Unit is a testing framework for Java that helps you automate the testing of your code â€” making sure it behaves correctly and doesn't break when you change something.

You write small test methods to test your code automatically, instead of testing manually with main().

Method	Description
assertEquals(expected, actual)	Passes if expected equals actual
assertNotEquals(expected, actual)	Passes if expected does not equal actual
assertTrue(condition)	Passes if the condition is true
assertFalse(condition)	Passes if the condition is false
assertNull(object)	Passes if the object is null
assertNotNull(object)	Passes if the object is not null
assertSame(expected, actual)	Passes if both refer to the same object reference (==)
assertNotSame(expected, actual)	Passes if both refer to different object references
fail("message")	Fails the test immediately with a message (useful in manual testing blocks)
assertArrayEquals(expected, actual)	Passes if two arrays are equal (deep equality check)
assertThrows(Class<T>, Executable)	âœ… JUnit 5 only: Passes if the given exception is thrown
assertDoesNotThrow(Executable)	âœ… JUnit 5: Passes if no exception is thrown
assertAll(...)	âœ… JUnit 5: Groups multiple assertions; fails if any fail
assertThat(actual, matcher)	âœ… JUnit 4 & 5: Uses Hamcrest matchers for fluent/assertive checks

Annotations:

Annotation	Description
@Test	Marks a method as a test method.
@Before	Runs before each test method.
@After	Runs after each test method.
@Ignore	Skips the test method.