
Name: Muhammad Abeer

Roll no: 19P0061

Section:6-B

MAP REDUCE on HADOOP

Steps

1. Open any code editor (I have used Eclipse) compatible with compiling Java.
2. Create a project **WordCount** or give it any name and create a class named WordCount.
3. Inside the class type in the following code of MapReduce

```
import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
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 WordCount {

    public static class Map extends 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, Context context) throws
IOException, InterruptedException {
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                word.set(tokenizer.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class Reduce extends Reducer<Text, IntWritable, Text,
IntWritable> {

        public void reduce(Text key, Iterable<IntWritable> values, Context
context)
            throws IOException, InterruptedException {
            int sum = 0;
```

```

        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}

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

    Job job = new Job(conf, "wordcount");

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

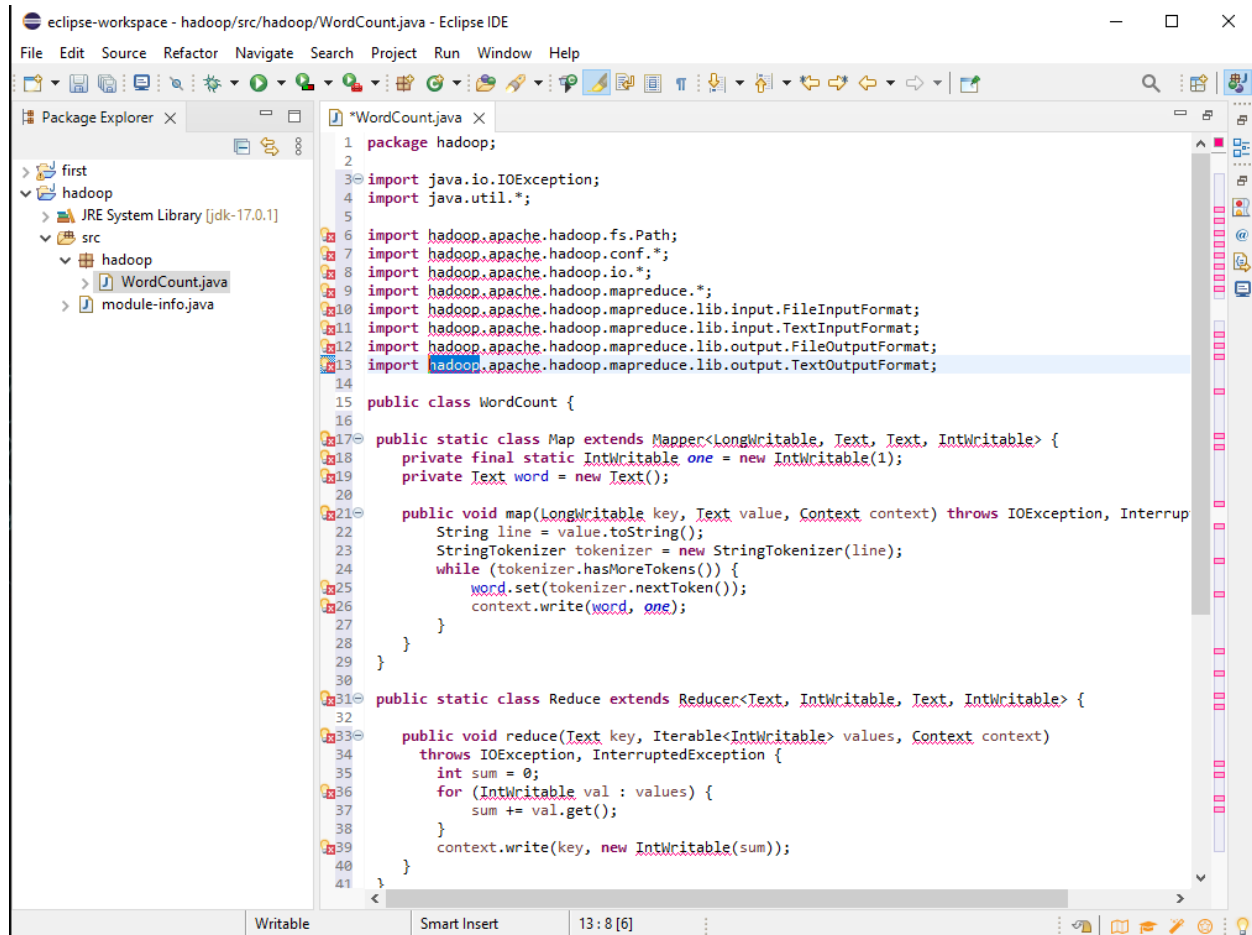
    job.setMapperClass(Map.class);
    job.setReducerClass(Reduce.class);

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

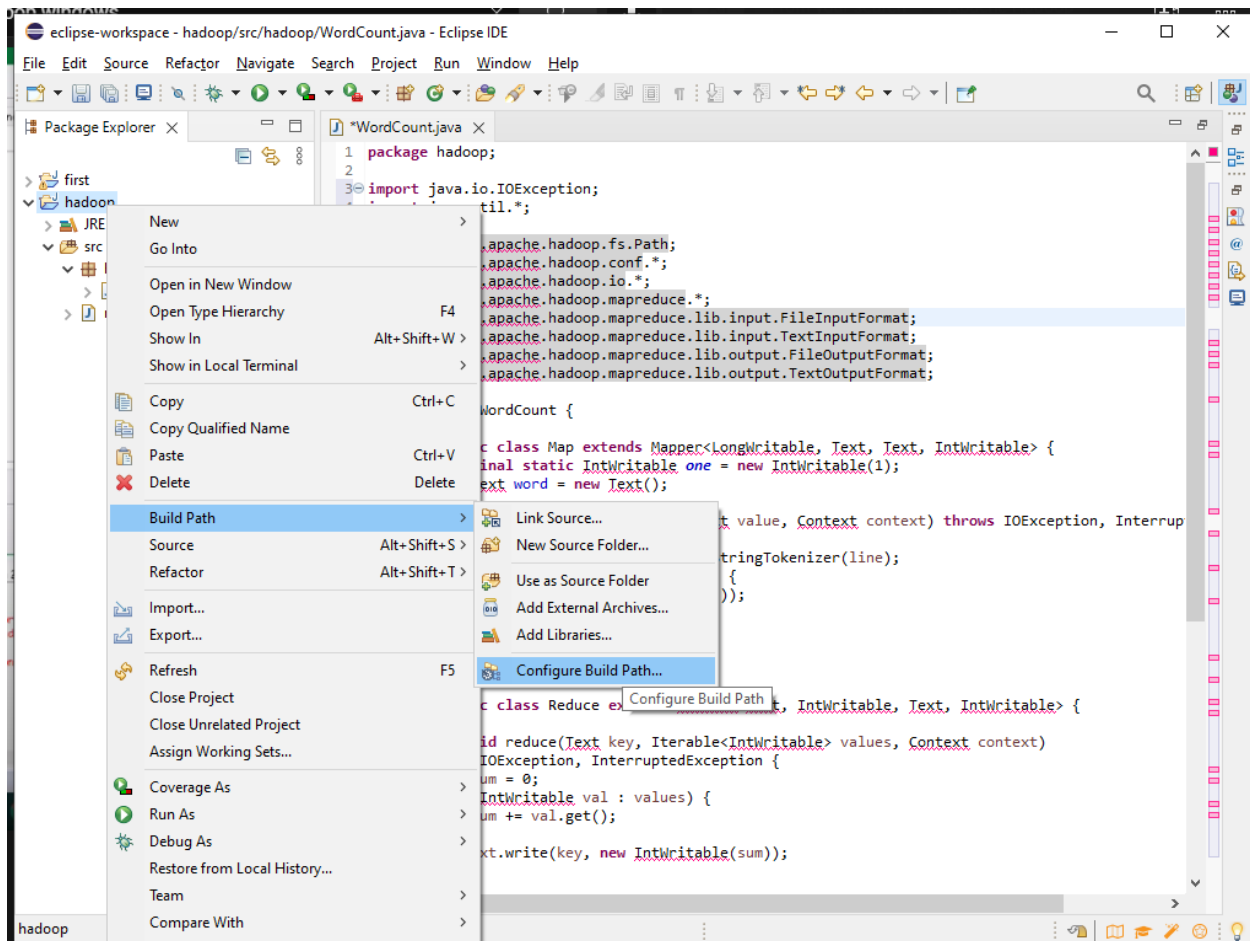
    job.waitForCompletion(true);
}
}

