

# Software Testing

## Basic Coverage Criteria

**Indranil Saha**

Department of Computer Science and Engineering  
Indian Institute of Technology Kanpur



# Types of Structural Coverage

- Function coverage
- Statement coverage
- Branch coverage
- Decision coverage
- Condition coverage
- Decision/condition coverage
- Multiple condition coverage
- Modified decision/condition coverage

# Function Coverage

- The function coverage of a program counts how many functions were called (and sometimes how often)
- If a function is called once during the execution, it is considered as covered

# Statement Coverage

- Measures how many statements in the code have been executed by a given test suite
- For a straight-line code, one test case is required for full statement coverage
- For the codes with branching, multiple tests are required

# Branch Coverage

- Branch coverage requires every exit from a conditional statement to be executed
- For an if statement, branch coverage requires the then part and the else part to be executed
- If there is no else part, the if statement should still execute the decision as true and false.

# Decision Coverage

- A decision is covered when it is evaluated to both `true` and `false`

# Example: Decision Coverage

Consider the following example:

```
If (a > 0) & (b > 0)
    cout << "Hello" << endl;
else
    cout << "Bye" << endl;
```

Test cases to achieve decision coverage:

$\langle a = 0, b = 0, \text{"Bye"} \rangle$ ,  $\langle a = 1, b = 1, \text{"Hello"} \rangle$

# Difference between Branch Coverage and decision Coverage

- Decision coverage is referenced by DO-178B/DO-178C whereas branch coverage is referenced by ISO 26262
- For a conditional statement, a test suite may provide partial branch coverage. But the decision coverage is full or zero.



- Each Boolean sub-expression of a Boolean expression has to be evaluated to both `true` and `false`

# Example: Condition Coverage

Consider the following example:

```
If (a >0) && (b >0)
    cout << "Hello" << endl;
else
    cout << "Bye" << endl;
```

Test cases to achieve condition coverage:

$\langle a = 0, b = 1, \text{"Bye"} \rangle$ ,  $\langle a = 1, b = 0, \text{"Bye"} \rangle$

# Condition Coverage Does not Imply Decision Coverage

Consider the following example:

```
If (a > 0) && (b > 0)
    cout << "Hello" << endl;
else
    cout << "Bye" << endl;
```

Test cases to achieve condition coverage:

$\langle a = 0, b = 1, \text{"Bye"} \rangle$ ,  $\langle a = 1, b = 0, \text{"Bye"} \rangle$

Does not achieve decision coverage

# Decision Coverage Does not Imply Condition Coverage

Consider the following example:

```
If (a >0) || (b >0)
    cout << "Hello" << endl;
else
    cout << "Bye" << endl;
```

Test cases to achieve decision coverage:

$\langle a = 0, b = 0, \text{"Bye"} \rangle$ ,  $\langle a = 1, b = 0, \text{"Hello"} \rangle$

Does not achieve conditional coverage

# Software Testing

## Basic Coverage Criteria

**Indranil Saha**

Department of Computer Science and Engineering  
Indian Institute of Technology Kanpur

