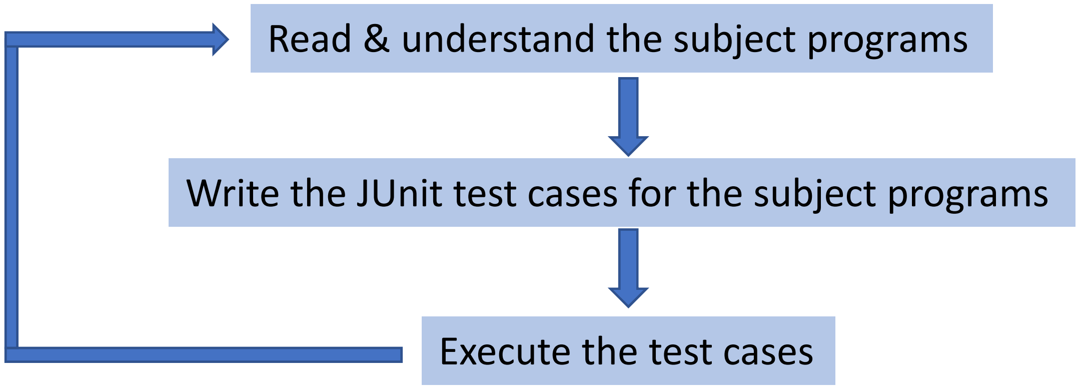
Tutorial for JUnit & Software Testing Contest

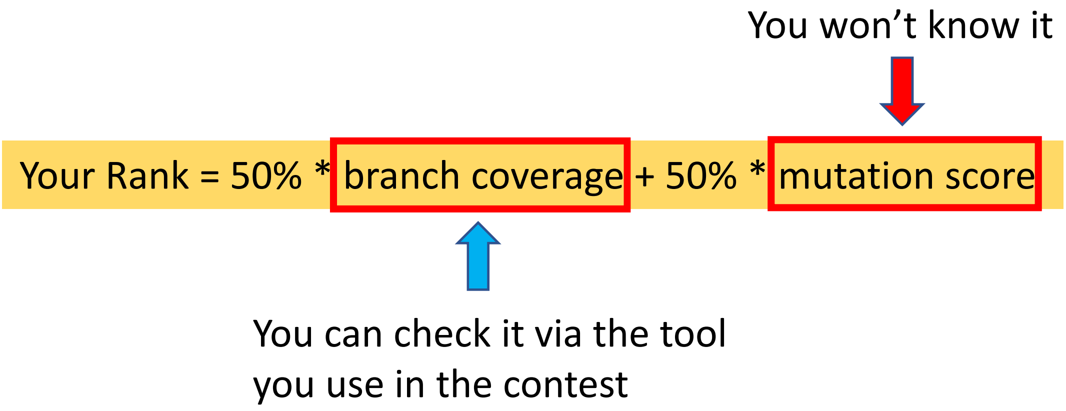
What Do You Need to Do In the Contest?



Read & understand the subject programs

Write the JUnit test cases for the subject programs

Execute the test cases

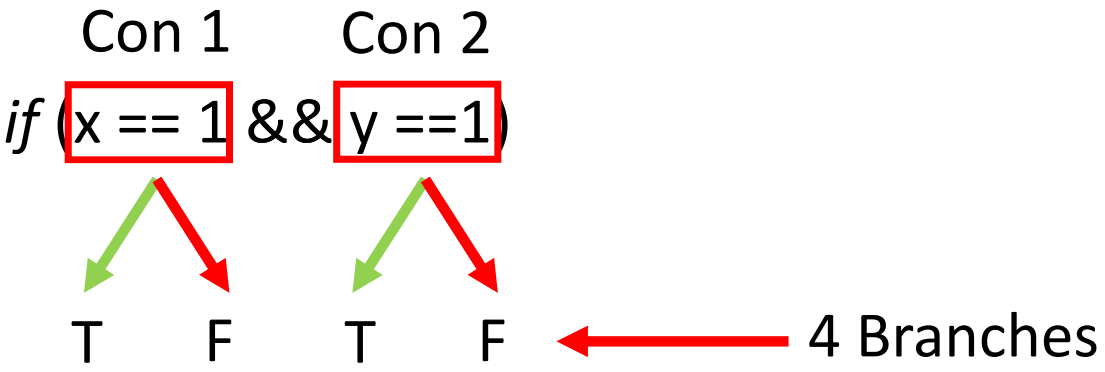


You won’t know it

Your Rank = 50% * branch coverage + 50% * mutation score

You can check it via the tool
you use in the contest

What is Branch Coverage



Con 1 Con 2

if (x == 1 && y ==1)

