Group B
Practical 2
Design a distributed application using MapReduce which processes a log file of a system.

LogLevelDriver.java :

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class LogLevelDriver {

   public static void main(String[] args) throws Exception {

      if (args.length != 2) {
         System.err.println("Usage: LogLevelCounter <input path> <output path>");
         System.exit(-1);
      }

      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "Log Level Counter");

      job.setJarByClass(LogLevelDriver.class);
      job.setMapperClass(LogLevelMapper.class);
      job.setReducerClass(LogLevelReducer.class);

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

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

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

LogMapper.java :

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class LogLevelMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

   private final static IntWritable one = new IntWritable(1);
   private Text logLevel = new Text();

   public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
      // Sample line: 2025-04-09 10:05:21 ERROR Unable to connect to database
      String line = value.toString();
      String[] parts = line.split(" ");
      if (parts.length >= 3) {
         logLevel.set(parts[2]); // Get the log level (INFO, ERROR, etc.)
```

```
            context.write(logLevel, one); // Emit (logLevel, 1)
        }
    }
}
```

LogLevelReducer.java :

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class LogLevelReducer 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(); // Sum all counts for each log level
        }
        context.write(key, new IntWritable(sum)); // Emit (logLevel, totalCount)
    }
}
```

system.log :

```
pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter$ cat system.log
2025-04-16 1:22:23 INFO User logged in
2025-04-16 1:22:43 ERROR Unable to connect to database
2025-04-16 1:24:22 WARN Low disk space
2025-04-16 1:30:51 INFO File uploaded successfully
```

Compile :

```
pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter$ javac -classpath `hadoop classpath` -d . LogLevelMapper.java LogLevelReducer.java LogLevelDriver.java

pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter$ jar -cvf loglevelcounter.jar *.class
```

Run On Hadoop :

```
pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter$ hadoop fs -mkdir /logdata

pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter$ hadoop fs -put system.log /logdata/

pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter$ hadoop jar loglevelcounter.jar LogLevelDriver /logdata /logdata/output
```

View Output :

```
pratik@DESKTOP-T62QGCD:~/TE_Practical/DSBDA-Group-B/LogLevelCounter hadoop fs -cat /logdata/output/part-r-00000
```

Sample Output :

```
ERROR   1
INFO    2
WARN    1
```