Contents

Advanced Testing Concepts

Test Suite

Comprises of multiple tests

```
import org.junit.runner.RunWith;

import org.junit.runners.Suite;

@RunWith(Suite.class)

@Suite.SuiteClasses({ TestCalAdd.class, TestCalSubtract.class,

TestCalMultiply.class, TestCalDivide.class })

public class CalSuite {

 // the class remains completely empty,

// being used only as a holder for the above annotations

}
```

ReUsing Tests

Allows you to run the same test with different data.
Syntax : **@RunWith(Parameterized.class)**

Testing in Isolation

While performing testing in isolation :

- It needs to be independent of expensive resources.
- Use mock objects to perform testing in isolation.

Mock Objects

- You get the ability to test code that is not yet written.
- They help teams to unit test one part of the code independently.
- They help to write focused tests that will test only a single method.
- They are helpful when the application integrates with expensive external resources.

Few examples of Mock frameworks : *EasyMock, DynaMock, JMock*