
| Module | Assessment Type |
| --- | --- |
| Distributed and Cloud Systems Programming | Individual Report |

# Workshop 9

Student Id : 2049867
Student Name : Roshan Parajuli
Section : L5CG3
Module Leader : Rupak Koirala
Lecturer /Tutor : Saroj Sharma
Submitted on : 2021-05-17

Table of Contents

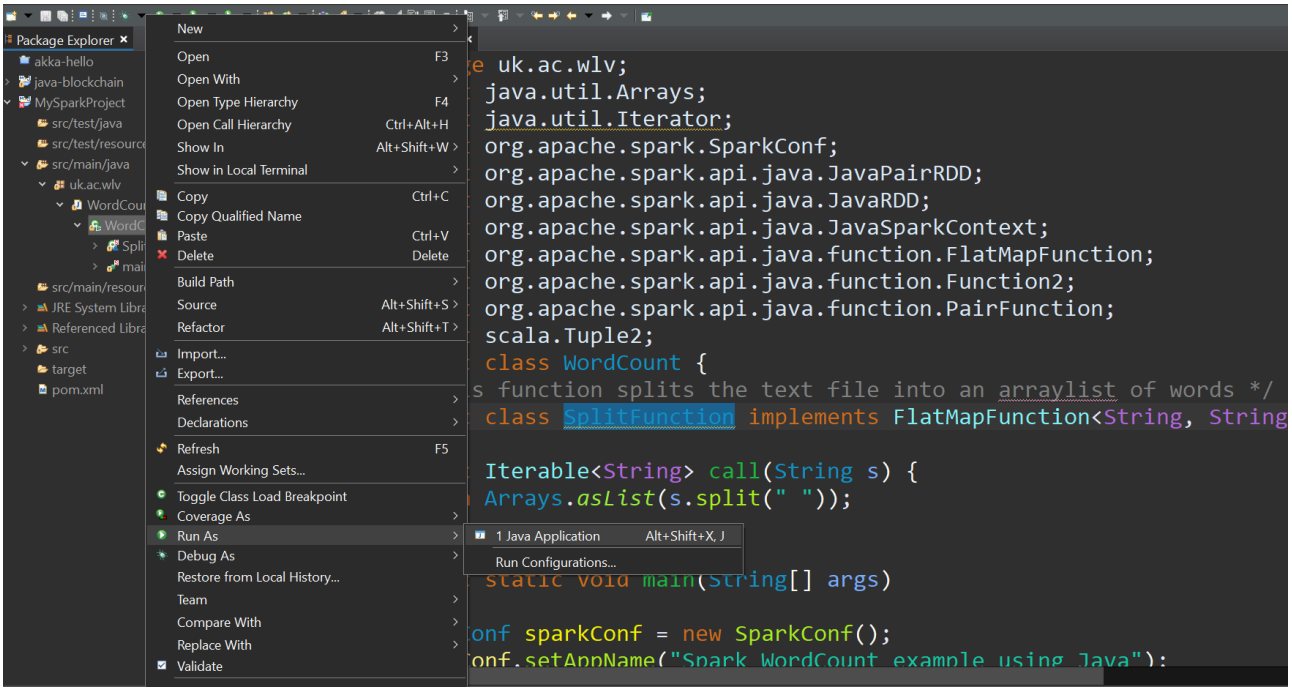
## Introduction

Apache Spark is an open-source data processing engine to store and process data in real-time across various clusters of computers using simple programming constructs. Supports multiple programming languages like: Java, Scala, python and R. In this workshop, a simple program is implemented in java which covers the content from the creation of spark context, creation of RDDs and all the simple spark terminologies with implementation.

