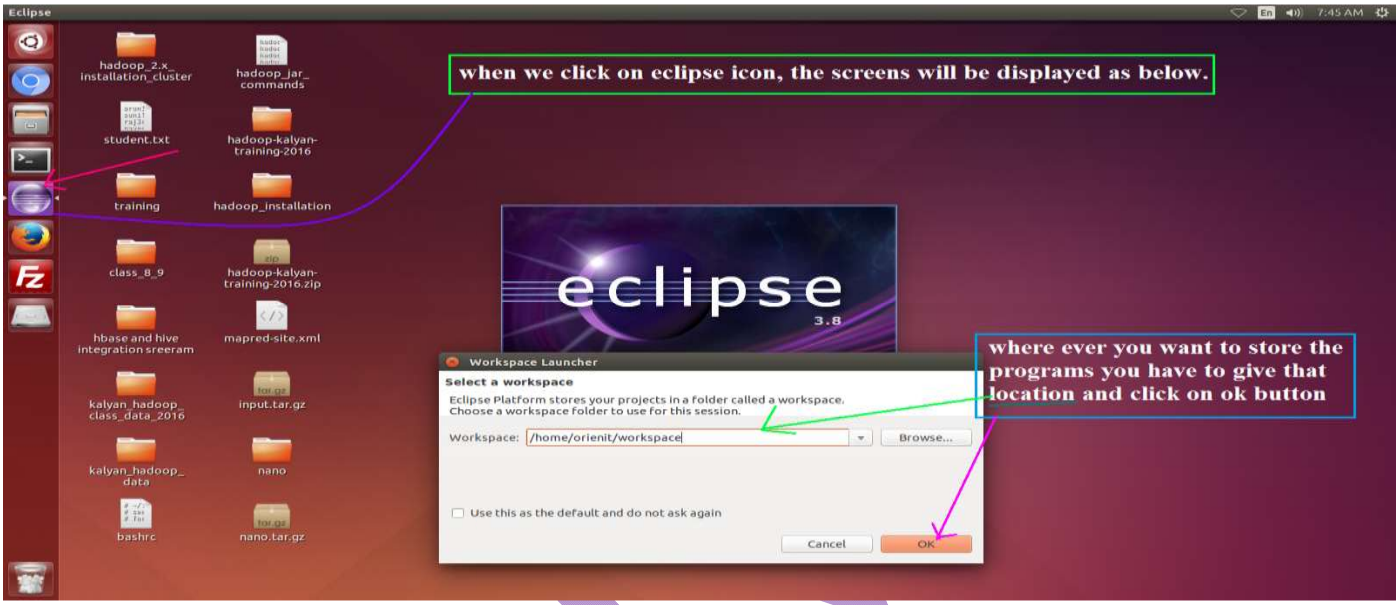
Eclipse Operations for Map Reduce Program

1. To do practice on **Map Reduce Programs** , first we need to start the **eclipse** and remaining steps follow the screen shots.
2. Provide the **workspace** path then click on **Ok** button.



when we click on eclipse icon, the screens will be displayed as below.

where ever you want to store the programs you have to give that location and click on ok button

Workspace Launcher

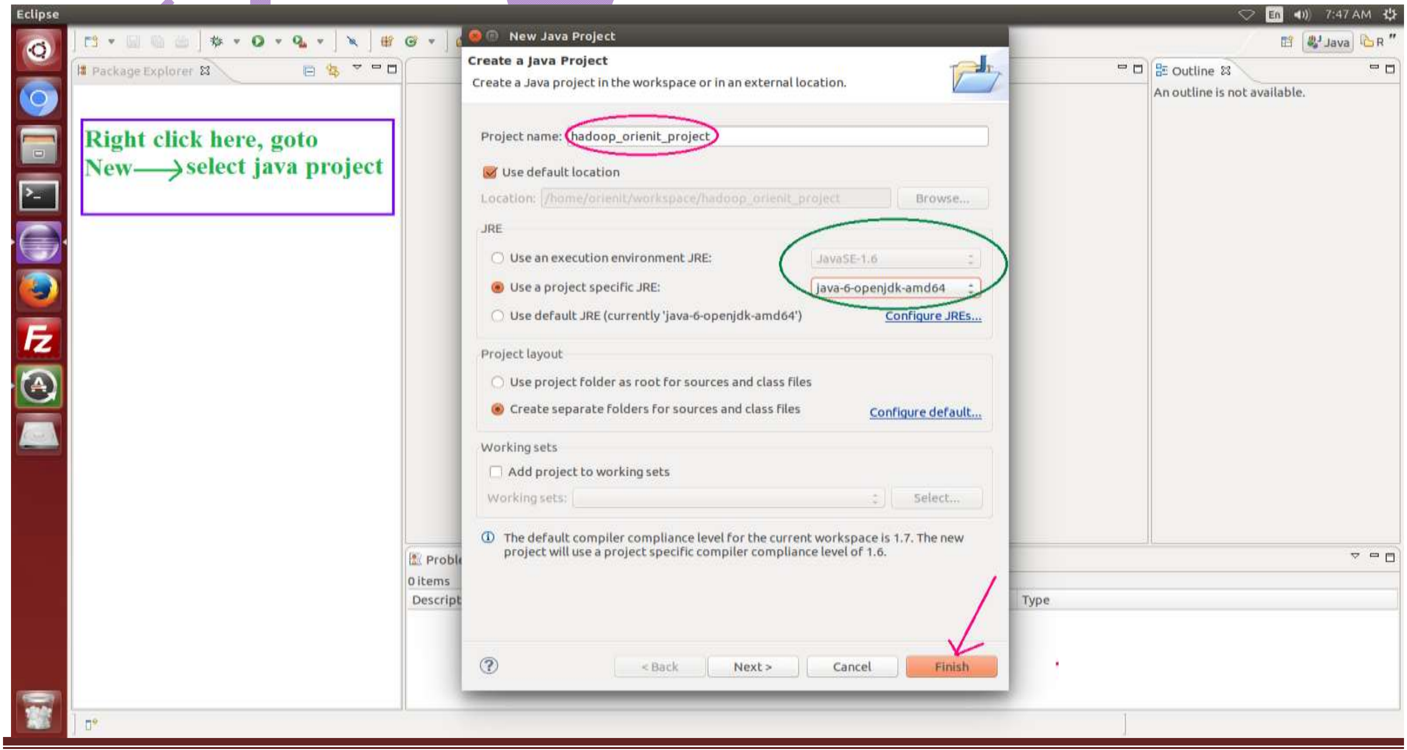
Select a workspace

Eclipse Platform stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /home/orienit/workspace

□ Use this as the default and do not ask again

3. Right click on **Package Explorer** prompt at left side of the window.
4. Go to New ⟶ select **Java Project** then provide the project name and verify **JRE**
5. Click on **Finish** button.



