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 - MusicShared).
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.Text;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;

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 MusicShared {
        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, "music shared");
                job.setJarByClass(MusicShared.class);
                job.setMapperClass(MapForMusicShared.class);
                job.setReducerClass(ReduceForMusicShared.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(Text.class);
        FileInputFormat.addInputPath(job, input);
        FileOutputFormat.setOutputPath(job, output);

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

    public static class MapForMusicShared extends Mapper<LongWritable, Text, Text, Text>
{
        public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
            String line = value.toString();
            String[] fields = line.split("[,\t]");

            if (fields.length >= 5) {
                String userId = fields[0].trim();
                String trackId = fields[1].trim();
                String shared = fields[2].trim();

                context.write(new Text(trackId), new Text(userId + "," + shared));
            }
        }
    }

    public static class ReduceForMusicShared extends Reducer<Text, Text, Text, Text> {
        public void reduce(Text trackId, Iterable<Text> values, Context context) throws
IOException, InterruptedException {
            int listenerCount = 0;
            int sharedCount = 0;

            for (Text val : values) {
                String[] parts = val.toString().split(",");
                if (parts.length == 2) {
                    listenerCount++;
                    if (parts[1].equals("1")) {
                        sharedCount++;
                    }
                }
            }

            Text result = new Text("Listeners: " + listenerCount + ", Shared: " +
sharedCount);
            context.write(trackId, result);
        }
    }
}
```

6. Make a jar file
Right Click on Project> Export> Select export destination as Jar File > next> Finish.

```
7. hadoop fs -put Music_dataset.csv music.csv

8. hadoop jar MRProgramsDemo.jar PackageDemo.MusicShared music.csv MRDir1
9. hadoop fs -ls MRDir1
10. hadoop fs -cat MRDir1/part-r-00000
```