EXPERIMENT NO 5

BE SEM VII Computer Engineering (TSEC)
Name : WORDCOUNT Program in haddop

Step 1 :
Create New Java Project in Eclipse
Name project : WordCountJob
click : finish

Step 2:
Right Click on project -> New -> Class
create class files as
Name : WordCount
Name : WordMapper
Name : WordReducer

Add the respective program contents to the file which is given below :
Run The program as java application

Step 3 :
Right Click Project -> build path -> add expertnal archievs -> filesystem -> usr -> lib -> hadoop-0.20 -> hadoop-core.jar
click add

Step 4 : export the jar file in same folder (training/workspace/WordCountJob/src)

Step 5 : open terminal

```
[training@localhost ~]$ ls -l
```

Step 6 : create sample file

```
[training@localhost ~]$ cat sample.txt
hi
how are you
bye
see you soon
hi
hi
hi
```

Put the file in Hadoop

Step 7 : Put the content of sample.txt to samplehadoop.txt

```
[training@localhost ~]$ hadoop fs -put sample.txt /user/training/samplehadoop.txt
[training@localhost ~]$ hadoop fs -ls
```

Step 8 : change the directory to

```
[training@localhost ~]$ cd /home/training/workspace/WordCountJob/src
```

Step 9: Listing the contents of that directory

```
[training@localhost WordCount22]$ ls
sample.txt  wordCount.jar  WordCount.java  WordMapper.java  WordReducer.java
```

Step 10 : Run the wordcount program and collect the output in sampleoutdir

```
[training@localhost src]$ hadoop jar wordCount.jar wordCount samplehadoop.txt sampleoutdir
```

Step 11 : chk the output file

```
[training@localhost src]$ hadoop fs -ls /user/training/sampleoutdir
Found 3 items
-rw-r--r--   1 training supergroup          0 2019-08-21 00:40 /user/training/sampleoutdir/_SUCCESS
drwxr-xr-x   - training supergroup          0 2019-08-21 00:40 /user/training/sampleoutdir/_logs
-rw-r--r--   1 training supergroup         27 2019-08-21 00:40 /user/training/sampleoutdir/part-r-00000
```

Step 12: Displaying the Output

```
[training@localhost src]$ hadoop fs -cat /user/training/sampleoutdir/part-r-00000
are     1
bye     1
hi      4
how     1
see     1
```

```
soon    1
you     2
```

Prog 1 : WordCount .java

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

public class wordCount
{
public static void main(String[] args) throws Exception
{
if (args.length != 2)
{
        System.out.printf("Usage: WordCount <input dir> <output dir>\n");
        System.exit(-1);
}
Job job = new Job();
job.setJarByClass(wordCount.class);
job.setJobName("wordCount");
FileInputFormat.setInputPaths(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.setMapperClass(wordMapper.class);
job.setReducerClass(wordReducer.class);
job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(IntWritable.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
boolean success = job.waitForCompletion(true);
System.exit(success ? 0 : 1);
}
}
```

Prog 2 : WordMapper.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 wordMapper extends Mapper<LongWritable, Text, Text, IntWritable>
{
@Override
public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException
{
String line = value.toString();
for (String word : line.split("\\W+"))
{
                                        if (word.length() > 0)
                                                      context.write(new Text(word), new IntWritable(1));
                           }
}
}
```

Program 3 : WordReducer.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 wordReducer extends Reducer<Text, IntWritable, Text, IntWritable>
{
@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context)  throws IOException, InterruptedException
{
                int wordCount = 0;
                for (IntWritable value : values)
                        wordCount += value.get();
                context.write(key, new IntWritable(wordCount));
}
}
```