Right click here, goto New⟶select java project

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: hadoop_orienit_project

☑ Use default location

Location: /home/orienit/workspace/hadoop_orienit_project

JRE

○ Use an execution environment JRE: JavaSE-1.6

● Use a project specific JRE: java-6-openjdk-amd64

○ Use default JRE (currently 'java-6-openjdk-amd64')

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

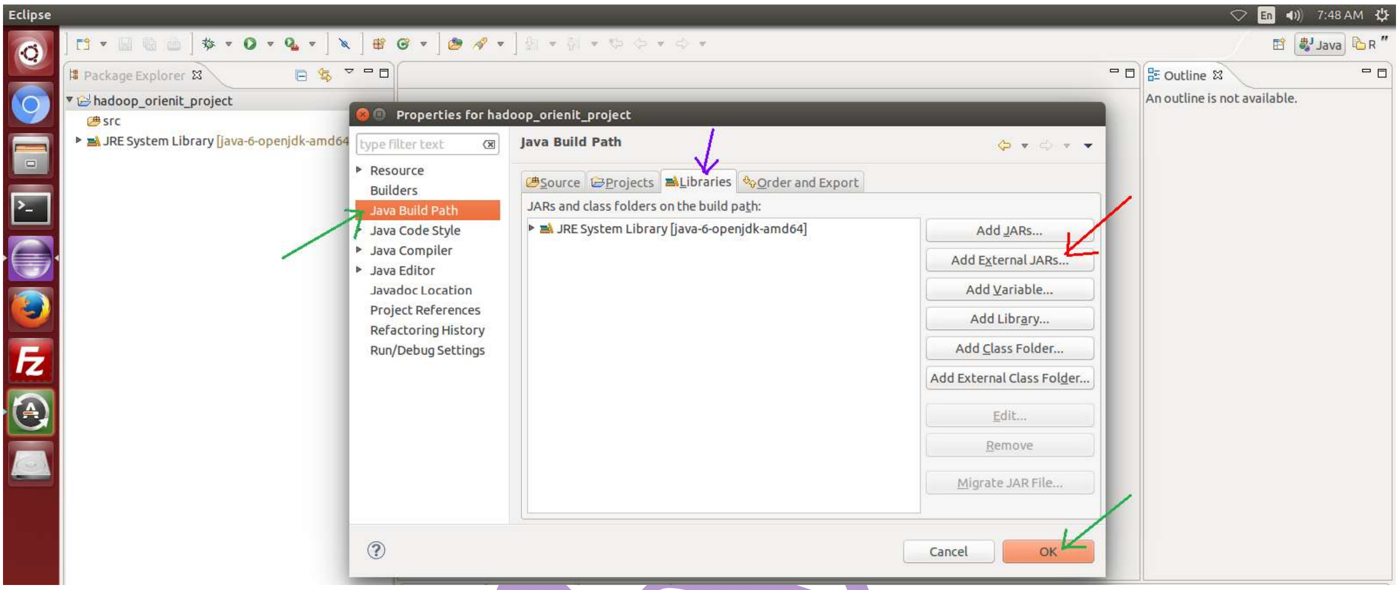
Working sets

□ Add project to working sets

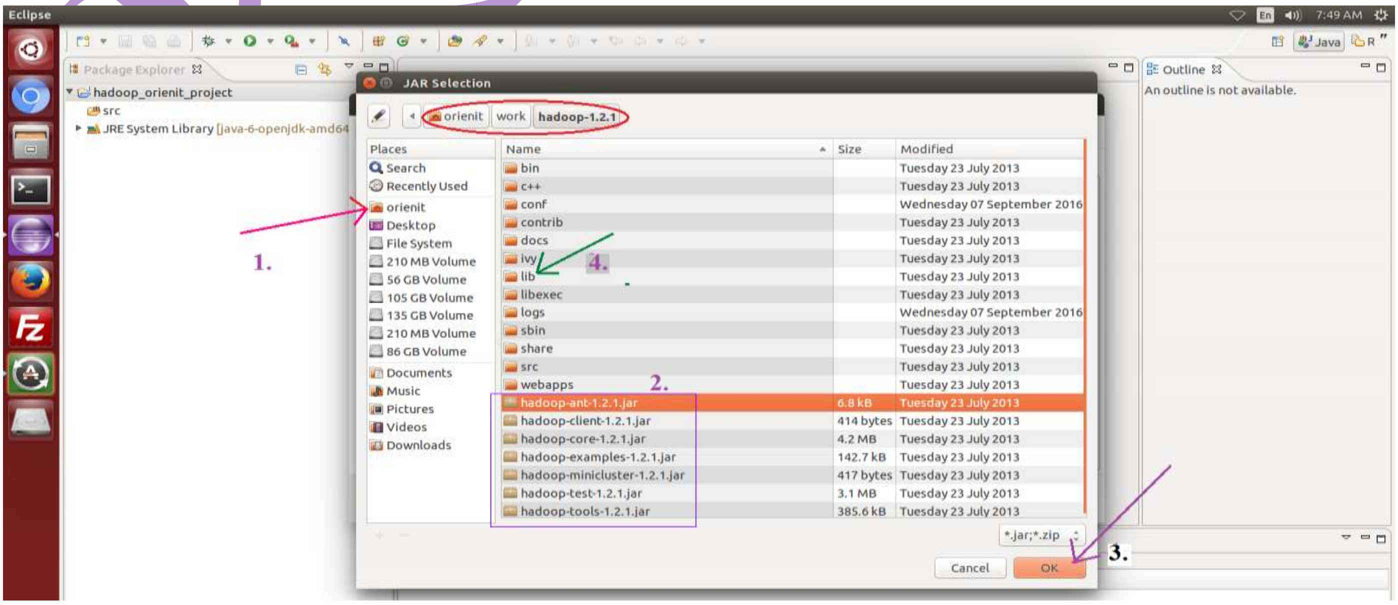
Working sets:

The default compiler compliance level for the current workspace is 1.7. The new project will use a project specific compiler compliance level of 1.6.

6. **Hadoop Local Mode Setup** follow the below steps
   a. add the **$HADOOP_HOME** folder jar files
   b. add the **$HADOOP_HOME/lib** folder jar files
7. Right click on **Project Name** goto **Properties** > **Java Build Path** then below screen open.