```

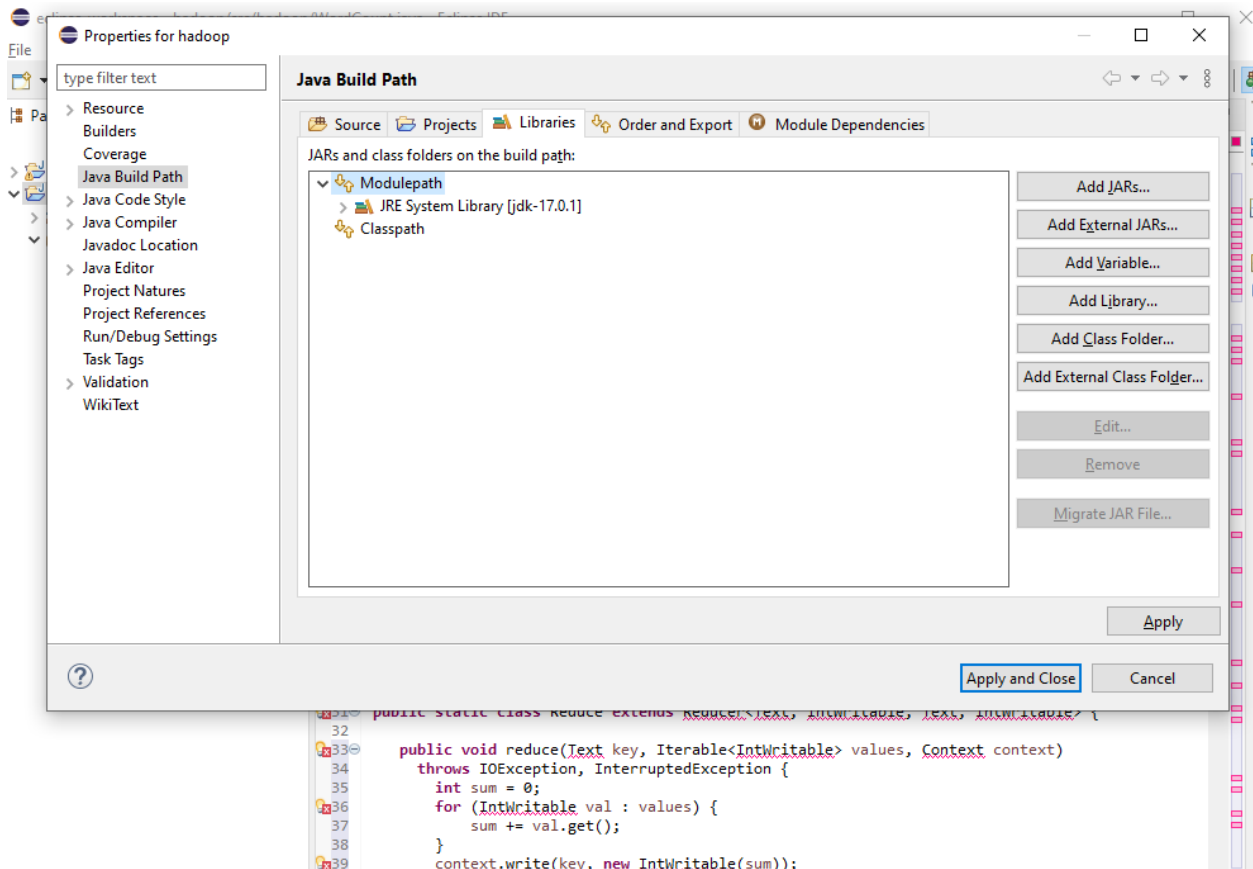


You can see a lot of errors in it. We shall fix them.

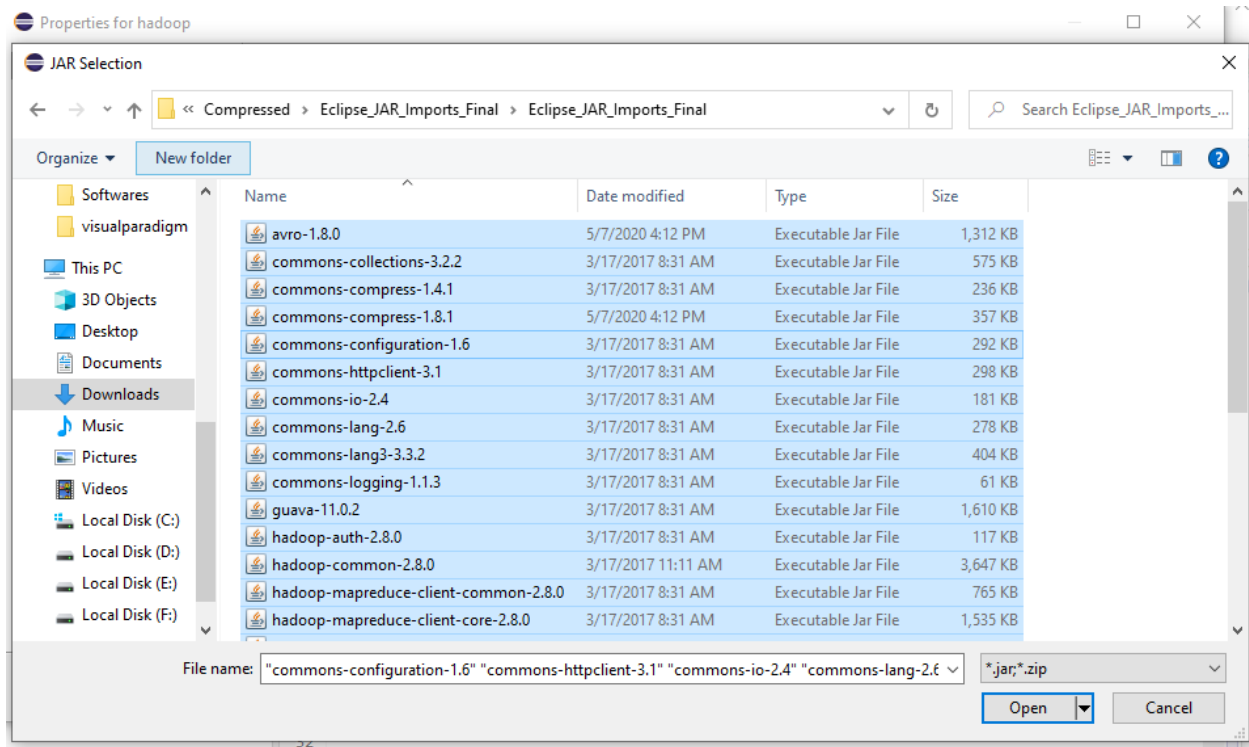
4. Go to [https://github.com/jijim/HADOOP-Windows10/blob/master/Eclipse JAR Imports Final.rar](https://github.com/jijim/HADOOP-Windows10/blob/master/Eclipse%20JAR%20Imports%20Final.rar) and download Jar Files required to successfully run the program and after downloading, extract them.
5. On left side of window of Eclipse, right click on Hadoop, click on build Path and click on configure build path.