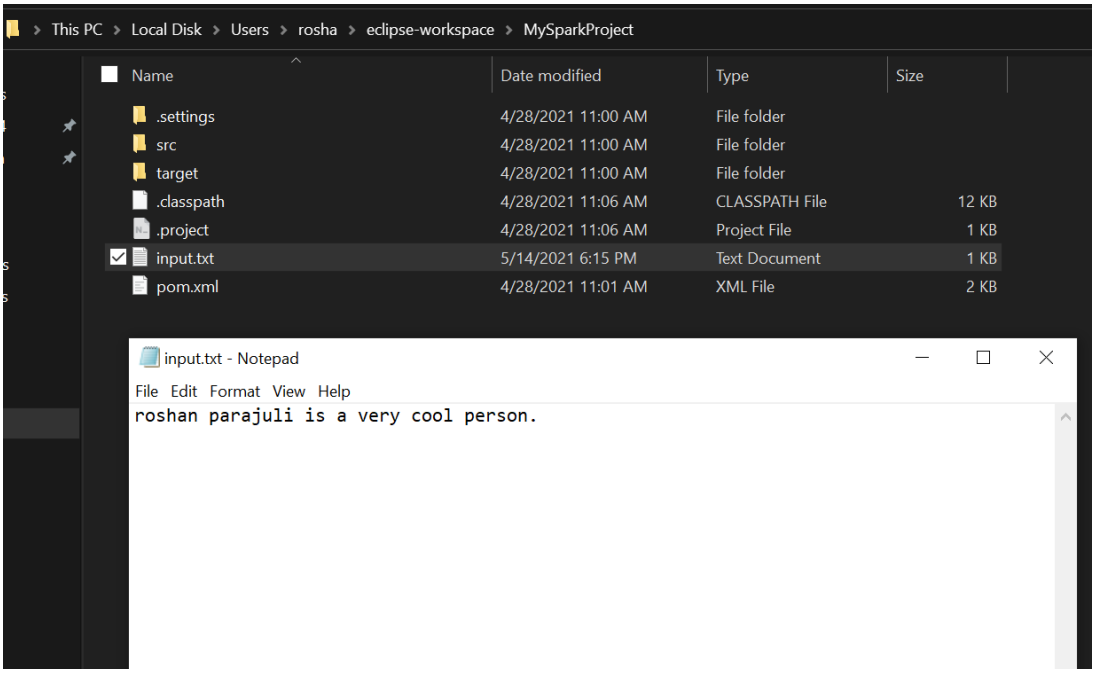
## Workshop

First of all, Hadoop was configured with the instructions provided in the workshop file.

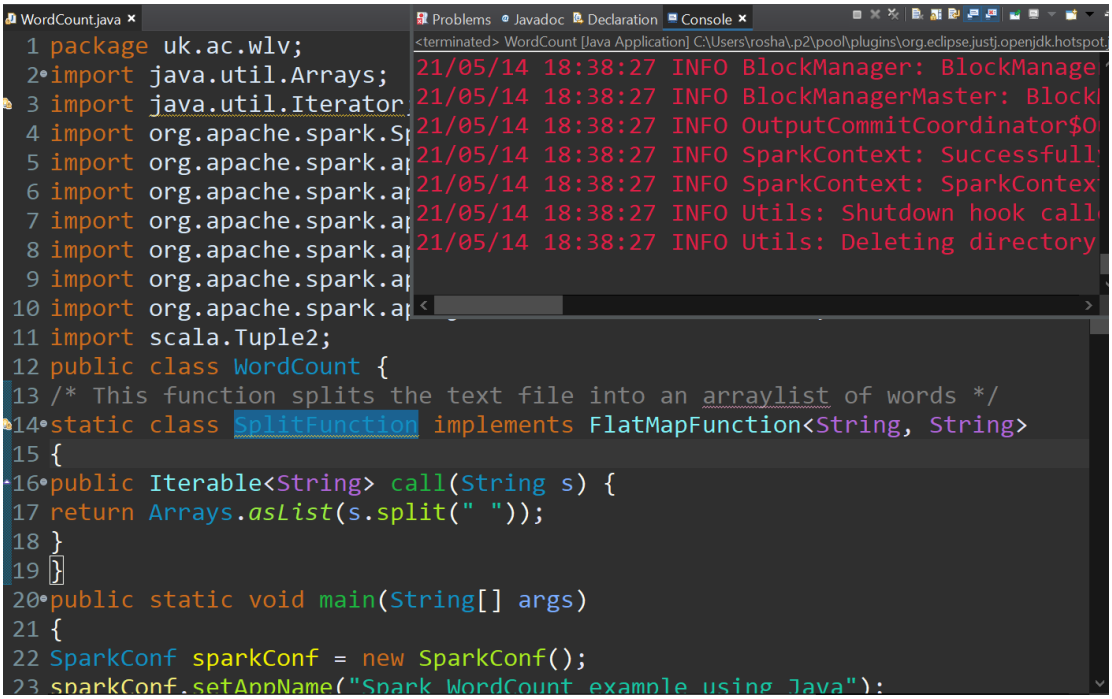
1. A new system variable called HADOOP_HOME was added whose job was to tell Apace Spark where to find Hadoop.
2. A new maven project was created in eclipse.
3. A simple project was created with the instructions provided.
4. The Project Object Model (POM) file was replaced as described. POM file is a fundamental XML file containing information about the project and configuration details used by maven to build the project. It contained the default values for the project.
5. The project was run. After running the application for the first time, Maven build configuration was synchronized with Eclipse and the Eclipse was brought up to date. All the necessary packages were downloaded from the internet.
6. The WordCount.java code was pasted and the application was re-run.



