Create a text file in your local machine and write some text into it.  
$ nano data.txt

Check the text written in the data.txt file.  
$ cat data.txt

Create a directory in HDFS, where to kept text file.  
$ hdfs dfs -mkdir /test

Upload the data.txt file on HDFS in the specific directory.  
$ hdfs dfs -put /home/codegyani/data.txt /test

 WC\_Mapper.java

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;

public class WC\_Mapper extends MapReduceBase implements Mapper**<LongWritable**,Text,Text,IntWritable**>**{

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

    private Text word = new Text();

    public void map(LongWritable key, Text value,OutputCollector**<Text**,IntWritable**>** output,

           Reporter reporter) throws IOException{

        String line = value.toString();

        StringTokenizer  tokenizer = new StringTokenizer(line);

        while (tokenizer.hasMoreTokens()){

            word.set(tokenizer.nextToken());

            output.collect(word, one);

        }

    }

}

### WC\_Reducer.java

import java.io.IOException;

    import java.util.Iterator;

    import org.apache.hadoop.io.IntWritable;

    import org.apache.hadoop.io.Text;

    import org.apache.hadoop.mapred.MapReduceBase;

    import org.apache.hadoop.mapred.OutputCollector;

    import org.apache.hadoop.mapred.Reducer;

    import org.apache.hadoop.mapred.Reporter;

    public class WC\_Reducer  extends MapReduceBase implements Reducer**<Text**,IntWritable,Text,IntWritable**>** {

    public void reduce(Text key, Iterator**<IntWritable>** values,OutputCollector**<Text**,IntWritable**>** output,

     Reporter reporter) throws IOException {

    int sum=0;

    while (values.hasNext()) {

    sum+=values.next().get();

    }

    output.collect(key,new IntWritable(sum));

    }

    }

### WC\_Runner.java

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.FileInputFormat;

    import org.apache.hadoop.mapred.FileOutputFormat;

    import org.apache.hadoop.mapred.JobClient;

    import org.apache.hadoop.mapred.JobConf;

    import org.apache.hadoop.mapred.TextInputFormat;

    import org.apache.hadoop.mapred.TextOutputFormat;

    public class WC\_Runner {

        public static void main(String[] args) throws IOException{

            JobConf conf = new JobConf(WC\_Runner.class);

            conf.setJobName("WordCount");

            conf.setOutputKeyClass(Text.class);

            conf.setOutputValueClass(IntWritable.class);

            conf.setMapperClass(WC\_Mapper.class);

            conf.setCombinerClass(WC\_Reducer.class);

            conf.setReducerClass(WC\_Reducer.class);

            conf.setInputFormat(TextInputFormat.class);

            conf.setOutputFormat(TextOutputFormat.class);

            FileInputFormat.setInputPaths(conf,new Path(args[0]));

            FileOutputFormat.setOutputPath(conf,new Path(args[1]));

            JobClient.runJob(conf);

        }

    }

Download the source code.

* Create the jar file of this program and name it **countworddemo.jar.**
* Run the jar file  
  hadoop jar /home/codegyani/wordcountdemo.jar com.javatpoint.WC\_Runner /test/data.txt /r\_output
* The output is stored in /r\_output/part-00000