

Name: Antonio Cousin, Jr

Prompts

1. What is the difference between TDD and BDD?

The difference between TDD and BDD is the method in which way you enact the process of coding. TDD stands for Test Driven Development, in which you program the testing cases for writing code, and base the main code off of the test cases in the Junit file. Following this example makes it so that you're always eliminating errors and exceptions as they would normally appear in the code (which in turn makes the code more flexible/reusable), but at the expense of the development potentially taking longer, which could make a project's timeline much longer than anticipated. BDD stands for Behaviour Driven Development, in which you program the main behaviour of the code before writing the test cases in the Junit file to ensure the code works. Following this example can make it easier to view the coding process from the perspective of an end-user, though at the expense of the code potentially being easier to break and not as flexible/reusable as a result.

2. What does mocking a class allow you to do?

Mocking a class allows for the programmer to create a controlled testing environment to ensure that you can test out coding for certain aspects of functionality that might not be fully reliable to access at any time. Such an example would be if you're connecting to a database that you might not have the authority to access whenever you wish to, or if the code you're testing might remove irreplaceable data. Using a mocked class from Mockito would ensure that your testing of code for external sources would make it localized while you examine the different ways the code you're testing is impacting the external sources, and makes the testing of said code easier to implement as a result.

Sources

- <https://www.softwaretestinghelp.com/tdd-vs-bdd/>
- <https://semaphoreci.com/community/tutorials/stubbing-and-mocking-with-mockito-2-and-junit#:~:text=Mocking%20is%20the%20act%20of,that%20we%20want%20to%20test.>