What is the difference between TDD and BDD?

(TDD) Test-Driven Development – is unit testing that the testing is done first before the coding. Typically, it is done with two people (this is not necessary, but it is usually done this way). One person writes the test and passes it the second person to write the code to satisfy the test. Then they keep passing it back and forth calling it eXtreme Programing – XP. It is also known as Red, Green, Refactor – Red is writing test code, green is writing app code and refractor is testing until complete and you just keep repeating until it is done.

Advantages of TDD

No extra code, usually faster to code, hence this saves on cost. You and another coder talk it out, so the project is being developed quickly and the best it can be the first time you work on it.

(BDD) Behavior-Driven Development – is results based driven that allows you to test the configuration and testing SQL statements work. This allows you to write fewer unit tests.

The nice thing is you can combine TDD and BDD and get the best of both worlds. TDD for complex methods and BDD for overall functionality. This installs confidence that the app will perform as desired.

*Information from class video notes.

What does mocking a class allow you to do?

https://www.springboottutorial.com/programming-basics-introduction-to-mocking-in-unit-tests#:~:text=Mocking%20is%20a%20technique%20of,fail%20when%20external%20stuff%20changes.

What is the value in separating your code into controller, service, and data access layers rather than keeping it all in the same files?

https://www.coreycleary.me/why-should-you-separate-controllers-from-services-in-node-rest-apis

Why would you want to avoid putting credentials in plaintext in your code?

Anyone can hack and gain access to the files and see all the passwords. This could cause a serious security breach.

What is one method that can be used to avoid putting plaintext database usernames and passwords into your code?

https://medium.com/twodigits/keep-passwords-out-of-source-code-why-and-how-e84f9004815a

What is your favorite thing you learned this week?
I thought it was unique that the compareTo method only cares about the first thing that is different not anything after that. It knows to stop after the first thing and returns the difference from object 1 to object 2.
I watched the recording of my old class for week 11 to see I am missing anything and this was something cool I learned. He went deep into Java review!