

# Home

[Edit](#)[New Page](#)[Jump to bottom](#)

RajeshwariCh edited this page 1 minute ago · 3 revisions

---

## BIG DATA PROGRAMMING

## IN CLASS PROGRAMMING 3

NAME : RAJESHWARI CHOLLETI

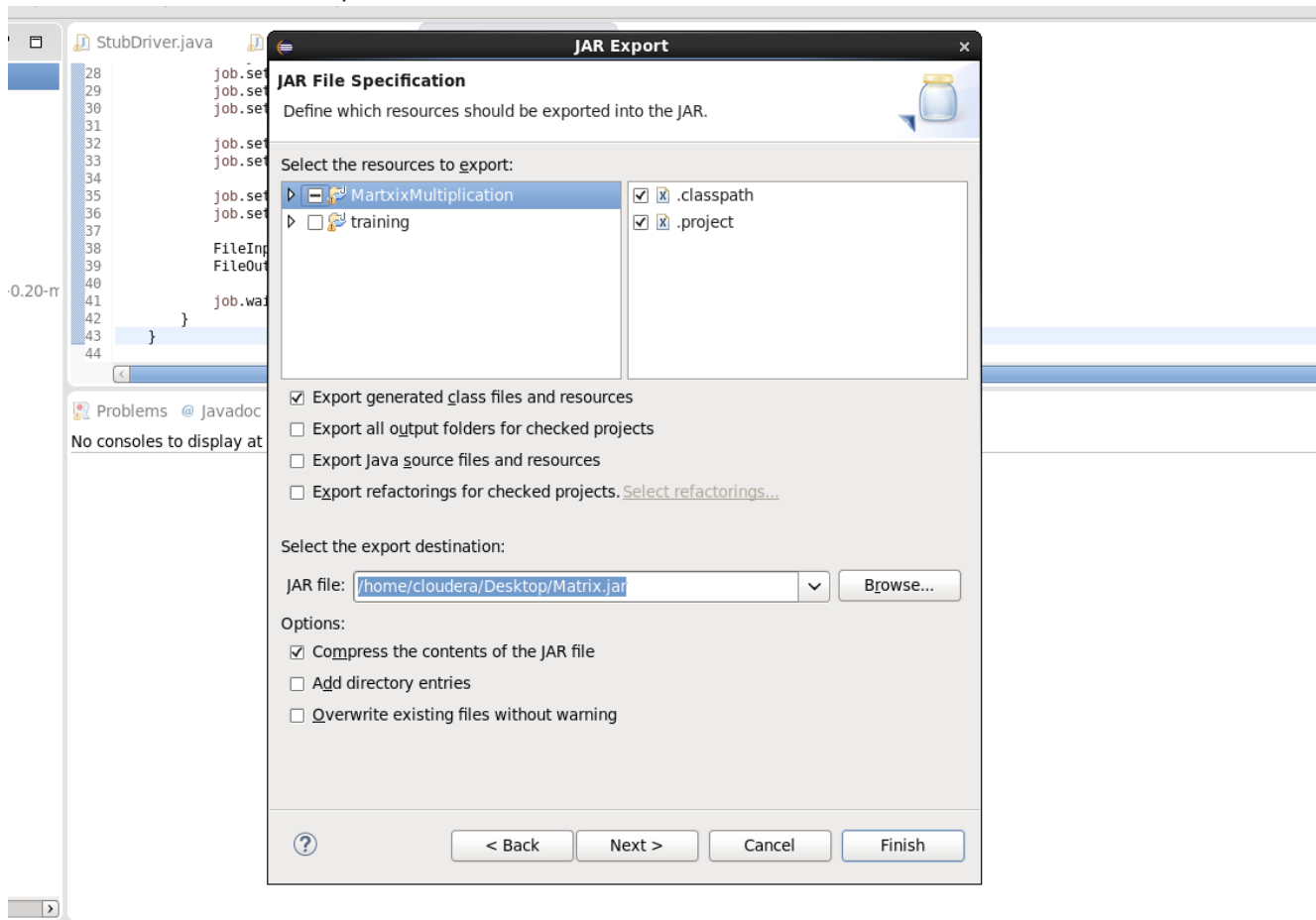
🔗 CLASS ID : 05

**OBJECTIVE : PROGRAM TO PERFORM MAP-REDUCE OPERATION USING  
THE CONCEPT OF MAP-REDUCE**

**IMPORT THE JAR FILES**

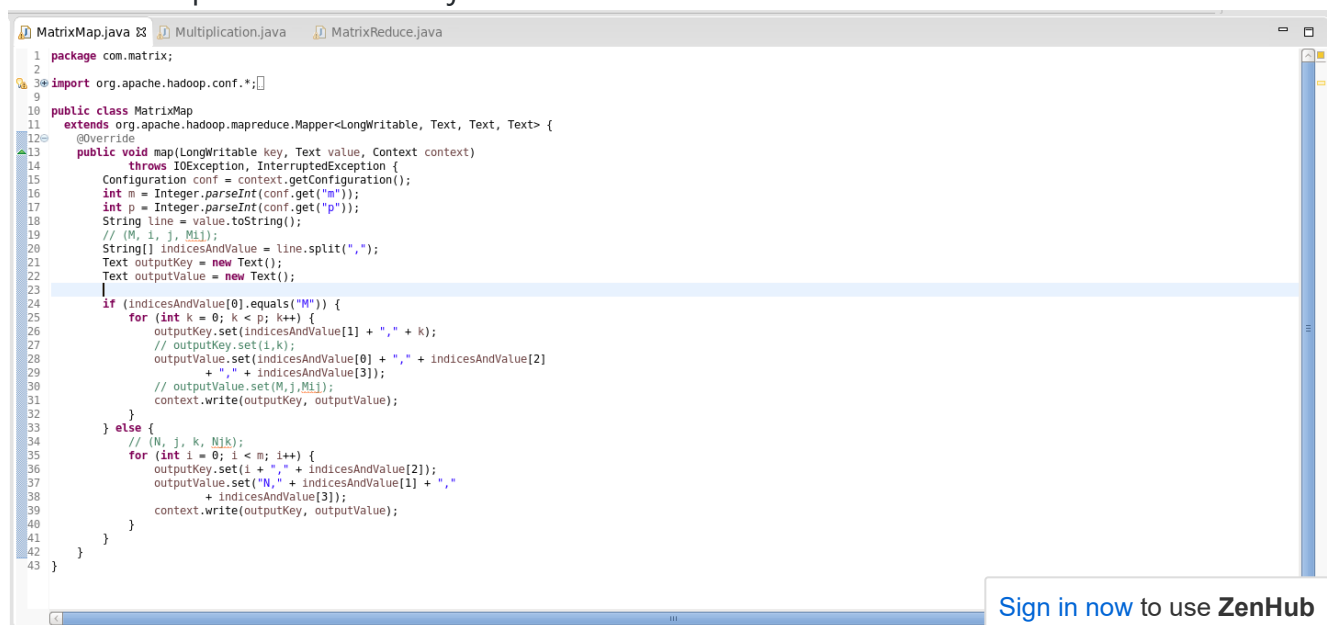
[Sign in now](#) to use **ZenHub**

We need to load the required Jar files



## MAPPER FUNCTION

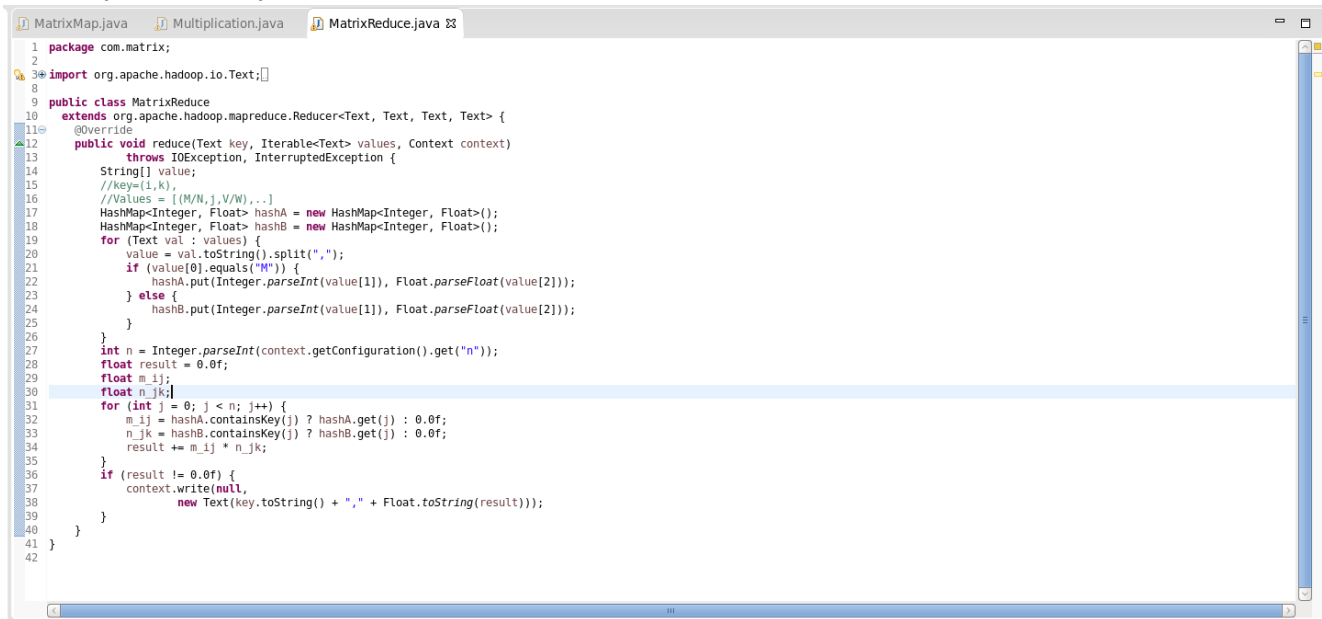
Mapper class extends `org.apache.hadoop.mapreduce.Mapper` class. Mapper function converts the input Matrix into a intermediate format such that each list (i,j) has values with M and N for all possible values of j.



Sign in now to use ZenHub

