Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-ten)

10.4: Fields

The getFields( ) class method helps you to discover all the public member variables of a class:

```
public class Foo {public String s = "A String";}
Object o = new Foo();

... (pass o to a function which doesn't
 know about Foo)

Class c = o.getClass();
Field f = c.getFields()[0];         // See Note #1 below
// Note #2
System.out.println(f.getType().getName());
System.out.println(f.getName());
System.out.println(f.get(o));       // Note #3
```

(prints)

```
java.lang.String
s
A String
```

Note #1: Field is a class in the package java.lang.reflect. You need to import it for this code to work. Class.getFields( ) returns an array of Field objects; here we're only looking at the first one. We should probably check the array length first.

Note #2: Field.getType( ) returns a Class object representing the type of the member in the object. There are special class objects to represent each primitive type as well; they are static members named TYPE inside the wrapper classes in java.lang. For example, calling getFields[0].getType( ) on the Class object java.awt.Rectangle would return the value Integer.TYPE.

Note #3: Field includes a whole series of methods that let you access the contents of the member represented by a Field object. In addition to get( ), there are functions named getInt( ), setShort( ), etc. Using these methods, you can read or set any public variable in an object whose type was unknown to your code at compile time!