Unit Testing

**What is Unit Testing?**

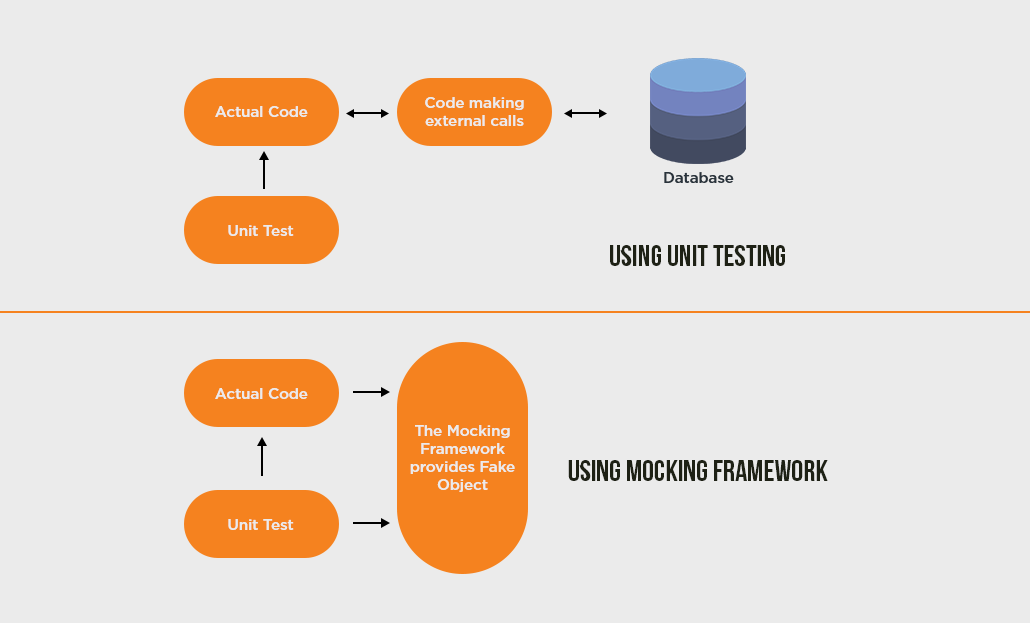
Unit Testing involves testing each and every component of your code to see if they work as expected. It isolates each individual method of your code and performs tests on it. Unit tests help make sure that your software is working as expected before releasing it.

**JUnit Test Cases:**

* @Test : Signify a test method
* @BeforeAll:
* @AfterAll :
* @BeforeEach : Run before each test method
* @AfterEach : Run after each test method
* assertEquals() : void assertEquals(boolean expected, boolean actual) : Checks that two primitives/objects are equal
* assertFalse() : void assertFalse(boolean condition) : Checks that a condition is false.
* assertTrue() : void assertTrue(boolean condition) : Checks that a condition is true.
* assertNull() : Checks that an object is null.
* assertNotNull(): Checks that an object is null.

**Mocking**

* when().thenReturn()
* doReturn().when().method\_name()
* verify().method\_name()
* @Mock : A mock is nullified class instance that does not play a role in a test
* Stub : A stub is a mock with additional instructions for how to behave when certain methods are called
* Spy : A spy is just like a regular class instance with the ability to also stub method calls



* Mocking is way to test the functionality of a class in isolation
* Mocking doesn’t require a database connection or properties file read or file server read to test a functionality
* Mock Objects do the mocking of the real service
* A mock object returns dummy data corresponding to some dummy input passed to it