Task 1: Create a Simple TestNG Class with Basic Annotations

Topic: Annotation and Execution with TestNG

Description: Create a basic TestNG class that demonstrates the use of fundamental annotations. This class will include two test methods: one that passes and one that intentionally fails to illustrate the assertion mechanism.

Steps:

Open your IDE and create a new Java class within your test project.

Import the TestNG library and set up the class with the @Test annotation.

Write the first test method testPass() that asserts a true condition, like Assert.assertTrue(1 == 1);.

Write the second test method testFail() that asserts a false condition, like Assert.assertTrue(1 == 2);.

Annotate both methods with @Test and give them meaningful names, such as shouldPass and shouldFail.

Execute the TestNG class using the IDE's TestNG plugin or via command line, and observe the output.

Briefly describe what the @BeforeMethod and @AfterMethod annotations do, and how they can be utilized in setting up and tearing down test conditions.

```
package Day4;


import org.junit.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;


public class Task1 {
   @BeforeMethod
   public void beforeMethod() {
         System.out.println("Before method is Executed");
   }
   @Test
   public void testPass() {
         System.out.println("Test is Passed");
         Assert.assertTrue(1==1);
   }
   @Test
   public void testFail() {
         System.out.println("Test is Failed");
         Assert.assertTrue(1==2);
```

```
    }
    @AfterMethod
    public void afterMethod() {
        System.out.println("After Method is Executed");
    }
}
```