**Exercise 1: Parameterized Tests**

**Steps:**

1. Create a Java class named EvenChecker with a method isEven(int number).

public class EvenChecker {

public boolean isEven(int number) {

return number % 2 == 0;

}

}

1. Write a parameterized test class EvenCheckerTest using JUnit 5’s @ParameterizedTest and @ValueSource.

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.ValueSource;

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

public class EvenCheckerTest {

private EvenChecker checker = new EvenChecker();

@ParameterizedTest

@ValueSource(ints = {2, 4, 6, 8, 10})

public void testIsEven(int number) {

assertTrue(checker.isEven(number));

}

}

**Exercise 2: Test Suites and Categories**

**Steps:**

1. Create a test suite class AllTests.
2. Add multiple test classes to the suite using JUnit 5.

Example using JUnit 5 @Suite:

import org.junit.platform.suite.api.SelectClasses;

import org.junit.platform.suite.api.Suite;

@Suite

@SelectClasses({CalculatorTest.class, AssertionsTest.class})

public class AllTests {

}

**Exercise 3: Test Execution Order**

**Steps:**

1. Create a test class OrderedTests.
2. Use @TestMethodOrder and @Order annotations.

Example:

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.MethodOrderer.OrderAnnotation;

import org.junit.jupiter.api.TestMethodOrder;

import org.junit.jupiter.api.Order;

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

@TestMethodOrder(OrderAnnotation.class)

public class OrderedTests {

@Test

@Order(2)

public void testSecond() {

assertTrue(true);

}

@Test

@Order(1)

public void testFirst() {

assertTrue(true);

}

}

**Exercise 4: Exception Testing**

**Steps:**

1. Create a class ExceptionThrower with a method throwException.

public class ExceptionThrower {

public void throwException() {

throw new IllegalArgumentException("Invalid input");

}

}

1. Write a test class ExceptionThrowerTest.

import org.junit.jupiter.api.Test;

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

public class ExceptionThrowerTest {

@Test

public void testException() {

ExceptionThrower et = new ExceptionThrower();

assertThrows(IllegalArgumentException.class, et::throwException);

}

}

**Exercise 5: Timeout and Performance Testing**

**Steps:**

1. Create a class PerformanceTester with a method performTask.

public class PerformanceTester {

public void performTask() throws InterruptedException {

Thread.sleep(500);

}

}

1. Write a test class PerformanceTesterTest to check the method finishes within 1 second.

import org.junit.jupiter.api.Test;

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

import java.time.Duration;

public class PerformanceTesterTest {

@Test

public void testPerformance() {

PerformanceTester tester = new PerformanceTester();

assertTimeout(Duration.ofSeconds(1), () -> tester.performTask());

}

}