

JUnit is used to test the individual components separately.

* **@Test** annotation helps to identify that the method is a test case and it should be executed separately as a testcase, when a java program is executed.
* Ecllipse has a library with name JUnit which can be used to add to the project for JUnit Testing.
* In eclipse we have a concept called **Content Assist,** this is used while programming code in eclipse.

To provide this details, Users should go to **Preferences🡪 Java🡪 Editor🡪Content Assist🡪Favorites.** Here we can add the library of JUnit. (i.e., org.junit.Assert)

* **TestCase** is the class used in JUnit for assert value testing. Ex:

TestCase.assertEquals("The sum correct", 6, sum);

* Without **@Test** annotation, the method written does not have any significance and will be executed as Test Case.
* No naming conventions are required for test case methods.
* **@Test** annotation also provides a facility of throwing specific type of Exception.

**Ex:** @Test(expected=ArithmeticException.class)

* **@Ignore** annotation will ignore the test case to get executed.
* **JUnit** test cases can be used to run **multiple classes**, to execute them we need to Runner instance called **JUnitCore.**
* Here we can use a class **Computer** to execute the classes sequentially and ParallelComputer for parallel.
* Test case methods will return **void** and it should be **public**.
* JUnit has assertions like **assertTrue** or **assertFalse** or **fail** to check whether our environment is correct or not, before we start actual testing.
* JUnit has assertion with numbers with **assertEquals, assertNotEquals** for comparing float or double we need to provide the delta value.
* To make the order of execution of test cases in JUnit, we can use an annotation **@FixMethodOrder** on the top of the class with values as **MethodSorters.DEFAULT, MethodDorters.JVM, MethodSorters.NAME\_ASCENDING.**
* Using **assertEquals** we can compare the value of the objects also, during which the objects equals method will be used.
* Using **assertSame**, we can compare whether two objects are same, i.e., having same reference.