White Box Testing

White Box Design

- Also called as glass-box testing.
- Uses the control structure of the procedural design to derive test cases.
- Using white-box testing we can derive test cases that
  - Guarantee that all independent paths within a module have been exercised at least once,
  - Exercise all logical decisions on their true and false sides,
  - Execute all loops at their boundaries and within their operational bounds,
  - Exercise internal data structures to ensure their validity.

White Box Design

- "Why spend time and energy worrying about (and testing) logics when we might better expend effort ensuring that program requirements have been met?“

OR

- Why don't we spend all of our energy on black-box tests?

White Box Design

- The answer lies in the nature of software defects:
  - Logical errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed
  - We often believe that a logical path is not likely to be executed when, in fact, it may be executed on a regular basis.
  - Typographical errors are random
- Each of these reasons provides an argument for conducting white-box tests.

Code Coverage Testing

Since a product is realized in terms of program code, if we can run test cases to exercise the different parts of the code, then that part of the product realized by the code gets tested. Code coverage testing involves designing and executing test cases and finding out the percentage of code that is covered by testing. The percentage of code covered by a test is found by adopting a technique called *instrumentation* of code. There are specialized tools available to achieve instrumentation. Instrumentation rebuilds the product, linking the product with a set of libraries provided by the tool

Code coverage testing is made up of the following types of coverage.

1. Statement coverage
2. Path coverage
3. Condition coverage
4. Function coverage

Statement coverage

- Programs constructs can be usually classified as:

1. Sequential Control Flow
2. Two way decision statements (if then else)
3. Multi way decision statements (switch)
4. Loops (while, do, repeat until, for)

- Statement coverage is a white box test design technique which involves execution of all the executable statements in the source code at least once.
- It is used to calculate and measure the number of statements in the source code which can be executed given the requirements.
- Statement coverage is used to derive scenario based upon the structure of the code under test.

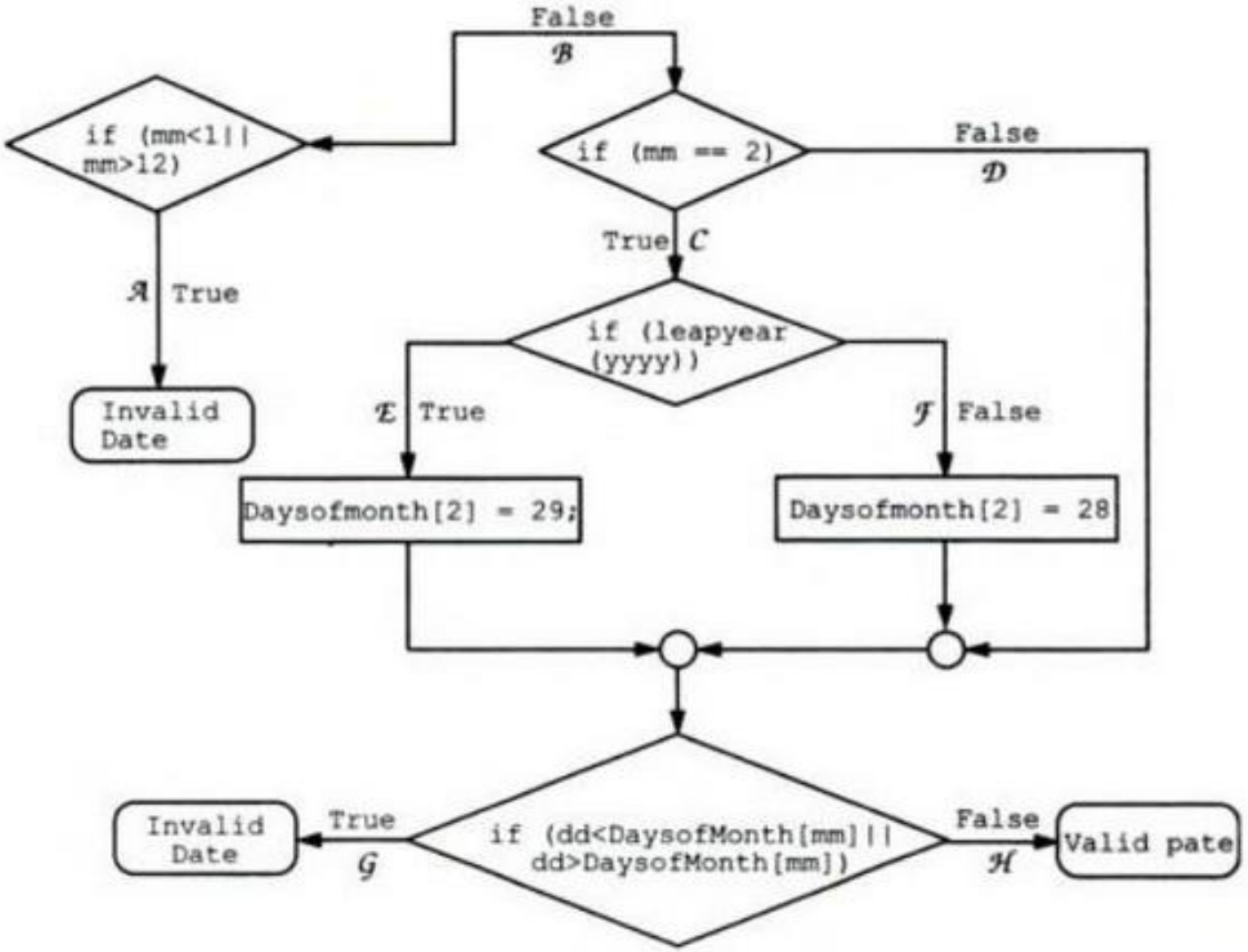
$$\text{Statement Coverage} = \frac{\text{Number of executed statements}}{\text{Total number of statements}} \times 100$$

