```
// Word Count Mapper program

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


public class WordCountMapper extends
            Mapper<LongWritable, Text, Text, LongWritable> {

      private final static LongWritable one = new LongWritable(1);

      @Override
      protected void map(LongWritable key, Text value, Context context)
                  throws IOException, InterruptedException {
            String line = value.toString();
            String[] words = line.split(" ");
            for (int i = 0; i < words.length; i++) {
                  context.write(new Text(words[i]), one);
            }
      }
}


// Word Count Reducer program


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

public class WordCountReducer extends
Reducer<Text,LongWritable,Text,LongWritable>
{
      @Override
      protected void reduce(Text key,Iterable<LongWritable> value,Context
context)throws IOException,InterruptedException
      {
            long sum=0;
            while(value.iterator().hasNext())
            {
                  sum+=value.iterator().next().get();
            }
            context.write(key,new LongWritable(sum));
      }
}


// Word count Driver Program

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.ToolRunner;
import org.apache.hadoop.util.Tool;
public class WordCountJob implements Tool{
      private Configuration conf;
      @Override
      public Configuration getConf()
      {
      return conf;
      }
      @Override
      public void setConf(Configuration conf)
      {
            this.conf=conf;
      }
      @Override
      public int run(String []args)throws Exception
      {
            Job wordcountjob=new Job(getConf());
            wordcountjob.setJobName("mat word count");
            wordcountjob.setJarByClass(this.getClass());
            wordcountjob.setMapperClass(WordCountMapper.class);
            wordcountjob.setReducerClass(WordCountReducer.class);

            wordcountjob.setMapOutputKeyClass(Text.class);
            wordcountjob.setMapOutputValueClass(LongWritable.class);
            wordcountjob.setOutputKeyClass(Text.class);
            wordcountjob.setOutputValueClass(LongWritable.class);
            FileInputFormat.setInputPaths(wordcountjob,new
Path(args[0]));
            FileOutputFormat.setOutputPath(wordcountjob,new
Path(args[1]));
            wordcountjob.setNumReduceTasks(2);
            return wordcountjob.waitForCompletion(true)==true? 0:1;
      }
      public static void main(String []args)throws Exception
      {
            ToolRunner.run(new Configuration(),new WordCountJob(),args);
      }

}
```

Execution procedure

```
CREATE JAR FILE IN ECLIPSE
=========================

CREATE AN NPUT FILE IN LOCAL FILESYSTEM
======================================

cloudera@localhost ~]$ gedit wc1.txt
Hadoop is the Elephant King!
A yellow and elegant thing.
He never forgets
```

Useful data, or lets
An extraneous element cling!

COPY INPUT FIL INTO HDFS
===

```
[cloudera@localhost ~]$ hadoop fs -put wc1.txt word
```

RUN THE PROGRAM
===============

```
[cloudera@localhost ~]$ hadoop jar wcount.jar WordCountJob wc1.txt wordout
```

OUTPUT::
========

```
[cloudera@localhost ~]$ hadoop fs -ls wordout

Found 4 items
-rw-r--r--   3 cloudera cloudera          0 2016-06-10 00:01
wordout/_SUCCESS
drwxr-xr-x   - cloudera cloudera          0 2016-06-10 00:00
wordout/_logs
-rw-r--r--   3 cloudera cloudera        135 2016-06-10 00:01
wordout/part-r-00000
-rw-r--r--   3 cloudera cloudera        200 2016-06-10 00:01
wordout/part-r-00001
```

SEE THE CONTENTS OF THE REDUCER OUTPUT FILES
==

```
[cloudera@localhost ~]$ hadoop fs -cat wordout/part-r-00000
But   1
HDFS  1
Hadoop      1
Hadoop.     1
He    1
Hive, 1
The   1
data, 1
forgets     1
helps 1
him   1
lets  1
never 1
plays 1
the   2
thing.      1
thrive      1
to    1
[cloudera@localhost ~]$ hadoop fs -cat wordout/part-r-00001
A     2
An    1
And   1
```

```
Are   1
Elephant    1
Impala,     1
King! 1
Sqoop.      1
Useful      1
and   2
cling!      1
elegant     1
element     1
elephant    1
extraneous  1
group.      1
in    1
is    2
king  1
or    1
well  1
what  1
with  1
wonderful   1
yellow      1
```