White Box Testing

What is Statement Coverage?

In statement coverage, the goal is to test every single line of code in the program.
To test, you must have the correct inputs.

Statement Coverage is a strategy that helps you figure out what inputs will enable you to test every line of code.

Choosing Inputs

Statement	Choose inputs that
`if`	Cause the condition to be `true` Cause the condition to be `false`

`for`	Force the loop to run
	    ● 0 times
	    ● 1 times
	    ● More than 1 time

`while`	Force the loop to run
	    ● 0 times
	    ● 1 times
	    ● More than 1 time

`switch`	Run every `case` statement Every `default` statement
`try-catch-finally`	Run the `try` clause Run every `catch` statement Run the `finally` clause

Example: Code with If-Statements

Function Requirements:
- Create a function that detects if a number is even
- Examples of even numbers: 2, 8, -40, 100, etc

Code:

```java
public void evenOdd(int n) {
      if (n % 2 == 0) {
            System.out.println("Number is even");
      }
      else {
            System.out.println("Number is odd");
      }
}
```

Question:
  a) What inputs will provide maximum statement coverage for this function?
  b) For each input, show:
  - The expected output
  - The actual output
  - If test case passes or fails

Solution:

a) What inputs will provide maximum statement coverage for this function?

The function is primarily an if-else statement. For if-else statements, you need to choose inputs that let you run both the (true) and (false) sides of the condition.

Statement	Choose inputs that
if	Cause the condition to be true Cause the condition to be false

Here is the condition:

```java
if (n % 2 == 0) {
```

Pick ANY input that causes the condition to be true
- *N = 20 causes it to be true*

Pick ANY input that causes condition to be false:
- *N = 5 causes it to be false*

Therefore, good inputs are:
- N = 20   (true)
- N = 5   (false)

b) For each input, show the expected output and actual output

Remember - expected output is based on REQUIREMENTS!

```java
1. public void evenOdd(int n) {
2.    if (n % 2 == 0) {
3.          System.out.println("Number is even");
4.    }
5.    else {
6.          System.out.println("Number is odd");
7.    }
8.}
```

For each input, show:
- The expected output
- The actual output
- If test case passes or fails

Line	Test	Input	Expected Output	Actual Output	Pass/Fail
2	Condition is true	n=20	Number is even	Number is even	**PASS**
2	Condition is false	n=5	Number is odd	Number is odd	**PASS**

Example: Code with a Loop

Code:

```
// Prints "hello" n times
 1. public static void printHello(int n) {
 2.    for (int i = 1; i < n; i++) {
 3.       System.out.println("HELLO");
 4.    }
 5. }
```

Question:
   a) What inputs will provide maximum statement coverage for this function?
   b) For each input, show:
       - The expected output
       - The actual output
       - If test case passes or fails

Solution:

a) What inputs will provide maximum statement coverage for this function?

The function is primarily a **for loop**, so:

Statement	Choose inputs that
for	Force the loop to run - 0 times - 1 times - More than 1 time

What inputs make the loop run 0 times? n = 0
What inputs make the loop run 1 times? n = 2
What inputs make the loop run more than 1 time? n = 3

b) For each input, show the expected output and actual output

Requierment: `// Prints "hello" n times`

Line	Test	Input	Expected Output	Actual Output	Pass/Fail
2	Loop 0 times	n=0	nothing	nothing	**PASS**
2	Loop 1 time	n=2	HELLO HELLO	HELLO	**FAIL**
2	Loop more than 1 time	n=3	HELLO HELLO HELLO	HELLO HELLO	**FAIL**

c) What is the total statement coverage?

100% - all lines were run

Example: Code with an if and Loop

Code:

```java
/**
 * Adds all the numbers between 1 and n.
 * @param n  n must be positive
 * @return
 *    the sum of all numbers from 1 to n
 *    -1  if there is an error
 */
1.   public static int sum(int n) {
2.    if (n <= 0) {
3.        return -1;
4.    }
5.
6.    int total = 1;
7.    for (int i = 1; i < n; i++) {
8.        total = total + i;
9.    }
10.    return total;
11. }
```

Question:
    c) What inputs will provide maximum statement coverage for this function?
    d) For each input, show:
- The expected output
- The actual output
- If test case passes or fails

Solution:

a) What inputs will provide maximum statement coverage for this function?

In this function, there is an:
- If statement
- For loop

Choose inputs that will test both.

Statement	Choose inputs that
if	Cause the condition to be `true` Cause the condition to be `false`
for	Force the loop to run ● 0 times ● 1 times ● More than 1 time

What inputs make the if-statement true?   n = -50
What inputs make the if-statement false?   n = 1

What inputs make the loop run 0 times? n = 1
      Why didn't I choose n = 0?
What inputs make the loop run 1 times? n = 2
What inputs make the loop run more than 1 time? n = 3

b) For each input, show the expected output and actual output

Line	Test	Input (n)	Expected Output	Actual Output	Pass/Fail
1	Condition true	-50	-1	-1	**PASS**
1	Condition false	1	1	1	**PASS**
7	Loop 0 times	1	1	1	**PASS**
7	Loop 1 time	2	3	2	**FAIL**
7	Loop more than 1 time	3	6	4	**FAIL**

c) What is the total statement coverage?

100% - all lines were run!