JUnit Assertion Statements

What is JUint assert ?

- Assert is a class offered by Junit that contains a number of assertion methods helpful for creating test cases and identifying test failure.
- Assert methods are used to determine whether the test case passed or failed.
- It is given by the class org.junit.Assert, which extends the java.lang.Object class.

Some examples of assert statements:

- assertEquals(Expected , actual) helps to check the equality of two primitives/objects are equal.

  assertEquals(“hello” , “hello”) 

  assertEquals(“hello” , ”hi”) 

- assertEquals(Expected, actual, 0.001): the delta parameter here helps to prevent issues with round-off errors when performing floating point comparisons.

Some examples of assert statements:

- assertArrayEquals(expected, actual) helps to check if the array is equal or not.

  Int[] arr1 = {1, 2, 3} Int[] arr2 = {4 ,5 , 6}

  assertArrayEquals(arr1, arr1) 

  assertArrayEquals(arr1, arr2) 

- You can also compare the values at an index:

  assertEquals(expected[i], actual[i])

Some examples of assert statements:

- assertTrue(condition) helps to check the if the condition is true.

  Eg: int a = 10 and b = 12

  assertTrue(a < b) ✔

  assertTrue(b < a) ✘

- assertFalse(condition) helps to check if the condition is false.

  Eg: int a =10 and b =12

  assertFalse(a < b) ✘

  assertFalse(b < a) ✔

Some examples of assert statements:

- assertNull(Object)

  It helps to check if the object is null.

- assertNotNull(Object)

  It helps to check if the object is not null.

- assertSame(Expected,Actual)

  It helps to check if the two objects are referring to same the object.

- assertNotsame(Expected,Actual)

  It helps to check if the two objects are not referring to the same object.