

# Week 4 Practice Quiz

## Test Doubles: Introduction

1. If a system under test needs to interact with other systems in order to complete a unit test, just skip the testing.
  - True
  - False
2. What do you call lightweight versions of components that a system under test interacts with that are necessary to do unit testing?
  - Dummy Objects
  - Test Doubles
  - Test Stubs
  - Mock Objects
3. What do you call the dummy values that are filled into objects that are required as parameters by the system i /1 point under test, but are otherwise irrelevant?
  - Dummy Values
  - Dummy Stubs
  - Dummy Mocks
  - Dummy Objects
4. What are the dummy input data sources used by a system under test?
  - Dummy Objects
  - Test Stubs
  - Fake Objects
  - Spy Objects
5. What would you use in order to check the indirect results produced by a system under test?
  - Mock Objects
  - Sky Objects
  - Fake Objects
  - Dummy Objects
6. There is essentially no difference between spy objects and mock objects.
  - True
  - False

*Mock objects are used to check indirect results produced by a system under test whereas spy objects wrap around a real object in order to monitor interactions between the system under test and other objects. While they are 'near' synonyms, they are not quite the same thing.*

7. The goal of Mockito is to:
- Automatically generate flakey tests
  - Create test doubles and inspect interactions
  - Assist in doing unit testing in a vacuum

*The goal is to create test doubles in java in order to mock away dependencies for unit tests and to inspect interactions between the system under test and mocks.*

# Test Doubles: Input

1. When providing test inputs for test doubles, we want to use \_\_\_\_\_ and \_\_\_\_\_.
  - Mock objects; Spy objects
  - Dummy objects; Mock objects
  - Test stubs; Spy objects
  - **Dummy objects; Test stubs**
2. During constructing test doubles, you wish to provide outputs for just a handful of values. What do you use?
  - Dummy Objects
  - Test Doubles
  - **Test Stubs**
  - Mock Objects

*Test stubs would provide outputs for a few values.*

3. During testing, you find that the unit tests require the use of a computationally expensive database. What could you use as a test double?
  - Test Stubs
  - **Fake Objects**

*Correct Fake objects would replace expensive with cheap (for instance, an in memory database)*

4. In Mockito, there is a distinction between the roles that a class plays and how you construct it. The roles of a test double are stub, mock, and dummy (or all three). How you construct them in Mockito is the same.
  - **True**
  - False

*Correct Yes; and that construction is a call to mock(Name.class).*

# Test Doubles: Output

1. \_\_\_\_\_ provide the ecosystem.
  - Test doubles
  - Inputs and outputs
  - Side effects
  - Mock objects
2. Mocks and spies will allow fine grain monitoring of the system under test.
  - True
  - False
3. Which of the following are not a part of testing interactions using mock objects and spy objects?
  - Methods are being called (or not)
  - Required fill-in values are being created for the system under test
  - Proper parameters are being used for method calls
  - Methods are being called in the proper order

*Monitoring interactions includes: methods being called in proper order, methods called or not called, and proper parameters to calls are being used.*

4. \_\_\_\_\_ objects allow us to observe interactions of fake objects. \_\_\_\_\_ objects allow us to observe interactions of real objects.
  - Mock; Dummy
  - Spy; Mock
  - Dummy; Mock
  - Mock; Spy

*Mock objects allow observation of the interaction of fake objects whereas spy objects allow us to observe the interactions of real objects.*

5. In Mockito, there is no capability to override the behavior of methods with a spy object.
  - True
  - False

*You are correct; since you can wrap the method in a spy, you can also override the method's behavior.*

6. In Mockito, it is possible to create an object that is both a stub and a mock.
  - True
  - False