

## CH2: Test cases for Simple Programs

### ★ Test case design techniques

- ① Specification-based Test case design Technique. [black-box] [white-box]
- ② Structure-based Test case Design Technique.
- ③ Experience-based Test case Design Technique.

★ The percentage of program statements that can be invoked during testing phase is called code coverage [white-box testing]

$$\text{code coverage} = \frac{\text{No. of lines of code exercised} \times 100}{\text{Total no. of lines of code}}$$

★ Statement coverage is used to calculate the total number of executed statements in the source code out of total statements present in the source code.

$$\text{Statement coverage} = \frac{\text{No. of executed statements} \times 100}{\text{Total number of statements}}$$

★ Eg:

```
1. input (int x, int y) {  
2.   sum = x + y;  
3.   if (sum > 0)  
4.     printf("This is positive result");  
5.   else  
6.     printf("This is negative result");  
7. }
```



Test case 1:  $x = 6, y = 2$

- ✓ 1. input (int x, int y) {
- ✓ 2.     int sum = x + y;
- ✓ 3.     if (sum > 0)
- ✓ 4.         printf (Positive);
- 5.     else
- 6.         printf (Negative);
- ✓ 7. }

Here sum = 8 which is positive so

Statement coverage =  $\frac{\text{No. executed stat.}}{\text{Total no. of stat.}} \times 100$

$$= \frac{5}{7} \times 100$$

$$= 71\%$$



0 25 50 71.75 100

Advantages:-

- ① Verifies what the written code is expected to do and not to do.
- ② measures quality of code.

Disadvantages:-

- ① cannot test false conditions.
- ② Does not understand logical operators.



## ★ Decision coverage.

Eg:-

```
1. Test (int x)
2. { if (x > 4) {
3.     x = x * 3
4.     print (x)
5. }
```

Test case 1:  $x = 9$ 

✓ 1.

✓ 2.

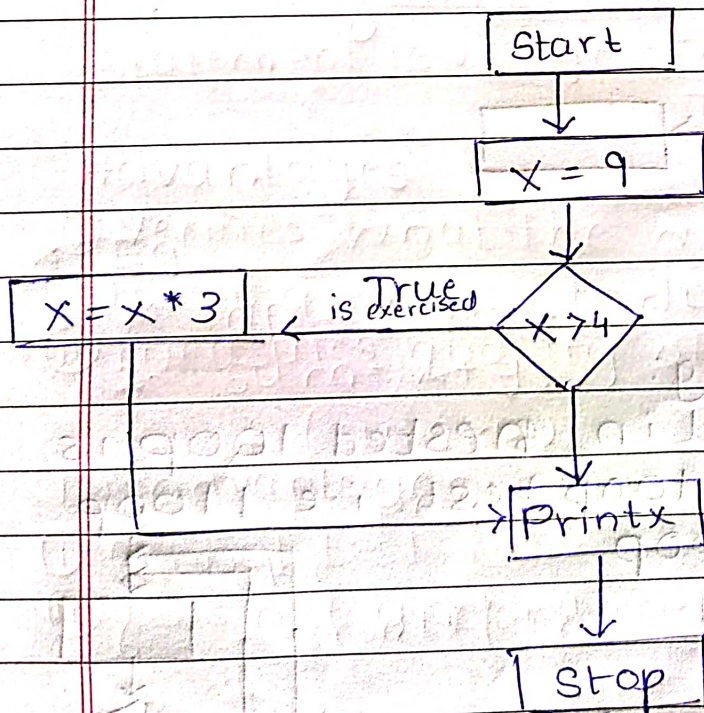
✓ 3.

✓ 4.

✓ 5.

Outcome is "True"

flowchart:-





$$\begin{aligned}
 \text{Decision coverage} &= \frac{\text{No. of decision outcomes exercised}}{\text{Total number of decision outcomes}} \times 100 \\
 &= \frac{1}{2} \times 100 \\
 &= \underline{\underline{50\%}}
 \end{aligned}$$

### Advantages

- ① Validates all branches in code are reached.
- ② Eliminates problems that occur with statement coverage testing.

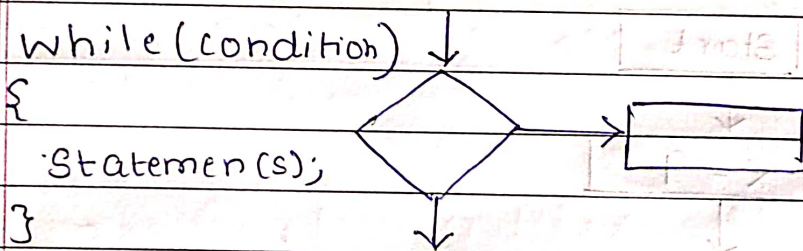
### Disadvantages

- ① Ignores branches within Boolean expressions.

### ★ Types of Loop Testing

#### ① Simple Loop Testing:

- Testing performed in a simple loop is known as simple loop testing.

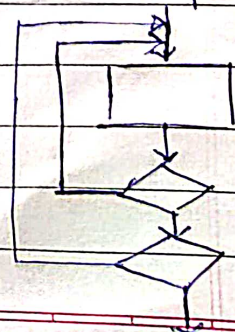


#### ② Nested loop testing:

- Testing performed in nested loop is known as nested loop test. i.e. 1 loop inside another loop.

```

while (cond. 1)
{
    while (condition2)
    {
        statement(s);
    }
}
  
```



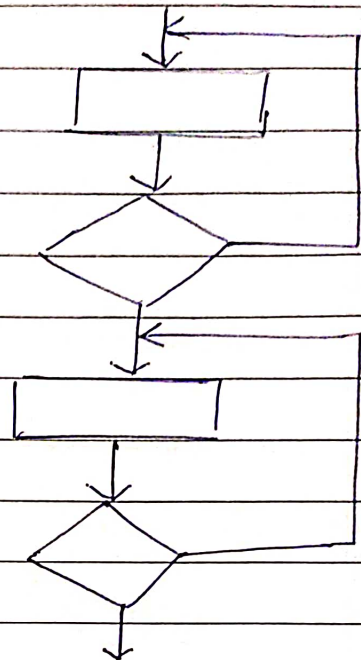


### iii) Concatenated Loop Testing:

- Testing performed in concatenated loop is concatenated loop testing. i.e loops after loops.

```

while (cond.1)
{
    statement(s);
}
while (cond.2)
{
    statement(s);
}
  
```



### iv) Unstructured Loop Testing:

- Testing performed on unstructured loop is known as unstructured loop testing.
- combination of concatenated and nested loops.

```

while ()
{
    for ()
    {
        // nested loop
    }
    while ()
    {
        // nested loop
    }
}
  
```