## REDUCER FUNCTION

Reducer class, extends org.apache.hadoop.mapreduce.Reducer class. Reducer function will combine the output from Mapper function. It sorts them with respect to M and N. It then sums up the multiplication results for both M and N.



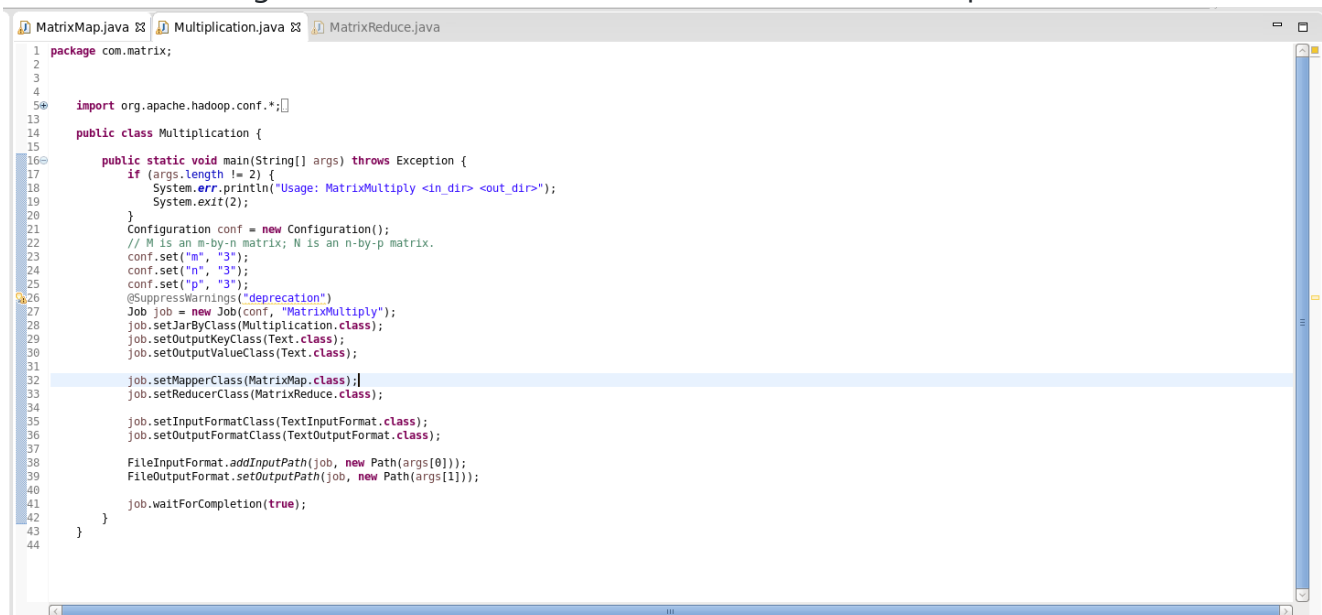
```

1 package com.matrix;
2
3 import org.apache.hadoop.io.Text;
4
5 public class MatrixReduce
6 extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> {
7     @Override
8     public void reduce(Text key, Iterable<Text> values, Context context)
9         throws IOException, InterruptedException {
10         String[] value;
11         //key=[i,k],
12         //Values = [(M/N,j,V/W),...]
13         HashMap<Integer, Float> hashA = new HashMap<Integer, Float>();
14         HashMap<Integer, Float> hashB = new HashMap<Integer, Float>();
15         for (Text val : values) {
16             value = val.toString().split(",");
17             if (value[0].equals("W")) {
18                 hashA.put(Integer.parseInt(value[1]), Float.parseFloat(value[2]));
19             } else {
20                 hashB.put(Integer.parseInt(value[1]), Float.parseFloat(value[2]));
21             }
22         }
23         int n = Integer.parseInt(context.getConfiguration().get("n"));
24         float result = 0.0f;
25         float m_ij;
26         float n_jk;
27         for (int j = 0; j < n; j++) {
28             m_ij = hashA.containsKey(j) ? hashA.get(j) : 0.0f;
29             n_jk = hashB.containsKey(j) ? hashB.get(j) : 0.0f;
30             result += m_ij * n_jk;
31         }
32         if (result != 0.0f) {
33             context.write(new Text(key.toString() + "," + Float.toString(result)));
34         }
35     }
36 }

