Software Testing

Tutorial 3

equivalence class partitioning program

Design and develop a program in C or Java language to solve the triangle problem defined as follows :

Accept three integers which are supposed to be the three sides of triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Derive test cases for your program based on equivalence class partitioning, execute the test cases and discuss the results

Test Case Name : Equivalence class Analysis for triangle problem

Test Data : Enter the 3 Integer Value(a , b And c)
Pre-condition :

$1 \leq a \leq 10$, $1 \leq b \leq 10$ and $1 \leq c \leq 10$ and $a < b + c$, $b < a + c$ and $c < a + b$

Brief Description : Check whether given value for a equilateral, isosceles , Scalene triangle or can't from a triangle

Triangle Problem -Equivalence Class Test cases for input data

Weak Equivalence class Testing								
Case Id	**Description**	**Input Data**			**Expected Output**	**Actual Output**	**Status**	**Comments**
		a	**b**	**c**				
1	Enter the min value for a , b and c	5	5	5	Should display the message Equilateral triangle			
2	Enter the min value for a , b and c	2	2	3	Should display the message Isosceles triangle			
3	Enter the min value for a , b and c	3	4	5	Should display the message Scalene triangle			
4	Enter the min value for a , b and c	4	1	2	Message should be displayed can't form a triangle			

Weak Robust Equivalence class Testing								
5	Enter one invalid input and two valid value for a , b and c	-1	5	5	Should display value of a is not in the range of permitted values			
6	Enter one invalid input and two valid value for a , b and c	5	-1	5	Should display value of a is not in the range of permitted values			
7	Enter one invalid input and two valid value for a , b and c	5	5	-1	Should display value of a is not in the range of permitted values			
8	Enter one invalid input and two valid value for a , b and c	11	5	5	Should display value of a is not in the range of permitted values			
9	Enter one invalid input and two valid value for a , b and c	5	11	5	Should display value of a is not in the range of permitted values			
10	Enter one invalid input and two valid value for a , b and c	5	5	11	Should display value of a is not in the range of permitted values			
Strong Robust Equivalence class Testing								
11	Enter one invalid input and two valid value for a , b and c	-1	5	5	Should display value of a is not in the range of permitted values			
12	Enter one invalid input and two valid value for a , b and c	5	-1	5	Should display value of a is not in the range of permitted values			
13	Enter one invalid input and two valid value for a , b and c	5	5	-1	Should display value of a is not in the range of permitted values			
14	Enter two invalid input and two valid value for a , b and c	-1	-1	5	Should display value of a is not in the range of permitted values			
					Should display value of b is not in the range of permitted values			
14	Enter two invalid input	5	-1	-1	Should display value of b is not in the range of permitted			

	and two valid value for a , b and c				values			
					Should display value of c is not in the range of permitted values			
14	Enter two invalid input and two valid value for a , b and c	-1	5	-1	Should display value of a is not in the range of permitted values			
					Should display value of c is not in the range of permitted values			
15	Enter all invalid inputs	-1	-1	-1	Should display value of a is not in the range of permitted values			
					Should display value of b is not in the range of permitted values			
					Should display value of c is not in the range of permitted values			

Equivalence Class Testing Types:

The equivalence class testing can be categorized into four different types, which are integral part of testing and cater to different data set.

Weak Normal Equivalence Class Testing: In this first type of equivalence class testing, one variable from each equivalence class is tested by the team. Moreover, the values are identified in a systematic manner. Weak normal equivalence class testing is also known as **single fault assumption**.

1. **Strong Normal Equivalence Class Testing:** Termed as **multiple fault assumption**, in strong normal equivalence class testing the team selects test cases from each element of the Cartesian product of the equivalence. This ensures the notion of

completeness in testing, as it covers all equivalence classes and offers the team one of each possible combinations of inputs.

2. **Weak Robust Equivalence Class Testing:** Like weak normal equivalence, weak robust testing too tests one variable from each equivalence class. However, unlike the former method, it is also focused on testing test cases for invalid values.

3. **Strong Robust Equivalence Class Testing:** Another type of equivalence class testing, strong robust testing produces test cases for all valid and invalid elements of the product of the equivalence class. However, it is incapable of reducing the redundancy in testing.

Advantages & Disadvantages of Equivalence Class Testing:

Equivalence class testing or equivalence partitioning plays a potent role in reducing redundancy in testing and making the process agile and powerful. It is among those testing techniques that offer numerous benefits to the team and ensures compliance of the product with customer requirements. However, there are few drawbacks associated to this type of testing, which are listed below along with its various advantages.

Advantages:

- Equivalence class testing helps reduce the number of test cases, without compromising the test coverage.

- Reduces the overall **test execution** time as it minimizes the set of **test data**.

- It can be applied to all levels of testing, such as **unit testing**, **integration testing**, **system testing**, etc.

- Enables the testers to focus on smaller data sets, which increases the probability to uncovering more defects in the software product.

- It is used in cases where performing **exhaustive testing** is difficult but at the same time maintaining good coverage is required.

Disadvantages:

- It does not consider the conditions for boundary value.
- The identification of equivalence classes relies heavily on the expertise of testers.
- Testers might assume that the output for all input data set are correct, which can become a great **hurdle in testing.**