Path Coverage Testing

- Path testing is a structural testing method that involves using the source code of a program in order to find every possible executable path.
- It helps to determine all faults lying within a piece of code.
- This method is designed to execute all or selected path through a computer program.

Path Coverage = (Total paths exercised/ Total number of paths in program) * 100

Let us take an example of a date validation routine. The date is accepted as three fields `mm`, `dd` and `yyyy`. We have assumed that prior to entering this routine, the values are checked to be numeric. To simplify the discussion, we have assumed the existence of a function called `leapyear` which will return `TRUE` if the given year is a leap year. There is an array called `DayofMonth` which contains the number of days in each month. A simplified flow chart


False
B
if (mm<1||
mm>12)
if (mm == 2)
False
D
True C
A True
if (leapyear
(yyyy))
Invalid
Date
E True
F False
Daysofmonth[2] = 29;
Daysofmonth[2] = 28
Invalid
Date
True
G
if (dd<DaysofMonth[mm]||
dd>DaysofMonth[mm])
False
H
Valid pate

Condition Coverage

- Conditional coverage or expression coverage will reveal how the variables or subexpressions in the conditional statement are evaluated.
- In this coverage expressions with logical operands are only considered.
- Conditional coverage offers better sensitivity to the control flow than decision coverage.
- Condition coverage does not give a guarantee about full decision coverage

$$Condition\ Coverage = \frac{Number\ of\ Executed\ Operands}{Total\ Number\ of\ Operands} * 100$$

BLACK BOX TESTING	WHITE BOX TESTING
It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it.	It is a way of testing the software in which the tester has knowledge about the internal structure r the code or the program of the software.
It is mostly done by software testers.	It is mostly done by software developers.
No knowledge of implementation is needed.	Knowledge of implementation is required.
It can be referred as outer or external software testing.	It is the inner or the internal software testing.

It is functional test of the software.	It is structural test of the software.
This testing can be initiated on the basis of requirement specifications document.	This type of testing of software is started after detail design document.
No knowledge of programming is required.	It is mandatory to have knowledge of programming.
It is the behavior testing of the software.	It is the logic testing of the software.
It is applicable to the higher levels of testing of software.	It is generally applicable to the lower levels of software testing.

It is also called closed testing.	It is also called as clear box testing.
It is least time consuming.	It is most time consuming.
It is not suitable or preferred for algorithm testing.	It is suitable for algorithm testing.
Can be done by trial and error ways and methods.	Data domains along with inner or internal boundaries can be better tested.