

# Advanced Programming Techniques

## Assignment 2

### Reflective Object Inspector

The goal of this assignment is to create a reflective object inspector that does a complete introspection of an object at runtime. The inspector will be implemented in a Java class called `Inspector`, and will be invoked using the method:

```
public void inspect(Object obj, boolean recursive)
```

This method will introspect on the object specified by the first parameter, printing what it finds to standard output. You should find the following information about the object:

- The name of the declaring class
- The name of the immediate superclass
- The name of the interfaces the class implements
- The methods the class declares. For each, also find the following:
  - The exceptions thrown
  - The parameter types
  - The return type
  - The modifiers
- The constructors the class declares. For each, also find the following:
  - The parameter types
  - The modifiers
- The fields the class declares. For each, also find the following:
  - The type
  - The modifiers
- The current value of each field. If the field is an object reference, and *recursive* is set to false, simply print out the “reference value” directly (this will be the name of the object’s class plus the object’s “identity hash code”).

You must also traverse the inheritance hierarchy to find all the methods, constructors, fields, and field values that each superclass and superinterface declares. Be sure you can also handle any array you might encounter, printing out its name, component type, length, and all its contents.

### Recursive Inspection

If the `inspect` method is invoked with *recursive* set to false, simply find information for the object specified. If it is set to true, then fully inspect each field that is an object.

### Other Requirements

Your TA will provide a driver program that creates an object to inspect, and then invokes your *inspect* method. Capture the output of the driver program using the *script* UNIX command, saving to a file called *script.txt*. The TA will compile and run your code to verify that everything works. You must also use version control, unit testing, and refactoring throughout this assignment.

### Submit the following:

1. An electronic copy of your `Inspector` class (in a file called *Inspector.java*), the *script.txt* file, your unit tests, your version control logs, and a record of your refactorings (in a Word, PDF, or text file called *refactorings*). Use the *Assignment 2* Dropbox Folder in D2L to submit electronically.

## Advanced Programming Techniques

### Assignment 2 Grading

Student: \_\_\_\_\_

#### Introspection

Name of declaring class	2	_____
Name of superclass	2	_____
Names of interfaces	2	_____
Methods (name, exceptions, parameters, return type, modifiers)	10	_____
Constructors (name, parameters, modifiers)	6	_____
Fields (name, type, modifiers)	6	_____
Field values	2	_____
Traversal of class and interface hierarchy	4	_____
Handles arrays	2	_____
Well-formatted output (shown in <i>script.txt</i> )	2	_____

<b>Recursive Inspection</b>	6	_____
-----------------------------	---	-------

#### Other Requirements

Version control (show log files)	4	_____
Unit Tests	4	_____
Refactoring (described in <i>refactorings</i> file)	4	_____
Design Quality	4	_____

<b>Total</b>	<b>60</b>	_____	_____ %
--------------	-----------	-------	---------