**TestNG**

**TestNG** is a testing framework inspired from **JUnit** and **NUnit** but introducing some new functionality that make it more powerful and easier to use.

It is an open source automated testing framework; where NG of TestNG means Next Generation. TestNG is similar to JUnit but it is much more powerful than JUnit but still it’s inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes. CedricBeust *who is the creator of* TestNG.

TestNG eliminates most of the limitations of the older framework and gives the developer the ability to write more flexible and powerful tests with help of easy annotations, grouping, sequencing & parametrizing.

Example:

*//This method will provide data to any test method that declares that its Data Provider is named "provider1".*

@DataProvider(name = "provider1")

**public** Object[][] createData1() {

**return** **new** Object[][] {

{ "Cedric", **new** Integer(36) },

{ "Anne", **new** Integer(37) }

};

}

*// This test method declares that its data should be supplied by the Data Provider named "provider1".*

@Test(dataProvider = "provider1")

**public** void verifyData1(String n1, Integer n2) {

System.out.println(n1 + " " + n2);

}

*// A data provider which returns an iterator of parameter arrays.*

@DataProvider(name = "provider2")

**public** Iterator<Object[]> createData() {

**return** **new** MyIterator(...);

}

*// A data provider with an argument of the type java.lang.reflect.Method.*

*// It is particularly useful when several test methods use the same*

*// provider and you want it to return different values depending on*

*// which test method it is serving.*

@DataProvider(name = "provider3")

**public** Object[][] createData(Method m) {

System.out.println(m.getName());

**return** **new** Object[][] { **new** Object[] { "Cedric" } };

}