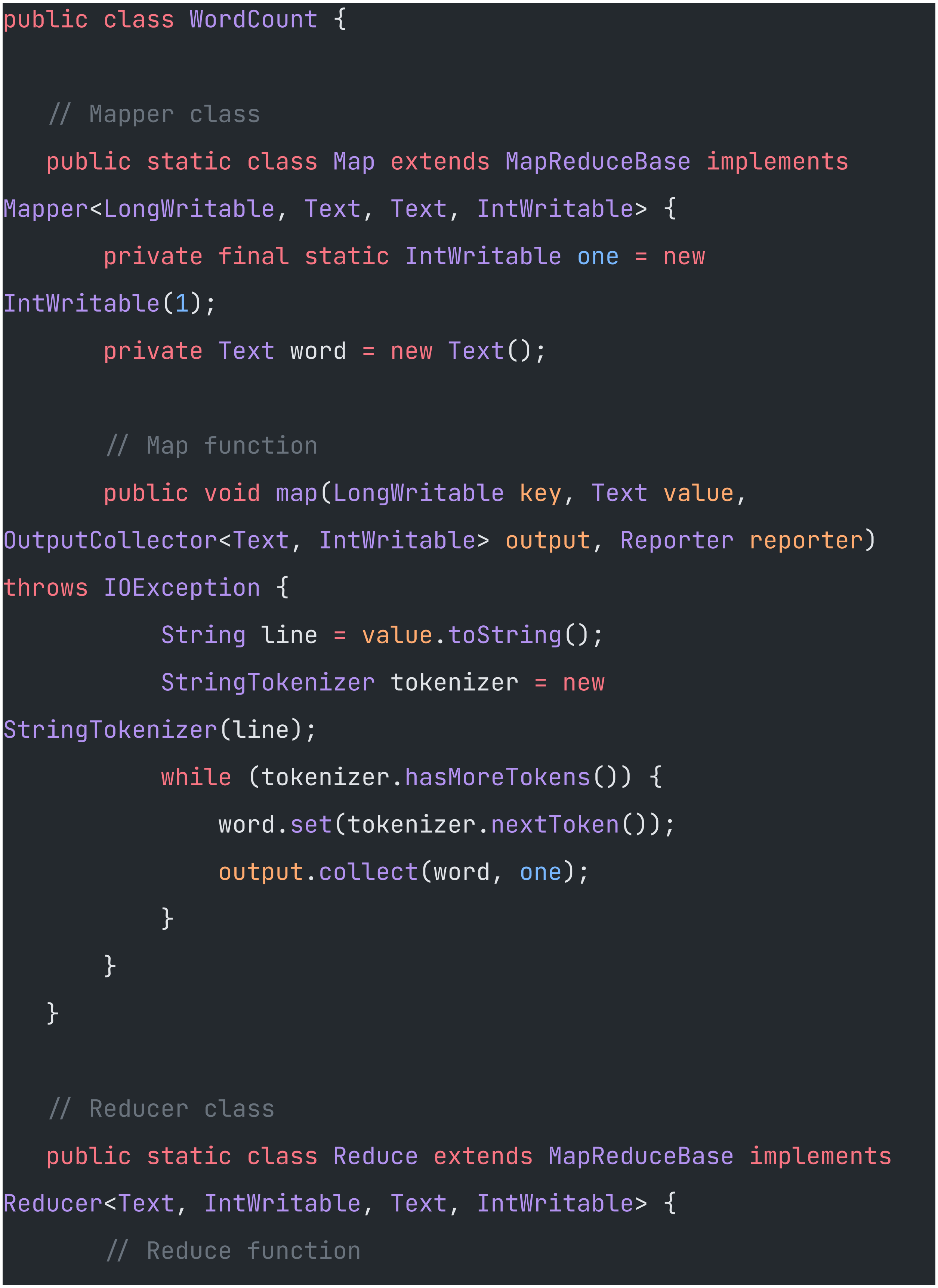
BDA
Practical 5
MapReduce algorithms

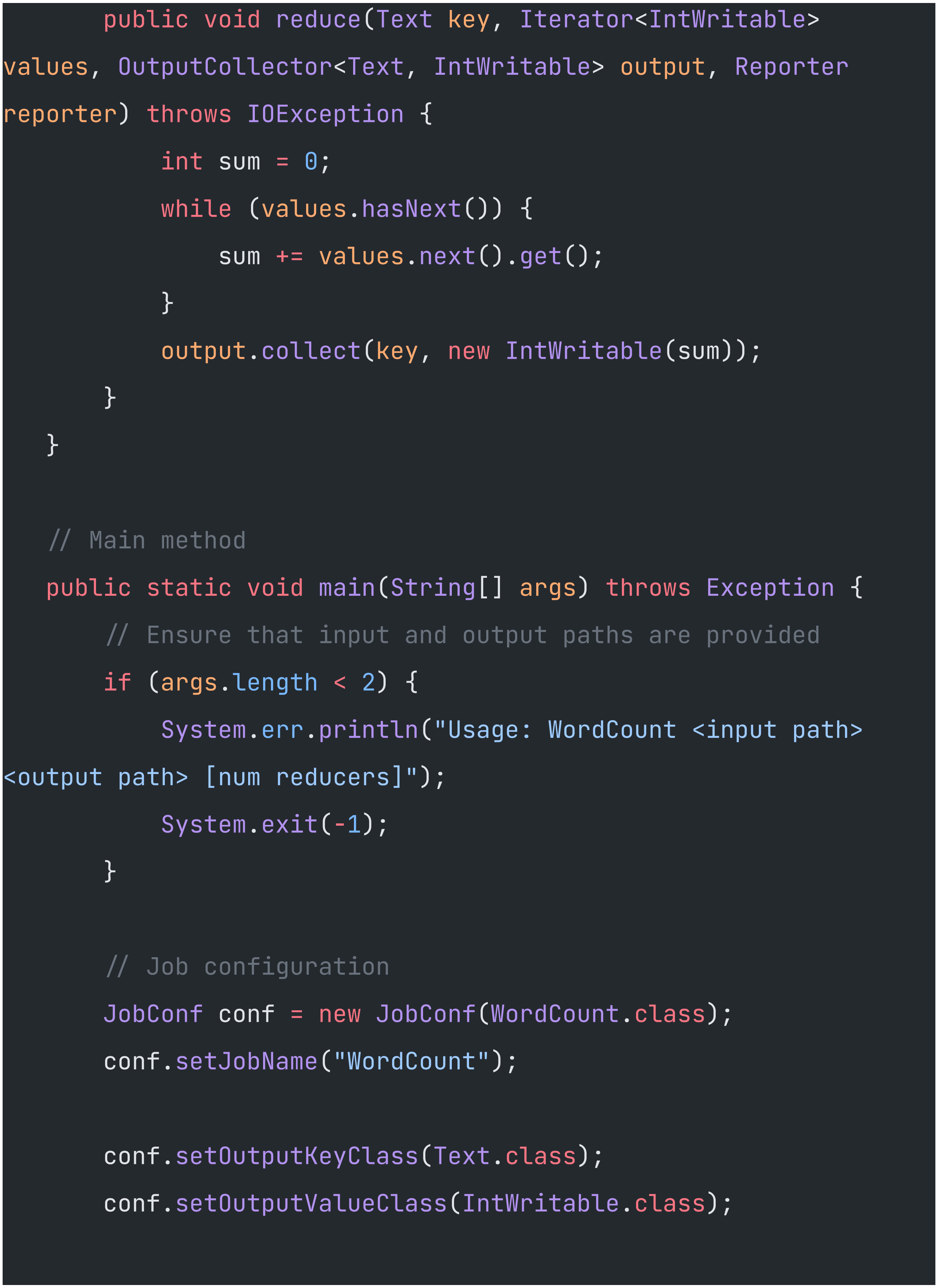
21BCE020

MapReduce Code

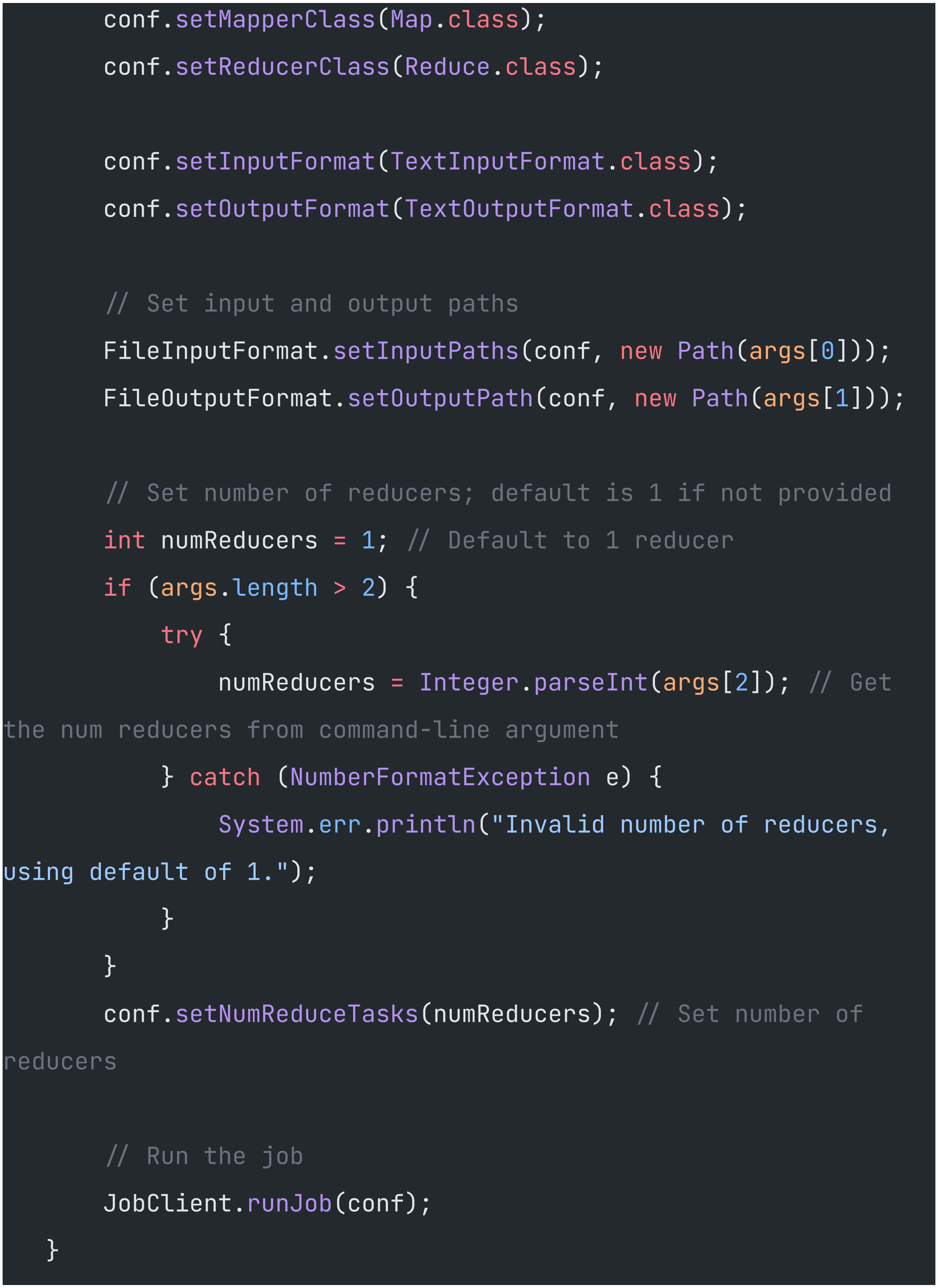
```
package com.code.dezyre;
import java.io.IOException;
import java.util.Iterator;
import java.util.StringTokenizer;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;
```



```
public class WordCount {

   // Mapper class
   public static class Map extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {
       private final static IntWritable one = new
IntWritable(1);
       private Text word = new Text();

       // Map function
       public void map(LongWritable key, Text value,
OutputCollector<Text, IntWritable> output, Reporter reporter)
throws IOException {
           String line = value.toString();
           StringTokenizer tokenizer = new
StringTokenizer(line);
           while (tokenizer.hasMoreTokens()) {
               word.set(tokenizer.nextToken());
               output.collect(word, one);
           }
       }
   }

   // Reducer class
   public static class Reduce extends MapReduceBase implements
Reducer<Text, IntWritable, Text, IntWritable> {
       // Reduce function
```



