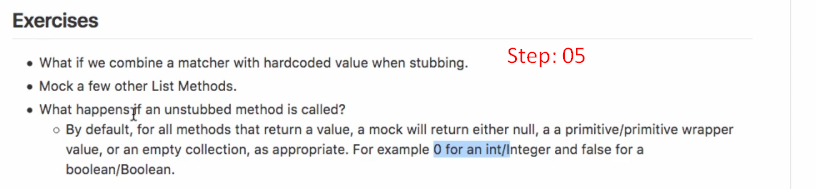
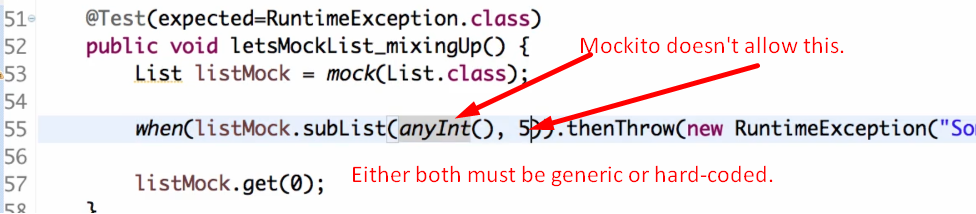
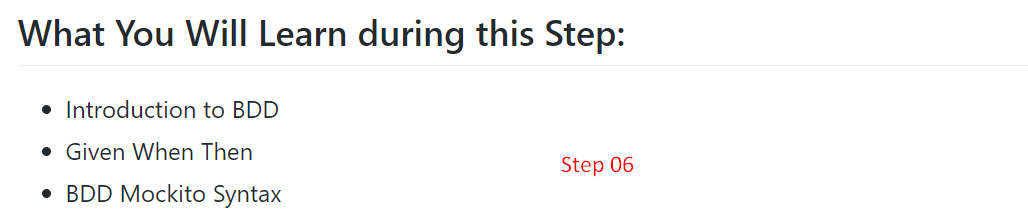
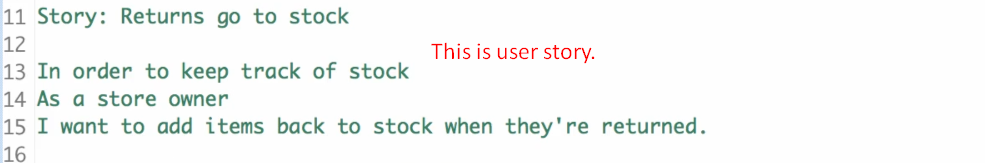
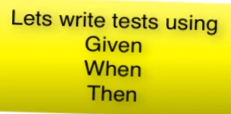
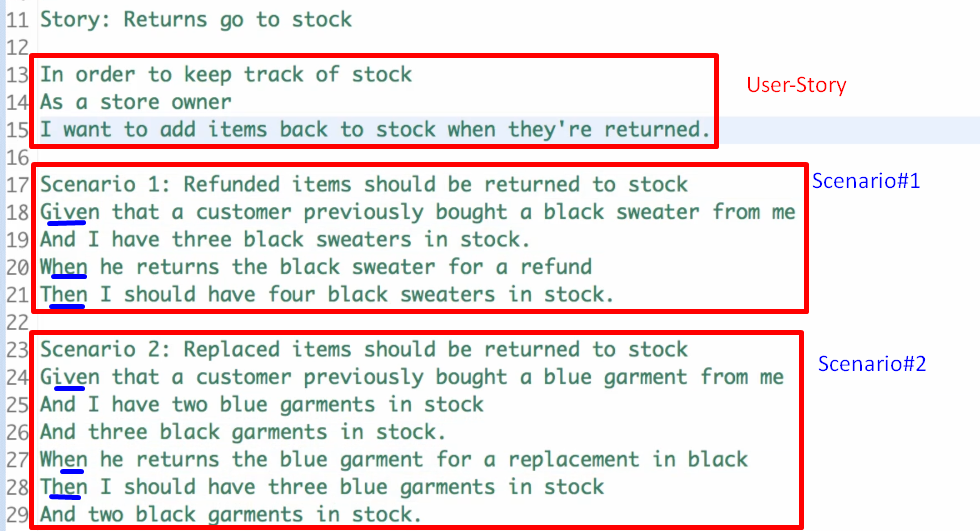
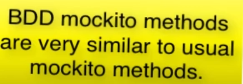
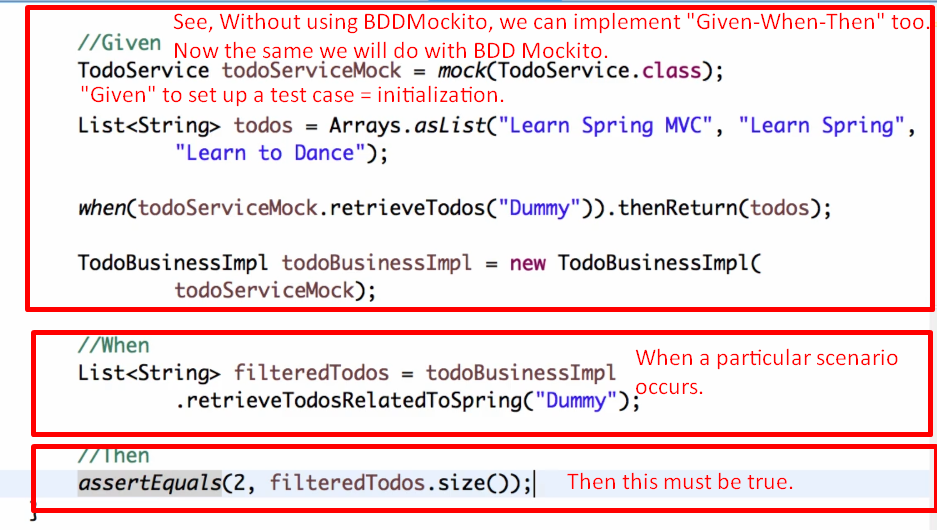
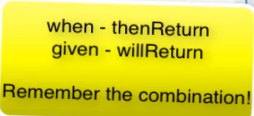
1. 
2. In the below snippet, we’re trying to pass two arguments when stubbing method List.subList(int,int) where the first argument is matcher whereas the second argument is hard-coded value which is not allowed by Mockito. Either both of them must be matchers or hard-coded values.   
   