T F T F

4 Branches

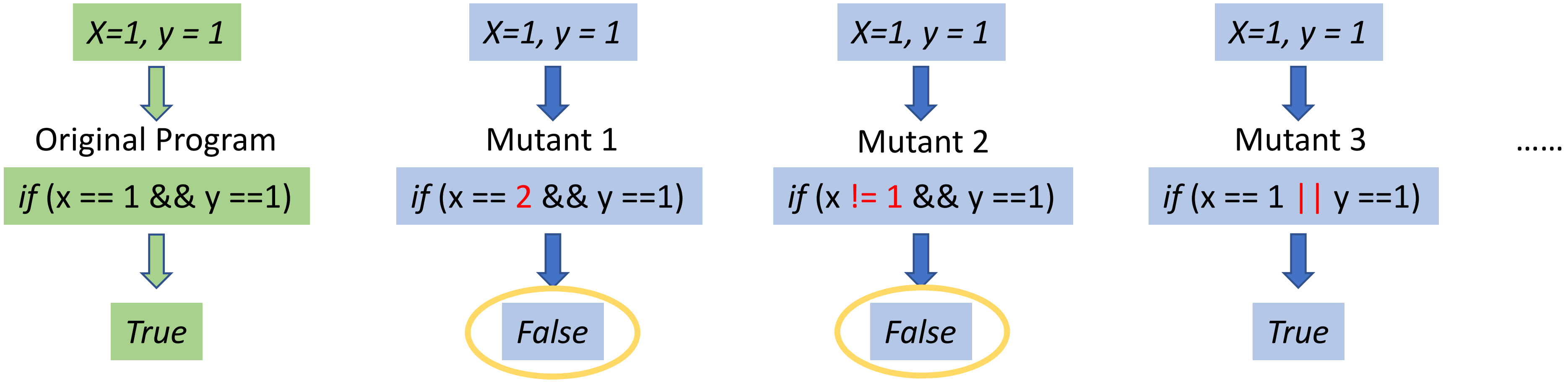
$$B_{cov} = \frac{no.\ executed\ \ branches}{no.\ total\ \ branches}$$

x=1, y=1 → 2/4 = 50%

x=1, y=0 → 2/4 = 50%

x=0, y=0 → 1/4 = 25%

What is Mutation Score



X=1, y = 1

Original Program

if (x == 1 && y ==1)

True

X=1, y = 1

Mutant 1

if (x == 2 && y ==1)

False

X=1, y = 1

Mutant 2

if (x != 1 && y ==1)

False

X=1, y = 1

Mutant 3

if (x == 1 || y ==1)