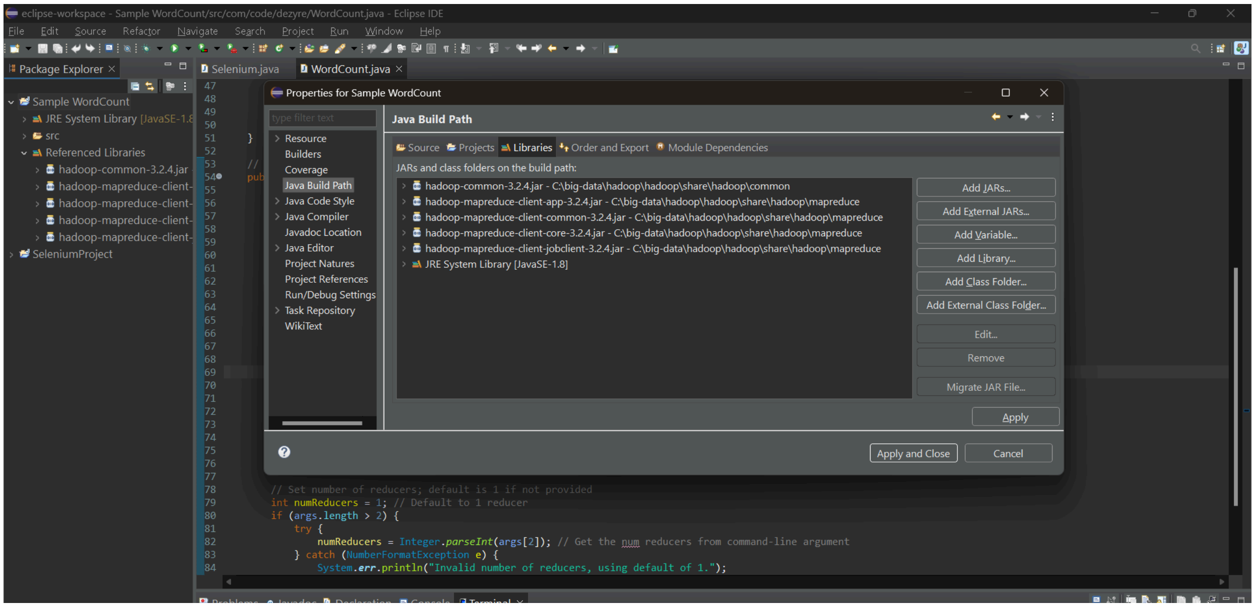
```java
        public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
            int sum = 0;
            while (values.hasNext()) {
                sum += values.next().get();
            }
            output.collect(key, new IntWritable(sum));
        }
    }

    // Main method
    public static void main(String[] args) throws Exception {
        // Ensure that input and output paths are provided
        if (args.length < 2) {
            System.err.println("Usage: WordCount <input path> <output path> [num reducers]");
            System.exit(-1);
        }

        // Job configuration
        JobConf conf = new JobConf(WordCount.class);
        conf.setJobName("WordCount");

        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
```



```java
        conf.setMapperClass(Map.class);
        conf.setReducerClass(Reduce.class);

        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        // Set input and output paths
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));

        // Set number of reducers; default is 1 if not provided
        int numReducers = 1; // Default to 1 reducer
        if (args.length > 2) {
            try {
                numReducers = Integer.parseInt(args[2]); // Get the num reducers from command-line argument
            } catch (NumberFormatException e) {
                System.err.println("Invalid number of reducers, using default of 1.");
            }
        }
        conf.setNumReduceTasks(numReducers); // Set number of reducers

        // Run the job
        JobClient.runJob(conf);
    }
```

```
}
```