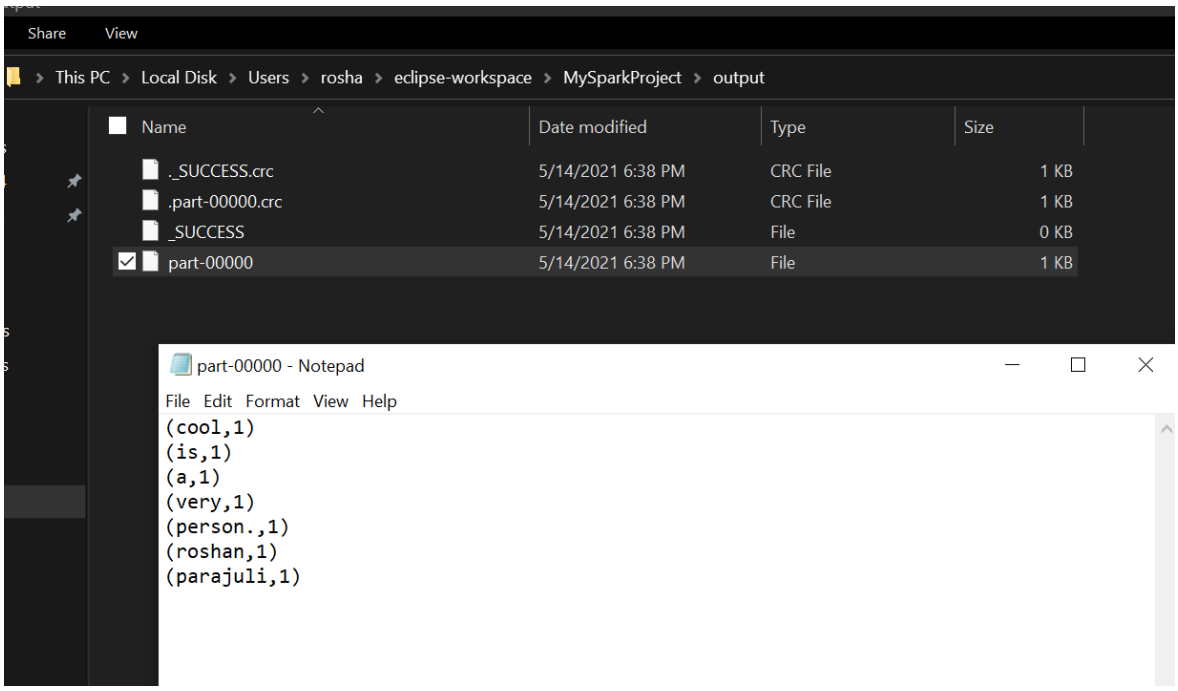
7. A new input text file in the same directory as pom.xml file. The contents of the file are shown below:




8. The program was run successfully with a bunch of log info messages.




9. The output file contains the expected output. Every word is counted and the count is stored in the form of a tuple in the file named part-00000.




## Creating a Spark program to count letter instead of words

To modify the program to count words instead of letters, the split method needs to be modified. As the method is splitting the input text based on a space between the words, the words are split and counted.

