Lab 3 – Test Case Design

Consider the following triangle problem and the associated component Java implementation called Determine Triangle.



**Figure 1. The Triangle Problem**

1. Assuming the range of a,b,c take the following ranges (with minimum increments of 1):

10 < a ≤ 50

0 ≤ b < 38

0 ≤ c ≤ 100

Derive the most minimum set of test cases based on Equivalence Partitioning as well as Boundary Value Analysis to test the method ***public static void triangle (int a, int b, int c)***

1. Consider the following conditions

Scalene: a + b > c, where a<b<c

Isosceles: a + a > c, where b=a

Equilateral: a = a = a, where b=a, c=a, and a >0

If necessary, develops additional test cases to cover the aforementioned conditions.

1. Based on the developed test cases, derive the appropriate test oracle. (Hint: use Excel table)

*class DetermineTriangle*

*{*

*public static void triangle (int a, int b, int c)*

*{*

*int min,med, max;*

*if (a>b)*

*{*

*max=a;*

*min = b;*

*}*

*else*

*{*

*max = b;*

*min = a;*

*}*

*if (c>max)*

*max = c;*

*else if (c<max)*

*min = c;*

*med = a+b+c-min-max;*

*if (max>min+med)*

*System.out.println( "Impossible triangle");*

*else if (max==min)*

*System.out.println( "Equilateral triangle");*

*else if (max==med||med==min)*

*System.out.println( "Isoceles triangle");*

*else if (max\*max==min\*min + med\*med)*

*System.out.println( "Rightangled triangle");*

*else*

*System.out.println(“Any triangle”);;*

*}*

*}*

**Figure 2. Code Unit to DetermineTriangle.java**