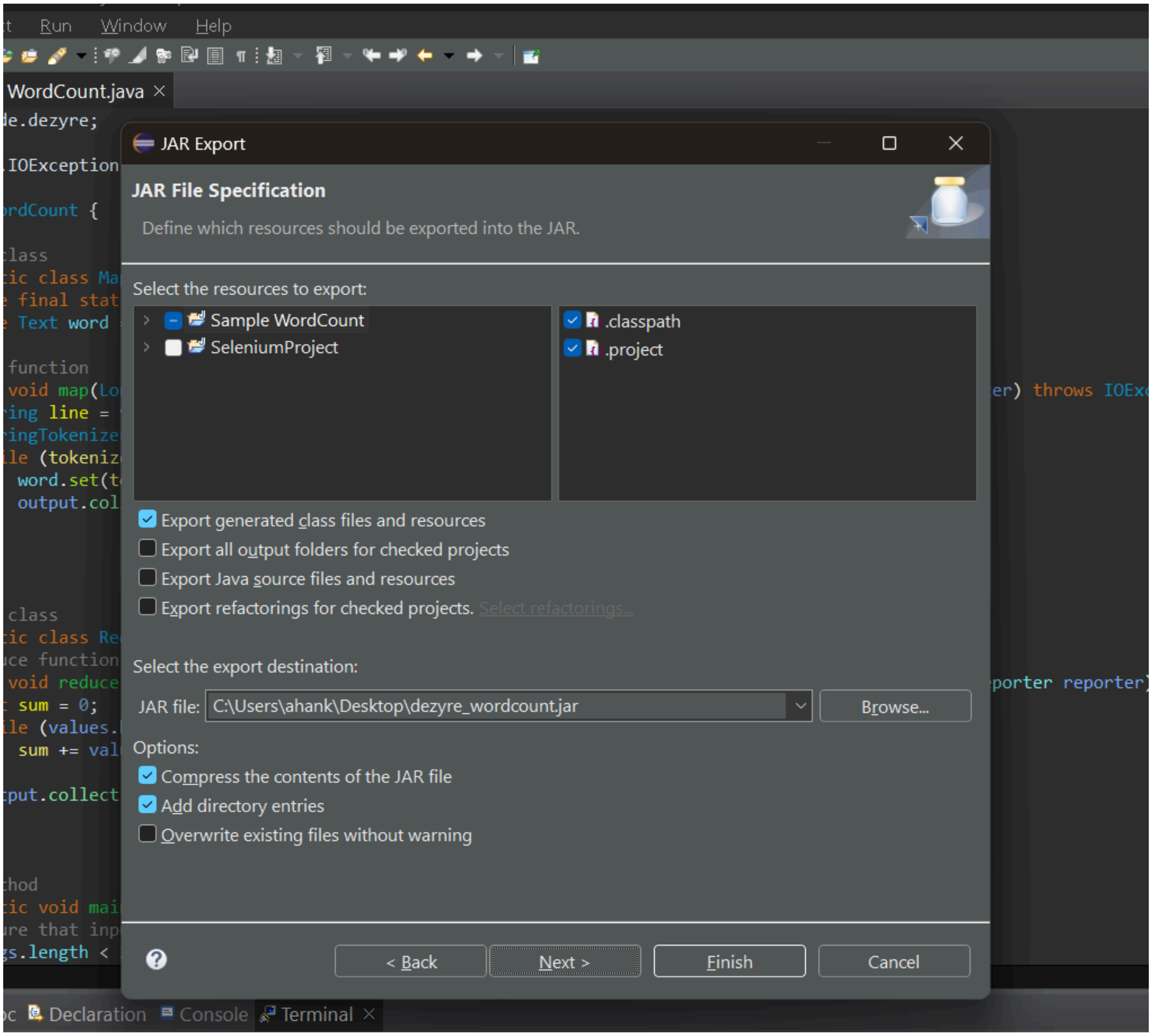
Jar Files



JARs and class folders on the build path:

- hadoop-common-3.2.4.jar - C:\big-data\hadoop\hadoop\share\hadoop\common
- hadoop-mapreduce-client-app-3.2.4.jar - C:\big-data\hadoop\hadoop\share\hadoop\mapreduce
- hadoop-mapreduce-client-common-3.2.4.jar - C:\big-data\hadoop\hadoop\share\hadoop\mapreduce
- hadoop-mapreduce-client-core-3.2.4.jar - C:\big-data\hadoop\hadoop\share\hadoop\mapreduce
- hadoop-mapreduce-client-jobclient-3.2.4.jar - C:\big-data\hadoop\hadoop\share\hadoop\mapreduce
- JRE System Library [JavaSE-1.8]

```
    // Set number of reducers; default is 1 if not provided
    int numReducers = 1; // Default to 1 reducer
    if (args.length > 2) {
        try {
            numReducers = Integer.parseInt(args[2]); // Get the num reducers from command-line argument
        } catch (NumberFormatException e) {
            System.err.println("Invalid number of reducers, using default of 1.");
```



JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- ▣ Sample WordCount
- □ SeleniumProject

- ☑ .classpath
- ☑ .project

☑ Export generated class files and resources

□ Export all output folders for checked projects

□ Export Java source files and resources

□ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: C:\Users\ahank\Desktop\dezyre_wordcount.jar   Browse...

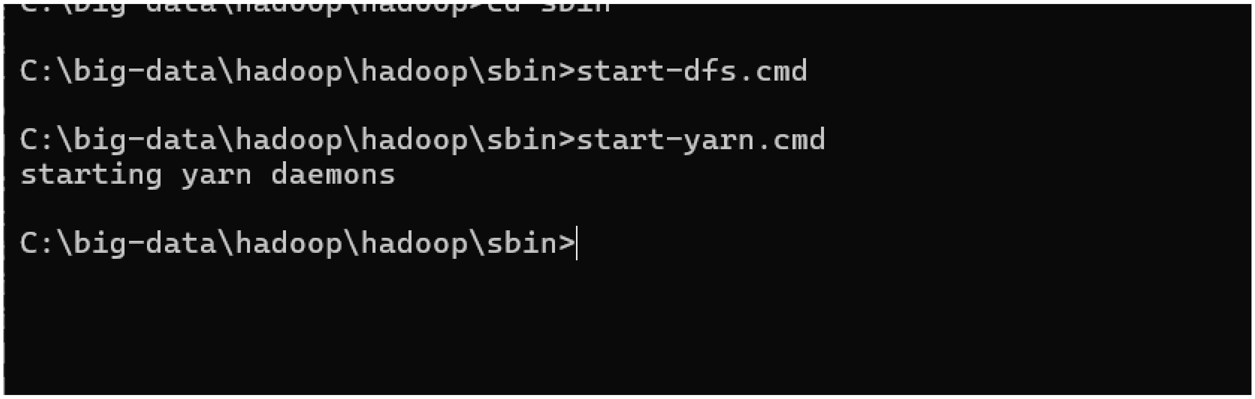
Options:

