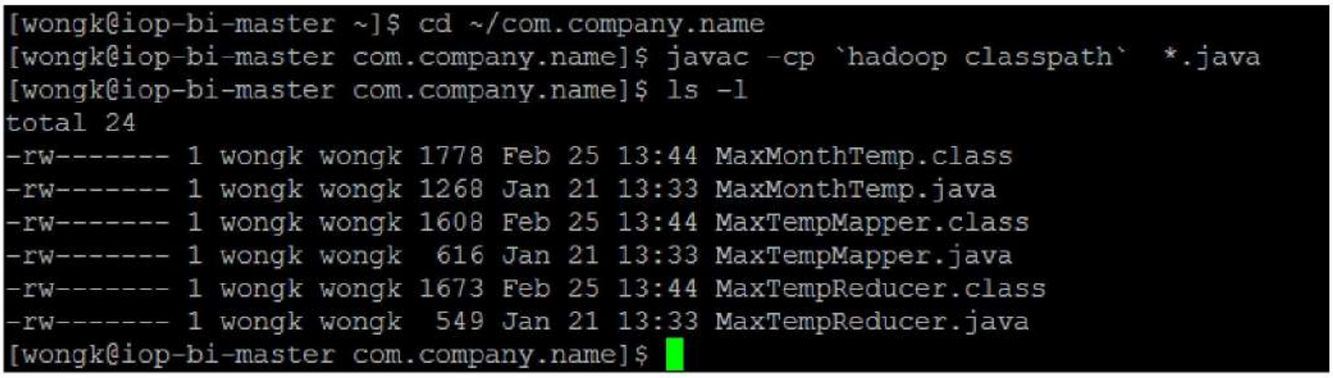
__2. Assemble your file in the vi editor any way you choose, and remember to save your work.

1.8 Compile your Java files & create the JAR file

__1. Compile all three Java files with one statement as the root user and then list your directory to see that you now have three Java source files and three Java class files. Type:

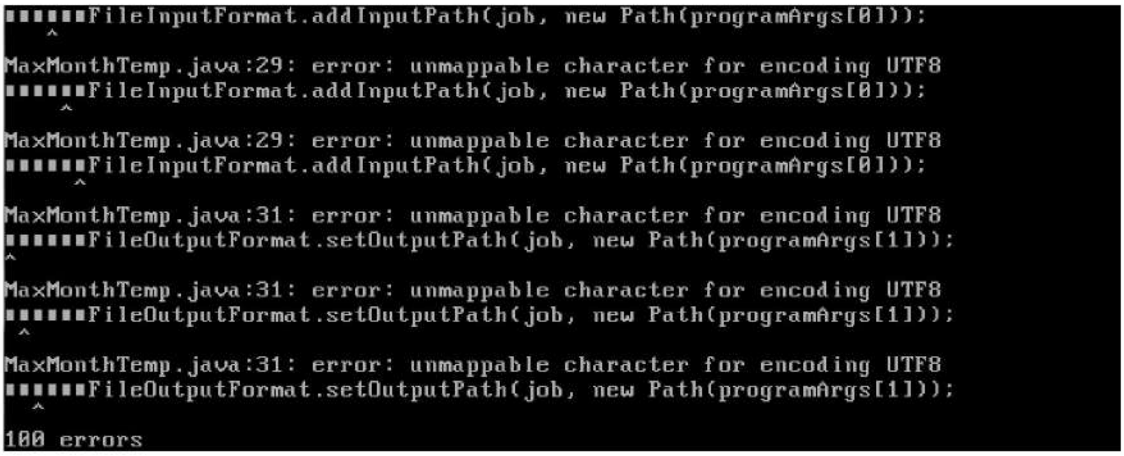
```
cd ~/com.company.name
javac -cp `hadoop classpath`  *.java
ls -l
```