```

## MAIN FUNCTION

Here we are calling all the libraries and functions for the matrix multiplication.



```

1 package com.matrix;
2
3
4
5 import org.apache.hadoop.conf.*;
6
7 public class Multiplication {
8
9     public static void main(String[] args) throws Exception {
10         if (args.length != 2) {
11             System.err.println("Usage: MatrixMultiply <in_dir> <out_dir>");
12             System.exit(2);
13         }
14         Configuration conf = new Configuration();
15         // M is an m-by-n matrix; N is an n-by-p matrix.
16         conf.set("m", "3");
17         conf.set("n", "3");
18         conf.set("p", "3");
19         @SuppressWarnings("deprecation")
20         Job job = new Job(conf, "MatrixMultiply");
21         job.setJarByClass(Multiplication.class);
22         job.setOutputKeyClass(Text.class);
23         job.setOutputValueClass(Text.class);
24         job.setMapperClass(MatrixMap.class);
25         job.setReducerClass(MatrixReduce.class);
26
27         job.setInputFormatClass(TextInputFormat.class);
28         job.setOutputFormatClass(TextOutputFormat.class);
29
30         FileInputFormat.addInputPath(job, new Path(args[0]));
31         FileOutputFormat.setOutputPath(job, new Path(args[1]));
32
33         job.waitForCompletion(true);
34     }
35 }

```

## OUTPUT

[Sign in now](#) to use ZenHub

Below is the output.

[Home](#)[/ user / cloudera / ICP3 / part-r-00000](#)Page  to  of 1 

```
0,0,16.0
0,1,31.0
0,2,24.0
1,0,50.0
1,1,65.0
1,2,60.0
2,0,34.0
2,1,60.0
2,2,41.0
```

## REFERENCES

<https://lendap.wordpress.com/2015/02/16/matrix-multiplication-with-mapreduce/>

+ Add a custom footer

### ▼ Pages 1

[Home](#)

+ Add a custom sidebar

## Clone this wiki locally

[https://github.com/RajeshwariCh/BDP\\_ICP3.wiki.git](https://github.com/RajeshwariCh/BDP_ICP3.wiki.git)



[Sign in now](#) to use **ZenHub**