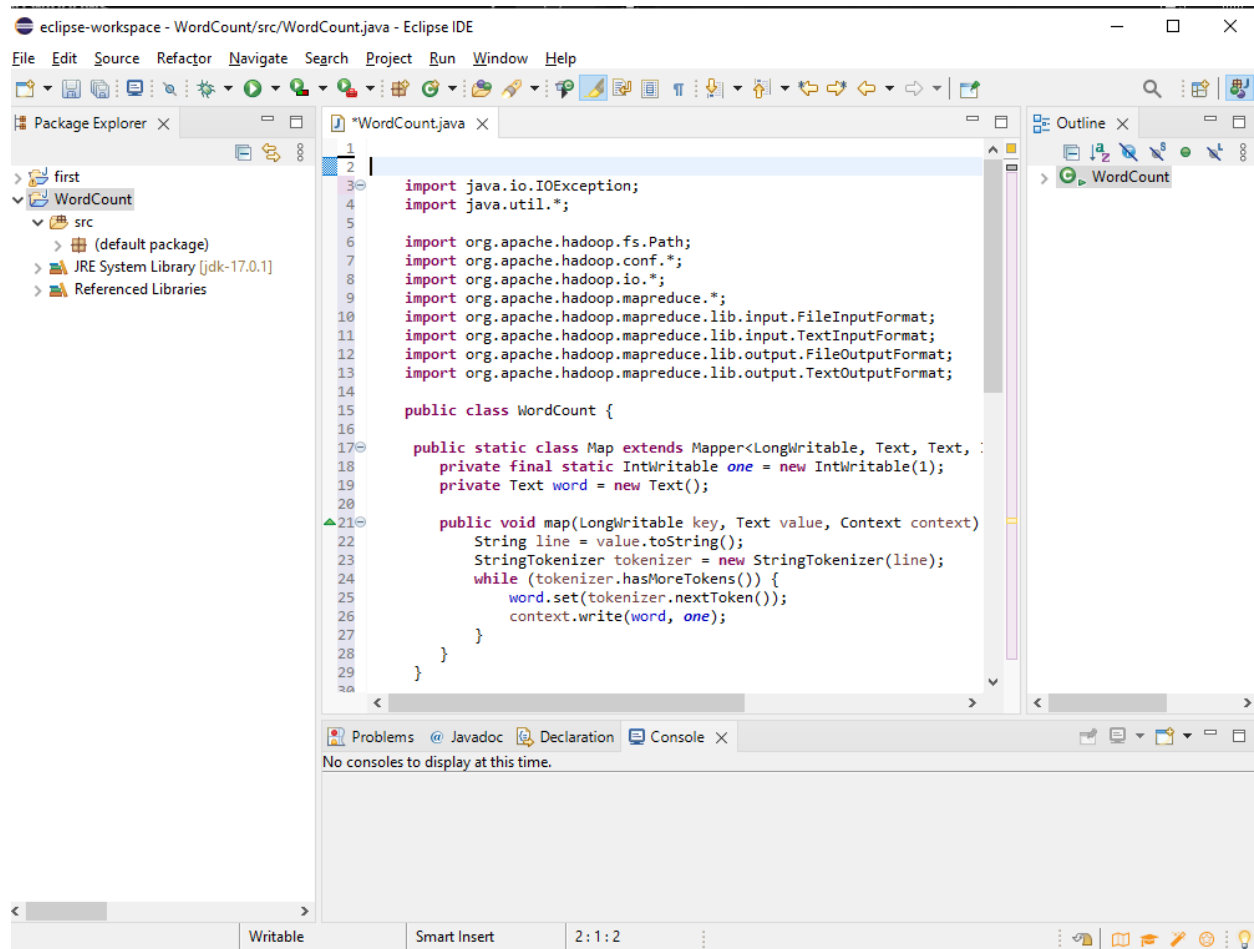
A WINDOW WILL BE OPENED



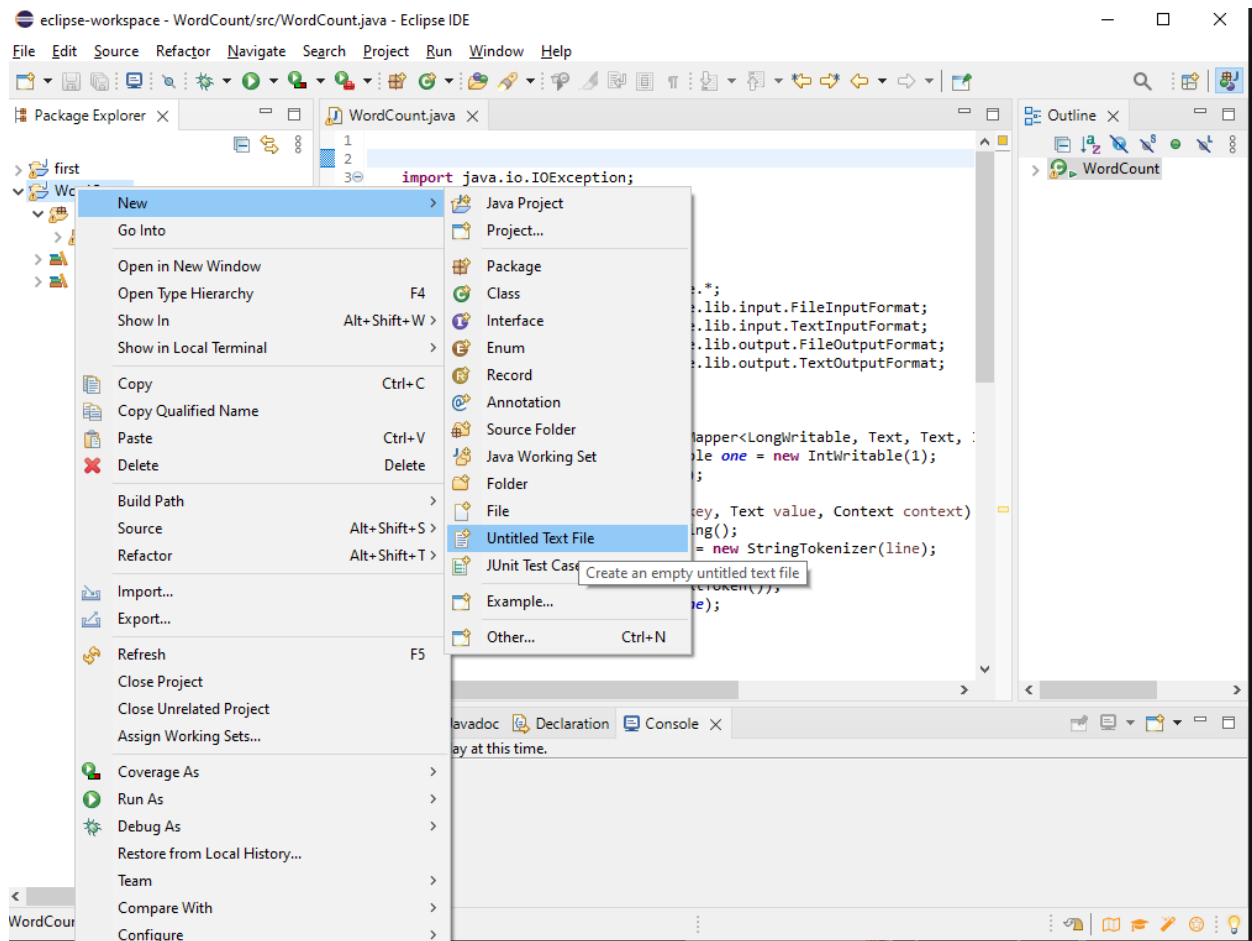
6. Click on Add External JARs.. and select all jar files which you downloaded earlier.