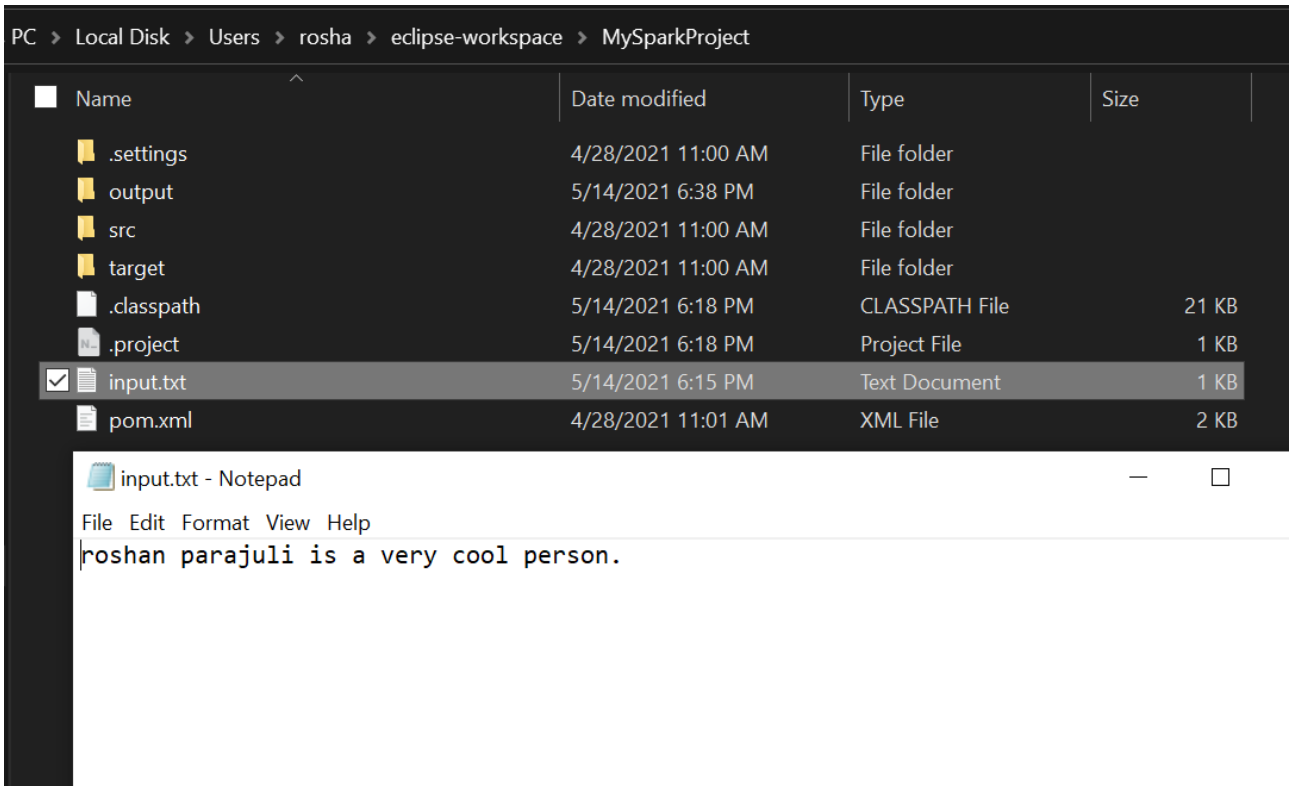
First, the space is removed in the parameter of the split function so that every letter is counted. The main problem with that approach would be that it would count the spaces as letters.

```
16 public class WordCount {
17
18 /* This function splits the text file into an arraylist of words */
19 static class SplitFunction implements FlatMapFunction<String, String>
20 {
21 public Iterable<String> call(String s) {
22     List<String> list = new ArrayList<String>(Arrays.asList(s.split("")));
23     list.removeAll(Collections.singleton(null));
24     list.removeAll(Collections.singleton(" "));
25
26     return list;
27 }
28
```

