

# Fault Detection:

**Here are the following Faults that was determine during the Junit Test runs of the 53 test cases. Note the Compile output is at the start of test suite of initial run without corrections of the source code Printtokens.java. Please refer below first Part of testing (Unit Testing):**

```
ant -f C:\\Users\\dienadel\\Documents\\NetBeansProjects\\ProjectCSE4321 "-Drun.jvmargs= -
javaagent:\\\"C:\\Users\\dienadel\\AppData\\Roaming\\NetBeans\\8.2\\modules\\ext\\jacocoagent.jar\\
\"=includes=JUnit4Printtokens.*,destfile=\\\"C:\\Users\\dienadel\\Documents\\NetBeansProjects\\Project
CSE4321\\jacoco.exec-1468127461679600\\\"\" test
```

init:

deps-jar:

Updating property file: C:\\Users\\dienadel\\Documents\\NetBeansProjects\\ProjectCSE4321\\build\\built-  
jar.properties

compile:

Created dir: C:\\Users\\dienadel\\Documents\\NetBeansProjects\\ProjectCSE4321\\build\\test\\classes

Compiling 10 source files to

C:\\Users\\dienadel\\Documents\\NetBeansProjects\\ProjectCSE4321\\build\\test\\classes

compile-test:

Created dir: C:\\Users\\dienadel\\Documents\\NetBeansProjects\\ProjectCSE4321\\build\\test\\results

Testsuite: JUnit4Printtokens.CommentParameterTest

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.111 sec

Testsuite: JUnit4Printtokens.KeywordParameterTest

Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.101 sec

Testsuite: JUnit4Printtokens.charParameterTest

Tests run: 4, Failures: 4, Errors: 0, Skipped: 0, Time elapsed: 0.179 sec

Testcase: charParameterTestActual[0: #a](JUnit4Printtokens.charParameterTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.charParameterTest.charParameterTestActual(charParameterTest.java:39)

Testcase: charParameterTestActual[1: #b](JUnit4Printtokens.charParameterTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.charParameterTest.charParameterTestActual(charParameterTest.java:39)

Testcase: charParameterTestActual[2: #c](JUnit4Printtokens.charParameterTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.charParameterTest.charParameterTestActual(charParameterTest.java:39)

Testcase: charParameterTestActual[3: #d](JUnit4Printtokens.charParameterTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.charParameterTest.charParameterTestActual(charParameterTest.java:39)

Test JUnit4Printtokens.charParameterTest FAILED

Testsuite: JUnit4Printtokens.handleTokenTest

Tests run: 8, Failures: 3, Errors: 5, Skipped: 0, Time elapsed: 0.131 sec

Testcase: ungetcharToken(JUnit4Printtokens.handleTokenTest): Caused an ERROR

null

java.lang.NullPointerException

at JUnit4Printtokens.Printtokens.get\_char(Printtokens.java:53)

at JUnit4Printtokens.handleTokenTest.ungetcharToken(handleTokenTest.java:68)

Testcase: openCharStreamNoFile(JUnit4Printtokens.handleTokenTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.handleTokenTest.openCharStreamNoFile(handleTokenTest.java:42)

Testcase: getCharToken(JUnit4Printtokens.handleTokenTest): Caused an ERROR

null

java.lang.NullPointerException

at JUnit4Printtokens.Printtokens.get\_char(Printtokens.java:53)

at JUnit4Printtokens.handleTokenTest.getCharToken(handleTokenTest.java:52)

Testcase: tokenType(JUnit4Printtokens.handleTokenTest): Caused an ERROR

null

java.lang.NullPointerException

at JUnit4Printtokens.Printtokens.get\_char(Printtokens.java:53)

at JUnit4Printtokens.Printtokens.get\_token(Printtokens.java:110)

at JUnit4Printtokens.handleTokenTest.tokenType(handleTokenTest.java:112)

