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 - LogFile).

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 java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

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 LoginTimeAnalyzer {
  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, "Login Time Analyzer");
    job.setJarByClass(LoginTimeAnalyzer.class);
    job.setMapperClass(MapForLoginTime.class);
    job.setReducerClass(ReduceForLoginTime.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 MapForLoginTime extends Mapper<LongWritable, Text, Text,
DoubleWritable> {
    private static final SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy
HH:mm");

    public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
      try {
        String[] fields = value.toString().split(",");
        if (fields.length >= 8) {
          String mac = fields[1].trim();
          String loginTimeStr = fields[5].trim();
          String logoutTimeStr = fields[7].trim();

          Date loginTime = formatter.parse(loginTimeStr);
          Date logoutTime = formatter.parse(logoutTimeStr);

          long durationMillis = logoutTime.getTime() - loginTime.getTime();
          double durationMinutes = durationMillis / (1000.0 * 60);

          if (durationMinutes >= 0) {
            context.write(new Text(mac), new DoubleWritable(durationMinutes));
          }
        }
      } catch (Exception e) {
        // Ignore any parse or format errors silently
      }
    }
  }

  public static class ReduceForLoginTime extends Reducer<Text, DoubleWritable, Text,
DoubleWritable> {
    public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
        throws IOException, InterruptedException {
      double total = 0;
      for (DoubleWritable val : values) {
        total += val.get();
      }
      context.write(key, new DoubleWritable(total));
    }
  }
}
```

6. Make a jar file

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

7. hadoop fs -put logTime22.csv logTime22.csv

8. hadoop jar MRProgramsDemo.jar PackageDemo.LogFile logTime22 MRDir1

9. hadoop fs -ls MRDir1

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