8. Click on **Libraries** tab
9. Click on **Add External JARS...** then below screen open.
10. Click on **orienit** folder at left side of the Window
11. Click on **work** folder
12. Click on **hadoop-1.2.1** folder
13. Select the **all the jars** from **hadoop-1.2.1** folder then click on **OK** button

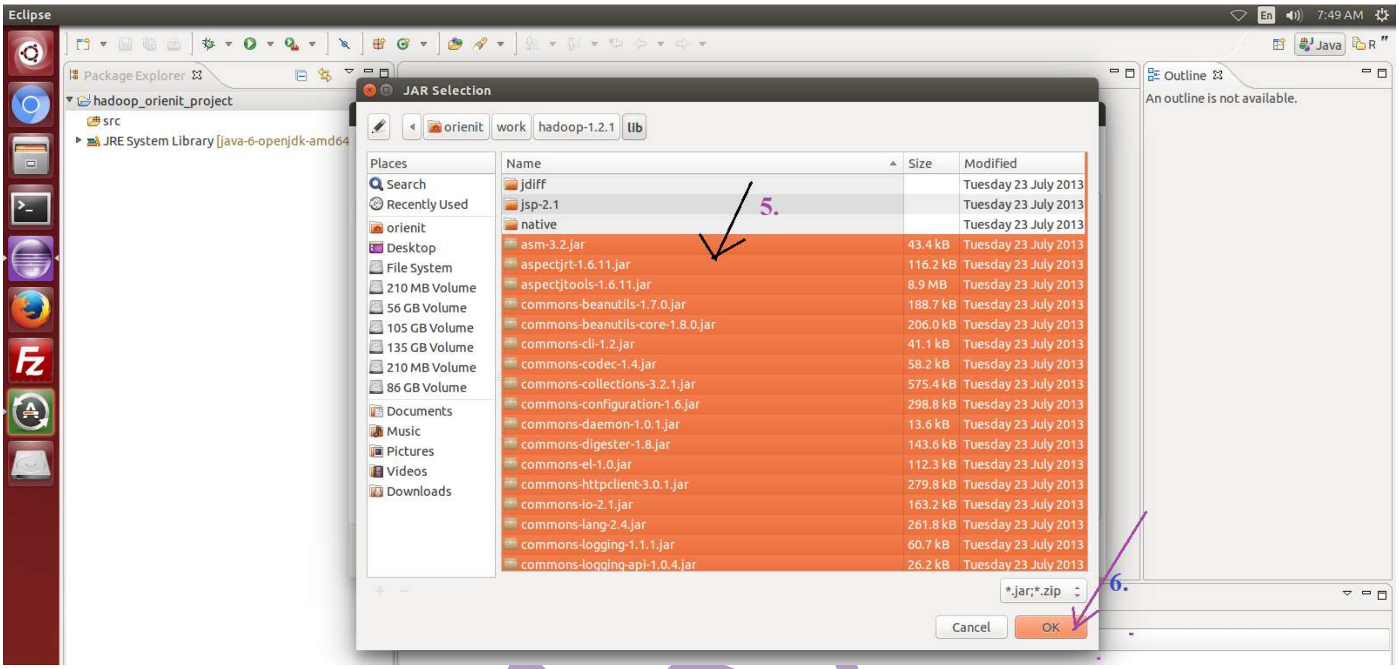


Name	Size	Modified
bin		Tuesday 23 July 2013
c++		Tuesday 23 July 2013
conf		Wednesday 07 September 2016
contrib		Tuesday 23 July 2013
docs		Tuesday 23 July 2013
ivy		Tuesday 23 July 2013
lib		Tuesday 23 July 2013
libexec		Tuesday 23 July 2013
logs		Wednesday 07 September 2016
sbin		Tuesday 23 July 2013
share		Tuesday 23 July 2013
src		Tuesday 23 July 2013
webapps		Tuesday 23 July 2013
hadoop-ant-1.2.1.jar	6.8 kB	Tuesday 23 July 2013
hadoop-client-1.2.1.jar	414 bytes	Tuesday 23 July 2013
hadoop-core-1.2.1.jar	4.2 MB	Tuesday 23 July 2013
hadoop-examples-1.2.1.jar	142.7 kB	Tuesday 23 July 2013
hadoop-minicluster-1.2.1.jar	417 bytes	Tuesday 23 July 2013
hadoop-test-1.2.1.jar	3.1 MB	Tuesday 23 July 2013
hadoop-tools-1.2.1.jar	385.6 kB	Tuesday 23 July 2013

14. Repeat once again **8 step** to **12 step**.

15. Click on **lib** folder

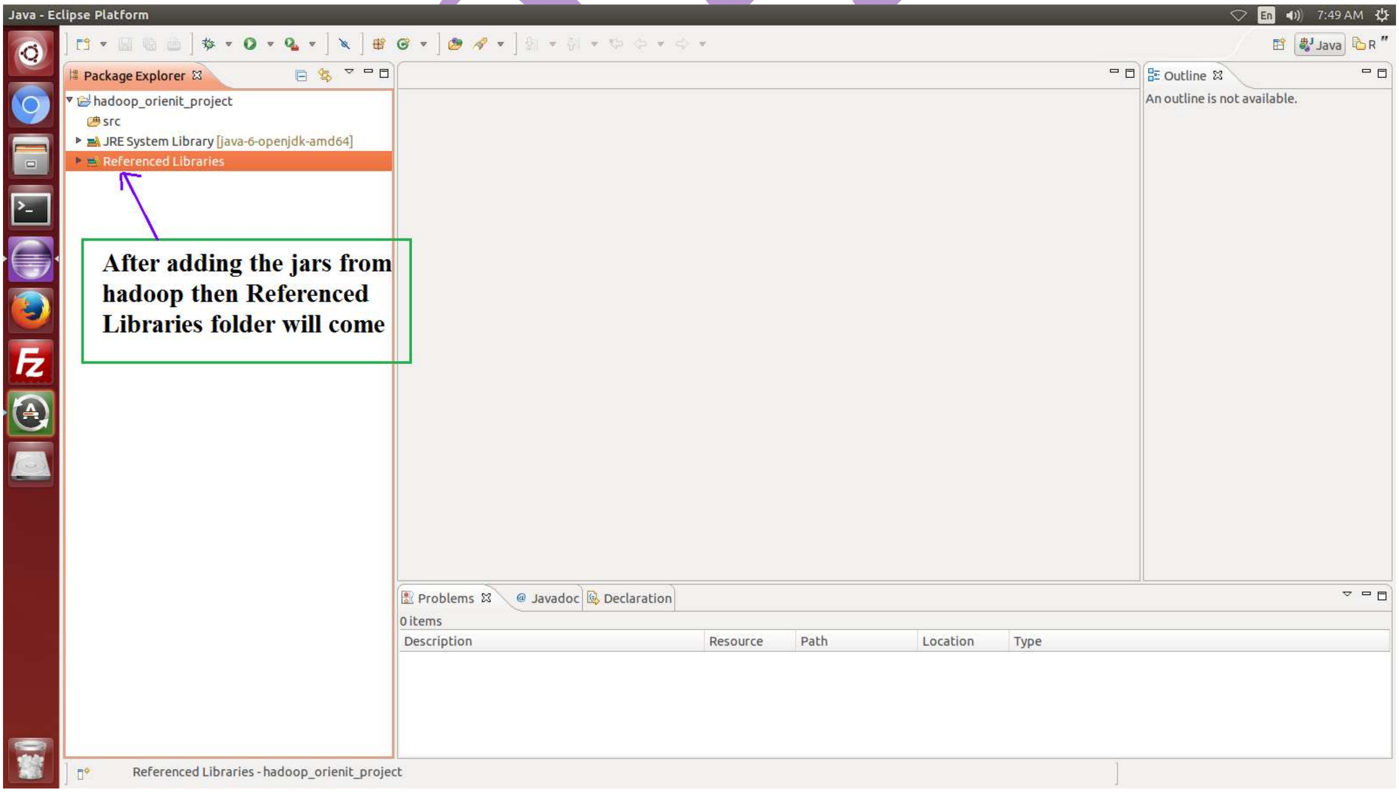
16. Select the **all the jars** from **lib** folder then click on **OK** button



