

1. Open Eclipse> File > New > Java Project >( Name it – MRProgramsDemo) > Finish.
2. Right Click > New > Package ( Name it - PackageDemo) > Finish.
3. Right Click on Package > New > Class (Name it - MovieRating).
4. Add Following Reference Libraries:

1. Right Click on Project > Build Path> Add External

1. /usr/lib/hadoop-0.20/hadoop-core.jar
2. Usr/lib/hadoop-0.20/lib/Commons-cli-1.2.jar
3. Usr/lib/Hadoop-0.20/hadoop-common-2.6.0-cdh5.13.0.jar

```
package PackageDemo;
```

```
import java.io.IOException;
```

```
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.Path;
```

```
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.io.DoubleWritable;
```

```
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;
```

```
import org.apache.hadoop.util.GenericOptionsParser;
```

```
public class MovieRating {  
    public static void main(String[] args) throws Exception {  
        Configuration conf = new Configuration();  
        String[] files = new GenericOptionsParser(conf, args).getRemainingArgs();  
        Path input = new Path(files[0]);  
        Path output = new Path(files[1]);  
  
        Job job = new Job(conf, "movie rating");  
        job.setJarByClass(MovieRating.class);  
        job.setMapperClass(MapForMovieRating.class);  
        job.setReducerClass(ReduceForMovieRating.class);  
        job.setOutputKeyClass(Text.class);  
        job.setOutputValueClass(DoubleWritable.class);  
    }  
}
```

```

        FileInputFormat.addInputPath(job, input);
        FileOutputFormat.setOutputPath(job, output);

        System.exit(job.waitForCompletion(true) ? 0: 1);
    }

    public static class MapForMovieRating extends Mapper<LongWritable, Text, Text,
DoubleWritable> {
        public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
            String[] fields = value.toString().split("[,\\t]");
            if (fields.length >= 3 && !fields[0].equals("userId")) {
                String movieId = fields[1].trim();
                double rating = Double.parseDouble(fields[2].trim());
                context.write(new Text(movieId), new DoubleWritable(rating));
            }
        }
    }

    public static class ReduceForMovieRating extends Reducer<Text, DoubleWritable, Text,
DoubleWritable> {
        public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
throws IOException, InterruptedException {
            double sum = 0;
            int count = 0;
            for (DoubleWritable val : values) {
                sum += val.get();
                count++;
            }

            double average = sum / count;
            context.write(key, new DoubleWritable(average));
        }
    }
}

```

6. Make a jar file

Right Click on Project> Export> Select export destination as Jar File > next> Finish.

7. `hadoop fs -put ratings_small.csv ratings.csv`

8. `hadoop jar MRProgramsDemo.jar PackageDemo.MovieRating ratings.csv MRDir1`

9. `hadoop fs -ls MRDir1`

10. `hadoop fs -cat MRDir1/part-r-00000`

`Hadoop fs -cat MRDir1/part-r-00000 | sort -k2 -n | tail -n 10`

sort -k2 -n: Sorts based on the **second column** (average rating), numerically.  
tail -n 1: Gets the **last 10 lines, top 10 movies based on average rating**