



test case  
design

## What is White Box Testing?

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

## White Box Testing Techniques:

- **Statement Coverage** - This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage** - This technique is running a series of tests to ensure that all branches are tested at least once.
- **Path Coverage** - This technique corresponds to testing all possible paths which means that each statement and branch is covered.

## Calculating Structural Testing Effectiveness:

$\text{Statement Testing} = (\text{Number of Statements Exercised} / \text{Total Number of Statements}) \times 100 \%$

$\text{Branch Testing} = (\text{Number of decisions outcomes tested} / \text{Total Number of decision Outcomes}) \times 100 \%$

$\text{Path Coverage} = (\text{Number paths exercised} / \text{Total Number of paths in the program}) \times 100 \%$

## Advantages of White Box Testing:

- Forces test developer to reason carefully about implementation.
- Reveals errors in "hidden" code.
- Spots the Dead Code or other issues with respect to best programming practices.

## Disadvantages of White Box Testing:

- Expensive as one has to spend both time and money to perform white box testing.
- Every possibility that few lines of code are missed accidentally.
- In-depth knowledge about the programming language is necessary to perform white box testing.

# What is White Box Testing?

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

## White Box Testing Techniques:

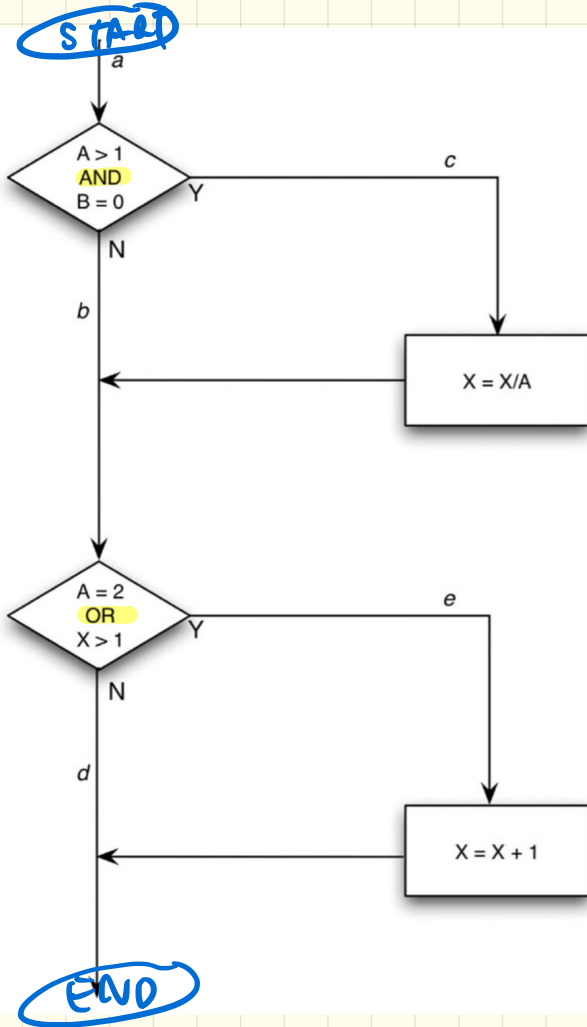
- **Statement Coverage** - This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage** - This technique is running a series of tests to ensure that all branches are tested at least once.
- **Path Coverage** - This technique corresponds to testing all possible paths which means that each statement and branch is covered.

## Advantages of White Box Testing:

- ▣ Forces test developer to reason carefully about implementation.
- ▣ Reveals errors in "hidden" code.
- ▣ Spots the Dead Code or other issues with respect to best programming practices.

## Disadvantages of White Box Testing:

- ▣ Expensive as one has to spend both time and money to perform white box testing.
- ▣ Every possibility that few lines of code are missed accidentally.
- ▣ In-depth knowledge about the programming language is necessary to perform white box testing.