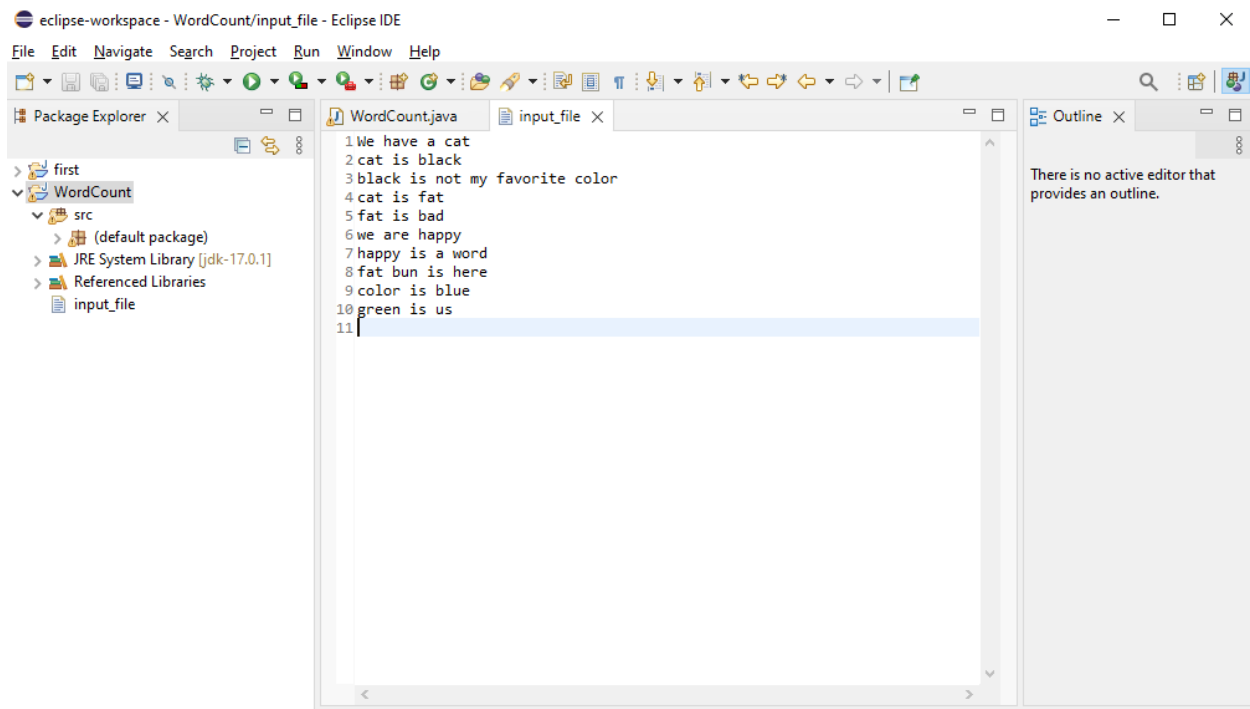
7. After adding, click on apply and you'll see all the errors will be removed.



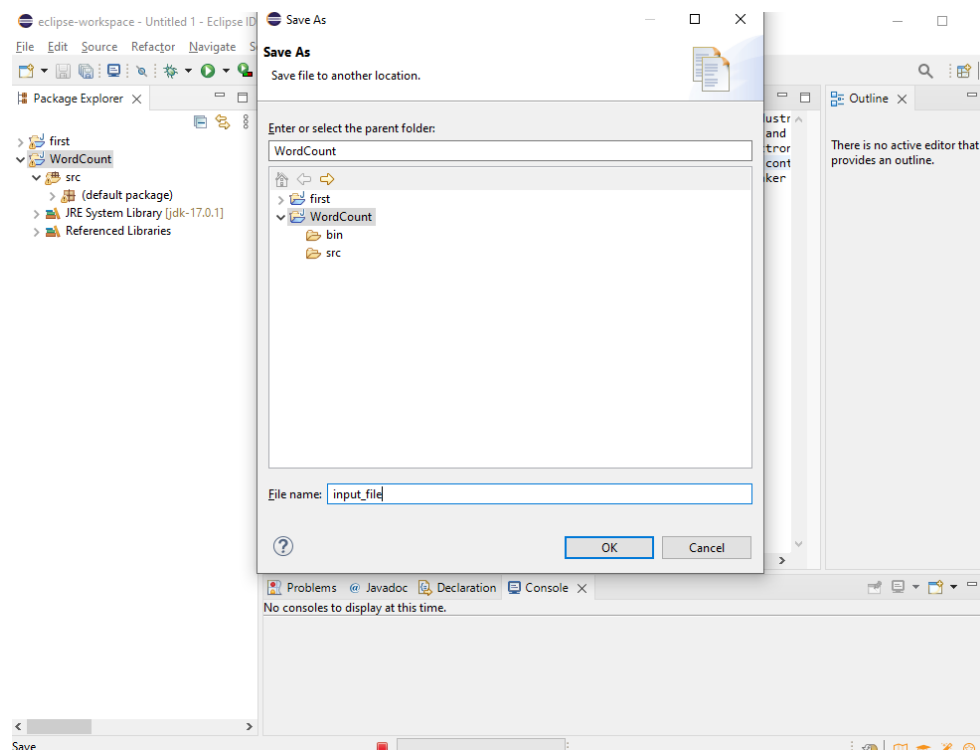
8. Now in order to create an input file for WordCount program, right click on WordCount, Click on new and then click on Untitled Text File



