# Exercise 3: Assertions in JUnit

## Objective

To explore and use various types of assertions in JUnit for validating test outcomes.

## Common JUnit Assertions

- assertEquals(expected, actual): Verifies that two values are equal  
- assertNotEquals(expected, actual): Checks that values are not equal  
- assertTrue(condition): Validates that the condition is true  
- assertFalse(condition): Validates that the condition is false  
- assertNull(object): Asserts that the object is null  
- assertNotNull(object): Asserts that the object is not null  
- assertThrows(expectedException.class, () -> code): Verifies that an exception is thrown

## Step-by-Step Instructions

## Step 1: Create a Simple Utility Class

// MathUtils.java  
public class MathUtils {  
 public int divide(int a, int b) {  
 return a / b;  
 }  
  
 public int add(int a, int b) {  
 return a + b;  
 }  
}

## Step 2: Write a Test Class Using Assertions

// MathUtilsTest.java  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class MathUtilsTest {  
  
 MathUtils math = new MathUtils();  
  
 @Test  
 void testAdd() {  
 assertEquals(4, math.add(2, 2));  
 assertNotEquals(5, math.add(2, 2));  
 }  
  
 @Test  
 void testDivide() {  
 assertThrows(ArithmeticException.class, () -> math.divide(1, 0));  
 assertEquals(2, math.divide(4, 2));  
 }  
  
 @Test  
 void testNullCheck() {  
 String str = null;  
 assertNull(str);  
 str = "JUnit";  
 assertNotNull(str);  
 }  
}

## Expected Output

All assertions should pass, and test results should show 0 failures.

## Notes

- Assertions make your tests readable and reliable.  
- Use descriptive messages if needed: assertEquals(4, result, "Expected sum is 4");