Note that the quotes here are ***back-quotes*** (the key to top left corner of your keyboard, to the left of the one-key [“1”}). The command *hadoop classpath* is executed to list the required classpath information needed by the compiler; the result of this is passed to javac with the classpath option (-cp).



```
[wongk@iop-bi-master ~]$ cd ~/com.company.name
[wongk@iop-bi-master com.company.name]$ javac -cp `hadoop classpath`  *.java
[wongk@iop-bi-master com.company.name]$ ls -l
total 24
-rw------- 1 wongk wongk 1778 Feb 25 13:44 MaxMonthTemp.class
-rw------- 1 wongk wongk 1268 Jan 21 13:33 MaxMonthTemp.java
-rw------- 1 wongk wongk 1608 Feb 25 13:44 MaxTempMapper.class
-rw------- 1 wongk wongk  616 Jan 21 13:33 MaxTempMapper.java
-rw------- 1 wongk wongk 1673 Feb 25 13:44 MaxTempReducer.class
-rw------- 1 wongk wongk  549 Jan 21 13:33 MaxTempReducer.java
[wongk@iop-bi-master com.company.name]$ 
```

Note: If you get this error:

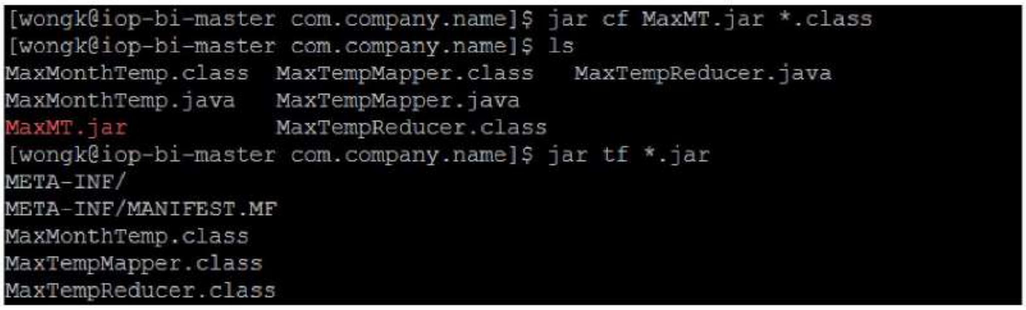


```
■■■■■■FileInputFormat.addInputPath(job, new Path(programArgs[0]));
   ^
MaxMonthTemp.java:29: error: unmappable character for encoding UTF8
■■■■■■FileInputFormat.addInputPath(job, new Path(programArgs[0]));
    ^
MaxMonthTemp.java:29: error: unmappable character for encoding UTF8
■■■■■■FileInputFormat.addInputPath(job, new Path(programArgs[0]));
     ^
MaxMonthTemp.java:31: error: unmappable character for encoding UTF8
■■■■■■FileOutputFormat.setOutputPath(job, new Path(programArgs[1]));
^
MaxMonthTemp.java:31: error: unmappable character for encoding UTF8
■■■■■■FileOutputFormat.setOutputPath(job, new Path(programArgs[1]));
 ^
MaxMonthTemp.java:31: error: unmappable character for encoding UTF8
■■■■■■FileOutputFormat.setOutputPath(job, new Path(programArgs[1]));
  ^
100 errors
```

This meant that you copied & pasted directly from this document while using a program that converted some of the whitespaces into a different format. To correct this, open notepad from windows and/or vi and delete the characters that give you an error and replace it with a typed character of itself. (Ex, replace the above boxes, which show up as spaces on your editor with spaces typed from your keyboard).

__2. Create a Java Archive File (jar cf, where c = create, f = file) from the three class files. Then list the manifest (jar tf) of the archive file:

```
jar cf MaxMT.jar *.class

ls

jar tf *.jar
```



```
[wongk@iop-bi-master com.company.name]$ jar cf MaxMT.jar *.class
[wongk@iop-bi-master com.company.name]$ ls
MaxMonthTemp.class  MaxTempMapper.class   MaxTempReducer.java
MaxMonthTemp.java   MaxTempMapper.java
MaxMT.jar           MaxTempReducer.class
[wongk@iop-bi-master com.company.name]$ jar tf *.jar
META-INF/
META-INF/MANIFEST.MF
MaxMonthTemp.class
MaxTempMapper.class
MaxTempReducer.class
```