Testcase: getTokenString(JUnit4Printtokens.handleTokenTest): Caused an ERROR

null

java.lang.NullPointerException

at JUnit4Printtokens.Printtokens.get\_char(Printtokens.java:53)

at JUnit4Printtokens.Printtokens.get\_token(Printtokens.java:110)

at JUnit4Printtokens.handleTokenTest.getTokenString(handleTokenTest.java:100)

Testcase: opentokenStreamNotNull(JUnit4Printtokens.handleTokenTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.handleTokenTest.opentokenStreamNotNull(handleTokenTest.java:79)

Testcase: openCharStreamWithFile(JUnit4Printtokens.handleTokenTest): FAILED

null

junit.framework.AssertionFailedError

at JUnit4Printtokens.handleTokenTest.openCharStreamWithFile(handleTokenTest.java:33)

Testcase: opentokenStreamNull(JUnit4Printtokens.handleTokenTest): Caused an ERROR

null

java.lang.NullPointerException

at JUnit4Printtokens.Printtokens.open\_token\_stream(Printtokens.java:86)

at JUnit4Printtokens.handleTokenTest.opentokenStreamNull(handleTokenTest.java:87)

Test JUnit4Printtokens.handleTokenTest FAILED

Testsuite: JUnit4Printtokens.identifierParameterTest

Tests run: 4, Failures: 4, Errors: 0, Skipped: 0, Time elapsed: 0.127 sec

Testcase: identifierParameterTestActual[0: a](JUnit4Printtokens.identifierParameterTest): FAILED

null

junit.framework.AssertionFailedError

at  
JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual(identifierParameterTest.java:37)

Testcase: identifierParameterTestActual[1: aa](JUnit4Printtokens.identifierParameterTest): FAILED  
null  
junit.framework.AssertionFailedError

at  
JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual(identifierParameterTest.java:37)

Testcase: identifierParameterTestActual[2: a1](JUnit4Printtokens.identifierParameterTest): FAILED  
null  
junit.framework.AssertionFailedError

at  
JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual(identifierParameterTest.java:37)

Testcase: identifierParameterTestActual[3: var](JUnit4Printtokens.identifierParameterTest): FAILED  
null  
junit.framework.AssertionFailedError

at  
JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual(identifierParameterTest.java:37)

Test JUnit4Printtokens.identifierParameterTest FAILED

Testsuite: JUnit4Printtokens.numParameterTest

Tests run: 4, Failures: 0, Errors: 4, Skipped: 0, Time elapsed: 0.172 sec

Testcase: numParameterTestActual[0: 0](JUnit4Printtokens.numParameterTest): Caused an ERROR

String index out of range: 1

java.lang.StringIndexOutOfBoundsException: String index out of range: 1

at java.lang.String.charAt(String.java:658)

at JUnit4Printtokens.Printtokens.is\_num\_constant(Printtokens.java:306)

at JUnit4Printtokens.numParameterTest.numParameterTestActual(numParameterTest.java:36)

Testcase: numParameterTestActual[1: 01](JUnit4Printtokens.numParameterTest): Caused an ERROR

String index out of range: 2

java.lang.StringIndexOutOfBoundsException: String index out of range: 2

at java.lang.String.charAt(String.java:658)

at JUnit4Printtokens.Printtokens.is\_num\_constant(Printtokens.java:308)

at JUnit4Printtokens.numParameterTest.numParameterTestActual(numParameterTest.java:36)

Testcase: numParameterTestActual[2: 123](JUnit4Printtokens.numParameterTest): Caused an ERROR

String index out of range: 3

java.lang.StringIndexOutOfBoundsException: String index out of range: 3

at java.lang.String.charAt(String.java:658)

at JUnit4Printtokens.Printtokens.is\_num\_constant(Printtokens.java:308)

at JUnit4Printtokens.numParameterTest.numParameterTestActual(numParameterTest.java:36)

Testcase: numParameterTestActual[3: 4321](JUnit4Printtokens.numParameterTest): Caused an ERROR

String index out of range: 4

java.lang.StringIndexOutOfBoundsException: String index out of range: 4

at java.lang.String.charAt(String.java:658)

at JUnit4Printtokens.Printtokens.is\_num\_constant(Printtokens.java:308)

at JUnit4Printtokens.numParameterTest.numParameterTestActual(numParameterTest.java:36)

Test JUnit4Printtokens.numParameterTest FAILED

Testsuite: JUnit4Printtokens.printoutTokenTest

It can not get charcter

\*\*\*testing printout 2 for above statement\*\*.

lparen.

