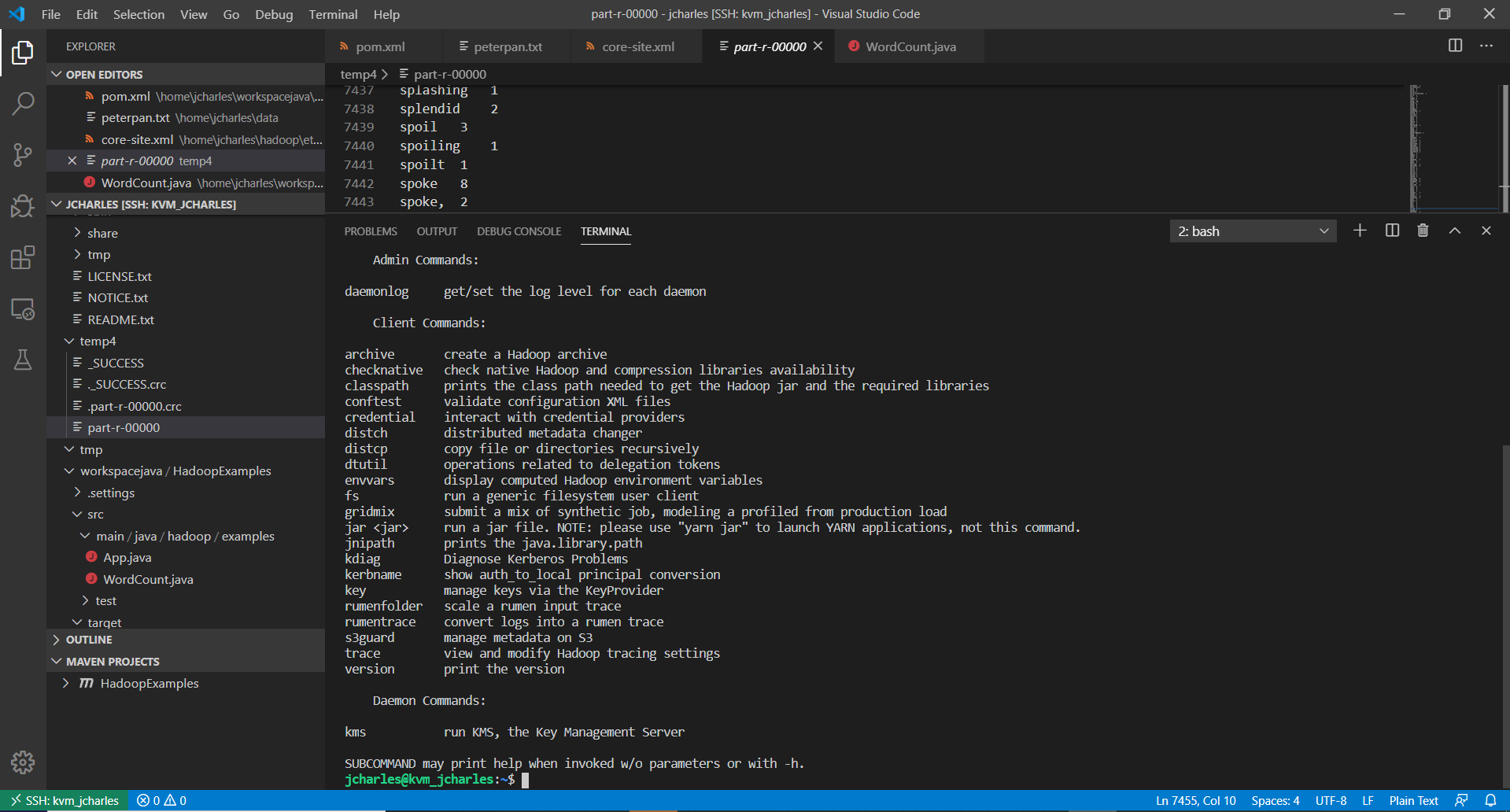
