Array Declarations (OCA Objectives 4.1 and 4.2)

- ❑ Arrays can hold primitives or objects, but the array itself is always an object.
- ❑ When you declare an array, the brackets can be to the left or to the right of the variable name.
- ❑ It is never legal to include the size of an array in the declaration.
- ❑ An array of objects can hold any object that passes the IS-A (or `instanceof`) test for the declared type of the array. For example, if `Horse` extends `Animal`, then a `Horse` object can go into an `Animal` array.

Static Variables and Methods (OCA Objective 6.2)

- ❑ They are not tied to any particular instance of a class.
- ❑ No class instances are needed in order to use `static` members of the class.
- ❑ There is only one copy of a `static` variable/class and all instances share it.
- ❑ `static` methods do not have direct access to nonstatic members.

enums (OCA Objective 1.2 and OCP Objective 2.5)

- ❑ An `enum` specifies a list of constant values assigned to a type.
- ❑ An `enum` is NOT a `String` or an `int`; an `enum` constant's type is the `enum` type. For example, `SUMMER` and `FALL` are of the `enum` type `Season`.
- ❑ An `enum` can be declared outside or inside a class, but NOT in a method.
- ❑ An `enum` declared outside a class must NOT be marked `static`, `final`, `abstract`, `protected`, or `private`.
- ❑ `enum`s can contain constructors, methods, variables, and constant-specific class bodies.
- ❑ `enum` constants can send arguments to the `enum` constructor, using the syntax `BIG(8)`, where the `int` literal 8 is passed to the `enum` constructor.
- ❑ `enum` constructors can have arguments and can be overloaded.
- ❑ `enum` constructors can NEVER be invoked directly in code. They are always called automatically when an `enum` is initialized.
- ❑ The semicolon at the end of an `enum` declaration is optional. These are legal:
  - ❑ `enum Foo { ONE, TWO, THREE}`
    `enum Foo { ONE, TWO, THREE};`
- ❑ `MyEnum.values()` returns an array of `MyEnum`'s values.

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all of the choices carefully, as there may be more than one correct answer. Choose all correct answers for each question. Stay focused.

If you have a rough time with these at first, don't beat yourself up. Be positive. Repeat nice affirmations to yourself like, "I am smart enough to understand `enums`" and "OK, so that other guy knows `enums` better than I do, but I bet he can't <insert something you *are* good at> like me."

1. Which are true? (Choose all that apply.)

A. "X extends Y" is correct if and only if X is a class and Y is an interface.

B. "X extends Y" is correct if and only if X is an interface and Y is a class.

C. "X extends Y" is correct if X and Y are either both classes or both interfaces.

D. "X extends Y" is correct for all combinations of X and Y being classes and/or interfaces.

2. Given:

```
class Rocket {
  private void blastOff() { System.out.print("bang "); }
}
public class Shuttle extends Rocket {
  public static void main(String[] args) {
    new Shuttle().go();
  }
  void go() {
    blastOff();
    // Rocket.blastOff();  // line A
  }
  private void blastOff() { System.out.print("sh-bang "); }
}
```

Which are true? (Choose all that apply.)

A. As the code stands, the output is `bang`

B. As the code stands, the output is `sh-bang`

C. As the code stands, compilation fails.

D. If line A is uncommented, the output is `bang bang`

E. If line A is uncommented, the output is `sh-bang bang`

F. If line A is uncommented, compilation fails.

3. Given that the `for` loop's syntax is correct, and given:

```
import static java.lang.System.*;
class _ {
```