Sean Cavanaugh
John Krasich

1. We created a mock database. We then told the mock what calls to expect, and what to return when those are calls are made. We then made the calls and checked to see if the results were as expected.
2. You can use LastCall.Throw to simulate an exception being thrown by the mock, if you make the call during the test that would require an exception. This can be used to test exception handling.
3. You do not need to use the stub if it does not return a value. You could replace the stub with a dynamic mock.
4. It creates a mock Database, fills it with rooms, and adds the database to a hotel. It then tests the room count of the hotel against the number of rooms added to the mock database.
5. A service locator is created, and two cars are added. One car is booked by a user. Reflection is used to set ServiceLocator.Instance to access private values stored in the ServiceLocator to verify that a car has been removed, and then to check that the car remaining is the correct one.