**TEST DOUBLES**

TEST STUB: Consider a computer program that queries a database to obtain the sum price total of all products stored in the database. In this example, the query is slow and consumes a large number of system resources. This reduces the number of test runs per day. Secondly, tests may need include values outside those currently in the database. The method (or call) used to perform this is *get\_total()*. For testing purposes, the source code in *get\_total()* can be temporarily replaced with a simple statement that returns a specific value. This would be a test stub.

MOCK: In a unit test, mock objects can simulate the behavior of complex, real objects and are therefore useful when a real object is impractical or impossible to incorporate into a unit test. If an actual object has any of the following characteristics, it may be useful to use a mock object in its place.

If the real object:

* supplies non-deterministic results (e.g., the current time or the current temperature);
* has states that are difficult to create or reproduce (e.g., a network error);
* is slow (e.g., a complete database, which would have to be initialized before the test);
* does not yet exist or may change behavior;
* would have to include information and methods exclusively for testing purposes (and not for its actual task).

For example, an alarm clock program which causes a bell to ring at a certain time might get the current time from the outside world. To test this, the test must wait until the alarm time to know whether it has rung the bell correctly. If a mock object is used in place of the real object, it can be programmed to provide the bell-ringing time (whether it is actually that time or not) so that the alarm clock program can be tested in isolation.

MOCKS, FAKES AND STUBS

Some authors[[1]](http://en.wikipedia.org/wiki/Fake_object#cite_note-1) draw a distinction between *fake* and *mock* objects. Fakes are the simpler of the two, simply implementing the same [interface](http://en.wikipedia.org/wiki/Interface_(computing)#Software_interfaces_in_object_oriented_languages) as the object that they represent and returning pre-arranged responses. Thus a fake object merely provides a set of [method stubs](http://en.wikipedia.org/wiki/Method_stub).

Mock objects in this sense do a little more: their method implementations contain [assertions](http://en.wikipedia.org/wiki/Assertion_(computing)) of their own. This means that a true mock, in this sense, will examine the context of each call— perhaps checking the order in which its methods are called, perhaps performing tests on the data passed into the method calls as arguments.

## Role Descriptions

Here is a summary of what I mean by each of the major [*Test Double*](http://xunitpatterns.com/Test%20Double.html) pattern names:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pattern | Purpose | Has Behavior | Injects [indirect inputs](http://xunitpatterns.com/indirect%20input.html) into SUT | Handles [indirect outputs](http://xunitpatterns.com/indirect%20output.html) of SUT | Values provided by test(er) | Examples |
| [*Test Double*](http://xunitpatterns.com/Test%20Double.html) | Generic name for family |  |  |  |  |  |
| [*Dummy Object*](http://xunitpatterns.com/Dummy%20Object.html)*(page X)* | Attribute or Method Parameter | no | no, never called | no, never called | no | Null, "Ignored String", new Object() |
| [*Test Stub*](http://xunitpatterns.com/Test%20Stub.html)*(page X)* | Verify indirect inputs of SUT | yes | yes | ignores them | inputs |  |
| [*Test Spy*](http://xunitpatterns.com/Test%20Spy.html)*(page X)* | Verify indirect outputs of SUT | yes | optional | captures them for later verification | inputs (optional) |  |
| [*Mock Object*](http://xunitpatterns.com/Mock%20Object.html)*(page X)* | Verify indirect outputs of SUT | yes | optional | verifies correctness against expectations | outputs & inputs (optional) |  |
| [*Fake Object*](http://xunitpatterns.com/Fake%20Object.html)*(page X)* | Run (unrunnable) tests (faster) | yes | no | uses them | none | In-memory database emulator |
| [*Temporary Test Stub*](http://xunitpatterns.com/Test%20Stub.html#Temporary Test Stub)*(see Test Stub)* | Stand in for procedural code not yet written | yes | no | uses them | none | In-memory database emulator |

## 