Discussion Questions for "Java Fields"

1. What are the names and data types of fields you might choose to model a car, a ball, a bank account?
2. What are the advantages and disadvantages of restricting access to a field? If you were designing the Java language, would you leave out the access modifiers? If so, would you make everything public, protected, package-private, private, or some other access mechanism?
3. The default access of a field is package-private. If you were designing Java, would you choose a different default? Why?
4. In Java, a numeric literal is a 32 bit two's-complement integer by default. Would you choose a different default for a numeric literal? What are the advantages and disadvantages of the default that Java uses?
5. In the examples of literals in the lecture, we needed to use the "L" suffix to make a long literal, but we used an "F" suffix to make a shorter floating point number, and Java does not have a suffix to make a short or byte literal, but used casting instead. Would you choose a different compiler implementation to reduce the inconsistency and complexity of literal type conversion? If so, what would your implementation do?
6. We represent an instance of an object schematically by stating the class name, the list of fields, and the values associated with those fields, but we don't list the methods. Why not?