# Chapter 10. Implementing and Using Classes

[10-1] . What is called a client of a class X?

 A.✓ Any subclass of X.

 B.✓ A class whose objects serve as X’s fields.

 C.✓ Any class above or below X in the inheritance line.

 **D.✓ Any class that invokes X’s constructors and/or calls its methods.**

[10-2] . Consider the following class:

public class Ticket

{

private int price;

public int getPrice()

{

return price;

}

public int compareTo(Ticket other)

{

return < *expression* >;

}

< *constructors and other fields and methods not shown* >

}

True or false? The following can replace < *expression* > in the compareTo method, so that the class compiles with no errors:

** (a)✓ price - other.price**

** (b)✓ this.price - other.price**

** (c)✓ price - other.getPrice()**

** (d)✓ getPrice() - other.getPrice()**

** (e)✓ this.getPrice() - other.getPrice()**

[10-3] . Given

public class SomeClass

{

private int x, y;

...

public char someFun(SomeClass o)

{

...

}

}

Which of the following statements would be illegal in someFun’s body?

 A.✓ x = y;

 B.✓ x = o.y;

** C.✓ o.x = this.y;**

 D.✓ All of the above would be legal

[10-4] . Which of the following is the appropriate name for a constructor in the class Flower?

** A.✓ Flower**

 B.✓ flower

 C.✓ flowerConstructor

 D.✓ newFlower

[10-5] . True or false?

T (a)✓ Constructors never return any values.

F (b)✓ A constructor must explicitly initialize all the fields of its class.

F (c)✓ All constructors of a class take the same number and types of parameters.

T (d)✓ A class’s constructor can call that class’s methods.

T (e)✓ A “copy constructor” takes one parameter that is an object of the same class and creates a new object, a copy of the parameter.

[10-6] . The class Animal has a constructor that takes one String parameter. Which of the following is a valid statement that declares and creates an instance of an Animal in a program?

 A.✓ Animal elephant = new("Dumbo");

 B.✓ Animal dumbo = new Elephant("Dumbo");

** C.✓ Animal elephant = new Animal("Dumbo");**

 D.✓ Animal dumbo = new Animal(String "Dumbo");

[10-7] . True or false?

T (a)✓ You must either specify the return type of a method or declare it void.

F (b)✓ The name of a method that returns the value of a class’s field must always start with “get.”

F (c)✓ The order of a method’s parameters is important only when they have the same type.

T (d)✓ In a method’s body, its parameters act like local variables.

[10-8] 8. What happens if you forget to initialize a field that is an object and start calling its methods?

 A.✓ A syntax error is reported.

** B.✓ A run-time NullPointerException error is reported.**

 C.✓ A new object is created with the default values for its fields.

 D.✓ A new object is created with unpredictable values for its fields.

[10-9] . True or false?

T (a)✓ A method that is not void must have at least one return.

F (b)✓ A void method must have at least one return.

T (c)✓ A method may have several return statements.

F (d)✓ One return statement in a method may return a char value, while another return in the same method may return a String.

T (e)✓ If the return type of a method is a class, the method returns a reference to an object of that class.

[10-10] . What are “overloaded” methods?

 A.✓ Methods in a derived class that redefine a method from the base class

 B.✓ Methods of a class that create objects of the same class

 C.✓ Methods of different classes that have the same name

** D.✓ Methods of the same class that have the same name but different numbers or types of parameters**

[10-11] . Why doesn’t Java let you create an object of the Math class?

 A.✓ Because Math has no fields

 B.✓ Because Math has no constructors

** C.✓ Because all Math’s methods and fields are static, so all Math objects would be identical**

 D.✓ Because Math does not represent a real-world object

[10-12] . Which of the following is NOT a good reason for making a class’s field static?

 A.✓ The field is final, and the same value must be used in all objects of the class.

 B.✓ The field represents a symbolic constant used in many places in the program.

** C.✓ The field is used to count the number of times any of the methods of a particular object are called.**

 D.✓ The field is used to count the number of times a particular constructor of the class is invoked.

[10-13] . True or false?

F (a)✓ You must construct at least one instance of a class to use that class’s static methods.

F (b)✓ A static method can call an instance method of the same class using the this-dot prefix.

T (c)✓ An instance method of any object can call a static method of the same class.

F (d)✓ A static method can set a non-static field of the same class using the this-dot prefix.

T (e)✓ An instance method of any object can modify a static field of the same class.

[10-14] 14. Recall that the *Snack Bar* program models three vending machines selling different and possibly differently priced items. True or false?

T (a)✓ Each vending machine has its own Vendor object.

F (b)✓ The myPrice field in Vendor could be declared static.

F (c)✓ The setStock method in Vendor could be declared static.

T (d)✓ The FULL\_STOCK field in VendingMachine can be declared final static.

**Questions 15 - 18 refer to the following class:**

public class SomeClass

{

private static int dflt = -1;

private int x;

public static void reset(int val)

{

dflt = val;

}

public static void reset(SomeClass o)

{

o.x = dflt;

}

public void reset()

{

x = dflt;

}

public int getX()

{

return x;

}

public String toString()

{

return "" + x;

}

}

These questions assume, also, that main in SomeOtherClass declares and initializes an object of SomeClass:

SomeClass obj = new SomeClass();

[10-15] . True or false? The following declaration would be valid in main in SomeOtherClass:

T (a)✓ SomeClass z1;

T (b)✓ SomeClass z2 = new SomeClass();

F (c)✓ SomeClass z3 = new SomeClass(-1);

F**** (d)✓ SomeClass z4 = new SomeClass(obj);

[10-16] . True or false? The following statement would be valid in main in SomeOtherClass:

F (a)✓ System.out.print(obj.dflt);

F (b)✓ System.out.print(obj.x);

T (c)✓ System.out.print(obj.getX());

F (d)✓ System.out.print(SomeClass.dflt);

F (e)✓ System.out.print(SomeClass.getX());

[10-17] . True or false? The following call would be valid in main in SomeOtherClass:

F (a)✓ SomeClass.reset();

T (b)✓ SomeClass.reset(-1);

T (c)✓ SomeClass.reset(obj);

T (d)✓ obj.reset();

[10-18] . What is displayed when the following code fragment is compiled/executed?

SomeClass.reset(2000);

SomeClass t = new SomeClass();

t.reset();

System.out.print(t);

 A.✓ Syntax error

** B.✓ 2000**

 C.✓ 0

 D.✓ -1

[10-19] . Consider the following statements that involve the class Fraction described in Chapter 10:

Fraction f1 = new Fraction(1, 3), f2 = new Fraction(1, 6);

Fraction sum = f1.add(f2);

Which of the following best describes what happens?

** A.✓ sum becomes 1/2; f1 and f2 remain unchanged.**

 B.✓ sum and f1 become 1/2; f2 remains 1/6.

 C.✓ sum and f2 become 1/2; f1 remains 1/3.

 D.✓ sum becomes 1/3; f1 becomes 1/2; f2 remains 1/6.

[10-20] . Who invented Java?

 A.✓ Alan Kay

** B.✓ James Gosling**

 C.✓ Tim Berners-Lee

 D.✓ Bjarne Stroustrup



[10-21]