Jeel Viradiya
202101164
Lab 10 Software Report

Question 1

Test Case	Day	Month	Year	Expected Outcome	Reason for Invalid Date
Equivalence Partitioning Test Cases: **1. Valid Month Equivalence Class:**					
1	15	2	2005	Yes	
2	15	1	2005	Yes	
3	15	12	2005	Yes	

4	15	0	2005	An invalid date	Invalid month (0)
5	15	13	2005	An invalid date	Invalid month (13)
2. Valid Day Equivalence Class:					
6	5	6	1999	Yes	
7	1	6	1999	Yes	
8	31	6	1999	Yes	
9	0	6	1999	An invalid date	Invalid day (0)
10	32	6	1999	An invalid date	Invalid day (32)

3. Valid Year Equivalence Class:					
11	10	4	1900	Yes	
12	10	4	1901	Yes	
13	10	4	2015	Yes	
14	10	4	1899	An invalid date	Invalid year (1899)
4. Combined Equivalence Classes:					
15	10	4	2016	An invalid date	Invalid year (2016)
16	29	2	2000	Yes	

17	5	15	2005	An invalid date	Invalid month (15)
18	31	6	1988	An invalid date	Invalid day (31)
19	20	9	2025	An invalid date	Invalid year (2025)
Boundary Value Analysis Test Cases: 5. Minimum Values:					
20	1	1	1900	Yes	
21	1	1	1899	An invalid date	Invalid year (1899)
6. Maximum Values:					
22	31	12	2015	Yes	

23	31	12	2016	An invalid date	Invalid year (2016)
7. Leap Year Testing:					
24	29	2	2000	Yes	
25	29	2	1900	An invalid date	Invalid year (1900)
8. Edge Cases:					
26	1	1	1900	Yes	
27	12	31	2015	Yes	

Question 2

.a) Equivalence Classes for the System:

1. Scalene Triangle: A triangle with no sides of equal length.

2. Isosceles Triangle: A triangle with two sides of equal length.

3. Equilateral Triangle: A triangle with all sides of equal length.

4. Right-Angled Triangle: A triangle where the Pythagorean theorem holds (A^2 + B^2 = C^2).

5. Non-Triangle: Impossible to form a triangle with the given sides (A + B <= C).

b) Test Cases to Cover Equivalence Classes:

1. Scalene Triangle: A=3.0, B=4.0, C=5.0

2. Isosceles Triangle: A=3.0, B=4.0, C=4.0

3. Equilateral Triangle: A=3.0, B=3.0, C=3.0

4. Right-Angled Triangle: A=5.0, B=12.0, C=13.0

5. Non-Triangle: A=2.0, B=3.0, C=10.0

c) Boundary Condition A + B > C (Scalene Triangle):

1. A=0.1, B=0.2, C=0.3 (Minimum values where A + B > C)

d) Boundary Condition A = C (Isosceles Triangle):

1. A=3.0, B=3.0, C=4.0 (A = C)

2. A=4.0, B=4.0, C=4.0 (A = B = C)

e) Boundary Condition A = B = C (Equilateral Triangle):

1. A=1.0, B=1.0, C=1.0 (Minimum values where A = B = C)

2. A=9.0, B=9.0, C=9.0 (Maximum values where A = B = C)

f) Boundary Condition A^2 + B^2 = C^2 (Right-Angled Triangle):

1. A=3.0, B=4.0, C=5.0 (A^2 + B^2 = C^2, minimum values)

2. A=5.0, B=12.0, C=13.0 (A^2 + B^2 = C^2, maximum values)

g) Non-Triangle Case (Boundary Exploration):

1. A=1.0, B=2.0, C=3.0 (A + B <= C)

2. A=3.0, B=2.0, C=1.0 (A + B equals C, not less than)

3. A=6.0, B=10.0, C=3.0 (A + C equals B, not less than)

h) Non-Positive Input:

1. A=-1.0, B=-2.0, C=-3.0

2. A=0.0, B=0.0, C=0.0

3. A=4.0, B=-5.0, C=6.0

4. A=7.0, B=8.0, C=-9.0

5. A=0.0, B=4.0, C=5.0

These test cases should cover the identified equivalence classes and boundary conditions. You can use them to test the program with floating-point inputs and ensure that the expected outcomes match the actual outcomes.

Program such that it runs on eclipse

```
public class TriangleClassifierApp {
    final static int EQUILATERAL = 0;
```

```
final static int ISOSCELES = 1;
final static int SCALENE = 2;
final static int INVALID = 3;

public static int classifyTriangle(int a, int b, int c) {
    if (a >= b + c || b >= a + c || c >= a + b)
        return INVALID;

    if (a == b && b == c)
        return EQUILATERAL;

    if (a == b || a == c || b == c)
        return ISOSCELES;

    return SCALENE;
}

public static void main(String[] args) {
    int side1 = 5;
    int side2 = 5;
    int side3 = 5;

    int result = classifyTriangle(side1, side2, side3);

    String classification = "";
    switch (result) {
        case EQUILATERAL:
            classification = "Equilateral";
            break;
        case ISOSCELES:
            classification = "Isosceles";
            break;
        case SCALENE:
            classification = "Scalene";
            break;
        case INVALID:
            classification = "Invalid";
            break;
```

```
        }

        System.out.println("The triangle is " + classification);
    }
}
```