

# Lab: Write a MapReduce Program

## Lab Overview

The labs for this lesson cover how to make modifications to an existing MapReduce template, compile it, run it, and examine the output. The goal of these lab exercises is to calculate the maximum, minimum, and mean values for sepal length, sepal width, petal length, and petal width from the famous Iris flower data set.

Here is the output expected from the map and reduce code for this lab (just 3 lines with tab-separated fields exactly as follows):

```
1.0  min-sw max-sw mean-sw min-sl max-sl mean-sl min-pw max-pw mean-pw min-pl max-pl mean-pl
2.0  min-sw max-sw mean-sw min-sl max-sl mean-sl min-pw max-pw mean-pw min-pl max-pl mean-pl
3.0  min-sw max-sw mean-sw min-sl max-sl mean-sl min-pw max-pw mean-pw min-pl max-pl mean-pl
```

## Lab Submission Details

Your lab submission must conform to the following format in order to receive credit for your work. The lab submission will be processed by an automated framework, so it's absolutely critical that everything requested is exactly where it needs to be.

1. Create a ZIP file called `CS286_FIRST_LAST_LAB_1.zip` containing your solution with the `IRIS_LAB` directory inside. Replace *FIRST* and *LAST* with your first and last name respectively.

```
$ zip -r CS286_JAMES_CASALETTO_LAB_1.zip IRIS_LAB
```

2. Email me (james.casaletto@sjsu.edu) and the class grader (guannan.zhong@sjsu.edu) your ZIP file as an attachment and put "CS286 Lab1 Submission" in the subject line. Leave the body of the email blank.

3. When we unzip this file, we should only see the output below.

```
$ unzip CS_286_JAMES_CASALETTO_LAB_1.zip
Archive:  CS_286_JAMES_CASALETTO_LAB_1.zip
  creating:  IRIS_LAB/
  inflating: IRIS_LAB/IrisMapper.java
  creating:  IRIS_LAB/OUT/
  inflating: IRIS_LAB/OUT/part-r-00000
  extracting: IRIS_LAB/OUT/_SUCCESS
```

```
creating: IRIS_LAB/DATA/  
inflating: IRIS_LAB/DATA/iris-data.txt  
creating: IRIS_LAB/classes/  
creating: IRIS_LAB/classes/Iris/  
inflating: IRIS_LAB/classes/Iris/IrisReducer.class  
inflating: IRIS_LAB/classes/Iris/IrisDriver.class  
inflating: IRIS_LAB/classes/Iris/IrisMapper.class  
inflating: IRIS_LAB/Iris.jar  
inflating: IRIS_LAB/rerun.sh  
inflating: IRIS_LAB/IrisReducer.java  
inflating: IRIS_LAB/IrisDriver.java  
inflating: IRIS_LAB/rebuild.sh
```

## Summary of Data

You can get a description of and download the data from the following site:  
[https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)

## Exercise 1.1: Copy the Lab Files

1. Copy the source code from your laptop into the `/user/user01` directory.

```
$ scp -P 2222 IRIS_LAB.zip user01@localhost:/user/user01
```

2. Login to the virtual machine as `user01` and change directory into the `user01` directory.

```
$ ssh -p 2222 user01@localhost  
$ cd /user/user01
```

3. Unzip the `IRIS_LAB.zip` file.

```
$ unzip IRIS_LAB.zip
```

## Exercise 1.2: Modify Code in the Driver

1. Change directory into the `IRIS_LAB` directory.

```
$ cd /user/user01/IRIS_LAB
```

2. Open the `IrisDriver.java` source file with your favorite text editor.

```
$ vi IrisDriver.java
```

3. Make the changes to the driver as described by the TODO markups.

## Exercise 1.3: Write, Build, and Run the Mapper Code

1. Change directory into the `IRIS_LAB` directory.

```
$ cd /user/user01/IRIS_LAB
```

2. Open the `IrisMapper.java` file with your favorite editor.

```
$ vi IrisMapper.java
```

3. Modify the code in the mapper as described in the TODO markups.

4. Execute the `rebuild.sh` script to compile your code.

```
$ ./rebuild.sh
```

5. Execute the `rerun.sh` script to run your code. This script will execute your code.

```
$ ./rerun.sh
```

6. Examine the output from your MapReduce job. Note you may need to wait a minute before the job output is completely written to the output files.

```
$ cat /user/user01/IRIS_LAB/OUT/part-m-00000
```

Here is partial output expected for this exercise:

```
0.0      4.9_3.0_1.4_0.2
0.0      4.7_3.2_1.3_0.2
0.0      4.6_3.1_1.5_0.2
0.0  5.0_3.6_1.4_0.2
```

If you did not obtain the results above, you'll need to revisit your Mapper class.

## Exercise 3.4: Implement Code in the Reducer

In this exercise, you will implement code in the reducer to calculate the max, min, and mean values. Recall that your mapper code will produce intermediate results. One such record looks like this. The output fields in order are: flower species and (sepal length, sepal width, petal length, petal width) all separated by the "\_" character.

**0.0      4.9\_3.0\_1.4\_0.2**

1. Change directory into the `IRIS_LAB` directory.

```
$ cd /user/user01/IRIS_LAB
```

2. Open the `IrisReducer.java` source file with your favorite text editor.

```
$ vi IrisReducer.java
```

3. Make the changes to the reducer as described by the TODO markups.

4. Open the `ReceiptsDriver.java` source file with your favorite text editor and uncomment the reducer definition.

## Exercise 3.5: Compile and Run Your Code

1. Change directory into the `IRIS_LAB` directory.

```
$ cd /user/user01/IRIS_LAB
```

2. Execute the `rebuild.sh` script to compile your code.

```
$ ./rebuild.sh
```

3. Execute the `rerun.sh` script to run your code.

```
$ ./rerun.sh
```

4. Examine the output from your MapReduce job.

```
$ cat /user/user01/IRIS_LAB/OUT/part-r-00000
```