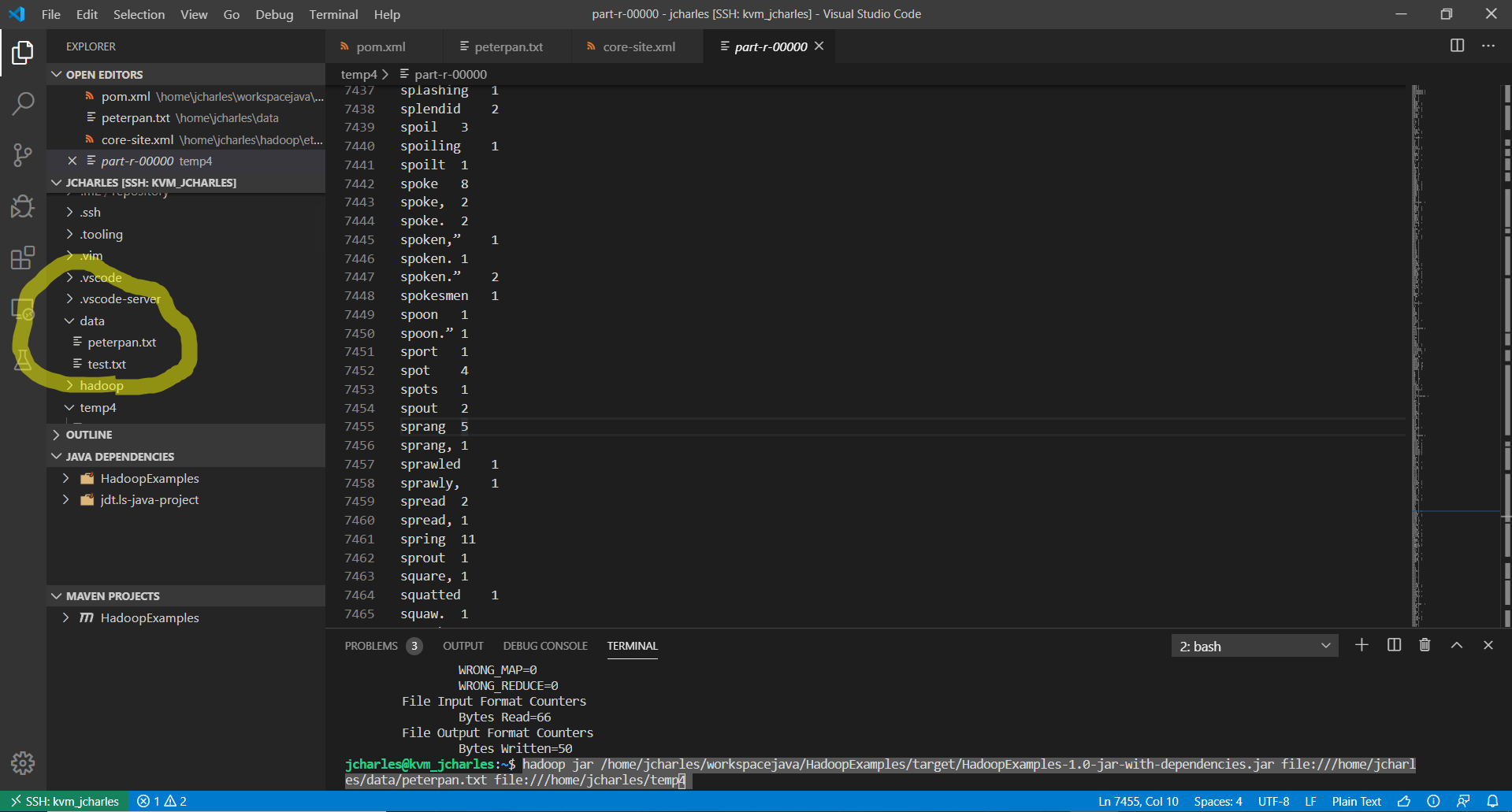
1. **Setting up Hadoop**