☑ Compress the contents of the JAR file

☑ Add directory entries

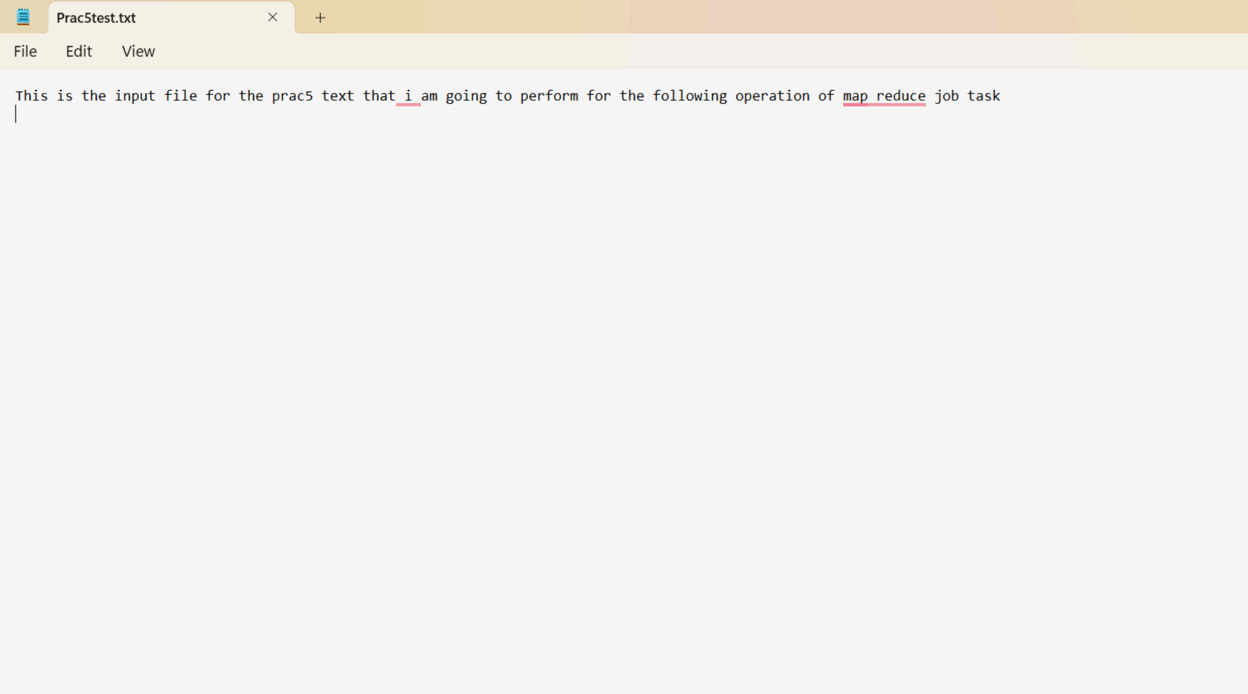
□ Overwrite existing files without warning

< Back   Next >   Finish   Cancel

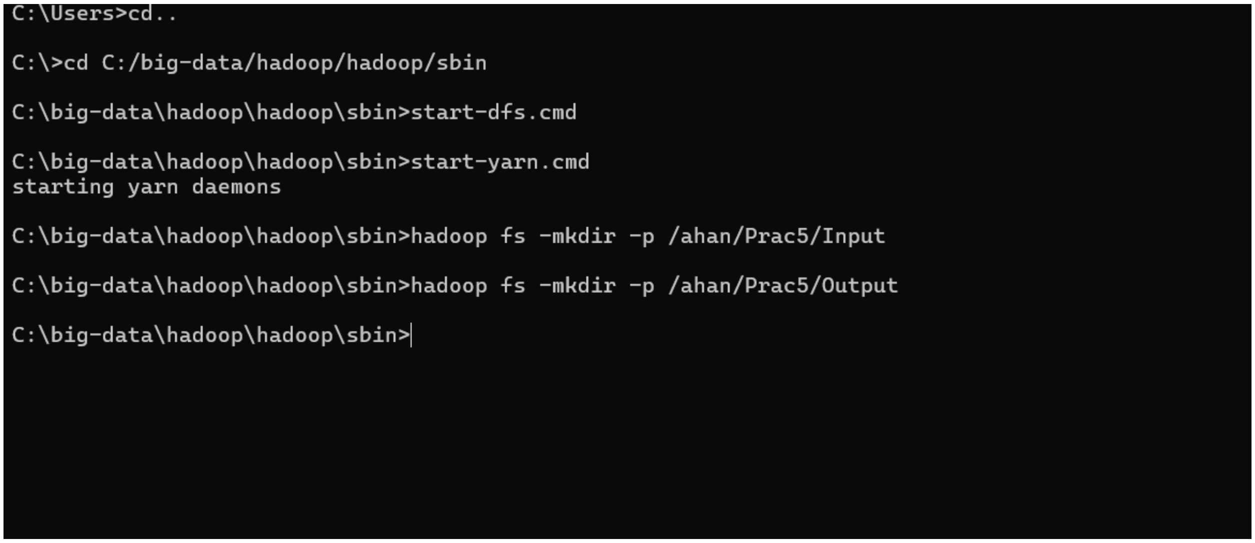


```
C:\big-data\hadoop\hadoop>cd sbin

C:\big-data\hadoop\hadoop\sbin>start-dfs.cmd

C:\big-data\hadoop\hadoop\sbin>start-yarn.cmd
starting yarn daemons

C:\big-data\hadoop\hadoop\sbin>
```