\*\*\*testing printout 3 for above statement\*\*.

rparen.

\*\*\*testing printout 4 for above statement\*\*.

lsquare.

\*\*\*testing printout 5 for above statement\*\*.

rsquare.

\*\*\*testing printout 6 for above statement\*\*.

quote.

\*\*\*testing printout 7 for above statement\*\*.

bquote.

\*\*\*testing printout 8 for above statement\*\*.

Tests run: 13, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.127 sec

----- Standard Output -----

It can not get charcter

\*\*\*testing printout 2 for above statement\*\*.

lparen.

\*\*\*testing printout 3 for above statement\*\*.

rparen.



\*\*\*testing printout 4 for above statement\*\*.

lsquare.

\*\*\*testing printout 5 for above statement\*\*.

rsquare.

\*\*\*testing printout 6 for above statement\*\*.

quote.

\*\*\*testing printout 7 for above statement\*\*.

bquote.

\*\*\*testing printout 8 for above statement\*\*.

-----  
Testcase: printoutTokenActual(JUnit4Printtokens.printoutTokenTest): Caused an ERROR  
null  
java.lang.NullPointerException  
at JUnit4Printtokens.Printtokens.get\_char(Printtokens.java:53)

```
at JUnit4Printtokens.Printtokens.get_token(Printtokens.java:110)
at JUnit4Printtokens.printoutTokenTest.printoutTokenActual(printoutTokenTest.java:28)
```

Test JUnit4Printtokens.printoutTokenTest FAILED

Testsuite: JUnit4Printtokens.specsymbolParameterTest

Tests run: 6, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.152 sec

Testcase: is\_special\_symbol[5: '](JUnit4Printtokens.specsymbolParameterTest): FAILED

null

junit.framework.AssertionFailedError

at

JUnit4Printtokens.specsymbolParameterTest.is\_special\_symbol(specsymbolParameterTest.java:35)

Test JUnit4Printtokens.specsymbolParameterTest FAILED

Testsuite: JUnit4Printtokens.stringParameterTest

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.122 sec

test-report:

C:\Users\dienadel\Documents\NetBeansProjects\ProjectCSE4321\nbproject\build-impl.xml:1296: Some tests failed; see details above.

BUILD FAILED (total time: 12 seconds)

**Here are the screenshots of the Junit test cases statistics. JaCoCoverage was used to determine the coverage analysis of the source code in lieu with the Junit Testing scheme.**

Tests passed: 59.26 %

32 tests passed, 12 tests failed, 10 tests caused an error. (1.222 s)

- JUnit4Printtokens.charParameterTest **Failed**
  - JUnit4Printtokens.charParameterTest.charParameterTestActual[0: #a] **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.charParameterTest.charParameterTestActual[1: #b] **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.charParameterTest.charParameterTestActual[2: #c] **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.charParameterTest.charParameterTestActual[3: #d] **Failed: junit.framework.AssertionFailedError**
- JUnit4Printtokens.handleTokenTest **Failed**
  - JUnit4Printtokens.handleTokenTest.openCharStreamNoFile **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.handleTokenTest.openTokenStreamNotNull **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.handleTokenTest.openCharStreamWithFile **Failed: junit.framework.AssertionFailedError**
- JUnit4Printtokens.identifierParameterTest **Failed**
  - JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual[0: a] **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual[1: aa] **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual[2: a1] **Failed: junit.framework.AssertionFailedError**
  - JUnit4Printtokens.identifierParameterTest.identifierParameterTestActual[3: var] **Failed: junit.framework.AssertionFailedError**
- JUnit4Printtokens.specsSymbolParameterTest **Failed**
  - JUnit4Printtokens.specsSymbolParameterTest.is\_special\_symbol[5: ] **Failed: junit.framework.AssertionFailedError**