Name	Size	Modified
jdiff		Tuesday 23 July 2013
jsp-2.1		Tuesday 23 July 2013
native		Tuesday 23 July 2013
asm-3.2.jar	43.4 kB	Tuesday 23 July 2013
aspectjrt-1.6.11.jar	116.2 kB	Tuesday 23 July 2013
aspectjtools-1.6.11.jar	8.9 MB	Tuesday 23 July 2013
commons-beanutils-1.7.0.jar	188.7 kB	Tuesday 23 July 2013
commons-beanutils-core-1.8.0.jar	206.0 kB	Tuesday 23 July 2013
commons-cli-1.2.jar	41.1 kB	Tuesday 23 July 2013
commons-codec-1.4.jar	58.2 kB	Tuesday 23 July 2013
commons-collections-3.2.1.jar	575.4 kB	Tuesday 23 July 2013
commons-configuration-1.6.jar	298.8 kB	Tuesday 23 July 2013
commons-daemon-1.0.1.jar	13.6 kB	Tuesday 23 July 2013
commons-digester-1.8.jar	143.6 kB	Tuesday 23 July 2013
commons-el-1.0.jar	112.3 kB	Tuesday 23 July 2013
commons-httpclient-3.0.1.jar	279.8 kB	Tuesday 23 July 2013
commons-io-2.1.jar	163.2 kB	Tuesday 23 July 2013
commons-lang-2.4.jar	261.8 kB	Tuesday 23 July 2013
commons-logging-1.1.1.jar	60.7 kB	Tuesday 23 July 2013
commons-logging-api-1.0.4.jar	26.2 kB	Tuesday 23 July 2013

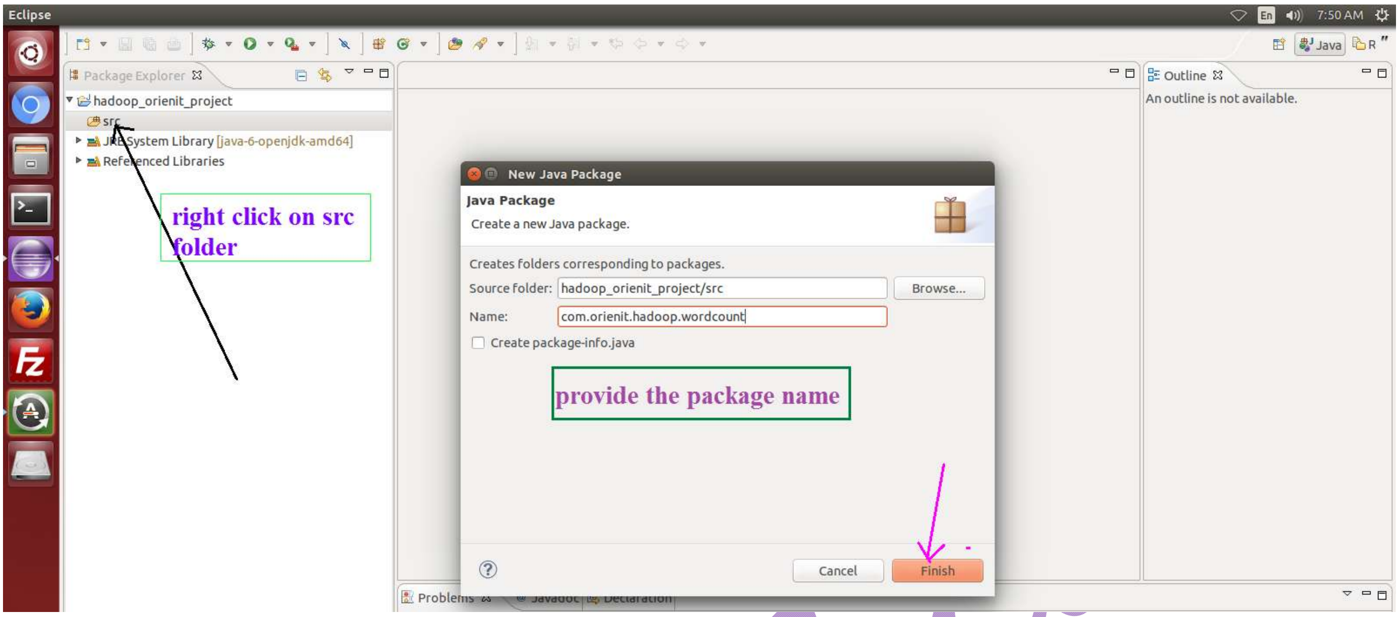
17. Now we are done with adding **all the required jars** to **classpath.**

18. Verify the **Package Explorer** screen & Expand the **Project Name** by **Double Clicking** then **Referenced Libraries** will be displayed.



After adding the jars from hadoop then Referenced Libraries folder will come

19. Right click on **src** folder and goto New ⟶ select **package** and provide the **package** name then Click on **Finish** button.



Package Explorer
- hadoop_orienit_project
  - src
  - JRE System Library [java-6-openjdk-amd64]
  - Referenced Libraries

New Java Package

Java Package

Create a new Java package.

Creates folders corresponding to packages.