True

……

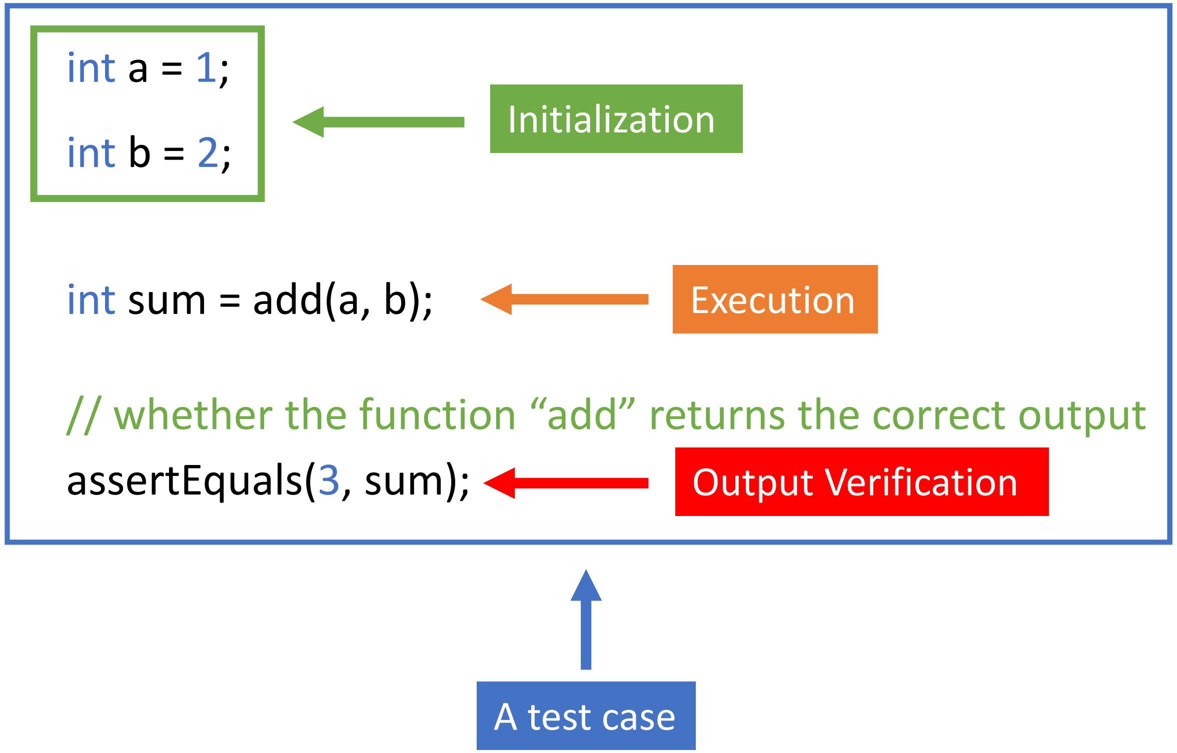
If the output of a mutant is different from that of the original program, we say that mutant has been killed

$$M_{score} = \frac{no.\ killed\ \ mutants}{no.\ total\ \ mutants}$$

Mooctest & JUnit in Our Contest

- Each contestant needs to use JUnit to write test cases
- Mooctest will execute these submitted JUnit test cases to evaluate the accumulative branch coverage and mutation score

How to write a JUnit test case



```
int a = 1;

int b = 2;


int sum = add(a, b);


// whether the function "add" returns the correct output
assertEquals(3, sum);
```

Initialization

Execution

Output Verification

A test case

Common Assertion Methods

- assertArrayEquals()
- assertEquals()
- assertTrue() & assertFalse()
- assertNull() & assertNotNull()
- assertSame() & assertNotSame()

Be Careful

- Make sure your JUnit code can pass all the assertions. Otherwise your branch coverage and mutation score will not be updated

- Make sure you submit your code to the Mooctest

assertArrayEquals(expected, actual)

Test whether two arrays are equal to each other.

It compares based on the order, if mismatch in order results in failure.

```
String[] str1 = {"apple", "mango", "grape"};
String[] str2 = {"apple", "mango", "grape"};
String[] str3 = {“banana", "mango", "grape"};
assertArrayEquals(str1, str2); //will pass the test
assertArrayEquals(str1, str3); //will fail the test
```

assertEquals(expected, actual)

Checks that two objects are equal.

```
String str1 = new String ("abc");

String str2 = new String ("abc");

String str3 = new String ("cba");

assertEquals(str1, str2); //will pass the test

assertEquals(str1, str3); //will fail the test
```

assertTrue() & assertFalse()

Test a single variable to see if its value is either true, or false.

```
int val1 = 5;

int val2 = 6;

assertTrue (val1 < val2); //will pass the test

assertFalse(val1 > val2); //will pass the test
```

assertNull() & assertNotNull()

Test a single variable to see if it is null or not null.

```
String str1 = new String ("abc");

String str2 = null;

assertNotNull(str1); //will pass the test

assertNull(str2);      //will pass the test
```

assertSame() & assertNotSame()

Test if two object references point to the same object or not.

```
String str1 = new String ("abc");
String str2 = new String ("abc");
String str3 = "abc";
String str4 = "abc";
assertSame(str3, str4);       //will pass the test
assertNotSame(str1, str2);  //will pass the test
```