Discussion Questions for "Unit Test"

1. How have you tested the code you wrote for assignments or labs in the past? Do you think JUnit testing would have been useful?
2. For most of the simple assignments we've worked on, we haven't bothered with any formal "Unit Testing". How big does a project need to be before formal unit testing becomes really useful, almost required to complete the project?
3. In the industry, do you think most companies that develop in Java use JUnit testing? Would companies that use other languages still use unit testing concepts? Would it be important for software development companies to hire programmers who have unit testing skills and experience?
4. The JUnit @BeforeAll (setUpBeforeClass) method must be static. (You can remove the static keyword, but then JUnit fails at runtime with messages that suggest even more complicated solutions.) This means that values that need to be initialized for all tests must be static, global variables instead of fields. Could that ever be a problem? What if the value of the global variable is changed by one of the tests? Would it be better to declare these fields as "final" and initialize them when they are declared?
5. There are separate tests for the Money constructor and the Money getter methods, but to test the constructor, you have to use the getter methods, and to test the getter methods, you have to use the constructors. If there is a bug in a getter method or a constructor, will you still catch that bug? Will the error messages and failure messages in this case still be correct? (Remember, JUnit may change the order of the tests arbitrarily.)