Output

ProjectCSE4321 (jacocoverage report) X ProjectCSE4321 (test) X

```

=== JaCoCoverage report (powered by JaCoCo from EcjEmma) ===
Covered | Partially covered | Not covered | Java Class

72  18  109  /JUnit4Printtokens/Printtokens.java

End of report
You can ask questions and report bugs by visiting: https://github.com/jonathanlermitage/tikione-jacocoverage

```

## JaCoCoverage analysis of project "ProjectCSE4321" (powered by JaCoCo from EcjEmma)

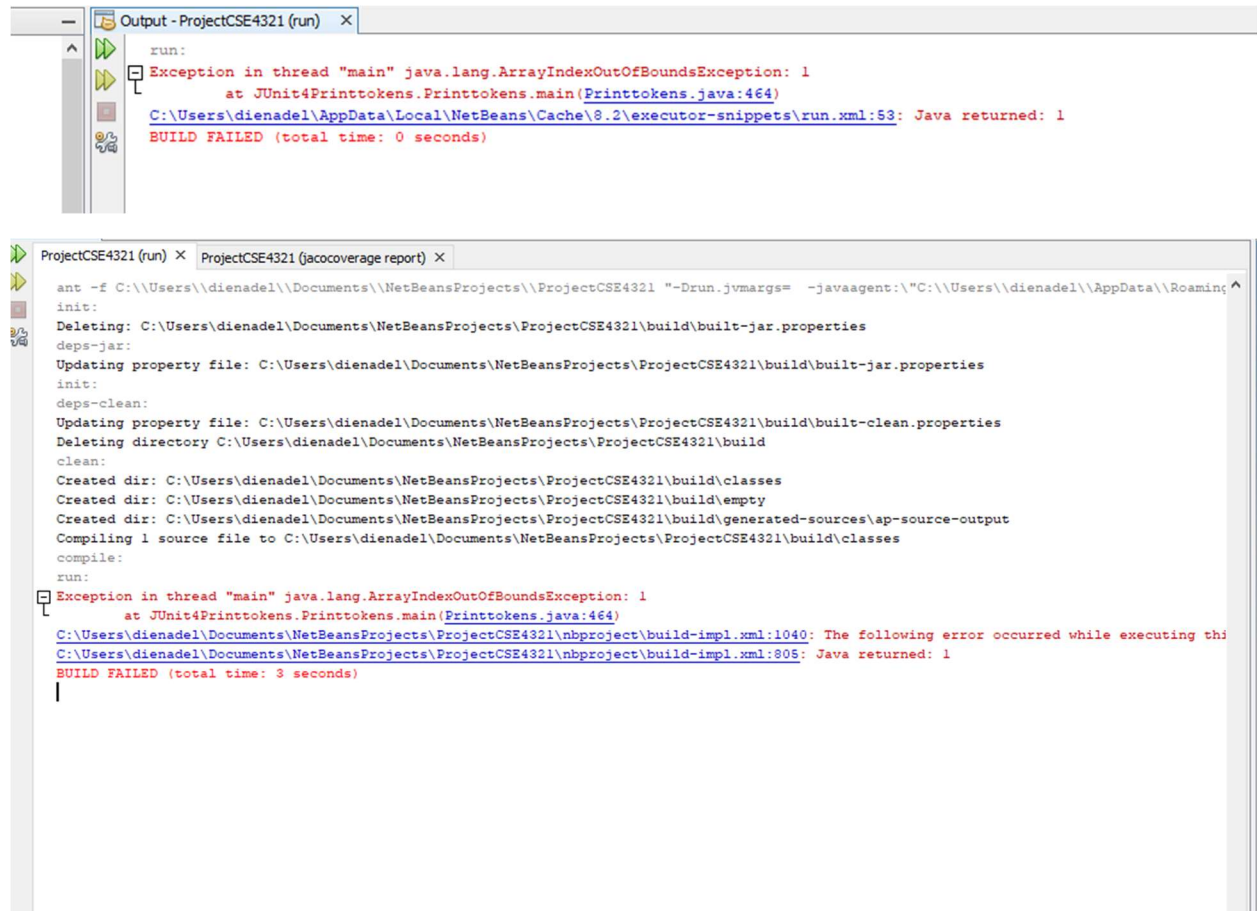
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
JUnit4Printtokens	<div><div></div></div>	42%	<div><div></div></div>	38%	69	105	109	199	4	20	0	1
Total	432 of 739	42%	105 of 170	38%	69	105	109	199	4	20	0	1