3. **Agenda**:



1. **What is BDD** (Behaviour-Driven Development)?
2. Let’s try to understand with an example.
3. You know that most of the Agile projects use user-story.
4. You’re given a user-story.
5. If we’re properly doing behaviour development, we will create scenarios around this user story.
6. 
7. **NOTE**: We write a scenario in “**Given-When-Then**” format.  
   Actually, we split a user story into multiple scenarios (“**Given-When-Then**”)
8. Then we start writing tests (in the same format “**Given-When-Then**”) for each of these scenarios.  
   
9. 
10. “**Given-When-Then**”:
    1. **Given**:
       1. It is setup for the scenario.
       2. **Example**: I’ve 3 black sweaters in stock.
    2. **When**:
       1. It is the actual method call means the System Under Test (SUT). The method which we want to test.
       2. **Example**: When a customer returns a black sweater for a refund.
    3. **Then**:
       1. It is a set of asserts that something has happened.
       2. **Example**: Now the stock must have 4 black sweaters.
11. “Given-When-Then” is a good way to write tests for a class.  
    Now what Mockito did is it came up with a specific class which is called BDD Mockito which contains specific methods to write test in “Given-When-Then” format.
12. Let’s use the BDD methods from Mockito.
13. 
14. **NOTE**: Even syntax of mockito (Not talking about BDD Mockito), allows us to write “Given-When-Then” format.
15. Following Code snippet is BDD Style without using BDDMockito.  
    
16. 
17. Graphical user interface, text, application

    Description automatically generated

