Topic #1 – Software Black-Box Testing

Instructor: Jerry Gao, Ph.D., Professor
San Jose State University

Topic #1 – Software Black-Box Testing

- What Is Black-Box Testing?
- Black-Box Testing Focuses
- Why Is Black-Box Testing Important?
- Who Does Black-Box Testing?
- A Black-Box Testing Example
- Black-Box Testing Coverage

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

What is software black-box testing?

Black-box testing, also known as requirement-based testing

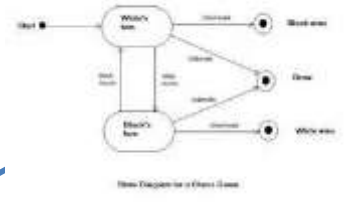
Definition: Software black-box testing refers to test design based on program requirements specifications to validate software functions, external behaviors, and external visible QoS requirements.

What do you need for black-box testing?

- Black-box testing models and test criteria
- Black-box test design and generation methods
- Software requirements specification and product specification

Black-Box Testing Focuses

System QoS Requirements



System Behaviors

Software

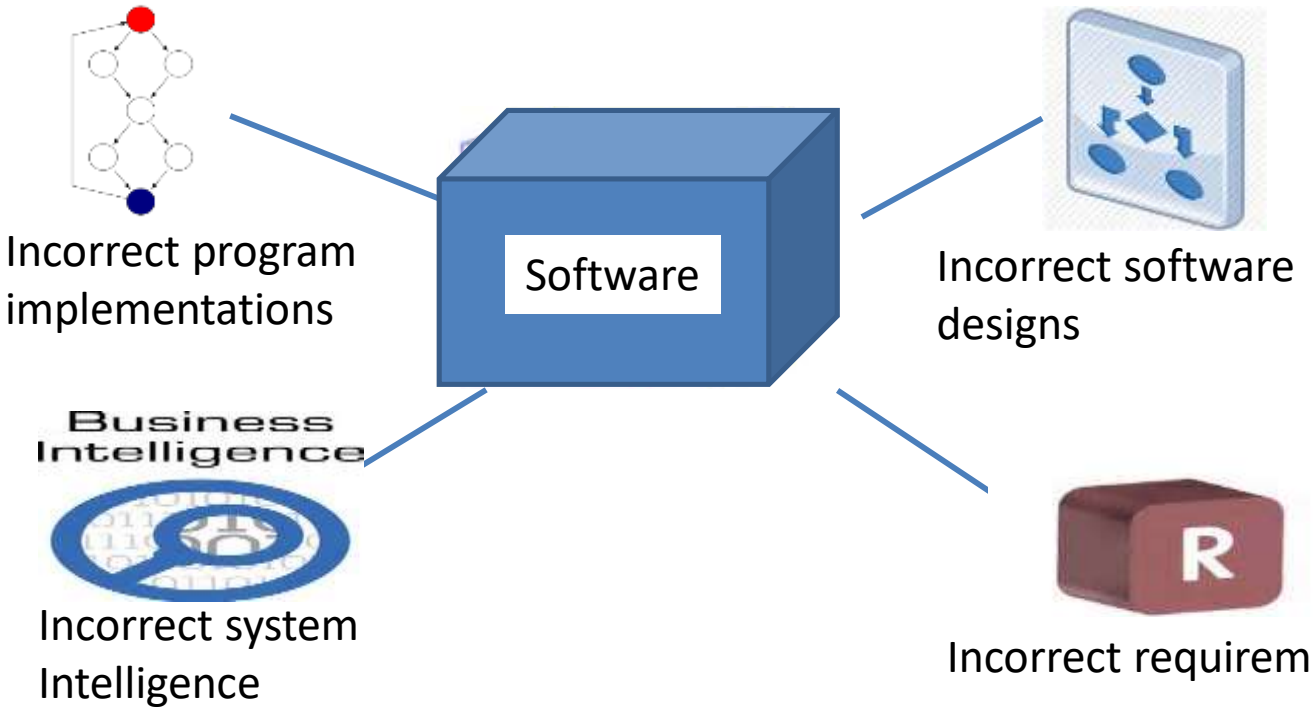
System Functions

Software GUI

- Check system QoS Requirements
- Software GUI
- System Functions
- System Behaviors

Why Is Black-Box Testing Important?

Answer: - Assure the quality of software functions, behaviors and QoS parameters.
- Achieve adequate system requirement validation criteria.



Who Does Black-Box Testing?

Several engineer groups perform black-box program testing:

- Function validation engineers
  – Perform function testing
- System test engineers
  – Conduct system testing for QoS parameters
- Quality assurance engineers
  – Performance system testing based on QA standards
- System users – Acceptance testing

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

A Black-Box Testing Example - Triangle Analyzer

Program specification:

Input: 3 numbers separated by commas or spaces

Processing:

Determine if three numbers make a valid triangle; if not, print message NOT A TRIANGLE.