READ A

READ B

IF A > 1 AND B == 0 THEN

$X \leftarrow X/A$

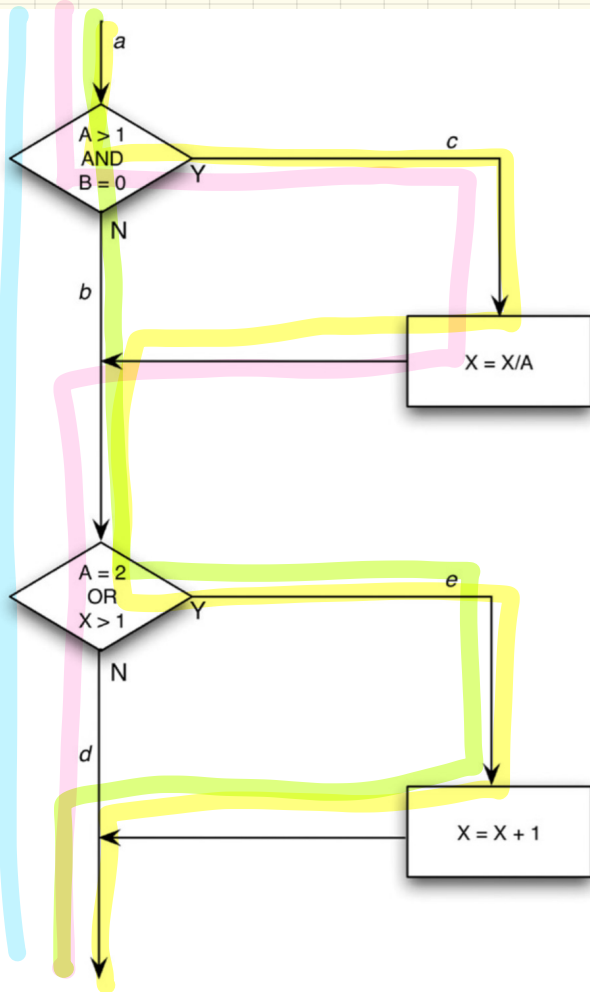
ELSE

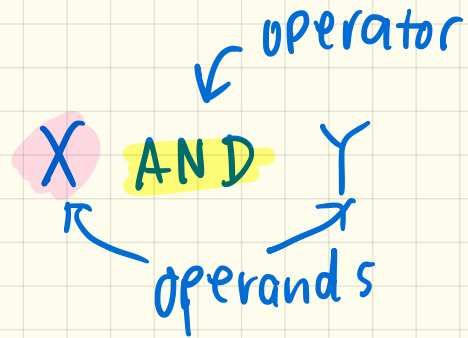
IF A == 2 OR X > 1 THEN

$X \leftarrow X + 1$

ENDIF

ENDIF





Y gets executed  
only when X is True

X OR Y

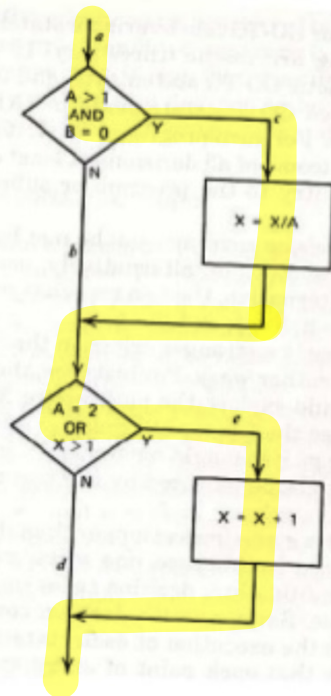
Y gets executed  
when?  
X is False



# Statement Coverage

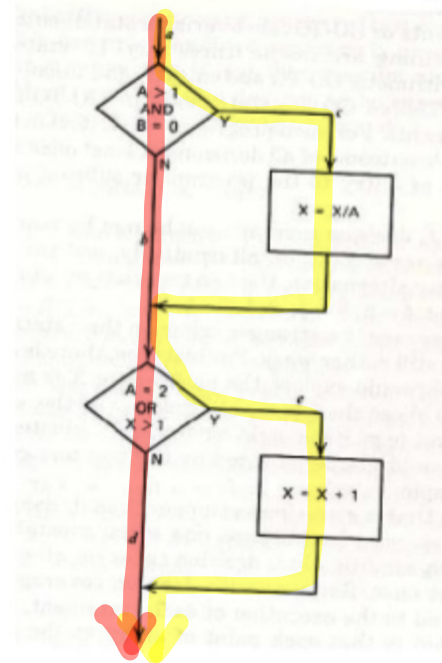
- Exercise all statements at least once
- How many test cases?

➤ A=2 and B=0 (ace)



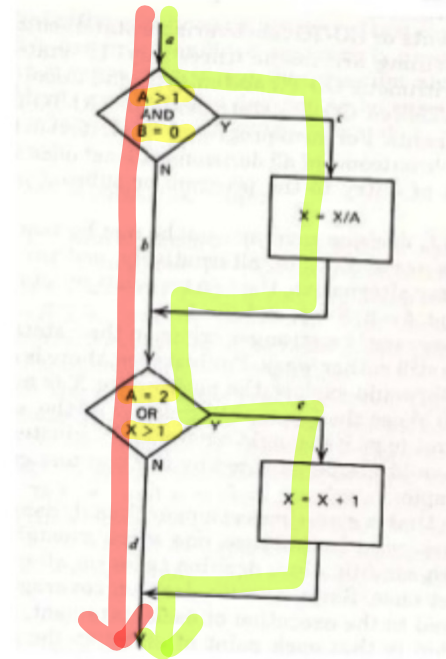
# Decision/Branch Coverage

- Each decision has a true and a false outcome at least once
- How many test cases?
  - A=2 and B=0 (ace)
  - A=1 and X=1 (abd)



# Condition Coverage

- Each condition in a decision takes on all possible outcomes at least once