Created with [JaCoCo](#) 0.7.6.201602180812

On the first initial Test run without corrections. Junit testing statistics has determined that it executed 54 test cases: 33 passed, 12 failed and 10 produced errors. In addition, JaCoCoverage analysis project using JaCOCO unit testing coverage has determine that 432 of 739 were missed instructions(achieving 42% coverage), 105 of 170 were missed branches (achieving 38% coverages )and 4 of 20 methods were missed from the unit test run. From this Unit testing of the initial run, it has determined that only 72 lines of code were covered, 18 lines partial coverage, and 109 lines were not covered.

Moreover, I have performed another test that measure the edge coverage of the entire program (End-To-End/System-Level Testing). It has determine that, from JaCOCO system run analysis, 722 of 739 were missed instructions and 170 of 170 were missed branches. The reason that it has a very low

coverage (0% – 2% ) is due to a build failed during the system compile run. When it fails, the whole system of the program will not execute and traverse through the branches. Refer to screenshots.



JaCoCoverage analysis of project "ProjectCSE4321" (powered by JaCoCo from EcIEmma) [Sessions](#)

## JaCoCoverage analysis of project "ProjectCSE4321" (powered by JaCoCo from EcIEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
JUnit4Printtokens		2%		0%	104	105	191	199	19	20	0	1
Total	722 of 739	2%	170 of 170	0%	104	105	191	199	19	20	0	1