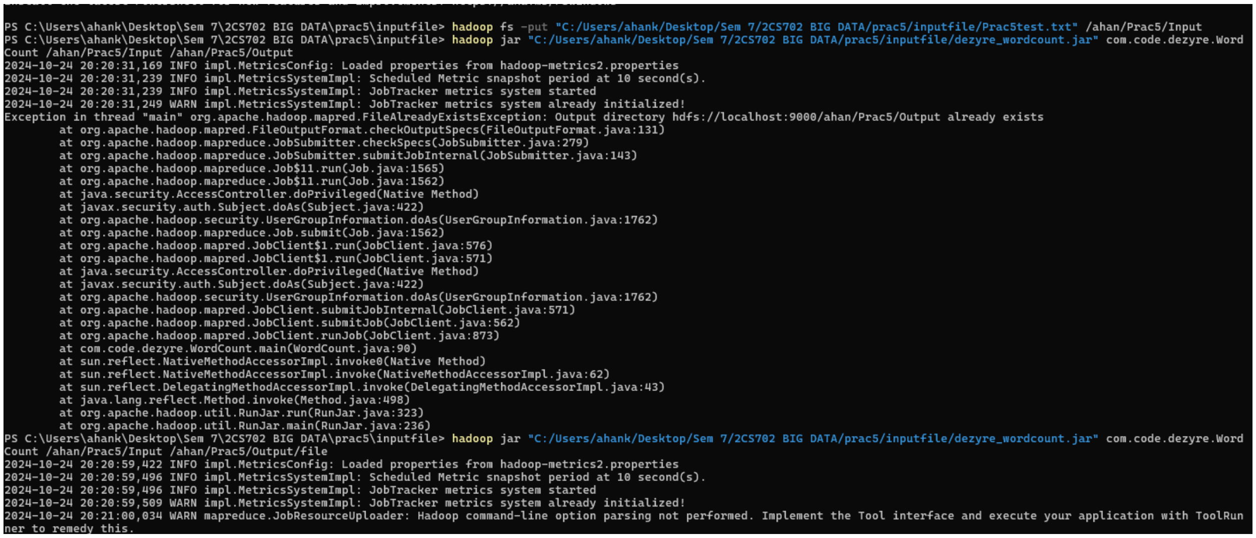
Input



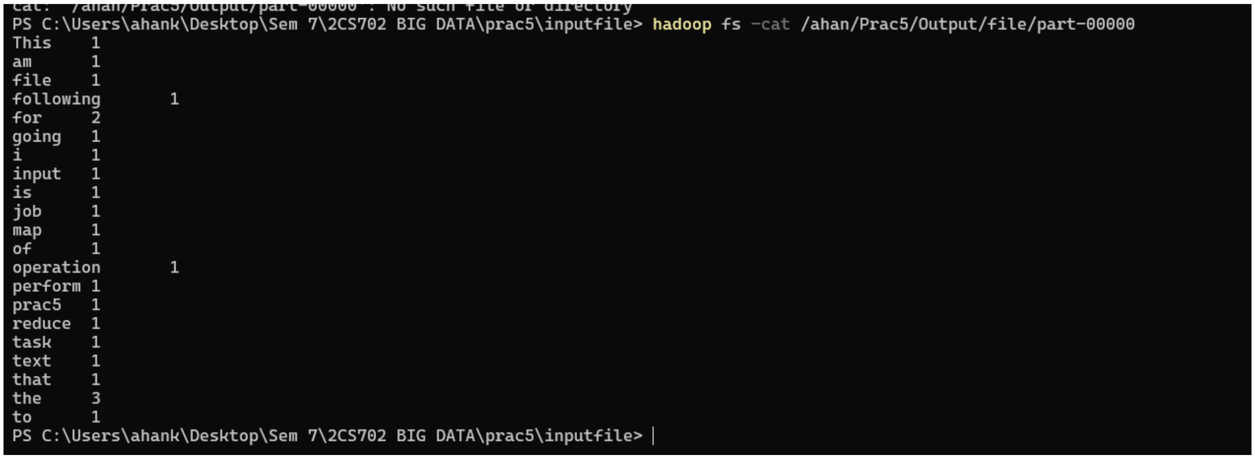
```
This is the input file for the prac5 text that i am going to perform for the following operation of map reduce job task
```