__3. The Java Archive File was created in the directory where the .java and .class files reside. But when we use Hadoop MapReduce to run the jar, Hadoop does not like to have the .class files in the same directory. Therefore you want to move the file to the parent directory, where we will run it in the next step:

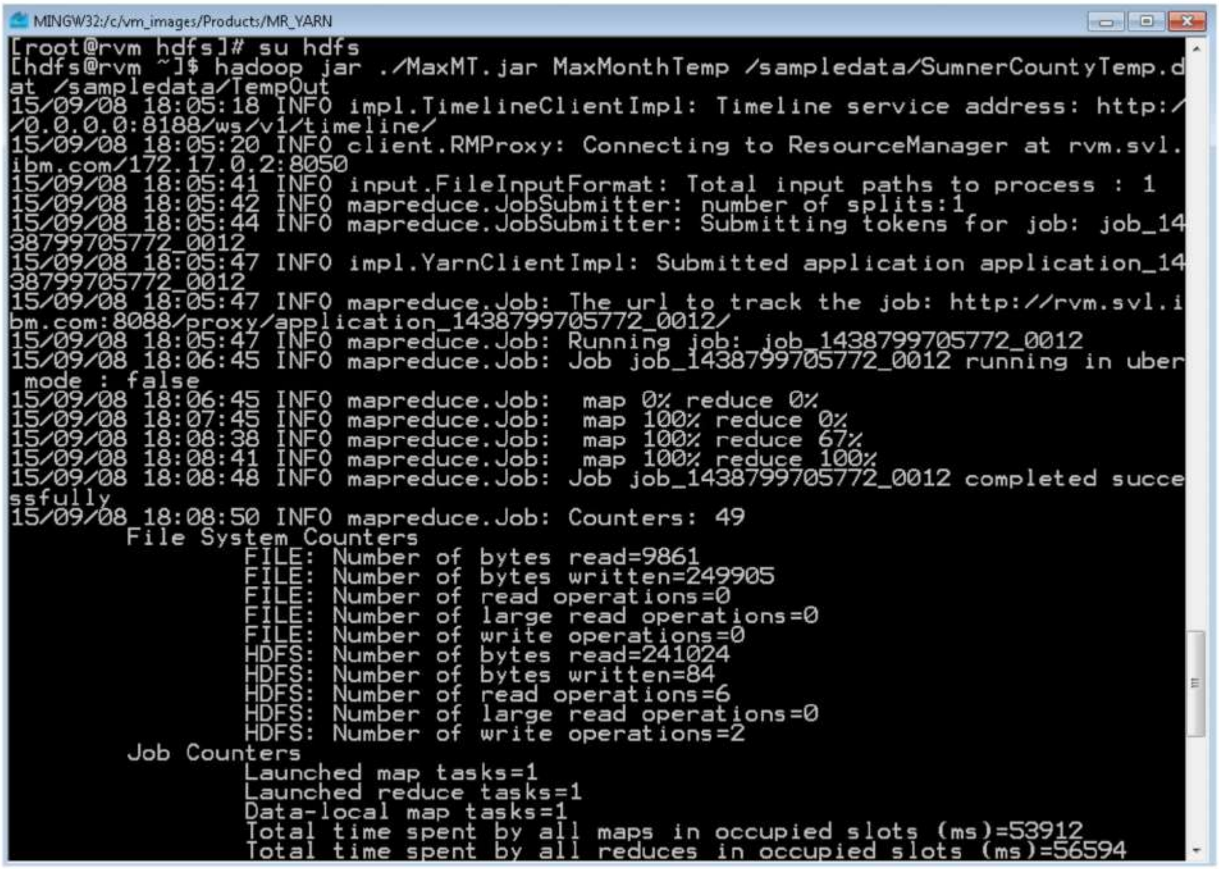
```
cp *.jar ..

cd ..
```

1.9 Run the JAR file

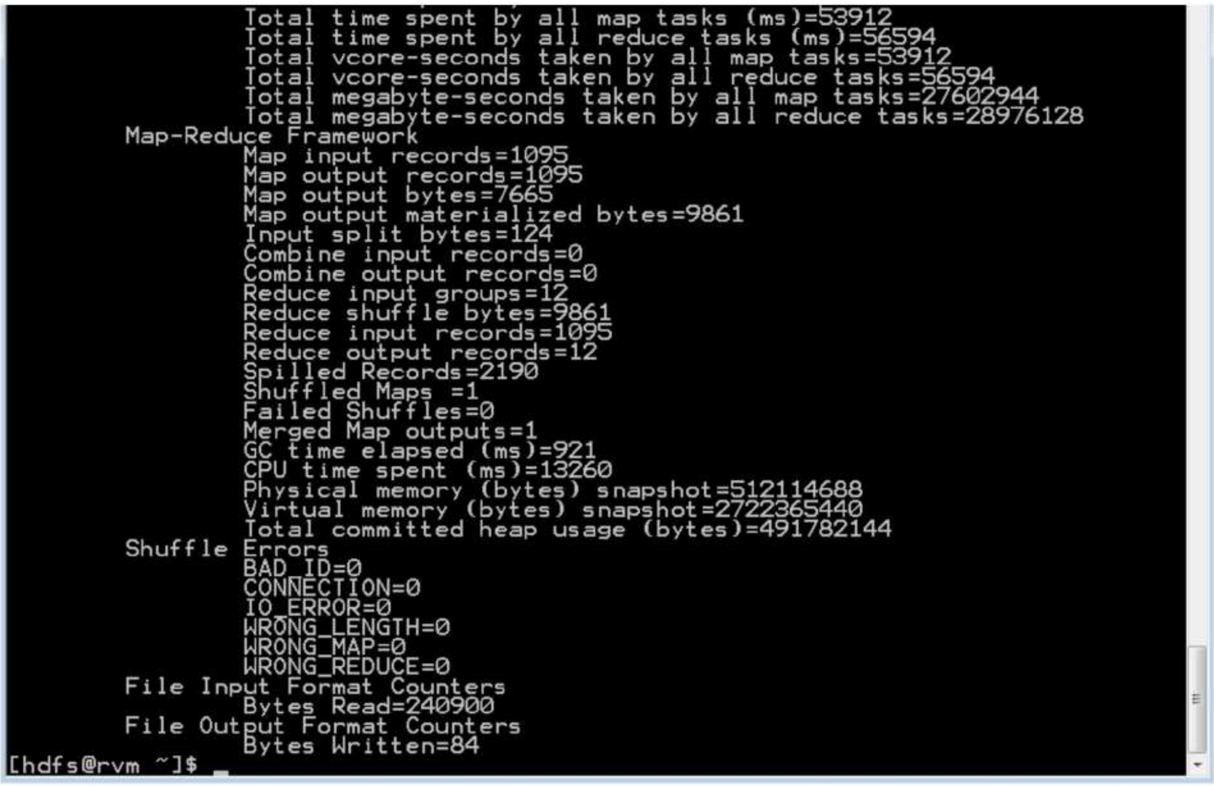
__1. Run the application. Type:

```
hadoop jar ./MaxMT.jar MaxMonthTemp sampledata/SumnerCountyTemp.dat
sampledata/TempOut
```

You will see the following output in that terminal window (but your results will be slightly different, of course):



```
[root@rvm hdfs]# su hdfs
[hdfs@rvm ~]$ hadoop jar ./MaxMT.jar MaxMonthTemp /sampledata/SumnerCountyTemp.d
at /sampledata/TempOut
15/09/08 18:05:18 INFO impl.TimelineClientImpl: Timeline service address: http:/
/0.0.0.0:8188/ws/v1/timeline/
15/09/08 18:05:20 INFO client.RMProxy: Connecting to ResourceManager at rvm.svl.
ibm.com/172.17.0.2:8050
15/09/08 18:05:41 INFO input.FileInputFormat: Total input paths to process : 1
15/09/08 18:05:42 INFO mapreduce.JobSubmitter: number of splits:1
15/09/08 18:05:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14
38799705772_0012
15/09/08 18:05:47 INFO impl.YarnClientImpl: Submitted application application_14
38799705772_0012
15/09/08 18:05:47 INFO mapreduce.Job: The url to track the job: http://rvm.svl.i
bm.com:8088/proxy/application_1438799705772_0012/
15/09/08 18:05:47 INFO mapreduce.Job: Running job: job_1438799705772_0012
15/09/08 18:06:45 INFO mapreduce.Job: Job job_1438799705772_0012 running in uber
 mode : false
15/09/08 18:06:45 INFO mapreduce.Job:  map 0% reduce 0%
15/09/08 18:07:45 INFO mapreduce.Job:  map 100% reduce 0%
15/09/08 18:08:38 INFO mapreduce.Job:  map 100% reduce 67%
15/09/08 18:08:41 INFO mapreduce.Job:  map 100% reduce 100%
15/09/08 18:08:48 INFO mapreduce.Job: Job job_1438799705772_0012 completed succe
ssfully
15/09/08 18:08:50 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=9861
                FILE: Number of bytes written=249905
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=241024
                HDFS: Number of bytes written=84
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=53912
                Total time spent by all reduces in occupied slots (ms)=56594
```



