

## Mocking/Mockito:

- Unit testing is dependent on testing just the smallest unit of code, generally a method in Java, but much code is highly dependent on other parts of the application: dependent objects for example.
- Mockito is a mocking framework that integrates well with JUnit to “mock” objects. These are “fake” objects that don’t actually execute the underlying code are instead given explicit instructions on what they should return, blackboxing the dependencies methods so we can isolate the code under test.
- Terms:
  - Stub – this is a fake object that has preprogrammed return values for the methods of the class that it is imitating.
  - Mock – A fake object that can be told how its methods should return while the object is under test.
  - Spy – This is a function of a testing support framework like Mockito that can analyze real objects and determine what methods on that object were called and how often. These are used in cases where Mocks are impossible: generally, static objects or methods.
- Mockito Specific Annotations and Methods:
  - Annotations:
    - `@Mock` – identify that an object should be mocked, not instantiated by the programmer.
    - `@RunWith(MockitoJUnitRunner.class)` – This goes above the class declaration and explains to Mockito that we are using JUnit and it needs to provide those implementations for integration.
  - Methods:
    - `.when/.then` – When indicates a method that is being called on a mocked object. Then is used to determine what is the result of that mocked method.
    - `.doThrow` – Can throw an exception on a method that returns void.
    - `.verify` – Checks to ensure that a mocked method was actually called. Generally used with void methods. It can check how many times the method was called.
    - There are many others for various use cases. The Mockito documentation or guides will have many details that you can look up.
- JUnit 4 and JUnit 5 are highly different in their implementations. This means that using Mockito with JUnit 5 is somewhat different, particularly in integration.