**Hands On Lab 2**

**Nice Mock Strict Mock**

EasyMock mock objects can have three types – default, strict and nice. We can specify mock object type using MockType [enum](https://www.journaldev.com/716/java-enum) while creating the mock object.

## EasyMock Mock Types

**Default**: Expects only recorded calls but in any order. Test won’t fail if any recorded calls are skipped.

**Strict**: Expects only recorded calls and in the same order they have been recorded.

**Nice**: Expects recorded calls in any order and return default empty values (0, null, false) for unexpected calls.

We can create strict or nice mock object by specifying it’s mock type.

Object mock = mock(MockType.STRICT, Object.class);

Object mock1 = mock(MockType.NICE, Object.class);

If you are using [@Mock annotation](https://www.journaldev.com/22220/easymock-annotations-junit-4), then you can specify the mock type by any of following ways.

@Mock(MockType.STRICT)

Object obj;

@Mock(type=MockType.NICE)

Object obj1;

### ****Instructions:****

* Create a simple Maven project (without archetype).
* Pull the provided source files into the src/main/java folder.
* Overwrite the pom.xml file with what is provided.
* Add a test class called EasyMockNiceMockTest.
* Edit the test class as in the below screenshot.

### 

* Add a test class called EasyMockStrictMockTest.
* Edit the test class as in the below screenshot.



* Run the tests. They should be green.

### ****Conclusion****

In this lab, we saw how to use easy and strict mocks using EasyMock.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*