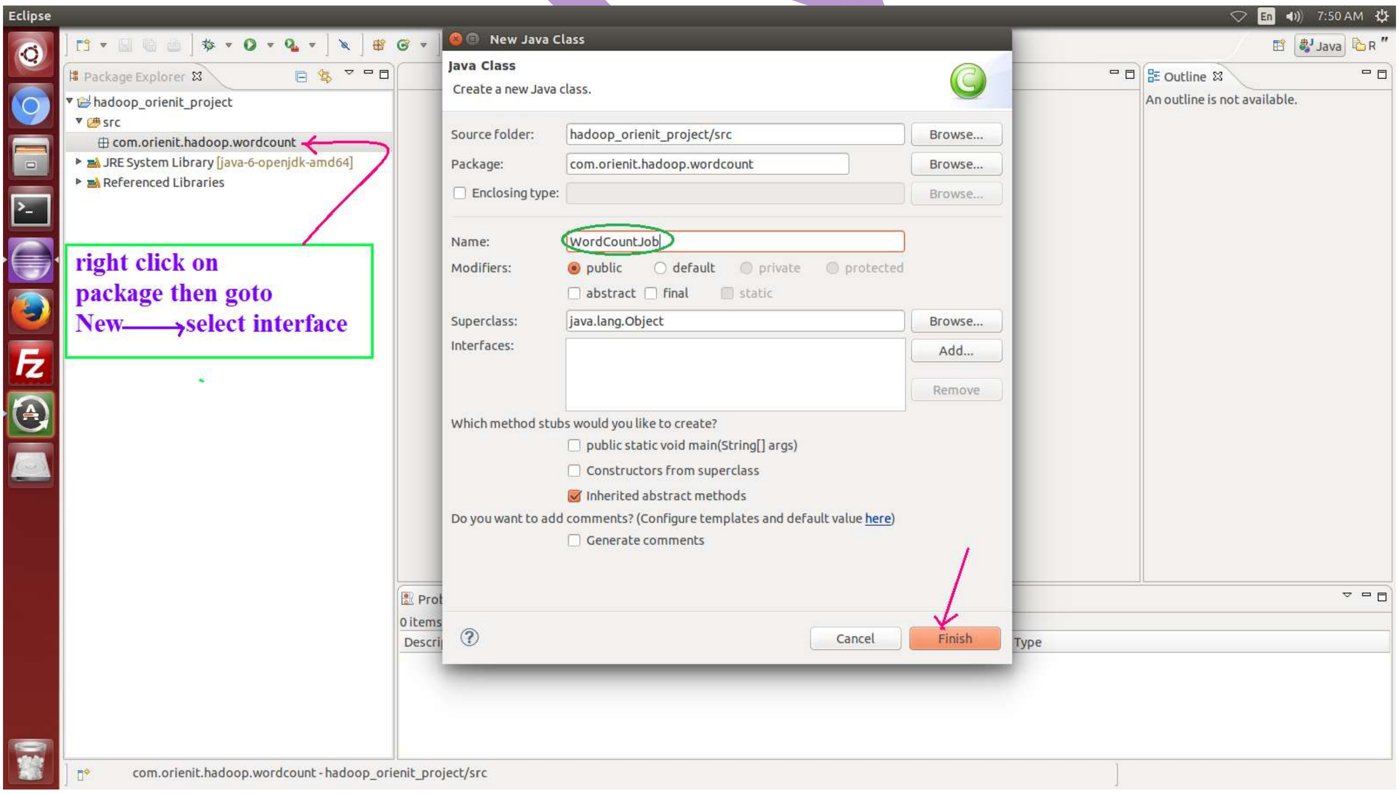
Source folder: hadoop_orienit_project/src

Name: com.orienit.hadoop.wordcount

□ Create package-info.java

Outline: An outline is not available.

20. Right click on **package** then goto New⟶ select **Class**, provide the **Class Name** (WordCountJob) then Click on Finish button.



Package Explorer
- hadoop_orienit_project
  - src
    - com.orienit.hadoop.wordcount
  - JRE System Library [java-6-openjdk-amd64]
  - Referenced Libraries

New Java Class

Java Class

Create a new Java class.

Source folder: hadoop_orienit_project/src

Package: com.orienit.hadoop.wordcount

□ Enclosing type:

Name: WordCountJob

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

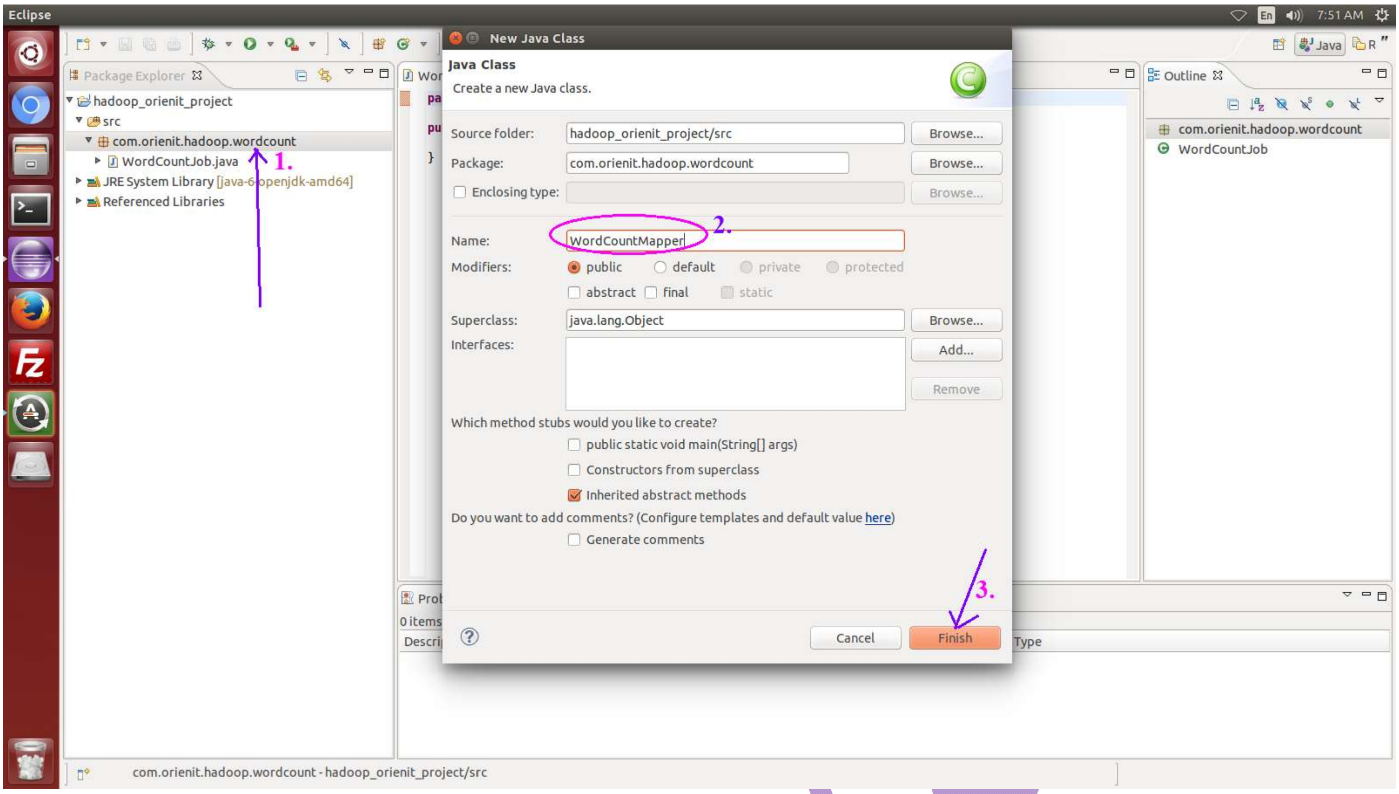
Do you want to add comments? (Configure templates and default value here)

□ Generate comments

Outline: An outline is not available.

com.orienit.hadoop.wordcount - hadoop_orienit_project/src

21. Right click on **package** then goto New⟶ select **Class**, provide the **Class Name** (WordCountMapper) then Click on Finish button.