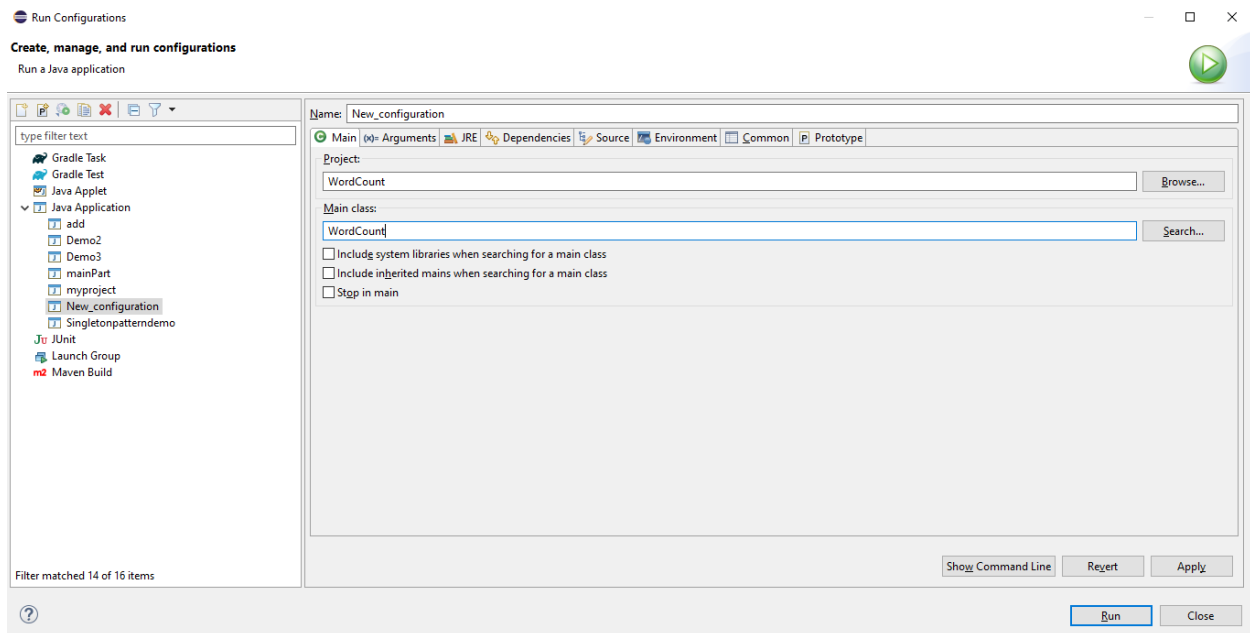
9. In that text file, add any random text.



