Wordcounter.java: Driver Program

```
package com. app;
import java.io.IOException;

import org. apache.hadoop.fs.Path,
import org. apache.hadoop.io.Intwritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.Jobconf;
import org.apache.hadoop.mapred.Mapper;
import org. apache.hadoop.mapreduce.Job;
import org. apache.hadoop.mapreduce.lib. input. FileInputFormat;
import org apache.hadoop.mapreduce.lib. input TextInputFormat;
import org apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.outputTextOutputFormat;

public class Wordcounter
public static void main (string (] args) throws IOException,
InterruptedException, ClassNotFoundException
Job job = new Job ().
job.setJobName ( "wordcounter")
job. setJarByClass (WordCounter class) ;
job. setMapperClass (WordCounterMap.class) ;
job, setReducerClass (WordCounterRed.class)
job.setOutputKeyClass (Text. class) ;
job.setOutputValueClass (IntWritable.class) ;
FileInputFormat.addInputPath (job, new Path ("/sample/word.
txt" )) ;
FileOutputFormat.setOutputPath (job, new Path ("/sample/
wordcount " ) ) ;
System.exit (job.waitForCompletion (true)? 0: 1);
}
}
```

WordCounterMap.java: Map Class

```
package com. app;
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 WordCounterMap extends Mapper<LongWritable, Text, Text,
Intwritable>
```

```
if
@Override
protected void map (LongWritable key, Text value, Context context)
throws IOException, InterruptedException
{
String [] words=value.toString () .split (",");
for (String word: words)
context .write (new Text (word), new IntWritable (1));
}
}
}
```

WordCountReduce.java: Reduce Class

```
package com.infosys;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer:
public class WordCounterRed extends Reducer<Text, Intwritable, Text,
Intwritable>
@Override
protected void reduce (Text word, Iterable<Intwritable> values,
Context context)
throws IOException, InterruptedException
Integer count = 0
for (Intwritable val: values) {
count += val.get ():}
context.write (word, new IntWritable (count) );
}}
```

```
job.setCombinerClass(WordCounterRed.class);
// Input and Output Path
FileInputFormat.addInpurPath(job, new Path("/mapreducedemos/lines.ext");
FileOutputFormat.setOutputPath(job, new Path("/ mapreducedemos/output/
wordcount/");
```

WordCountPartitioner.java

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io. Text;
import org.apache.hadoop.mapreduce.Partitioner;
public class WordCountPartitioner extends Partitioner<Text, IntWritable> {
@Override
public int getPartition (Text key, Int Writable value, it numPartitions) {
String word = key.toStringO;
char alphabet = word.toUpperCase0.charAt(0);
int partition Number = 0;
switch (alphabet) {
case 'A': partitionNumber = 1; break;
case 'B': partitionNumber = 2; break;
case 'C': partition Number = 3; break;
case 'D': partition Number = 4; break;
case 'E': partition Number = 5; break;
case 'F': partition Number = 6; break;
case 'G': partition Number = 7; break;
case 'H': partitionNumber = 8; break;
case 'I': partition Number = 9; break;
case "J: partitionNumber = 10; break;
case 'K': partitionNumber = 11; break;
case "L': partition Number = 12; break;
case 'M': partitionNumber = 13; break;
case 'N': partitionNumber = 14; break;
case 'O': partitionNumber = 15; break;
case "P': partitionNumber = 16; break;
case Q': partitionNumber = 17; break;
case 'R': partition Number = 18; break;
case 'S': partition.Number = 19; break;
case "T': partition Number = 20; break;
case 'U': partitionNumber = 21; break;
case V': partitionNumber = 22; break;
case 'W': partition Number = 23; break;
case "X': partitionNumber = 24; break;
case Y': partition Number = 25; break;
case 'Z': partition Number = 26; break;
default: partitionNumber = 0; break; }
return partitionNumber;}}
```

In the driver program, set the partitioner class as shown below:

```
job.set Num.Reduce Tasks(27);
job.setPartitionerClass(WordCountPartitioner.class);
I/ Input and Output Path
FileInputFormat.add.InputPath(job, new Path("/mapreducedemos/lines.txt");
FileOutputFormat.se*OutputPath(job, new Path("/mapreducedemos/output/
wordcountpartitioner/"));
```

Act:
WordSearcher.java

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
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.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
public class WordSearcher(
public static void main(String(] args) throws IOException,
InterruptedException, Class NotFoundException (
Configuration conf = new ConfigurationO;
Job job = new Job(conf);
job.setJarByClass(WordSearcher.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(Text.class);
job.setMapperClass(WordSearchMapper.class);
job.setReducerClass(WordSearchReducer.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
job.setNumReduce Tasks(1);
job.getConfiguration).set("keyword", "Jack");
FileInputFormat.setInputPaths(job, new Path("/mapreduce/student.csv"));

FileOutputFormat.setOutputPath(job, new Path("/mapreduce/output/search");
System.exit(job. waitForCompletion(true) ? 0: 1); }}
```

WordSearchMapper.java

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io. Int Writable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io. Text;
import org.apache.hadoop.mapreduce.InputSplit;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileSplit;
public class WordSearchMapper extends Mapper«Long Writable, Text, Text, Text>
{
static String keyword;
static int pos = 0;
protected void setup (Context context) throws IOException,
InterruptedFxception(
Configuration configuration = context.getConfiguration0;
keyword = configuration.get("keyword");
3
toString0 +
protected void map (Long Writable key, Text value, Context context)
throws IOException, Interrupted Exception (
InputSplit i = context,getInputSplitO; // Get the input split for this map.
FileSplit f= (FileSplit) i;
String fileName=f.getPathO.getNameO;
Integer wordPos;
post+;
if (value.toString).contains(keyword))f
wordPos = value.find(keyword);
context.write(value, new Text(fileName +
""+ new IntWritable(pos).
"+ wordPos.toString0)); }}}
```

```
WordSearchReducer.java
import java.io.IOException;
import org.apache.hadoop.io. Text;
import org.apache.hadoop.mapreduce.Reducer;
public class WordSearchReducer extends Reducer<Text, Text, Text, Text {
protected void reduce(Text key, Text value, Context context)
throws IOException, InterruptedException
context.write(key, value); }}
```

Sorting:

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io. Long Writable;
import org.apache.hadoop.io.NullWritable;
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;
```

```
public class SortStudNames{
public static class SortMapper extends
Mapper<Long Writable, Text, Text, Text> f
protected void map (Long Writable key, Text value, Context context)
throws IOException, InterruptedException(
String( ] token = value.toStringO.split(",");
context.write(new Text (token(1]), new Text(token[O]+" - "+token[1]));}}
/ Here, value is sorted.
public static class SortReducer extends
Reducer‹Text, Text, Null Writable, Text> (
public void reduce(Text key, Iterable<Text> values, Context context)
throws IOException, InterruptedException(
for (Text details : values) f
context, write(Null Writable.get), details);}}}
public static void main (String! ] args) throws IOException,
InterruptedException,ClassNotFoundException(
Configuration conf = new Configuration0;
Job job = new Job(conf);
job.setJarByClass(SortEmpNames.class);
job.set MapperClass(SortMapper.class);
job.setReducerClass(SortReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(Text.class);
FileInputFormat.setInputPaths(job, new Path("/mapreduce/student.csv"));
FileOutputFormat.setOutputPath(job, new
Path("/mapreduce/output/sorted/");
System.exit(job.waitForCompletion(true) ? 0: 1);}}
```