# Extending the Scope of Testing

To make sense of unit testing and mocking technologies all you need to do is download the completed demo application from the following link:

[QACS-TL/ExtendingTheScopeOfTestingCompletedDemos (github.com)](https://github.com/QACS-TL/ExtendingTheScopeOfTestingCompletedDemos)

* Review the code in the MissileLauncher project and try to work out what it’s doing.
* Review the code in the ControllerTests project
* Note, the first 14 tests are all good and ensure the state of the controller is set to expected values when the various operations occur. However, how can our unit test PROVE that the correct LaunchMissile sequence was actually followed?
* Look at the final 3 unit tests and note they make use of a mocking technology called FakeItEasy (other mocking technologies are available). Try to work out what additional benefits these tests bring to the party that the “ordinary” unit tests can never do.
* Build and run the tests and confirm they behave as you expected.
* Can you think of any other benefits a mocking technology can bring (think about creating a fake object based on just an interface, rather than a fully implemented class, and specifying “faked” return values to method calls). Could you use it to replace slow, costly calls to a real database (which could change the underlying structure of the database’s data and make the tests non-repeatable) with speedy faked results and not touch the database at all?