Setup Hadoop Setting up Hadoop allows us to later execute our java/jar files within hdfs. When you type “hadoop” in your command line, you should receive the following output to ensure that hadoop is set up properly.



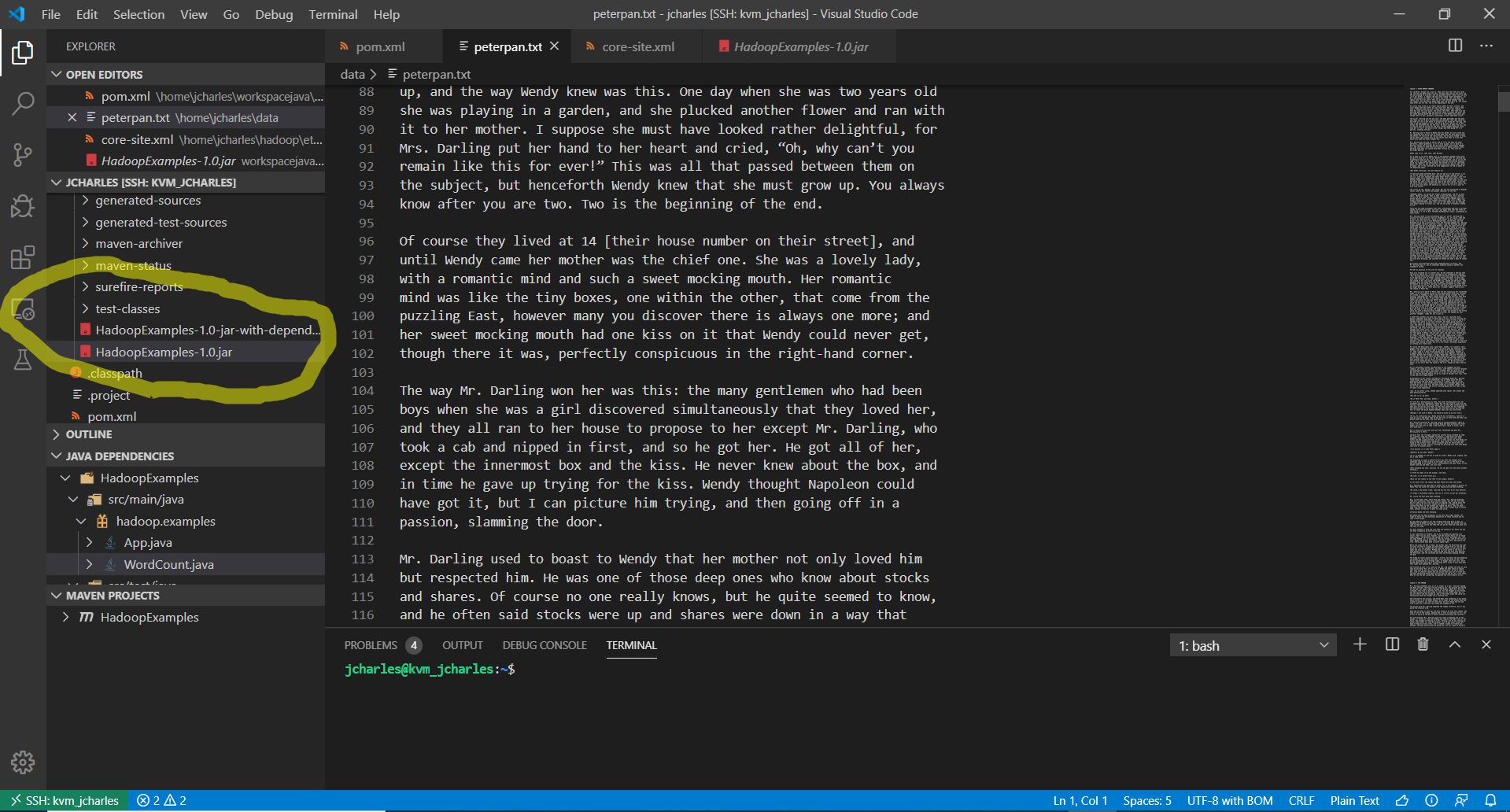
1. **Setting up HDFS**

Setup HDFS and upload the datasets “test.txt” and “peterpan.txt” into HDFS. The setuput of HDFS allows us to upload our files from our local system to our KVM. In the following output we see the files uploaded into our HDFS



