**Lab Exercise 1- Test-Driven Development (TDD) Java with GitHub Copilot & IntelliJ IDEA**

This Exercise will walk you through the **Test-Driven Development (TDD) process in Java** using **GitHub Copilot** in **IntelliJ IDEA**. It includes **Copilot prompts** at each stage to help generate code.

**Prerequisites**

Before starting, ensure you have the following installed:

1. **IntelliJ IDEA (Community or Ultimate)** – [Download Here](https://www.jetbrains.com/idea/download/)
2. **Java JDK (11 or later)** – [Install Guide](https://adoptopenjdk.net/)
3. **Maven (or Gradle)** – [Install Guide](https://maven.apache.org/install.html)
4. **JUnit 5** – Required for writing test cases
5. **Git** – [Download Git](https://git-scm.com/)
6. **GitHub Copilot** – Follow the official [GitHub Copilot Setup Guide](https://github.com/features/copilot)
7. **GitHub Account** – Create an account at [GitHub](https://github.com/)

**Step 1: Set Up a Java Project in IntelliJ IDEA**

1. Open **IntelliJ IDEA**.
2. Click **New Project**.
3. Select **Java** and click **Next**.
4. Choose **Maven** as the build system.
5. Set **GroupId** to com.example and **ArtifactId** to TDD\_Calculator.
6. Choose a project location and click **Finish**.
7. Open the **Terminal** (View → Tool Windows → Terminal) and initialize Git:

git init

**Step 2: Install and Enable GitHub Copilot**

1. Go to **File** → **Settings** → **Plugins**.
2. Search for **GitHub Copilot**, install it, and restart IntelliJ IDEA.
3. Sign in to GitHub when prompted.
4. Enable Copilot in **Settings** → **GitHub Copilot**.

**Step 3: Write the First Test (Red Phase)**

**Create a Test File**

1. In **Project Explorer**, navigate to src/test/java/com/example/.
2. Right-click and select **New → Java Class**. Name it CalculatorTest.java.
3. Start typing the following **GitHub Copilot prompt as a comment**:

// Write a JUnit test for a Calculator class that tests an addition method

1. **Press Enter** and let Copilot suggest the test case:

package com.example;

import org.junit.jupiter.api.Test;

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

class CalculatorTest {

@Test

void testAddition() {

Calculator calculator = new Calculator();

int result = calculator.add(2, 3);

assertEquals(5, result, "2 + 3 should equal 5");

}

}

1. **Run the test** (It should fail since Calculator does not exist).

**Step 4: Implement the Code (Green Phase)**

**Create the Calculator Class**

1. In **Project Explorer**, navigate to src/main/java/com/example/.
2. Right-click and select **New → Java Class**. Name it Calculator.java.
3. Start typing this **Copilot prompt as a comment**:

// Implement a Calculator class with an add method

1. **Press Enter** and let Copilot generate:

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

1. **Run the test again** (It should now pass).

**Step 5: Refactor and Add More Tests**

* **Write additional test cases with Copilot prompts**:

// Write JUnit tests for subtraction, multiplication, and division methods in Calculator

* Let Copilot generate the following in CalculatorTest.java:

@Test

void testSubtraction() {

Calculator calculator = new Calculator();

int result = calculator.subtract(5, 3);

assertEquals(2, result, "5 - 3 should equal 2");

}

@Test

void testMultiplication() {

Calculator calculator = new Calculator();

int result = calculator.multiply(4, 3);

assertEquals(12, result, "4 \* 3 should equal 12");

}

@Test

void testDivision() {

Calculator calculator = new Calculator();

int result = calculator.divide(10, 2);

assertEquals(5, result, "10 / 2 should equal 5");

}

* **Run the tests** (They should fail since the methods don’t exist yet).

**Step 6: Implement More Methods**

1. Open Calculator.java.
2. Use **Copilot prompts** to generate the missing methods:

// Implement subtraction, multiplication, and division methods in Calculator

* Let Copilot generate:

public int subtract(int a, int b) {

return a - b;

}

public int multiply(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) throw new ArithmeticException("Cannot divide by zero");

return a / b;

}

1. **Run all tests** (They should now pass).

**Step 7: Commit and Push to GitHub**

1. Create a GitHub repository at [GitHub](https://github.com/).
2. In IntelliJ IDEA, open the terminal and add the remote repository:

git remote add origin <your-repo-url>

git branch -M main

1. Commit and push your code:

git add .

git commit -m "Implemented Calculator with basic operations"

git push -u origin main