

Big Data: Architectures and Data Analytics

mm dd, yyyy

Student ID _____

First Name _____

Last Name _____

Use the following template for the Driver of Exercise 1

Fill in the missing parts. You can strikethrough the second job if you do not need it.

```
import ....
/* Driver class. */
public class DriverBigData extends Configured implements Tool {
    public int run(String[] args) throws Exception {
        Path inputPath = new Path(args[0]);
        Path outputDir = new Path(args[1]);
        Configuration conf = this.getConf();

        // First job
        Job job1 = Job.getInstance(conf);
        job1.setJobName("Exercise 1 - Job 1");
        // Job 1 - Input path
        FileInputFormat.addInputPath(job, _____);

        // Job 1 - Output path
        FileOutputFormat.setOutputPath(job, _____);

        // Job 1 - Driver class
        job1.setJarByClass(DriverBigData.class);

        // Job1 - Input format
        job1.setInputFormatClass(_____);

        // Job1 - Output format
        job1.setOutputFormatClass(_____);

        // Job 1 - Mapper class
        job1.setMapperClass(Mapper1BigData.class);
        // Job 1 – Mapper: Output key and output value: data types/classes
        job1.setMapOutputKeyClass(_____);

        job1.setMapOutputValueClass(_____);

        // Job 1 - Reducer class
        job1.setReducerClass(Reducer1BigData.class);

        // Job 1 – Reducer: Output key and output value: data types/classes
        job1.setOutputKeyClass(_____);

        job1.setOutputValueClass(_____);

        // Job 1 - Number of reducers
        job1.setNumReduceTasks( 0[ _ ] or 1[ _ ] or >=1[ _ ] ); /* Select only one of the three options */
```

```

// Execute the first job and wait for completion
if (job1.waitForCompletion(true)==true)
{
    // Second job
    Job job2 = Job.getInstance(conf);
    job2.setJobName("Exercise 1 - Job 2");
    // Set path of the input folder of the second job
    FileInputFormat.addInputPath(job2,_____);

    // Set path of the output folder for the second job
    FileOutputFormat.setOutputPath(job2,_____);

    // Class of the Driver for this job
    job2.setJarByClass(DriverBigData.class);

    // Set input format
    job2.setInputFormatClass(_____);

    // Set output format
    job2.setOutputFormatClass(_____);

    // Set map class
    job2.setMapperClass(Mapper2BigData.class);

    // Set map output key and value classes
    job2.setMapOutputKeyClass(_____);

    job2.setMapOutputValueClass(_____);

    // Set reduce class
    job2.setReducerClass(Reducer2BigData.class);

    // Set reduce output key and value classes
    job2.setOutputKeyClass(_____);

    job2.setOutputValueClass(_____);

    // Set number of reducers of the second job
    job2.setNumReduceTasks( 0[ _ ] or 1[ _ ] or >=1[ _ ] ); /*Select only one of the three
                                                                    options*/

    // Execute the job and wait for completion
    if (job2.waitForCompletion(true)==true)
        exitCode=0;
    else
        exitCode=1;
}
else
    exitCode=1;

return exitCode;
}
/* Main of the driver */
public static void main(String args[]) throws Exception {
int res = ToolRunner.run(new Configuration(), new DriverBigData(), args);
System.exit(res);
}
}

```