1. **Create a Project in VS Code**

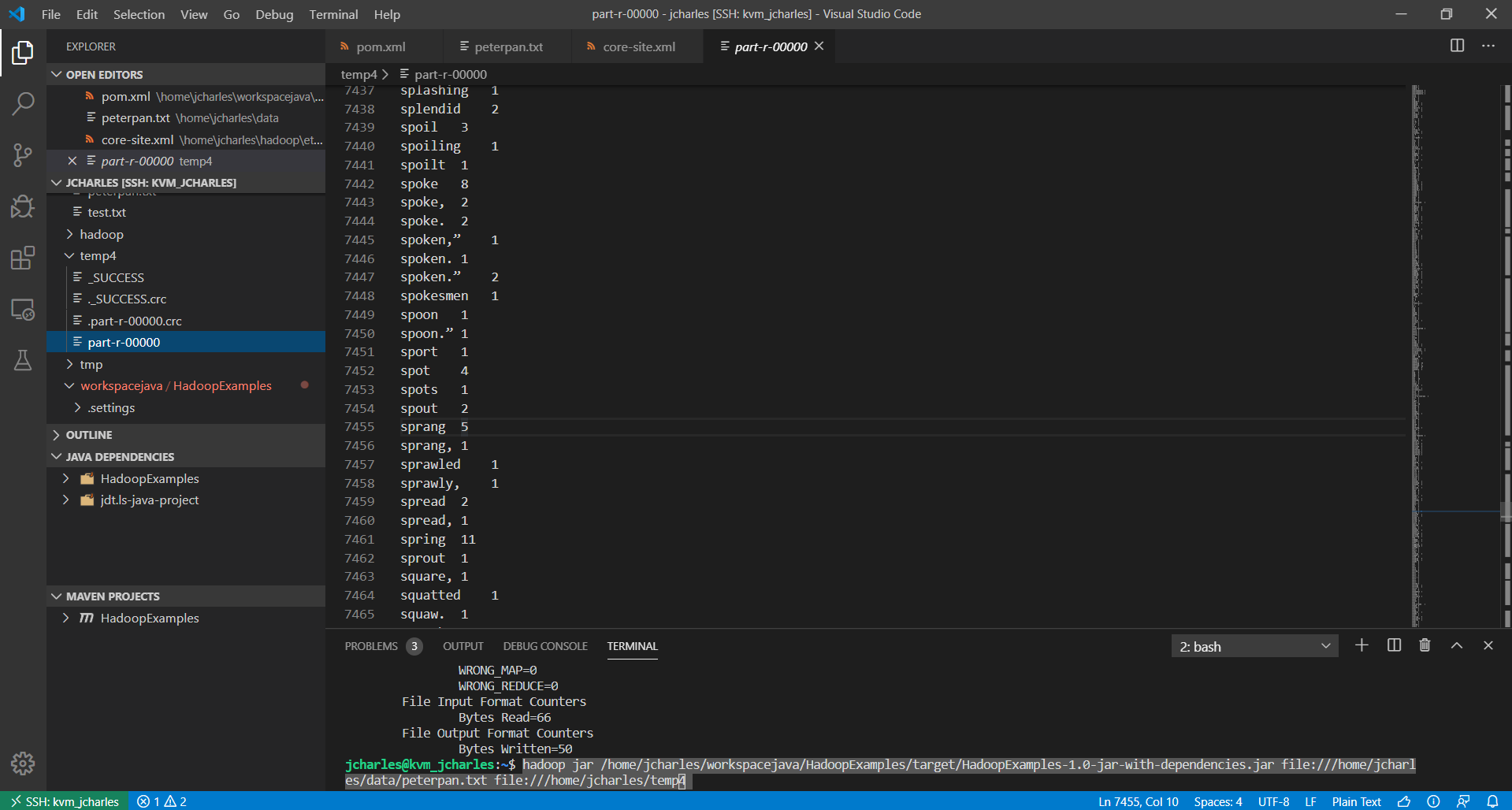
Create a project in VS Code, import the “WordCount.java”, configure the project, and export the “WordCount.jar” file. The WordCount.java file will help to create the WordCount.jar file in our target folder of the maven project. You need to create a custom build with “clean package assembly:single” as the keywords. This is what the jar file should look like after it is created.