Eclipse
Package Explorer
hadoop_orienit_project
src
com.orienit.hadoop.wordcount
WordCountJob.java
JRE System Library [java-6-openjdk-amd64]
Referenced Libraries
1.
2.
3.
Outline
com.orienit.hadoop.wordcount
WordCountJob
Cancel
Finish
com.orienit.hadoop.wordcount - hadoop_orienit_project/src

New Java Class

Java Class

Create a new Java class.

Source folder: hadoop_orienit_project/src

Package: com.orienit.hadoop.wordcount

☐ Enclosing type:

Name: WordCountMapper

Modifiers: ● public ○ default ○ private ○ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

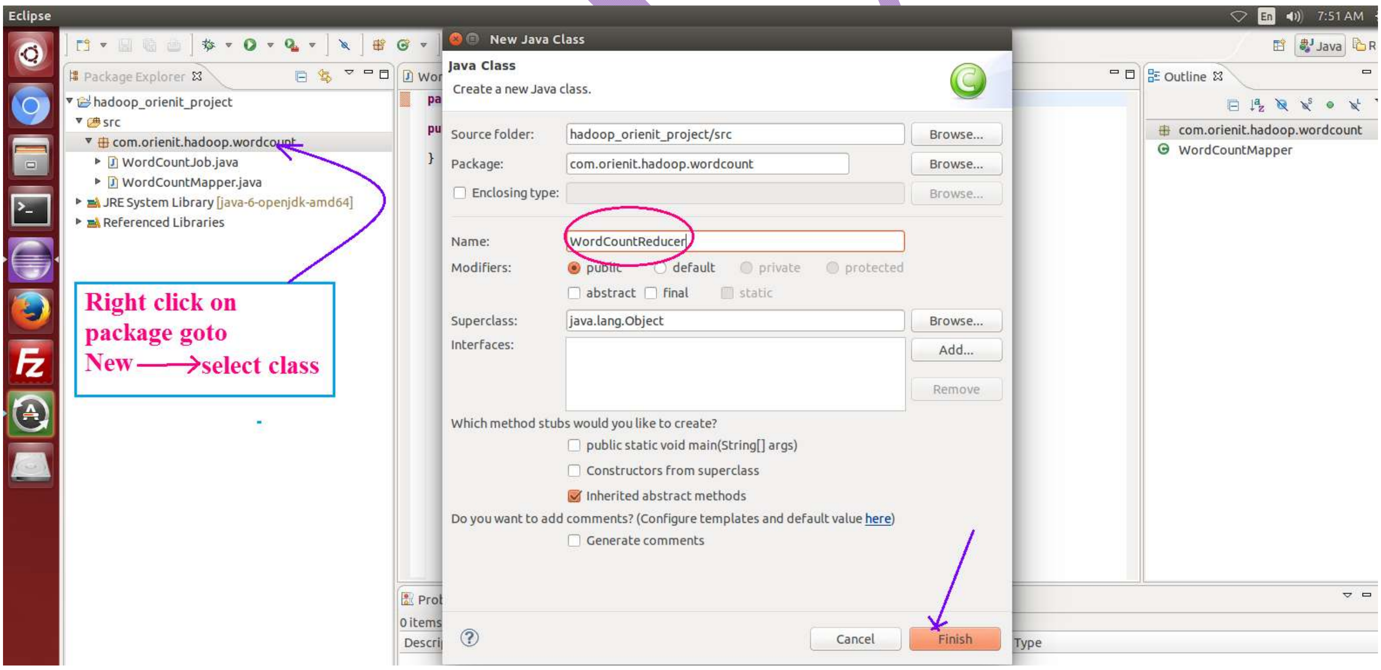
☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

22. Right click on **package** then goto New ⟶ select **Class**, provide the **Class Name** (WordCountReducer) then Click on Finish button.


Eclipse
Package Explorer
hadoop_orienit_project
src
com.orienit.hadoop.wordcount
WordCountJob.java
WordCountMapper.java
JRE System Library [java-6-openjdk-amd64]
Referenced Libraries
Right click on package goto New ⟶ select class
Outline
com.orienit.hadoop.wordcount
WordCountMapper
Cancel
Finish

New Java Class

Java Class

Create a new Java class.

Source folder: hadoop_orienit_project/src

Package: com.orienit.hadoop.wordcount

☐ Enclosing type:

Name: WordCountReducer

Modifiers: ● public ○ default ○ private ○ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

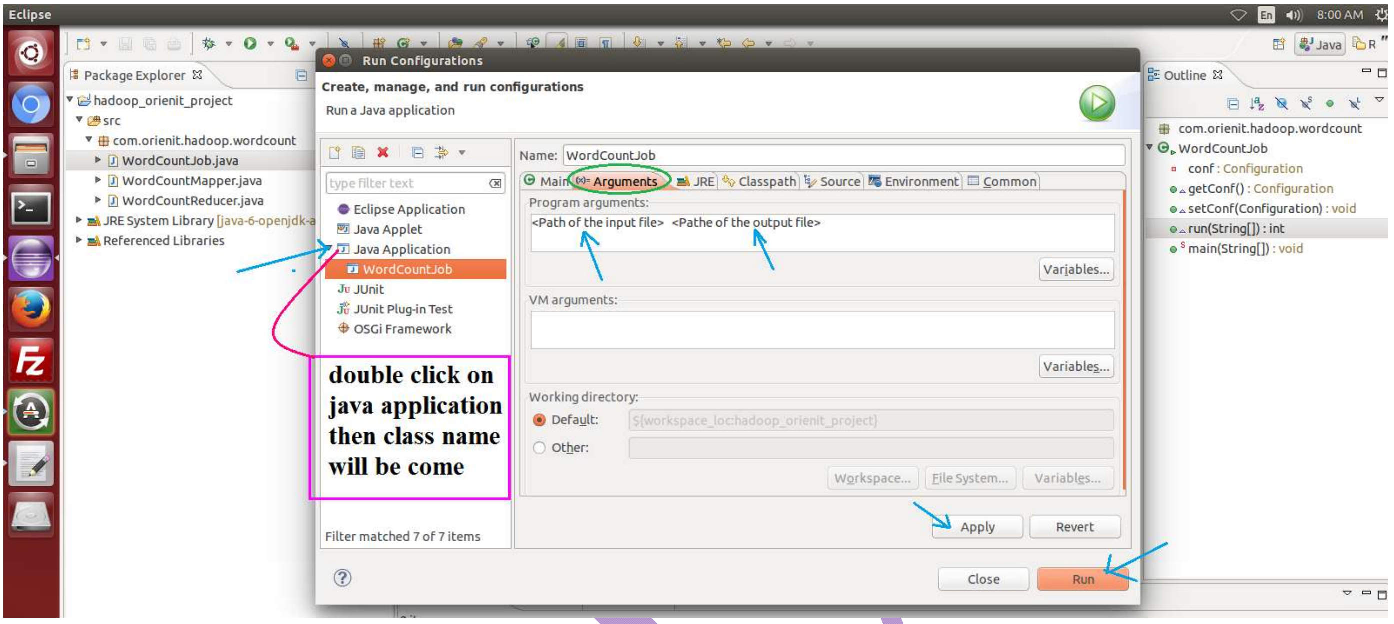
23. After writing the **complete code**, then we need to **Run** the program follow the below steps to proceed.