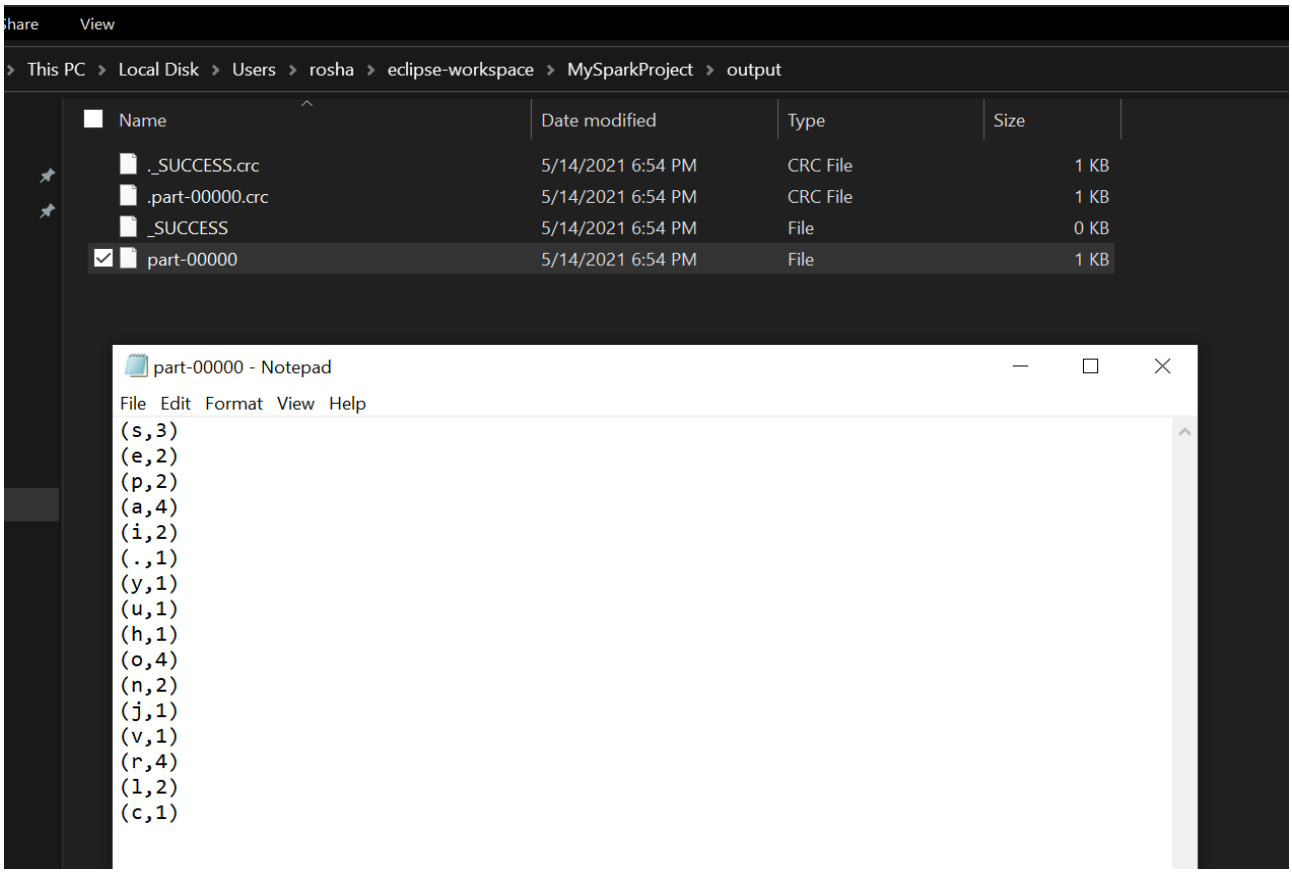
Here, to solve the problem of the inclusion of spaces as characters, all the split characters are stored in a list. From the list, all the null characters and spaces were removed with the

use of removeAll method. After the list was process this way, the same list was returned so as to place the content as tuples on the output file.

Input file:




Output file:



As seen in the output file, all the letters are counted, and the logic worked successfully. The workshop task has been successfully completed.

## Conclusion

In this task, the basics of Apache Spark was learnt. RDD were created to store the input file as well as the words after the split function. Spark configuration was made use of along with the spark context. A simple Spark project to count the words and the letters from the input file was run successfully.