Daanish Husain

Steve Lai

Anjani Mallampati

Paul Diaz

CMPE 187 Team 09

Assignment #8: Writing Test Cases II

1. **Sequential Testing**
2. Make sure there are 3 inputs
   1. <x != null, True>
   2. <y != null, True>
   3. <z != null, True>
3. Test if all 3 inputs are integers
   1. <x = integer, True>
   2. <y = integer, True>
   3. <z = integer, True>
4. Test if all input integers > 0
   1. <x > 0, True>
   2. <y > 0, True>
   3. <z > 0, True>
5. Then apply Triangle Inequality Theorem
   1. <x + y > z, is a triangle>
   2. <x + z > y, is a triangle>
   3. <y + z > x, is a triangle>
6. Then test what kind of triangle (equilateral, isosceles, or scalene)
   1. Equilateral Triangle
      1. <x = y, True>
      2. <y = z, Equilateral>
   2. Isosceles Triangle
      1. Case 1
         1. <x = y, True>
         2. <y != z, True>
         3. <y > z or y < z, Isosceles>
      2. Case 2
         1. <z = y, True>
         2. <y != x, True>
         3. <y > x or y < x, Isosceles>
      3. Case 3
         1. <x = z, True>
         2. <z != y, True>
         3. <y > x or y < x, Isosceles>

OR

1. <x = z, True>

2. <z != y, True>

3. <y > z or y < z, Isosceles>

* 1. Scalene Triangle
     1. <x!=y, true>
     2. <y!=z, Scalene>

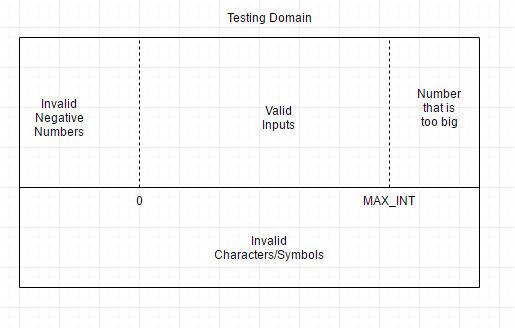
1. **Pseudocode**
2. Initialize x,y, and z as integers
3. Input three values for lengths of triangle sides
4. If values are null
   1. Inform user that a value is null and prompt for lengths again
5. If values are not integers
   1. Inform user that all values need to be integers and prompt for lengths again
6. If values are not greater than zero
   1. Inform user that all values need to be greater and prompt for lengths again
7. If (x+y)<z || (x+z)<y || (y+z)<x
   1. Inform user that the input lengths cannot form a triangle and restart program
8. If x==y && y==z
   1. Inform user that the given inputs form an equilateral triangle
9. If x==y && y != z
   1. Inform user that the given inputs form an isosceles triangle
10. If x==z && z != y
    1. Inform user that the given inputs form an isosceles triangle
11. If z==y && x != z
    1. Inform user that the given inputs form an isosceles triangle
12. If x==y && y != z
    1. Inform user that the given inputs form an isosceles triangle
13. If x!=y && y != z && x!=z
    1. Inform user that the given inputs form an scalene triangle
14. **Equivalence Partitioning**

Assumptions:

* In order or three sides to make a triangle the sum of two sides must be greater than and not equal to the third side.
* Input that is nonnumeric is allowed but will throw an exception
* Negative numbers can be inputted but will throw an exception
* Let MAX\_INT be the largest numeric value that an integer can be

|  |  |  |
| --- | --- | --- |
| **Test Input** | **Expected Result** | **Equivalence Class** |
| x==y && y==z | Equilateral triangle | Equilateral |
| x==y && y != z | Isosceles triangle | Isosceles |
| x==z && z != y | Isosceles triangle | Isosceles |
| z==y && x != z | Isosceles triangle | Isosceles |
| x!=y && y != z && x!=z | Scalene triangle | Scalene |
| x==y && z==0 | Not possible | None |

