Mockito

Introduction

Sensitivity: Internal & Restricted

Agenda

1 **Introduction to Mockito**

Objectives

At the end of this module, you will be able to explain:

- What is Mockito

Introduction to Mockito

Mockito

- Mockito is open source
- It is a mocking framework for testing Java Applications using Java – based library
- Mocking implementation of interfaces or classes enables the isolation during unit testing
- Side effects of other classes are controlled by mocking
  - Full service class implementation is not complete
  - DB connection to actual db is not yet up but test needs to be done on other behaviours

Mocking

- Mocking is creation of mock objects during unit testing

- Mock objects are used in situations like
  - To just record the interactions with the systems and validate it
  - To fill dummy parameter list
  - Used to connect with local sample internal database instead of actual database etc

Simple steps to test the code using mockito

- Mock away external dependencies by creating mock objects
- Inserting the mock codes into the test
- Execute the test
- Validate the code

Sample mock object

- Let us consider EmployeeDB class. In order to mock it we can use

```
empDB = mock(EmployeeED.class);
```

- Or by using annotation

```
@Mock
EmployeeDB empDB;
```

- In both the cases we don’t instantiate the object instance, its been mocked
- We can mock the behaviors with When Then Return statement

```
when(empDB.insert(empObj)).thenReturn(1);
when(empDB.insert(deptObj)).thenReturn(0);
```

For Further Reading and full demo visit:

https://www.tutorialspoint.com/mockito/mockito_junit_integration.htm

Summary

- Introduction to Mockito

Thank You