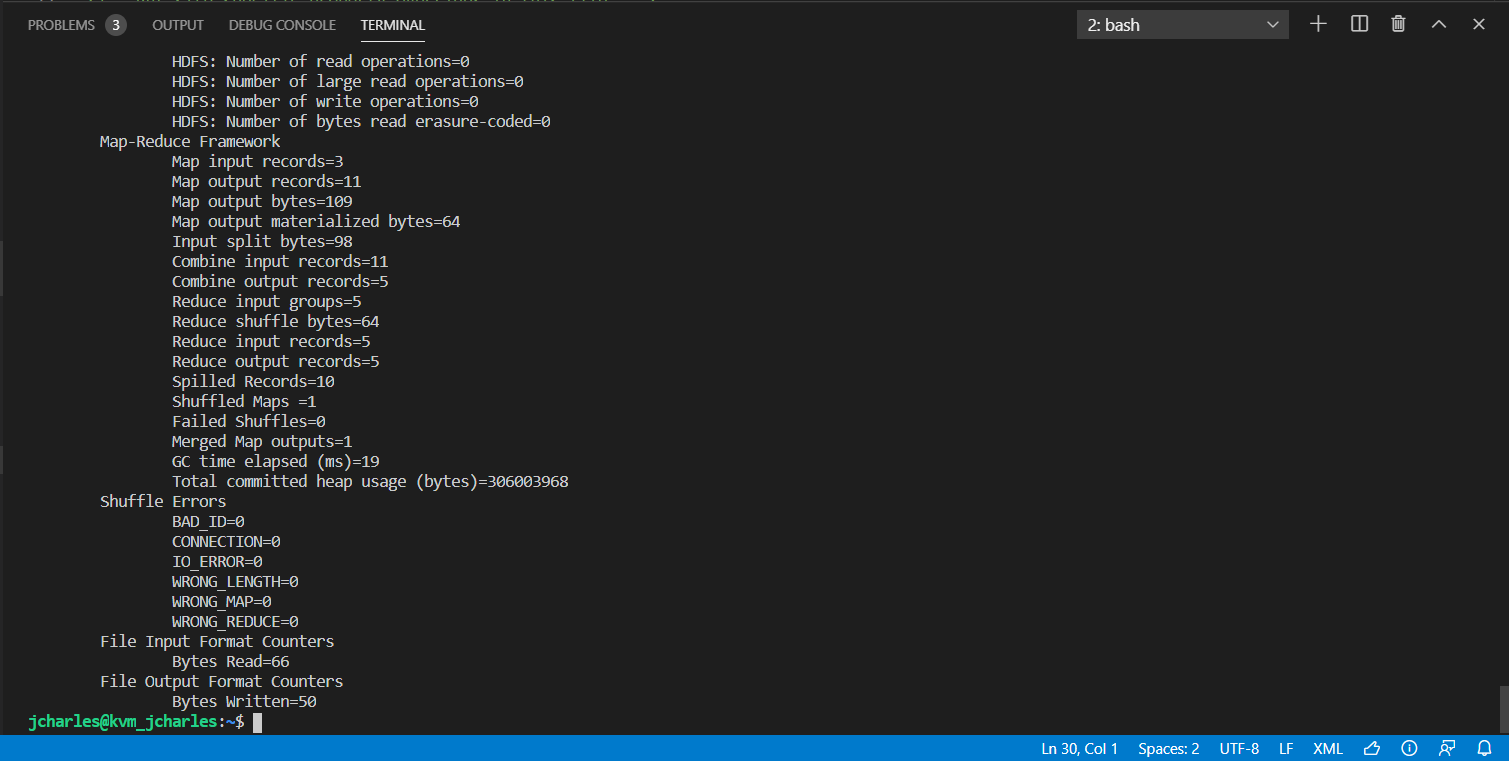
****

**4.**

**Open a terminal, and run the “WordCount.jar” file on “test.txt” and “peterpan.txt” respectively**

I ended up coming to office hours to get peterpan.txt to run and later getting test.txt to run as well. I made that happen by using the command “*hadoop jar /home/jcharles/workspacejava/HadoopExamples/target/HadoopExamples-1.0-jar-with-dependencies.jar file:///home/jcharles/data/test.txt file:///home/jcharles/temp2*” for test.txt and “*hadoop jar /home/jcharles/workspacejava/HadoopExamples/target/HadoopExamples-1.0-jar-with-dependencies.jar file:///home/jcharles/data/peterpan.txt file:///home/jcharles/temp4*” for peterpan.txt and I had their outputs saved to different folders.

****

****