1. **Boundary Value Testing**



|  |  |  |  |
| --- | --- | --- | --- |
| **Test Data** | **Actual output** | **Expected output** | **Fault detected** |
| x=1, y=1, z=1 | Valid equilateral triangle | Valid equilateral triangle | No |
| x=MAX\_INT-1, y=MAX\_INT-1, z=MAX\_INT-2 | Valid isosceles triangle | Valid isosceles triangle | No |
| x=MAX\_INT-1, y=MAX\_INT-3, z=MAX\_INT-2 | Valid scalene triangle | Valid scalene triangle | No |
| x=MAX\_INT+1, y=MAX\_INT-1, z=MAX\_INT-1 | Invalid input, throw number too large error | Invalid input, throw number too large error | No |
| x=MAX\_INT-1, y=MAX\_INT+1, z=MAX\_INT-1 | Invalid input, throw number too large error | Invalid input, throw number too large error | No |
| x=MAX\_INT-1, y=MAX\_INT-1, z=MAX\_INT+1 | Invalid input, throw number too large error | Invalid input, throw number too large error | No |
| x=-1, y=1, z=1 | Invalid input, throw no negative numbers allowed error | Invalid input, throw no negative numbers allowed error | No |
| x=1, y=-1, z=1 | Invalid input, throw no negative numbers allowed error | Invalid input, throw no negative numbers allowed error | No |
| x=1, y=1, z=-1 | Invalid input, throw no negative numbers allowed error | Invalid input, throw no negative numbers allowed error | No |
| x=a, y=1, z=1 | Invalid input, throw no characters allowed error | Invalid input, throw no characters allowed error | No |
| x=1, y=a, z=1 | Invalid input, throw no characters allowed error | Invalid input, throw no characters allowed error | No |
| x=1, y=1, z=a | Invalid input, throw no characters allowed error | Invalid input, throw no characters allowed error | No |
| x=!, y=1, z=1 | Invalid input, throw no symbols allowed error | Invalid input, throw no symbols allowed error | No |
| x=1, y=!, z=1 | Invalid input, throw no symbols allowed error | Invalid input, throw no symbols allowed error | No |
| x=1, y=1, z=! | Invalid input, throw no symbols allowed error | Invalid input, throw no symbols allowed error | No |

1. **Decision Tables**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Condition*** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| There are three inputs | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No |  |  |  |
| All three inputs are integers | Yes | Yes | Yes | Yes | Yes | Yes | Yes |  | No |  |  |
| All three inputs are greater than 0 | Yes | Yes | Yes | Yes | Yes | Yes | Yes |  |  | No |  |
| Triangle Inequality Theorem applicable | Yes | Yes | Yes | Yes | Yes | Yes | Yes |  |  |  | No |
| Lengths of the 3 sides? |  | x==y && y==z | x==y && y!=z | x==z && z!=y | z==y && x!=z | x==y && y!=z | x!=y && y!=z && x!=z |  |  |  |  |
| ***Type of Triangle*** |  |  |  |  |  |  |  |  |  |  |  |
| Invalid Triangle |  |  |  |  |  |  |  | Yes | Yes | Yes | Yes |
| Valid Triangle | Yes | Yes | Yes | Yes | Yes | Yes | Yes |  |  |  |  |
| Equilateral |  | Yes |  |  |  |  |  |  |  |  |  |
| Isosceles |  |  | Yes | Yes | Yes | Yes |  |  |  |  |  |
| Scalene |  |  |  |  |  |  | Yes |  |  |  |  |

1. **Mutation Testing**

Mutation 1: Change pseudocode line 6 from true to false.

Mutation 2: Change pseudocode line 7 from x=y to x=z.

Mutation 3: Change pseudocode line 4 to if values are integers.

Mutation 4: Change pseudocode line 5 to if values are greater than zero.

Mutation 5: Change pseudocode line 1 and assign x, y, z to null.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mutation** | **Input** | | | | **Expected Output** |
|  | **x** | **y** | **z** | **Response** |  |
| 1 | 5 | 6 | 17 | Found | An invalid triangle will be categorized as a scalene triangle |
| 2 | 1 | 1 | 1 | Not Found | If x= z instead of y = z, test still pass expected triangle will still be equilateral. |
| 3 | 8 | a | g | Found | System informs user that inputs must be all integers |
| 4 | 0 | 1 | -1 | Found | System tests the validity of all inputs, that is all inputs must be all greater than 0. System informs user an appropriate error. |
| 5 | null | null | null | Found | System halts since all inputs are all null. System informs user an appropriate error. |