**JUNIT Testing**

**JUNIT5 Framework use it**

**Unit Testing Framework (JUNIT5) ?**

**Steps :**

* Prepare (setup a Test Environment, write test method)
* Provide testing input
* Run the test
* Provide expected output
* Perform assertion (verify the result)
* Report test results (alert developer if test is failed or passed)

**JUNIT5 Architecture :**

JUNIT5 is not JUNIT4 \_ new features :

API

Jupitor

Vintage

Platform

**Create the Maven Project :**

First go to file 🡪 others (ctrl + N) 🡪Inside Maven select Maven Project 🡪 Create a simple project 🡪 Then create a new maven project 🡪 write the group id , Artifact id & last finish.

Then add the dependencies in our project for that purpose go through the <https://mvnrepository.com/artifact/junit/junit> this url 🡪 then click the testing framework & tool 🡪 then select the Junit 🡪 & select the any version 🡪 then select the maven repository & add our pom.xml file.

<dependencies>

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

----------------------------------------------------------------------------------------------------

Create the class write the package & class name. then write method –

Code :

**package** com.kiran.learning;

**public** **class** Calc {

**public** **int** divide(**int** num1, **int** num2)

{

**return** num1/num2;

}

}

Then write the test class for that purpose select the current class file right click 🡪 New 🡪 select Junit Test case 🡪 write test.java file.

**Code :**

package com.kiran.learning;

import static org.junit.Assert.\*;

import org.junit.Test;

public class CalcTest {

@Test

public void test() {

// System.out.println("This is my first test case example..");

//assertEquals("Kiran","Kiran");

Calc c = new Calc();

int r=c.divide(10, 5);

int expectedResult=2;

assertEquals(expectedResult,r);

System.out.println("Successfully Test the code..");

}

}

----------------------------------------------------------------------------------------------------

**Create the own project folder :**

First create the java project

Then create the folder select main folder then right click here , go to build path , new source folder. Ex : create the test folder.

**Code :**

**package** com.kiran;

**public** **class** ReverseString

{

**public** String reverseString(String str) {

**char**[] charArray = str.toCharArray();

**int** left =0;

**int** right = charArray.length -1;

**while**(left < right) {

//Swap characters at the left and right positions

**char** temp = charArray[left];

charArray[left] = charArray[right];

charArray[right] = temp;

// Move the pointers towards each other

left++;

right--;

}

**return** **new** String(charArray);

}

}

**Code for testcase :**

package com.kiran;

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

import org.junit.jupiter.api.Test;

class ReverseStringTest {

@Test

void test() {

ReverseString reverse=new ReverseString();

String actual = reverse.reverseString("Java");

String expected = "avaJ";

assertEquals(expected,actual);

}

}

**Note :** Shot the code line then right click 🡪 go through the refactor select inline option.

**Like that :**

@Test

**void** test() {

ReverseString reverse=**new** ReverseString();

*assertEquals*("avaJ",reverse.reverseString("Java"));

}

----------------------------------------------------------------------------------------------------

**@Test :**

* Applied over methods to mark method as test
* org.junit.jupiter.api
* visibility of @Test annotated method can be public, protected, default.
* Also inform test engine what method needs to run.

**void test() :**

* Check only failure , by default result set as success

**Assertion :**

* What is expectation & Reality match if match condition then result success otherwise result is failed.
* Static methods 🡪 eg :- assertEquals(expect,actual);

assertArrayEquals(expect, actual);

* org.junit.jupiter.Assertions - class
* It support fetures of Java 8(Lambda exp , StreamAPI)

**assertArrayEquals() method :**

* Actual and expected arrays are equal.
* Numbers of elements should equal.
* Elements of an array are equal.
* Order of element in an array.