**Negative Test Cases (Negatives) Calculations**  and excluded test cases

1. Negatives calculation

I used the following script for calculating number of good() methods in CWE476 folder of Juliet Test Suite:

~/Juliet1.3Last/src/testcases/CWE476\_NULL\_Pointer\_Dereference$ grep -o " goodB2G();\| goodG2B();\| goodG2B1();\| goodG2B2();\| goodB2G1();\| goodB2G2();\| good1();\| good2();\| good3();\| good4();\| good5();\| goodG2B(request, response);\| goodB2G(request, response);\| goodG2B1(request, response);\| goodB2G1(request, response);\| goodG2B2(request, response);\| goodB2G2(request, response);\| good1(request, response);\| good2(request, response);\| goodB2G3();\| goodG2B3();\| good3(request, response);\| good4(request, response);\| good5(request, response);" \*.java | wc -l

The total number of Negatives = 496

1. Excluded Test Cases

For double-checking the test cases ( Positives (198) + Negatives (496)), I reviewed them manually, and it turns out that 17 test cases were invalid (do not include Null Pointer Dereference Bug). Consequently, the related positives (17) and Negatives (30) have been excluded from the total test cases used in the study. Code snippets 1 (methods bad() and good1()) represent examples of excluded *Positive* and *Negative* respectively. The name of the test cases are: **CWE476\_NULL\_Pointer\_Dereference\_\_null\_check\_after\_deref\_01 till 17**

SO:

Positives = 198 - 17 = 181

Negatives = 496 - 30 = 466

***Code snippet 1***

public class CWE476\_NULL\_Pointer\_Dereference\_\_null\_check\_after\_deref\_16 {

public void bad() throws Throwable {

while(true) {

{

String myString = null;

myString = "Hello";

IO.writeLine(myString.length());

/\* FLAW: Check for null after dereferencing the object. This null check is unnecessary. \*/

if (myString != null) {

myString = "my, how I've changed";

}

IO.writeLine(myString.length());

}

break;

}

}

private void good1() throws Throwable {

while(true) {

{

String myString = null;

myString = "Hello";

IO.writeLine(myString.length());

/\* FIX: Don't check for null since we wouldn't reach this line if the object was null \*/

myString = "my, how I've changed";

IO.writeLine(myString.length());

}

break;

}

}

}

public class IO {

public static void writeString(String str) {

System.out.print(str);

}

public static void writeLine(String line) {

System.out.println(line);

}

}