Assignment 8 - Writing Test Cases - II

Team 16

Test Cases

#	Test-cases	Inputs (x, y, z)	Expected results
1	User doesn't write anything in the input fields.	(, ,)	User gets error message, saying to write an valid input. And the program returns to the idle state.
2	User enters 3 integer values	(1, 2, 3)	The program should return a statement saying the following "This is not a triangle as it does not satisfy the triangle inequality theorem."
3	User writes just one integer	(,5,)	The program should give the user a message saying that there are missing two inputs.
4	User enters one float, one double and one integer value	(12.3145, 2.0, 3)	This should not be allowed, and the program should not accept the inputs.
5	User enters 3 integer values	(2,2,2)	The program should return a statement saying the following "This triangle is a equilateral".
6	User writes a string value with two integers	(2, abc, 5)	The program should return a message stating that string values is not allowed, and tell the user to enter 3 integers.
7	User tries to enter more than 3 integers or values.	(1, 2.2, f, 124)	The system should pick up that there are too many values or that the values are not integer values. The

			system should return a decent message accordingly.
8	User enters 3 integer values	(2, 2, 3)	The program should return a statement saying the following "This triangle is isosceles"
9	User writes 3 integer values	(2, 3, 0)	The program should state that all integer values should be over 1 . ($x < 0$ and less than the integer boundary value)
10	User tries giving a negative value	(-1 ,2 ,2)	The program should state that it could not calculate the given input. Negative numbers should not be allowed. Eventually do a
11	User tries to input all non integer values	(a, b, c)	The program should state that this is an invalid entry because all the values are not integers.
12	User inputs invalid values	(. , w , -)	The program should state that this is an invalid entry because all the values are not integers.
13	User enters valid values	(4, 4, 4)	The program should return a statement saying the following "This triangle is an equilateral".

Testing Concept 1 - FSM



Testing Concept 2 - Boundary value testing

Assumption: All inputs for each side of the triangles should be within 0 and 40. 40 is the chosen boundary value for this particular problem.

Range: {0< a,b,c <40}

*The system should <u>NOT</u> accept values that are out of this range

*The system should <u>ONLY</u> accept positive numerical values as valid inputs

Test Case	Expected Output	Actual Output
(,,)	"Invalid Entry"	"Invalid Entry"
(1, 2, 3)	"Scalene Triangle"	"Scalene Triangle"
(,5,)	"Invalid Entry"	"Invalid Entry"
(12.3145, 2.0, 3)	"Invalid Entry"	"Invalid Entry"
(2,2,2)	"Equilateral Triangle"	"Equilateral Triangle"
(2, abc, 5)	"Invalid Entry"	"Invalid Entry"
(1, 2.2, f, 124)	"Invalid Entry"	"Invalid Entry"
(2, 2, 3)	"Isosceles Triangle"	"Isosceles Triangle"
(2, 3, 0)	"Scalene Triangle"	"Scalene Triangle"
(-1 ,2 ,2)	"Invalid Entry"	"Invalid Entry"
(a, b, c)	"Invalid Entry"	"Invalid Entry"
(. , w , -)	"Invalid Entry"	"Invalid Entry"
(4, 4, 4)	"Equilateral Triangle"	"Equilateral Triangle"

Equivalence Partitioning:

Valid: Integers

Invalid: Negative numbers String/Signs Double/Float

Test Input	Expected Output	Equivalence Class
(,,)	"Invalid Triangle"	String/Signs
(1, 2, 3)	"Scalene Triangle"	Integers
(,5,)	"Invalid Triangle"	String/Signs
(12.3145, 2.0, 3)	"Invalid Triangle"	Double/Float
(2,2,2)	"Equilateral triangle"	Integers
(2, abc, 5)	"Invalid Triangle"	String/Signs
(1, 2.2, f, 124)	"Invalid Triangle"	String/Signs
(2, 2, 3)	"Isosceles Triangle"	Integers
(2, 3, 0)	"Scalene Triangle"	Integers
(-1 ,2 ,2)	"Invalid Triangle"	Negative numbers
(a, b, c)	"Invalid Triangle"	String/Signs
(. , w , -)	"Invalid Triangle"	String/Signs
(4, 4, 4)	"Equilateral triangle"	Integers

This is how we decided to do the boundaries, since the program only takes integers we put double and float as invalid number, hence the error message. So the only valid boundary is

integers and all the rest are invalid, so every time we have a value that is not an integer the program should give the user an error message + what the error might be. Usually that the input is in the wrong format.