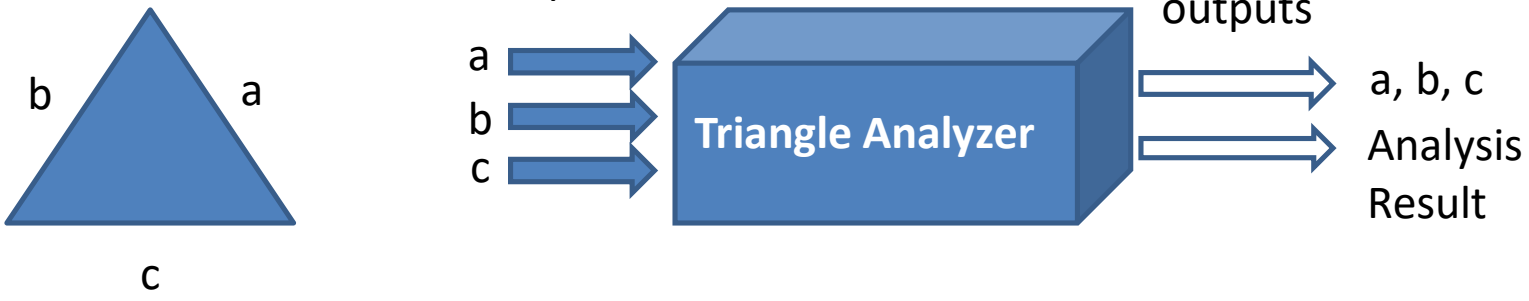
If it is a triangle, classify it according to the length of the sides as scalene (no sides equal), isosceles (two sides equal), or equilateral (all sides equal).

If it is a triangle, classify it according to the largest angle as acute (less than 90 degree), obtuse (greater than 90 degree), or right (exactly 90 degree).

Output: One line listing the three numbers provided as input and the classification or the not a triangle message.

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

A Black-Box Testing Example - Triangle Analyzer

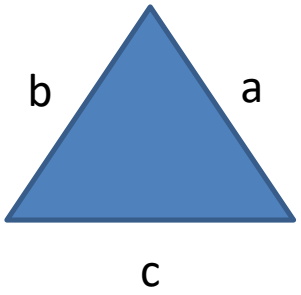


Output: One line listing the three numbers provided as input and the classification or the not a triangle message.

Test Example:	Inputs	Outputs	
	3,4,5	3,4,5	Scalene Right
	6,1,6	6,1,6	Isosceles Acute
	5,1,2	5,1,2	Not a triangle

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

A Black-Box Testing Example - Triangle Analyzer

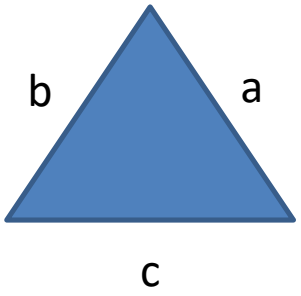


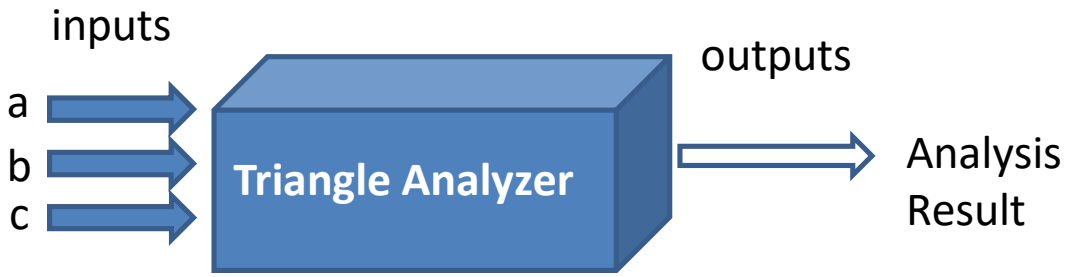
Test Set #1

Inputs	Expected Results	
4,4,4	4,4,4	Equilateral acute
6,5,3	6,5,3	Scalene acute
5,6,10	5,6,10	Scalene obtuse
3,4,5	3,4,5	Scalene right
6,1,6	6,1,6	Isosceles acute
7,4,4	7,4,4	Isosceles obtuse
1,2,2	1,2,2	Isosceles right