- Conditions:  $A > 1$ ,  $B = 0$ ,  $A = 2$ ,  $X > 1$
- How many test cases?
  - $A=2$ ,  $B=0$ , and  $X=4$  (ace)
  - $A=1$ ,  $B=1$ , and  $X=1$  (abd)



# Decision/Condition Coverage

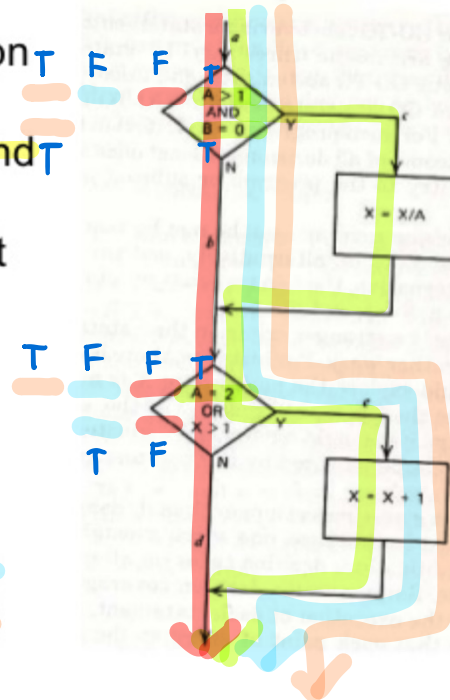
- Each condition in a decision takes on all possible outcomes at least once, and each decision takes on all possible outcomes at least once

- How many test cases?
  - A=2, B=0, and X=4 (ace)
  - A=1, B=1, and X=1 (abd)

- What about these?
  - A=1, B=0, and X=3 (abe)
  - A=2, B=1, and X=1 (abe)



cond. and decision T/N/F



Share



Like



Save

# Multiple Condition Coverage

- Exercise all possible combinations of condition outcomes in each decision
- Conditions:

$A > 1, B = 0$

$A > 1, B \neq 0$

$A \leq 1, B = 0$

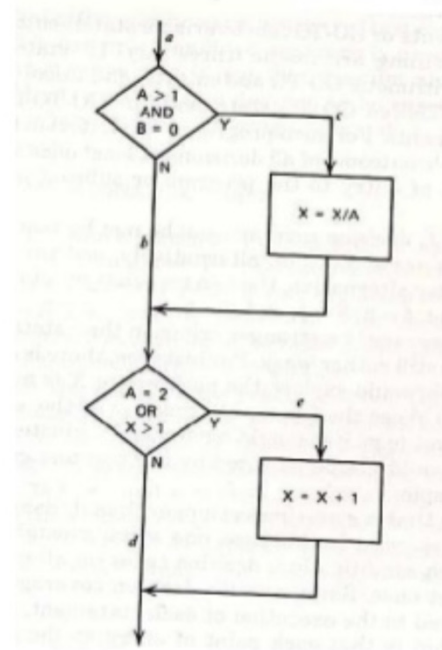
$A \leq 1, B \neq 0$

$A = 2, X > 1$

$A = 2, X \leq 1$

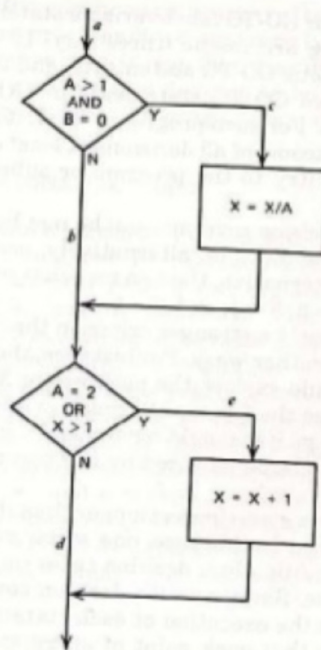
$A \neq 2, X > 1$

$A \neq 2, X \leq 1$



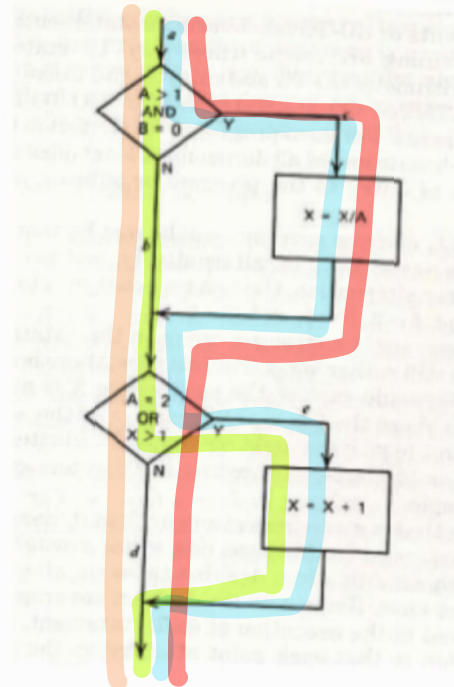
# Multiple Condition Coverage

- How many test cases?
  - A=2, B=0, X=4 (ace)
  - A=2, B=1, X=1 (abe)
  - A=1, B=0, X=2 (abe)
  - A=1, B=1, X=1 (abd)



# Path Coverage

- Every unique path through the program is executed at least once
- How many test cases?
  - A=2, B=0, X=4 (ace)
  - A=2, B=1, X=1 (abe)
  - A=3, B=0, X=1 (acd)
  - A=1, B=1, X=1 (abd)



# McCabe's Cyclomatic Complexity

- Software metric
- Developed by Tom McCabe (circa 1976)
- Directly measures the number of linearly independent paths through a program's source code, taking into account the various decision points
- Independent of implementation language

Source: Wikipedia