Created with [JaCoCo](#) 0.7.6.201602180812

## Code Corrections:

Code corrections of the source code (Printtokens.java) was performed. From the Unit testing, it has determined that there are 27 code line corrections that were corrected. Please refer to the source code file under comments to review the specific corrections that were made. Refer to screenshots for reference of code correction count.

```

3 import java.io.BufferedReader;
4 import java.io.FileNotFoundException;
5 import java.io.FileReader;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 //EDIT: comment out import java.io.StringReader; since its unused
9 //import java.io.StringReader;
10
11 public class Printtokens {
12
13     //EDIT:change the int values to fit in the if statements
14     static int error = 0;
15     static int keyword = 1;
16     static int spec_symbol = 2;
17     static int identifier = 3;
18     static int num_constant = 4;
19     static int str_constant = 5;
20     static int char_constant = 6;
21     static int comment = 7;
22
23
24
25     /**
26      * NMAE: open character stream
27      * INPUT: a filename
28      */

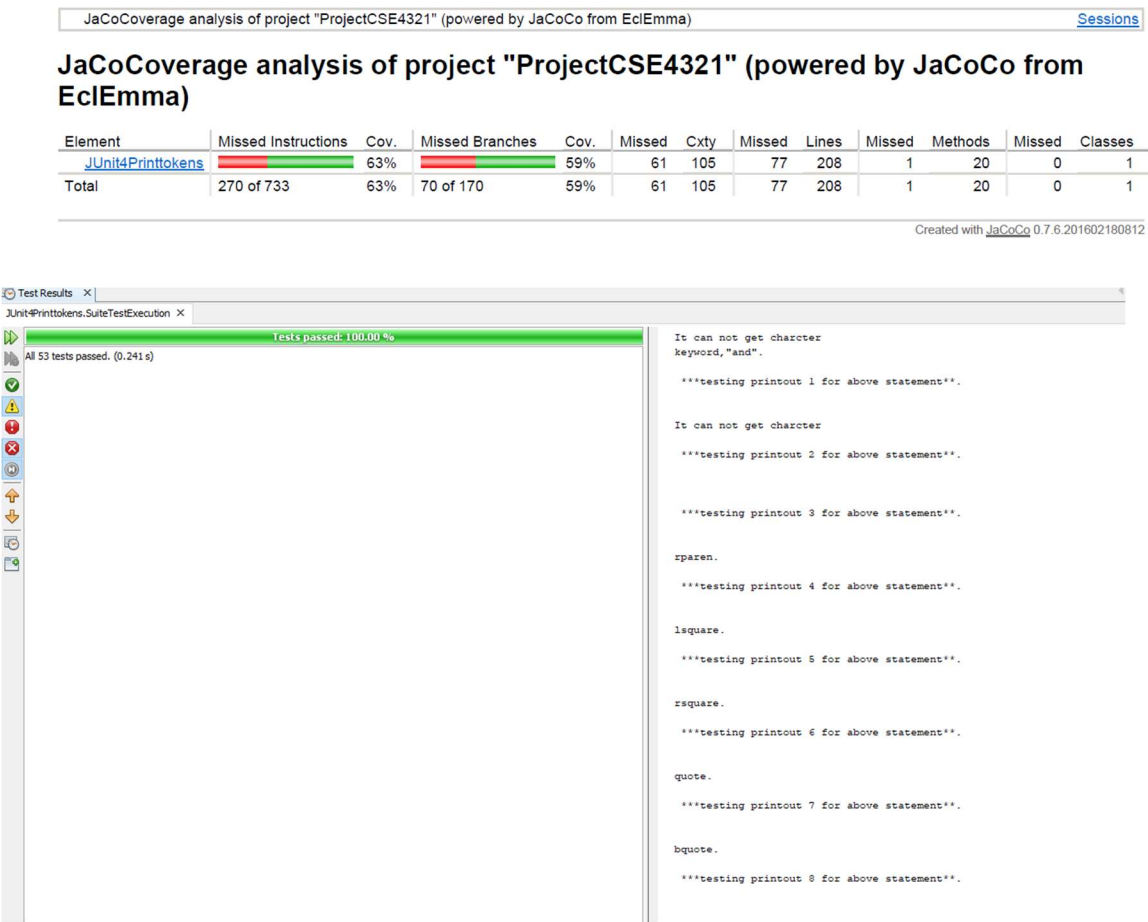
```

Tokens.txt file were updated and used for this final Unit and system testing. Refer to the txt file to examine the contents of the inputs.

```
tokens - Notepad
File Edit Format View Help
|and
|or
|if
|xor
|lambda
|=>
|(
|)
|[
|]
|'
|'
|,
|a
|aa
|a1
|a2
|123
|1
|321
|"asd"
```

After the source code correction, with high confidence, the final Unit and system testing were performed for this analysis.

The first portion of the test is to run the unit testing for the source code using Junit and JaCoCo analysis. Refer to the screenshot below.



All 53 unit tests cases were passed, all errors were found and corrected. **Junit testing statistics has determined that it executed 53 test cases: 53 passed, 0 failed and 0 produced errors. In addition, JaCoCoverage analysis project using JaCOCO unit testing coverage has determine that 132 of 739 were missed instructions (achieving 82% coverage), 105 of 170 missed branches (achieving 78% coverage) and 4 of 20 methods were missed from the unit test run. From this Unit testing of the initial run, it has determined that only 72 lines of code were covered, 18 lines partial coverage, and 109 lines were not covered.**

There was one unit test case that was omitted since it tests the file when it null, but when performing the Junit Assert method during file stream (either AssertTrue() or AssertNotTrue()), it still runs as an error but not a failure. For this case, this will be omitted since the test cases relied on test txt file that is not null. Here are the following printout output from the unit tests cases:

Testsuite: JUnit4Printtokens.SuiteTestExecution

It can not get charcter

keyword,"and".