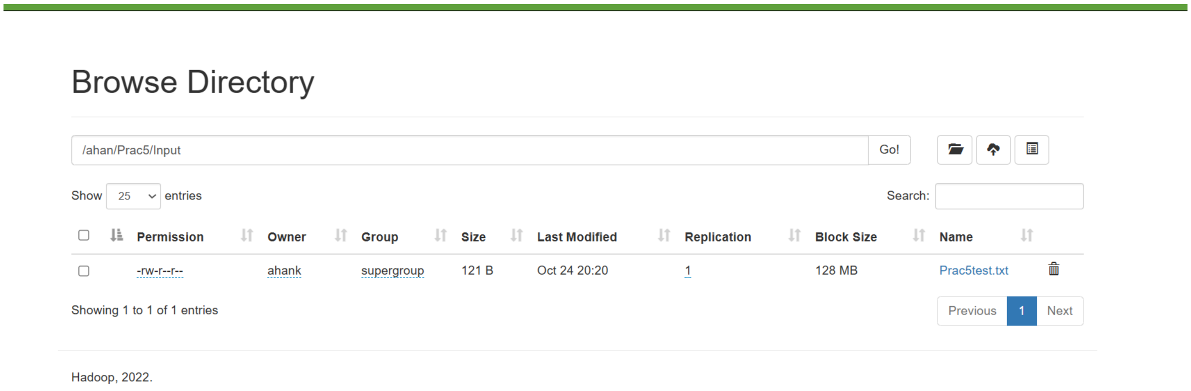
```
C:\Users>cd..

C:\>cd C:/big-data/hadoop/hadoop/sbin

C:\big-data\hadoop\hadoop\sbin>start-dfs.cmd

C:\big-data\hadoop\hadoop\sbin>start-yarn.cmd
starting yarn daemons

C:\big-data\hadoop\hadoop\sbin>hadoop fs -mkdir -p /ahan/Prac5/Input

C:\big-data\hadoop\hadoop\sbin>hadoop fs -mkdir -p /ahan/Prac5/Output

C:\big-data\hadoop\hadoop\sbin>
```



```
PS C:\Users\ahank\Desktop\Sem 7\2CS702 BIG DATA\prac5\inputfile> hadoop fs -put "C:/Users/ahank/Desktop/Sem 7/2CS702 BIG DATA/prac5/inputfile/Prac5test.txt" /ahan/Prac5/Input
PS C:\Users\ahank\Desktop\Sem 7\2CS702 BIG DATA\prac5\inputfile> hadoop jar "C:/Users/ahank/Desktop/Sem 7/2CS702 BIG DATA/prac5/inputfile/dezyre_wordcount.jar" com.code.dezyre.Word
Count /ahan/Prac5/Input /ahan/Prac5/Output
2024-10-24 20:20:31,169 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-10-24 20:20:31,239 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-10-24 20:20:31,239 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-10-24 20:20:31,249 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://localhost:9000/ahan/Prac5/Output already exists
        at org.apache.hadoop.mapred.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:131)
        at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java:279)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:143)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1565)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1562)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1762)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1562)
        at org.apache.hadoop.mapred.JobClient$1.run(JobClient.java:576)
        at org.apache.hadoop.mapred.JobClient$1.run(JobClient.java:571)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1762)
        at org.apache.hadoop.mapred.JobClient.submitJobInternal(JobClient.java:571)
        at org.apache.hadoop.mapred.JobClient.submitJob(JobClient.java:562)
        at org.apache.hadoop.mapred.JobClient.runJob(JobClient.java:873)
        at com.code.dezyre.WordCount.main(WordCount.java:90)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
PS C:\Users\ahank\Desktop\Sem 7\2CS702 BIG DATA\prac5\inputfile> hadoop jar "C:/Users/ahank/Desktop/Sem 7/2CS702 BIG DATA/prac5/inputfile/dezyre_wordcount.jar" com.code.dezyre.Word
Count /ahan/Prac5/Input /ahan/Prac5/Output/file
2024-10-24 20:20:59,422 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-10-24 20:20:59,496 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-10-24 20:20:59,496 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-10-24 20:20:59,509 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2024-10-24 20:21:00,034 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRun
ner to remedy this.
```



```
cat: /ahan/Prac5/Output/part-00000 : No such file or directory
PS C:\Users\ahank\Desktop\Sem 7\2CS702 BIG DATA\prac5\inputfile> hadoop fs -cat /ahan/Prac5/Output/file/part-00000
This    1
am      1
file    1
following       1
for     2
going   1
i       1
input   1
is      1
job     1
map     1
of      1
operation       1
perform 1
prac5   1
reduce  1
task    1
text    1
that    1
the     3
to      1
PS C:\Users\ahank\Desktop\Sem 7\2CS702 BIG DATA\prac5\inputfile> |
```



Browse Directory

/ahan/Prac5/Input

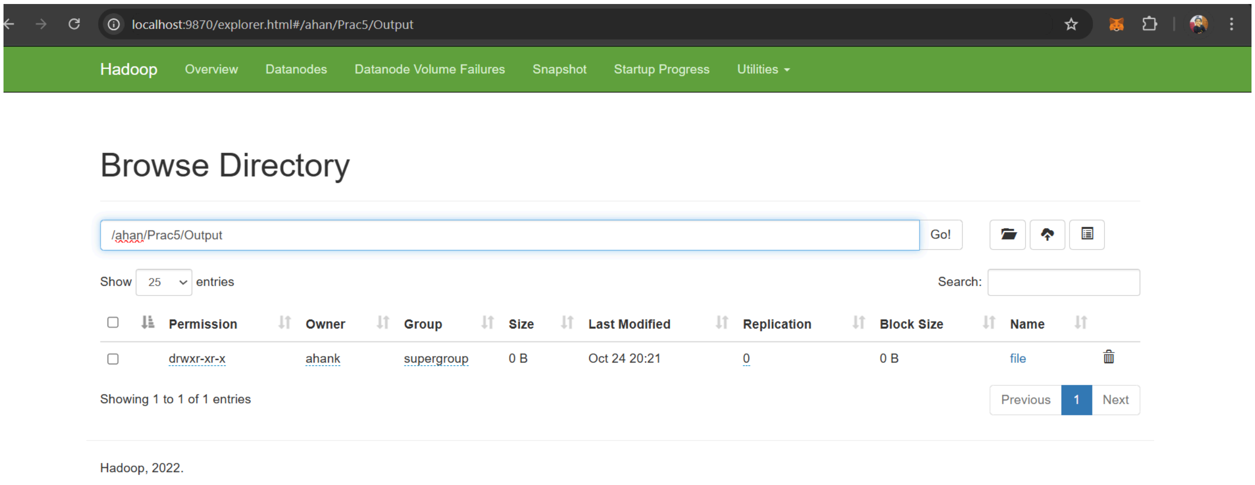
Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	ahank	supergroup	121 B	Oct 24 20:20	1	128 MB	Prac5test.txt

Showing 1 to 1 of 1 entries

Hadoop, 2022.