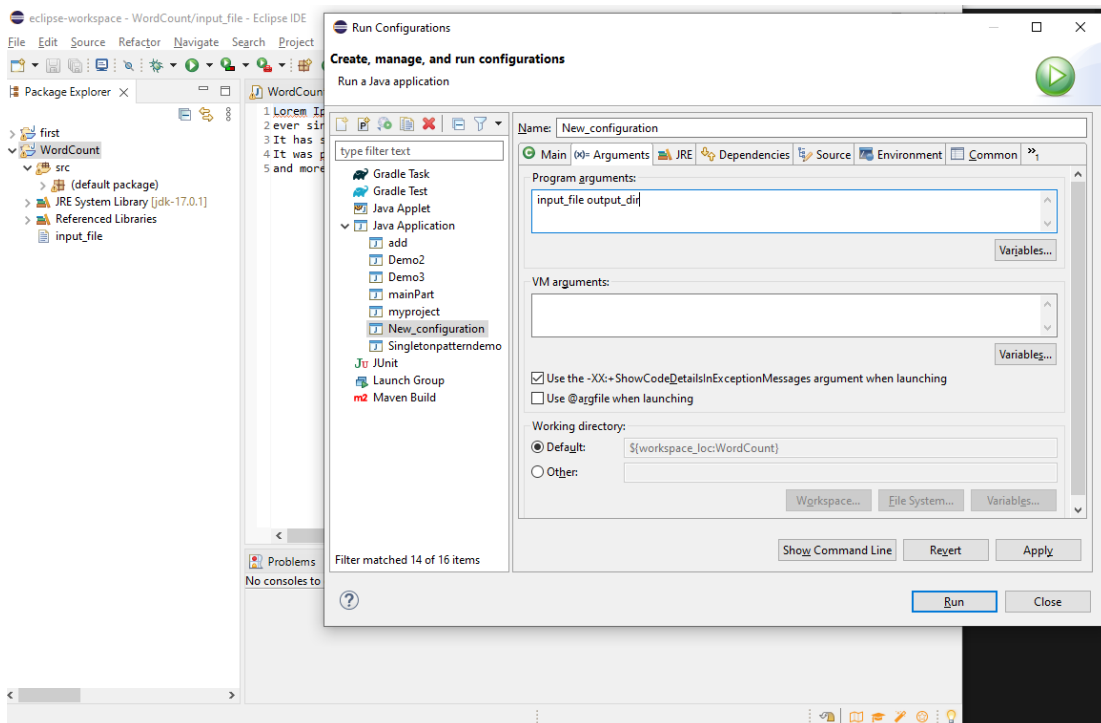
10. After that press Ctrl+S to save the file



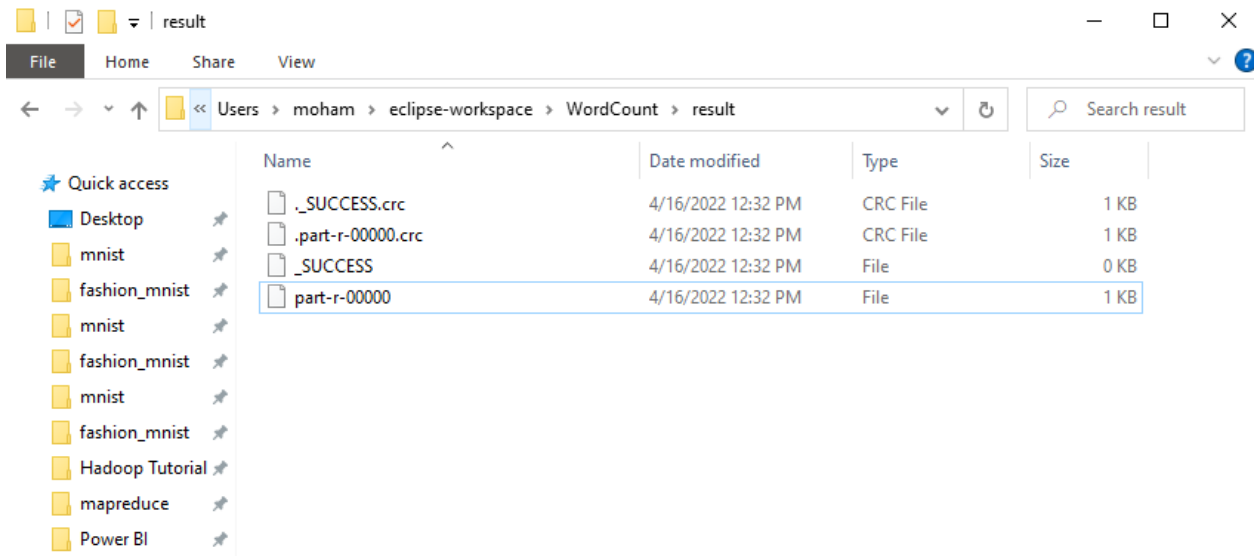
11. After that click on run in menu bar and then click on run configurations. A window will be opened. Give the main Class name.



12. After that click on Arguments and in Program Arguments, give the name of your input file along with output directory and click on run. Your program will build and run.



13. Go to your directory: **eclipse-workspace\WordCount** and open the output directory which you created for program. There will be a file **part-r-00000**. Open it with text editor. You'll be able to see your output of word count.



OUTPUT

part-r-00000 - Notepad

File	Edit	Format	View	Help
we	1			
a	2			
are	1			
bad	1			
black	2			
blue	1			
bun	1			
cat	3			
color	2			
fat	3			
favorite	1			
green	1			
happy	2			
have	1			
here	1			
is	8			
my	1			
not	1			
us	1			
we	1			
word	1			