24. Right Click on "**WordCountJob**" program select **Run > Run Configurations** then below window will be displayed.

25. Double click on **Java Application** then main class name will be come

26. Click on main class name (**WordCountJob**) and select **Arguments** tab then provide the **input path of the file / folder** and provide the **path of the output folder**.

27. Click on **Apply** button and then Click on **Run** button.



Run Configurations

Create, manage, and run configurations

Run a Java application

Name: WordCountJob

Program arguments:

```
<Path of the input file>  <Pathe of the output file>
```

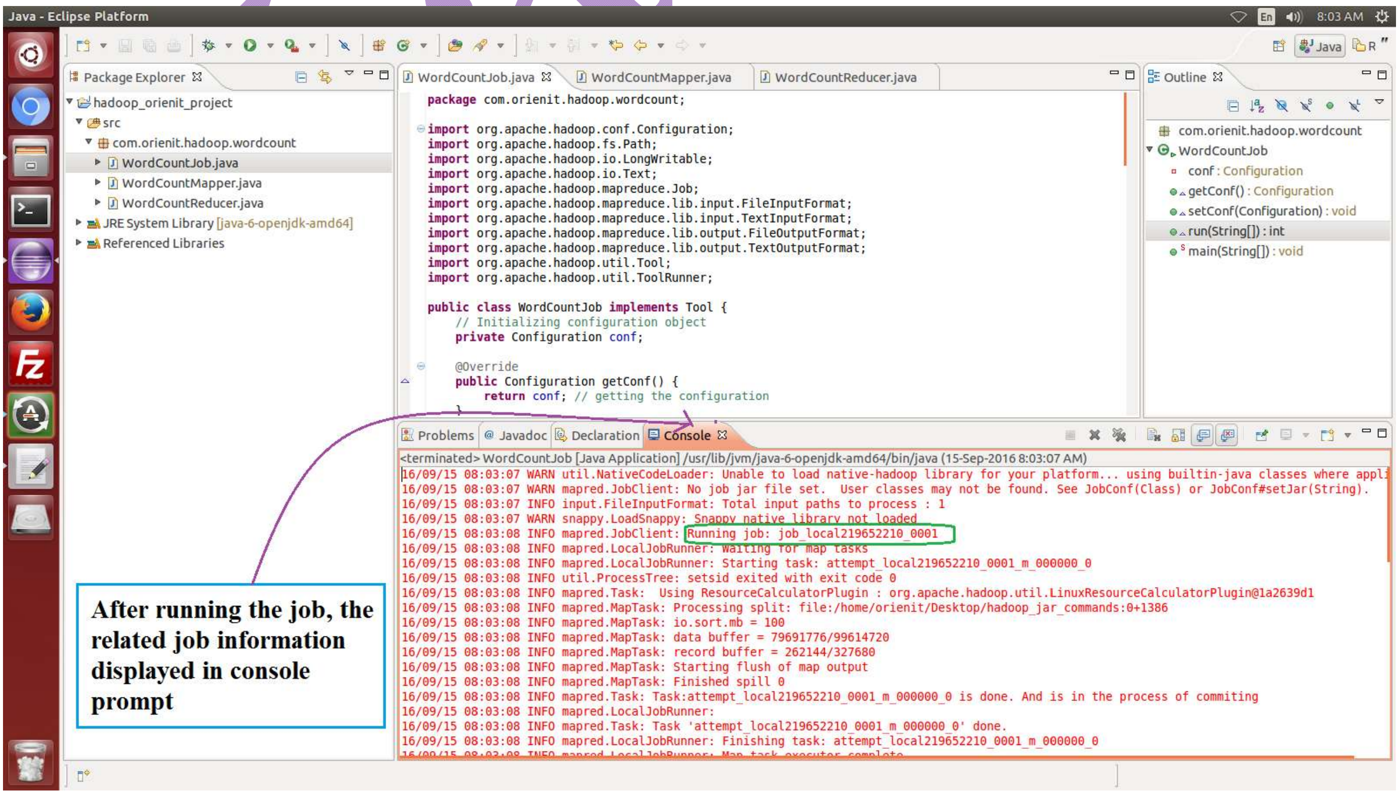
Working directory:

● Default: ${workspace_loc:hadoop_orienit_project}

○ Other:

double click on java application then class name will be come

28. After click on **Run** button, then the message will be displayed on **Console** prompt see below screen shot.

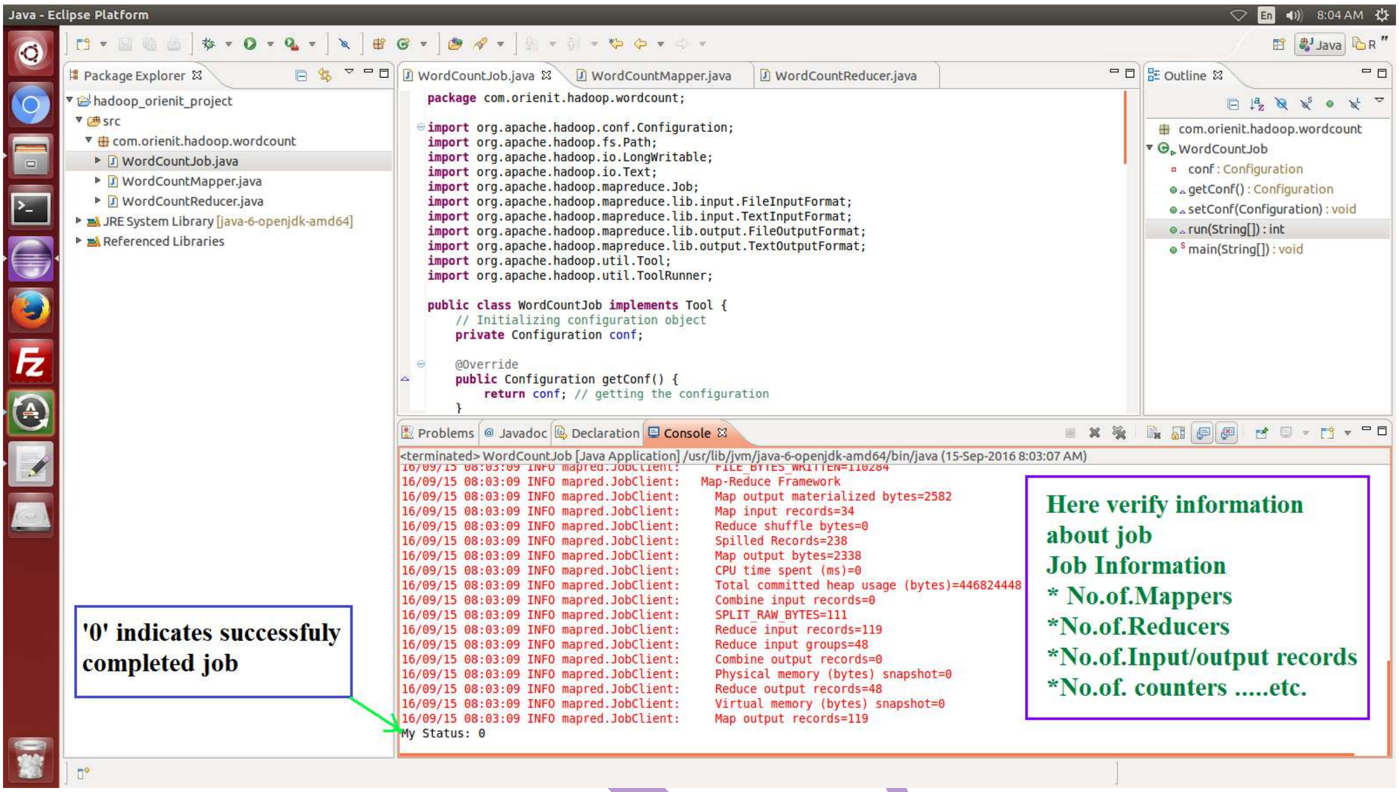


```java
package com.orienit.hadoop.wordcount;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
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.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WordCountJob implements Tool {
    // Initializing configuration object
    private Configuration conf;

    @Override
    public Configuration getConf() {
        return conf; // getting the configuration
    }
```

```
<terminated> WordCountJob [Java Application] /usr/lib/jvm/java-6-openjdk-amd64/bin/java (15-Sep-2016 8:03:07 AM)
16/09/15 08:03:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where appli
16/09/15 08:03:07 WARN mapred.JobClient: No job jar file set.  User classes may not be found. See JobConf(Class) or JobConf#setJar(String).
16/09/15 08:03:07 INFO input.FileInputFormat: Total input paths to process : 1
16/09/15 08:03:07 WARN snappy.LoadSnappy: Snappy native library not loaded
16/09/15 08:03:08 INFO mapred.JobClient: Running job: job_local219652210_0001
16/09/15 08:03:08 INFO mapred.LocalJobRunner: Waiting for map tasks
16/09/15 08:03:08 INFO mapred.LocalJobRunner: Starting task: attempt_local219652210_0001_m_000000_0
16/09/15 08:03:08 INFO util.ProcessTree: setsid exited with exit code 0
16/09/15 08:03:08 INFO mapred.Task:  Using ResourceCalculatorPlugin : org.apache.hadoop.util.LinuxResourceCalculatorPlugin@1a2639d1
16/09/15 08:03:08 INFO mapred.MapTask: Processing split: file:/home/orienit/Desktop/hadoop_jar_commands:0+1386
16/09/15 08:03:08 INFO mapred.MapTask: io.sort.mb = 100
16/09/15 08:03:08 INFO mapred.MapTask: data buffer = 79691776/99614720
16/09/15 08:03:08 INFO mapred.MapTask: record buffer = 262144/327680
16/09/15 08:03:08 INFO mapred.MapTask: Starting flush of map output
16/09/15 08:03:08 INFO mapred.MapTask: Finished spill 0
16/09/15 08:03:08 INFO mapred.Task: Task:attempt_local219652210_0001_m_000000_0 is done. And is in the process of commiting
16/09/15 08:03:08 INFO mapred.LocalJobRunner:
16/09/15 08:03:08 INFO mapred.Task: Task 'attempt_local219652210_0001_m_000000_0' done.
16/09/15 08:03:08 INFO mapred.LocalJobRunner: Finishing task: attempt_local219652210_0001_m_000000_0
16/09/15 08:03:08 INFO mapred.LocalJobRunner: Map task executor complete.
```

After running the job, the related job information displayed in console prompt

29. After **Complete the job** the **status** will be displayed '**0**' that means **Job is successful** otherwise **not successful**, see below screen shot.



```
package com.orienit.hadoop.wordcount;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
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.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WordCountJob implements Tool {
    // Initializing configuration object
    private Configuration conf;

    @Override
    public Configuration getConf() {
        return conf; // getting the configuration
    }
```

```
<terminated> WordCountJob [Java Application] /usr/lib/jvm/java-6-openjdk-amd64/bin/java (15-Sep-2016 8:03:07 AM)
16/09/15 08:03:09 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=110284
16/09/15 08:03:09 INFO mapred.JobClient:   Map-Reduce Framework
16/09/15 08:03:09 INFO mapred.JobClient:     Map output materialized bytes=2582
16/09/15 08:03:09 INFO mapred.JobClient:     Map input records=34
16/09/15 08:03:09 INFO mapred.JobClient:     Reduce shuffle bytes=0
16/09/15 08:03:09 INFO mapred.JobClient:     Spilled Records=238
16/09/15 08:03:09 INFO mapred.JobClient:     Map output bytes=2338
16/09/15 08:03:09 INFO mapred.JobClient:     CPU time spent (ms)=0
16/09/15 08:03:09 INFO mapred.JobClient:     Total committed heap usage (bytes)=446824448
16/09/15 08:03:09 INFO mapred.JobClient:     Combine input records=0
16/09/15 08:03:09 INFO mapred.JobClient:     SPLIT_RAW_BYTES=111
16/09/15 08:03:09 INFO mapred.JobClient:     Reduce input records=119
16/09/15 08:03:09 INFO mapred.JobClient:     Reduce input groups=48
16/09/15 08:03:09 INFO mapred.JobClient:     Combine output records=0
16/09/15 08:03:09 INFO mapred.JobClient:     Physical memory (bytes) snapshot=0
16/09/15 08:03:09 INFO mapred.JobClient:     Reduce output records=48
16/09/15 08:03:09 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=0
16/09/15 08:03:09 INFO mapred.JobClient:     Map output records=119
My Status: 0
```

'0' indicates successfuly completed job

Here verify information about job

Job Information

* No.of.Mappers

*No.of.Reducers

*No.of.Input/output records

*No.of. counters .....etc.