Browse Directory

/ahan/Prac5/Output

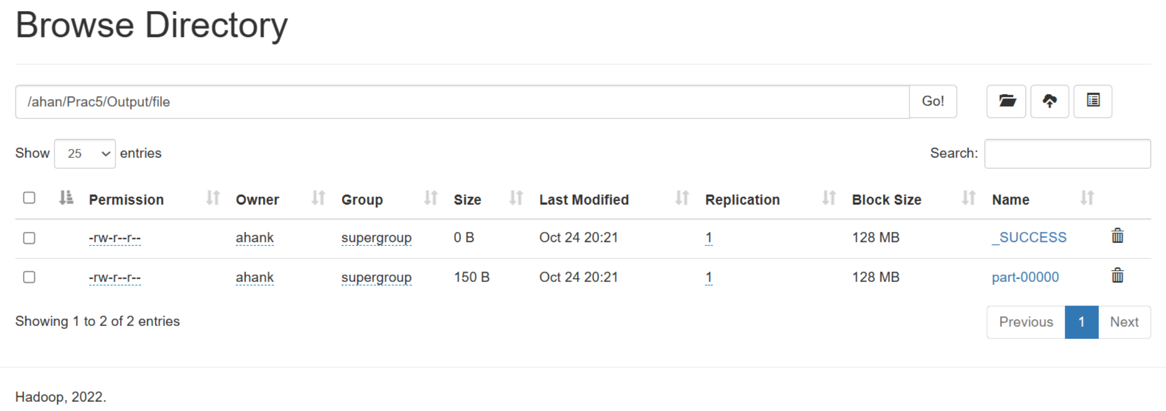
Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	drwxr-xr-x	ahank	supergroup	0 B	Oct 24 20:21	0	0 B	file	🗑

Showing 1 to 1 of 1 entries

Hadoop, 2022.



Browse Directory

/ahan/Prac5/Output/file

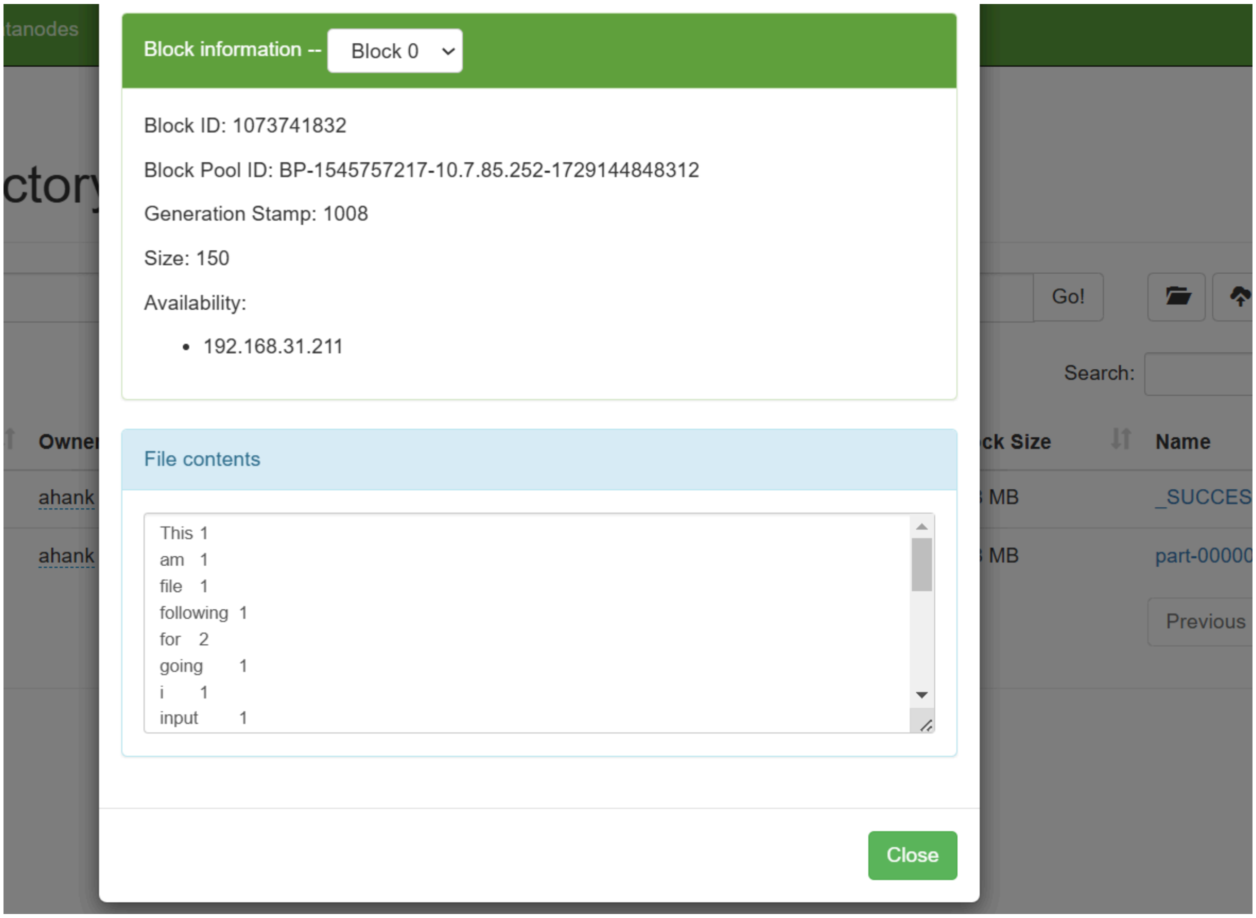
Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	ahank	supergroup	0 B	Oct 24 20:21	1	128 MB	_SUCCESS	🗑
□	-rw-r--r--	ahank	supergroup	150 B	Oct 24 20:21	1	128 MB	part-00000	🗑

Showing 1 to 2 of 2 entries

Hadoop, 2022.



Block information --

Block ID: 1073741832

Block Pool ID: BP-1545757217-10.7.85.252-1729144848312

Generation Stamp: 1008

Size: 150

Availability:

- 192.168.31.211

File contents

```
This 1
am   1
file   1
following  1
for   2
going      1
i      1
input      1
```