**TestNG (Test Next Generation) – Java-based Unit Testing Framework**

TestNG is a powerful testing framework inspired by JUnit and NUnit but designed to support a wide range of testing needs such as unit, functional, integration, and end-to-end testing.

✅ Advantages of TestNG:

1. Test Cases and Test Suites

* Organizes test cases into test suites using XML configuration files.

1. Grouping of Test Cases

* Test cases can be logically grouped (e.g., *smoke*, *regression*, *sanity*) using the *groups* attribute.
* Enables selective execution of grouped tests.

1. Prioritization of Test Cases

* Controls the order of execution using the *priority* attribute in the *@Test* annotation.

1. Parametrization

* TestNG allows passing parameters at runtime via the XML file using *<parameter>* tag or *@Parameters* annotation.
* Useful for dynamic data like: Browser name, Application URL

1. Parallel Testing

* Supports running tests in parallel on multiple threads or browsers.
* Reduces execution time.
* Configured via XML using *parallel* and *thread-count* attributes at the suite or test level.

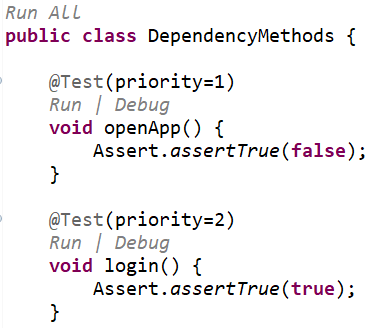
1. Reports

* Generates detailed HTML and XML reports by default.
* Supports integration with third-party reporting tools like: ExtentReports, Allure or ReportNG.
* Provides better visualization, logs, and customization.

**Part II: Dependency Methods**

🔗 <https://github.com/ManelDerbel/Test-Next-Generation--TestNG/blob/master/Udemy-Training/DependencyMethods-Demo.mp4>

🎯 Objective: Skip dependent methods if a test method fails using the *dependsOnMethods* attribute.

 A screenshot of a computer code

AI-generated content may be incorrect.

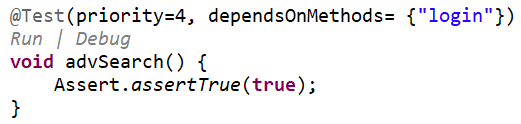
* Result of executing test methods using the XML file: the test method openApp failed, and all the remaining test methods passed.

A black and red text

AI-generated content may be incorrect.

* By using the *dependsOnMethods* attribute in the *@Test* annotation, the login test method was skipped because it depends on the failed openApp test method.

A black and white text

AI-generated content may be incorrect. 

* By using the *dependsOnMethods* attribute in the *@Test* annotation, the search and advSearch test methods were skipped because they depend on the skipped login test method.

A computer code with black text

AI-generated content may be incorrect. A close-up of a white background

AI-generated content may be incorrect.

* Even if the openApp test method passes and the login test method fails, the search and advSearch test methods still skipped because they depend on the failed login test method.

A computer code with text

AI-generated content may be incorrect. A black and white text

AI-generated content may be incorrect.

* Even if the login test method passes, the advSearch test method is still skipped because it depends on the failed search test method.