**Test Driven Development in C++**

**Chapter 6: Test Doubles Quiz**

1. Which of the following is not a type of test double?

A. Simpleton

B. Dummy

C. Fake

D. Mock

2. A dummy provides a functional (and usually simplified) implementation which may not be appropriate for production but works well for the unit testing.

A. True

B. False

3. A stub test double expects to be called and returns canned data.

A. True

B. False

4. This type of test double does not expect to be called and may actually throw an exception if it is called.

A. Fake

B. Spy

C. Dummy

D. Mock

5. You should always implement your own test doubles and avoid using mock frameworks.

A. True

B. False

**ANSWER KEY**

1. Which of the following is not a type of test double?

**A. Simpleton**

* Correct! The set of test doubles are: Dummy, Fake, Stub, Spies, and Mocks.

B. Dummy

* Incorrect. The set of test doubles are: Dummy, Fake, Stub, Spies, and Mocks

C. Fake

* Incorrect. The set of test doubles are: Dummy, Fake, Stub, Spies, and Mocks

D. Mock

* Incorrect. The set of test doubles are: Dummy, Fake, Stub, Spies, and Mocks

2. A dummy provides a functional (and usually simplified) implementation which may not be appropriate for production but works well for the unit testing.

A. True

* Incorrect. The Fake test double type actually implements an alternative functional implementation.

**B.** **False**

* Correct! The Fake test double type actually implements an alternative functional implementation.

3. A stub test double expects to be called and returns canned data.

**A.** **True**

* Correct! Stubs expect to be called and return canned data.

B. False

* Incorrect. Stubs expect to be called and return canned data.

4. This type of test double does not expect to be called and may actually throw an exception if it is called.

A. Fake

* Incorrect. A fake provides an actual implementation but perhaps one not suitable for production.

B. Spy

* Incorrect. A spy does expect to be called and saves the values of parameters that were passed in so they can be validated by the test.

**C. Dummy**

* Correct! Dummy test doubles rarely expect to actually be called and will often throw an exception if they are.

D. Mock

* Incorrect. Mock object are sophisticated objects that can be configured to fail if a function is called the wrong number of times or passes in the wrong data.

5. You should always implement your own test doubles and avoid using mock frameworks.

A. True

* Incorrect. Mock object frameworks can be very full featured and provide a lot of utilities for creating mock objects and setting sophisticated expectations.

**B. False**

* Correct. Creating your own mock objects is tedious and error prone.