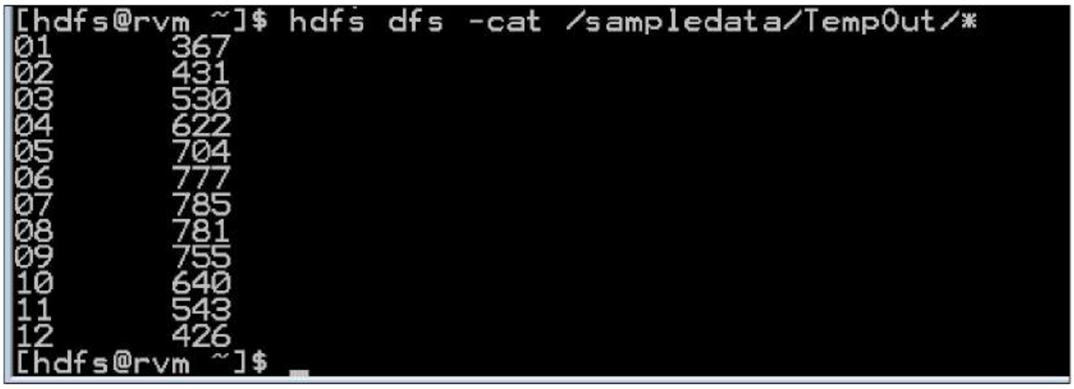
```
                Total time spent by all map tasks (ms)=53912
                Total time spent by all reduce tasks (ms)=56594
                Total vcore-seconds taken by all map tasks=53912
                Total vcore-seconds taken by all reduce tasks=56594
                Total megabyte-seconds taken by all map tasks=27602944
                Total megabyte-seconds taken by all reduce tasks=28976128
        Map-Reduce Framework
                Map input records=1095
                Map output records=1095
                Map output bytes=7665
                Map output materialized bytes=9861
                Input split bytes=124
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=9861
                Reduce input records=1095
                Reduce output records=12
                Spilled Records=2190
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=921
                CPU time spent (ms)=13260
                Physical memory (bytes) snapshot=512114688
                Virtual memory (bytes) snapshot=2722365440
                Total committed heap usage (bytes)=491782144
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=240900
        File Output Format Counters
                Bytes Written=84
[hdfs@rvm ~]$ _
```

Your results are certainly different, but the final lines of output will probably be similar.

__2. You want to examine the output that was produced:

```
hdfs dfs -cat sampledata/TempOut/*
```

The result of this run should be similar to:



```
[hdfs@rvm ~]$ hdfs dfs -cat /sampledata/TempOut/*
01      367
02      431
03      530
04      622
05      704
06      777
07      785
08      781
09      755
10      640
11      543
12      426
[hdfs@rvm ~]$ _
```

Be aware that if you cut & paste from this file, sometimes Microsoft and Adobe software change a single dash ("-") to an en-dash ("–") or an em-dash ("—"). Linux is not kind to these characters.

You can see that the maximum temperature for January (01) was 36.7 F — this is the coldest month of Winter in Sumner County as evidenced by the data — and the maximum temperature for July (07) was 78.5 F for the County (a rather cool summer, it appears).