❑ When you declare an array, the brackets can be to the left or 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 (Objective 1.4)

❑ They are not tied to any particular instance of a class.

❑ No classes 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 non-static members.

Enums (Objective 1.3)

❑ 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`.

❑ Enums can contain constructors, methods, variables, and constant class bo dies.

❑ `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 is 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. Which method names follow the JavaBeans standard? (Choose all that apply.) ?

A. `addSize`

B. `getCust`

C. `deleteRep`

D. `isColorado`

E. `putDimensions`

3. Given: ?

```
1. class Voop {
2.   public static void main(String[] args) {
```

```
3.      doStuff(1);
4.      doStuff(1,2);
5.    }
6.   // insert code here
7. }
```

Which, inserted independently at line 6, will compile? (Choose all that apply.)

A. `static void doStuff(int... doArgs) { }`

B. `static void doStuff(int[] doArgs) { }`

C. `static void doStuff(int doArgs...) { }`

D. `static void doStuff(int... doArgs, int y) { }`

E. `static void doStuff(int x, int... doArgs) { }`

4. Given:

```
 1. enum Animals {
 2.   DOG("woof"), CAT("meow"), FISH("burble");
 3.   String sound;
 4.   Animals(String s) { sound = s; }
 5. }
 6. class TestEnum {
 7.   static Animals a;
 8.   public static void main(String [] args) {
 9.     System.out.println(a.DOG.sound + " " + a.FISH.sound);
10.   }
11. }
```

What is the result?

A. `woof burble`

B. Multiple compilation errors

C. Compilation fails due to an error on line 2

D. Compilation fails due to an error on line 3

E. Compilation fails due to an error on line 4

F. Compilation fails due to an error on line 9

5. Given two files:

```
 1. package pkgA;
 2. public class Foo {
 3.   int a = 5;
 4.   protected int b = 6;
 5.   public int c = 7;
 6. }
```

```
 3. package pkgB;
 4. import pkgA.*;
 5. public class Baz {
 6.   public static void main(String[] args) {
 7.     Foo f = new Foo();
 8.     System.out.print(" " + f.a);
 9.     System.out.print(" " + f.b);
10.     System.out.print(" " + f.c);
11.   }
12. }
```

What is the result? (Choose all that apply.)

A. 5 6 7

B. 5 followed by an exception

C. Compilation fails with an error on line 7

D. Compilation fails with an error on line 8

E. Compilation fails with an error on line 9

F. Compilation fails with an error on line 10

6. Given:

```
 1. public class Electronic implements Device
          { public void doIt() { } }
 2.
 3. abstract class Phone1 extends Electronic { }
 4.
 5. abstract class Phone2 extends Electronic
      { public void doIt(int x) { } }
 6.
 7. class Phone3 extends Electronic implements Device
      { public void doStuff() { } }
 8.
 9. interface Device { public void doIt(); }
```

What is the result? (Choose all that apply.)

A. Compilation succeeds

B. Compilation fails with an error on line 1

C. Compilation fails with an error on line 3

D. Compilation fails with an error on line 5

E. Compilation fails with an error on line 7

F. Compilation fails with an error on line 9

7. Given:

```
 4. class Announce {
 5.   public static void main(String[] args) {
 6.     for(int __x = 0; __x < 3; __x++) ;
 7.     int #lb = 7;
 8.     long [] x [5];
 9.     Boolean []ba[];
10.     enum Traffic { RED, YELLOW, GREEN };
11.   }
12. }
```

What is the result? (Choose all that apply.)

A. Compilation succeeds

B. Compilation fails with an error on line 6

C. Compilation fails with an error on line 7

D. Compilation fails with an error on line 8

E. Compilation fails with an error on line 9

F. Compilation fails with an error on line 10

8. Given:

```
 3. public class TestDays {
 4.   public enum Days { MON, TUE, WED };
 5.   public static void main(String[] args) {
 6.     for(Days d : Days.values() )
 7.       ;
 8.     Days [] d2 = Days.values();
 9.     System.out.println(d2[2]);
10.   }
11. }
```

What is the result? (Choose all that apply.)

A. `TUE`

B. `WED`

C. The output is unpredictable

D. Compilation fails due to an error on line 4

E. Compilation fails due to an error on line 6

F. Compilation fails due to an error on line 8

G. Compilation fails due to an error on line 9

9. Given: ?

```
 4. public class Frodo extends Hobbit {
 5.   public static void main(String[] args) {
 6.     Short myGold = 7;
 7.     System.out.println(countGold(myGold, 6));
 8.   }
 9. }
10. class Hobbit {
11.   int countGold(int x, int y) { return x + y; }
12. }
```

What is the result?

A. 13

B. Compilation fails due to multiple errors

C. Compilation fails due to an error on line 6

D. Compilation fails due to an error on line 7

E. Compilation fails due to an error on line 11

Answers

1. Answer:

☑ **C** is correct.

☒ **A** is incorrect because classes implement interfaces, they don't extend them. **B** is incorrect because interfaces only "inherit from" other interfaces. **D** is incorrect based on the preceding rules. (Objective 1.2)

2. Answer:

☑ **B** and **D** use the valid prefixes `'get'` and `'is'`.

☒ **A** is incorrect because `'add'` can be used only with Listener methods. **C** and **E** are incorrect because '`delete`' and '`put`' are not standard JavaBeans name prefixes. (Objective 1.4)

3. Answer:

☑ **A** and **E** use valid var-args syntax.

☒ **B** and **C** are invalid var-arg syntax, and **D** is invalid because the var-arg must be the last of a method's arguments. (Objective 1.4)

4. Answer:

☑ **A** is correct; enums can have constructors and variables.

☒ **B, C, D, E,** and **F** are incorrect; these lines all use correct syntax. (Objective 1.3)

5. Answer:

☑ **D** and **E** are correct. Variable `a` has default access, so it cannot be accessed from outside the package. Variable `b` has protected access in pkgA.

☒ **A, B, C,** and **F** are incorrect based on the above information. (Objective 1.1)

6. Answer:

☑ **A** is correct; all of these are legal declarations.

☒ **B, C, D, E,** and **F** are incorrect based on the above information. (Objective 1.2)

7. Answer:

   ☑ **C, D,** and **F** are correct. Variable names cannot begin with a #, an array declaration can’t include a size without an instantiation, and enums can’t be declared within a method.

   ☒ **A, B,** and **E** are incorrect based on the above information. (Objective 1.3)

8. Answer:

   ☑ **B** is correct. Every `enum` comes with a `static values()` method that returns an array of the `enum`'s values, in the order in which they are declared in the `enum`.

   ☒ **A, C, D, E, F,** and **G** are incorrect based on the above information. (Objective 1.3)

9. Answer:

   ☑ **D** is correct. The `Short myGold` is autoboxed correctly, but the `countGold()` method cannot be invoked from a static context.

   ☒ **A, B, C,** and **E** are incorrect based on the above information. (Objective 1.4)