# 18.FactorialCalculationProgram(White-Box Testing)

Aim:

TowriteaJavaprogramtocalculatethefactorialofanumberandverifytheoutputusing white-box testing with JUnit.

Algorithm:

1. **Step1:**Accepttheintegernumber(n)as input.
2. **Step2:**Checkifthenumberislessthan0.Ifso,returnanerrormessagesince factorials are only defined for non-negative integers.
3. **Step 3:** Ifthenumberis 0,return1, asthefactorial of0is definedtobe1.
4. **Step4:**Foranypositiveintegern,initializeavariableresultto1,andmultiplyresult by all numbers from 1 to n (inclusive).
5. **Step5:**Returnthecalculated result.
6. **Step6:**WriteJUnittestcasestovalidatethefactorialcalculationlogic,including edge cases like 0, 1, and negative numbers.

**Code:**

public class FactorialCalculator {

public long factorial(int number) {

if (number < 0) {

throw new IllegalArgumentException("Factorial is not defined for negative numbers.");

}

long result = 1;

for (int i = 2; i <= number; i++) {

result \*= i;

}

return result;

}

}

**Step 2:**

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class FactorialCalculatorTest {

private FactorialCalculator calculator;

@BeforeEach

public void setUp() {

calculator = new FactorialCalculator();

}

@Test

public void testFactorialOf5() {

assertEquals(120, calculator.factorial(5)); // Test Case 1

}

@Test

public void testFactorialOf3() {

assertEquals(6, calculator.factorial(3)); // Test Case 2

}

@Test

public void testFactorialOf0() {

assertEquals(1, calculator.factorial(0)); // Test Case 3

}

@Test

public void testFactorialOf1() {

assertEquals(1, calculator.factorial(1)); // Test Case 4

}

@Test

public void testFactorialOfNegativeNumber() {

IllegalArgumentException thrown = assertThrows(

IllegalArgumentException.class,

() -> calculator.factorial(-5) // Test Case 5

);

assertEquals("Factorial is not defined for negative numbers.", thrown.getMessage());

}

}

Sample Input:

# TestCase1:

* + - Input:5
    - ExpectedOutput:
      * 120(since5!=5 ×4 ×3×2 ×1 =120)

# TestCase2:

* + - Input:3
    - ExpectedOutput:
      * 6(since3!=3 ×2 ×1=6)

# TestCase3:

* + - Input:0
    - ExpectedOutput:
      * 1(since0!=1 bydefinition)

# TestCase4:

* + - Input:1
    - ExpectedOutput:
      * 1(since1!=1)

# TestCase5:

* + - Input:-5(Invalidinput)
    - ExpectedOutput:
      * ThrowsIllegalArgumentExceptionwiththemessage:"Factorialisnot defined for negative numbers."

SampleOutput:

# TestCase1:

* + - Input:5
    - Output:120

# TestCase2:

* + - Input:3
    - Output:6

# TestCase3:

* + - Input:0
    - Output:1

# TestCase4:

* + - Input:1
    - Output:1

# TestCase5:

* + - Input:-5
    - Output: ThrowsIllegalArgumentException

Results:

* + **TestCase1:**Theprogram correctlycalculatesthefactorial of 5,outputting120.
  + **TestCase2:**Theprogram correctlycalculatesthefactorial of 3,outputting6.
  + **TestCase3:**Theprogram correctlyhandlesthe edgecaseof0, outputting 1.
  + **TestCase4:**Theprogram correctlycalculatesthefactorial of 1,outputting1.
  + **TestCase5:**Theprogramcorrectlyhandlesnegativeinput,throwingan IllegalArgumentException as expected.



