**JUNIT**

**INTRODUCTION:**

To do *Unit testing* in *Java* we have an excellent framework called ***Junit***. Junit is a *unit testing framework*. This framework provides us with the following facilities

* Base classes and Annotations to write unit tests
* Base class support to run tests, TestRunner class.
* Base class and Annotation support to write test suites, @RunWith(*Suite.Class*)
* And of course reporting of test results.

**Unit Testing:**

**Unit testing** is a testing methodology where individual units are tested in isolation from other units. This is usually done by **developers**. A unit can be considered as a class or method inside a class which needs to be tested individually. It is also known as **White Box** Testing, as developer is able to see the code for the functionality.

**Junit Annotations:**

* @BeforeClass – Run once before any of the test methods in the class, public static void.
* @AfterClass – Run once after all the tests in the class have been run, public static void.
* @Before – Run before @Test, public void.
* @After – Run after @Test, public void.

**MAVEN**

**INTRODUCTION:**

Maven is a software [project management build tool](https://www.simplilearn.com/tutorials/project-management-tutorial/project-management-tools) based on Project Object Model (POM). The tool is typically used for [Java](https://www.simplilearn.com/best-java-programs-article)-based projects. Every Java project requires certain dependencies, which are automatically downloaded when running a Maven build.

During the process, Maven takes care of the following elements:

* Builds
* Dependencies
* Reports
* Distribution
* Releases
* Mailing list

**Features of Maven:**

Apache Maven manages all the processes, like building, documentation, releasing, and distribution in [project management](https://www.simplilearn.com/project-management-skills-article).

* The tool simplifies the way of project building by increasing the performance of the project and the building process
* The task of downloading dependencies and JAR files is automated
* Maven provides easy access to all necessary information
* Maven makes it simple for the developer to build a project in different environments without worrying about the dependencies, processes, etc.
* In Maven, it is simple to add new dependencies—you only have to write the dependency code in the POM file.