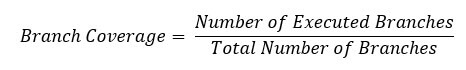
In the branch coverage, every outcome from a code module is tested. For example, if the outcomes are binary, you need to test both True and False outcomes.

It helps you to ensure that every possible branch from each decision condition is executed at least a single time.

By using Branch coverage method, you can also measure the fraction of independent code segments. It also helps you to find out which is sections of code don't have any branches.

The formula to calculate Branch Coverage:

[](https://www.guru99.com/images/1/102518_1122_CodeCoverag13.jpg)

### Example of Branch Coverage

To learn branch coverage, let's consider the same example used earlier

Consider the following code

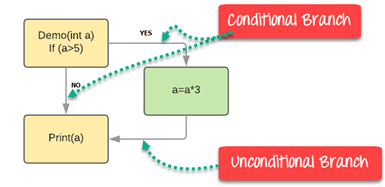
Demo(int a) {

If (a> 5)

a=a\*3

Print (a)

}



Branch Coverage will consider unconditional branch as well

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Value of A | Output | Decision Coverage | Branch Coverage |
| 1 | 2 | 2 | 50% | **33%** |
| 2 | 6 | 18 | 50% | **67%** |

**Advantages of Branch coverage:**

Branch coverage Testing offers the following advantages:

* Allows you to validate-all the branches in the code
* Helps you to ensure that no branched lead to any abnormality of the program's operation
* Branch coverage method removes issues which happen because of statement coverage testing
* Allows you to find those areas which are not tested by other testing methods
* It allows you to find a quantitative measure of code coverage
* Branch coverage ignores branches inside the Boolean expressions