\*\*\*testing printout 1 for above statement\*\*.

It can not get charcter

\*\*\*testing printout 2 for above statement\*\*.

\*\*\*testing printout 3 for above statement\*\*.

rparen.

\*\*\*testing printout 4 for above statement\*\*.

lsquare.

\*\*\*testing printout 5 for above statement\*\*.

rsquare.

\*\*\*testing printout 6 for above statement\*\*.

quote.

\*\*\*testing printout 7 for above statement\*\*.

bquote.

\*\*\*testing printout 8 for above statement\*\*.

Tests run: 53, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.4 sec

----- Standard Output -----

It can not get charcter

keyword,"and".

\*\*\*testing printout 1 for above statement\*\*.

It can not get charcter

\*\*\*testing printout 2 for above statement\*\*.

\*\*\*testing printout 3 for above statement\*\*.

rpren.



\*\*\*testing printout 4 for above statement\*\*.

lsquare.

\*\*\*testing printout 5 for above statement\*\*.

rsquare.

\*\*\*testing printout 6 for above statement\*\*.

quote.

\*\*\*testing printout 7 for above statement\*\*.

bquote.

\*\*\*testing printout 8 for above statement\*\*.

-----  
test:

Deleting: C:\Users\dienadel\AppData\Local\Temp\TEST-JUnit4Printtokens.SuiteTestExecution.xml

BUILD SUCCESSFUL (total time: 1 second)

In addition, I have performed another test that measure the edge coverages of the entire program (End-To-End/System-Level Testing). It has determine that, from JaCOCO system run analysis, 133 of 733 were missed instructions (achieving 82% coverage) and 38 of 170 were missed branches (achieving 78% coverage). When compared from the initial testing, it has significantly change and improved the execution of the program using the source code. Moreover, the compile run has proven to be successful after the 27 line code corrections. Refer to screenshots.

JaCoCoverage analysis of project "ProjectCSE4321" (powered by JaCoCo from EclEmma)

Sessions

### JaCoCoverage analysis of project "ProjectCSE4321" (powered by JaCoCo from EclEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">JUnit4Printtokens</a>	<div><div></div></div>	82%	<div><div></div></div>	78%	37	105	45	208	2	20	0	1
Total	133 of 733	82%	38 of 170	78%	37	105	45	208	2	20	0	1

Created with [JaCoCo](#) 0.7.6.201602180812

Output - ProjectCSE4321 (run)

```

keyword, "="
lparen.
rparen.
lsquare.
rsquare.
quote.
bquote.
identifier, "a".
identifier, "aa".
identifier, "a1".
identifier, "a2".
numeric, 123.
numeric, 1.
numeric, 321.
string, "asd".
string, "123".
character, "a".
character, "b".
character, "c".
character, "d".
string, ";gutenberg".
error, "0$1$".
error, "!".
error, "0".
error, "8".
error, "5".
error, "4".
error, "-".
error, "6".
error, "+".
error, "-".
error, "+".
error, "-".
error, "4".
error, "a".
error, "?".
error, "!".
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output

ProjectCSE4321 (jacocoverage report) × ProjectCSE4321 (run) ×

```

=== JaCoCoverage report (powered by JaCoCo from EclEmma) ===
Covered | Partially covered | Not covered | Java Class

  94    38    76    /JUnit4Printtokens/Printtokens.java

End of report
You can ask questions and report bugs by visiting: https://github.com/jonathanlermitage/tikione-jacocoverage

```