20. Demonstrate the TDD with TestNG

Step 1: Performing a TDD test:

- To perform a TDD test, follow the below steps:
    a. Add the test.
    b. Execute the test and see if the new one fails.
    c. Write the code.
    d. Execute the test.
    e. Refactor the code.
    f. Now, repeat the steps mentioned above.

Now, let's look at the above steps in detail:

a. Firstly, write the code that will be based on the requirements in Eclipse. It should look something like:

```
package test.testing;

import org.testng.Assert;
import org.testng.annotations.Test;

public class AddNumbers {

        @Test

        public void addIntegerNumbers()
        {
            Calculator myCalculator = new Calculator();
            int expected= 30;
            int actual= myCalculator.add(10,20);
            Assert.assertEquals(actual, expected);
        }
}
```

b. Now, if we execute our test for the first time, we should get the below error:
FAILED: addIntegerNumbers
java.lang.Error: Unresolved compilation problems:
Calculator cannot be resolved to a type
Calculator cannot be resolved to a type

c. Write the code shown below to resolve the above error in Eclipse. It will look something like:

```
package test.testing;

public class Calculator {

    public int add(int number1, int number2)
    {
            return 0;
    }
}
```

d. Now, execute our test:
FAILED: addIntegerNumbers
java.lang.AssertionError: expected [30] but found [0]

e. Refactor the code in Eclipse. It will look like:

```
package test.testing;

public class Calculator {

    public int add(int number1, int number2)
    {
            return (number1+number2);
    }
}
```

f. Now, if execute our test again, it will show the below message:
PASSED: addIntegerNumbers

Step 2: Running the code:

- Run the code through Eclipse.

Step 3: Pushing the code to your GitHub repositories:

Open your command prompt and navigate to the folder where you have created your files:

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master