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

A Black-Box Testing Example - Triangle Analyzer





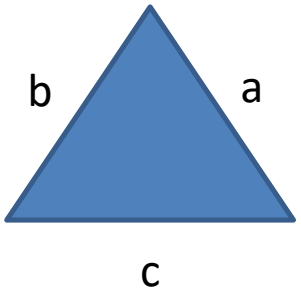
Test Set #2

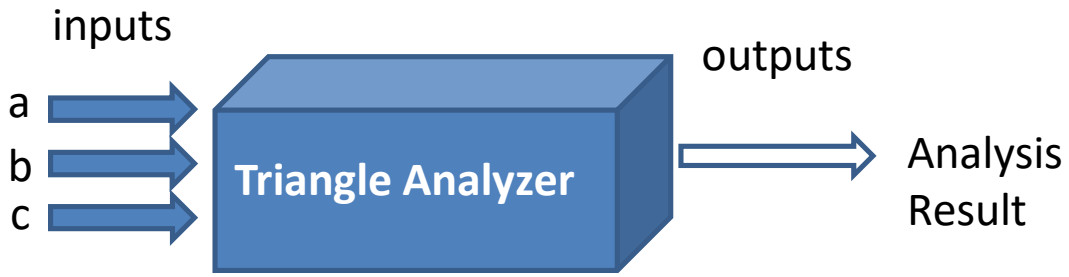
Test cases for special inputs and invalid formats:

Inputs	Descriptions
3,4,5,6	Four sides
646	Three-digit single number
3,,4,5	Two commas
3 4,5	Missing comma
3.14.6,4,5	Two decimal points
4,6	Two sides
5,5,A	Character as a side
6,-4,6	Negative number as a side
-3,-3,-3	All negative numbers
	Empty input

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

A Black-Box Testing Example - Triangle Analyzer





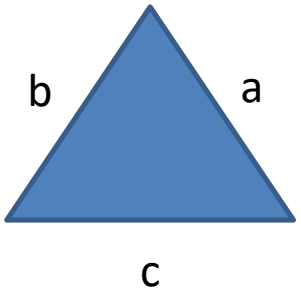
Test Set #3

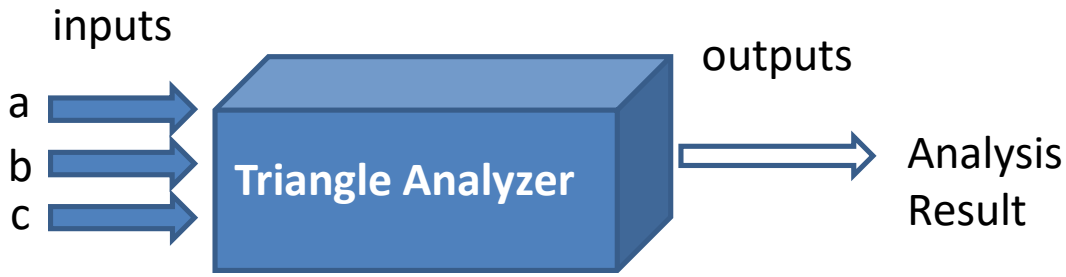
Boundary conditions for legitimate triangles:

Inputs	Descriptions
1,1,2	Makes a straight line, not a triangle
0,0,0	Makes a point, not a triangle
4,0,3	A zero side, not a triangle
1,2,3.00001	Close to a triangle but still not a triangle
9170,9168,3	Very small angle (scalene, acute)
.0001,.0001,.0001	Very small triangle (equilateral, acute)
83127168,74326166,96652988	Very large triangle, scalene, obtuse

TOPIC #1 – SOFTWARE BLACK-BOX TESTING

A Black-Box Testing Example - Triangle Analyzer





Test Set #3

Boundary conditions for sides classification:

Inputs	Descriptions
3.0000001,3,3	Very close to equilateral (isosceles, acute)
2.999999,4,5	Very close to isosceles (scalene, acute)

Boundary conditions for angles classification:

Inputs	Descriptions
3,4,5.000000001	Near right triangle (scalene, obtuse)
1,1,1.41141414141414	Near right triangle (isosceles, acute)

Software Testing Principles

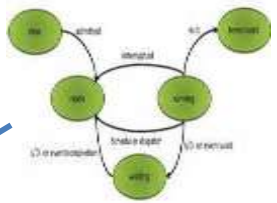
A set of testing principles listed below:

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- Testing should begin “in the small” and progress toward testing “in the large”.
- Exhaustive testing is not possible.
- Testing should be conducted by an independent third party.

Black-Box Software Testing Coverage

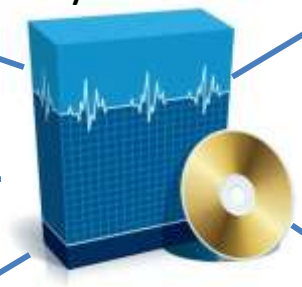
System Function Requirement Coverage

Under-test System



System Behavior Coverage

System QoS Parameter Coverage



Business Roles & Intelligence Coverage

System GUI Coverage