

# CS302: Self-Check Quiz 1

name: \_\_\_\_\_

## Part I True or False

**For these questions, is the statement true or false?**

Assume the statements are about the Java programming language.

- 1.) `7.11` is an example of a *literal* constant.
- 2.) `Math.PI()` accesses the *class constant* `PI` in the `Math` class.
- 3.) *Compilation* errors include mistakes in syntax.
- 4.) *Class* methods are declared with the `static` modifier.
- 5.) *Reserved words* can, but shouldn't, be used as *identifiers*.
- 6.) Execution for all Java applications begins in the method named *start*.
- 7.) A *message* is sent to an object to have it perform a task.
- 8.) A Java *compiler* translates bytecodes into source code.
- 9.) A *constructor* must have a return type specified.
- 10.) In the *design* phase of software engineering a list of requirements is specified.
- 11.) The code `x = y;` means *y* is *assigned* the value in *x*.
- 12.) Indentation of code affects its execution.
- 13.) In Java, classes can be grouped into *packages*.
- 14.) Numerical data types such as `int` are *primitive* data types.
- 15.) *Visibility modifiers* control access to the members of a class.
- 16.) Variables declared within a method declaration are called *local variables*.
- 17.) `3CPO` is a valid *identifier*.
- 18.) It is a good programming practice to capitalize all of the letters in a constant's name.

**For these questions, determine if the message valid for the given method definition.** Assume the variables below exist, the messages listed on the left would be in the main method, and the partial method definitions listed on the right would be in the `Quiz` class.

```
int    i = 11;
double d = 2.2;
Quiz   o = new Quiz();
```

- |                                      |  |
|--------------------------------------|--|
| 19.) <code>o.showvalues()</code>     | <code>public void showValues ()</code>                 |
| 20.) <code>o.setData(22, d)</code>   | <code>public void setData (double f, double g)</code>  |
| 21.) <code>o.find(d, i)</code>       | <code>public void find (double d, int i, int j)</code> |
| 22.) <code>i = o.setMax(11)</code>   | <code>private int setMax (int m)</code>                |
| 23.) <code>o.compute(i, i)</code>    | <code>public void compute (int i, int x)</code>        |
| 24.) <code>o.doIt(new Quiz())</code> | <code>public void doIt (Quiz q)</code>                 |
| 25.) <code>o.jump(i, d)</code>       | <code>public void jump (double d, int i)</code>        |

## Part II Multiple Choice

For these questions, choose the one best answer after reading all of the choices.

26.) If multiplication had lower precedence than addition the result of the expression:

$$2 * 3 + 4 * 3 + 2$$

would be which one of the following?

- A. 14
- B. 20
- C. 32
- D. 70
- E. 144

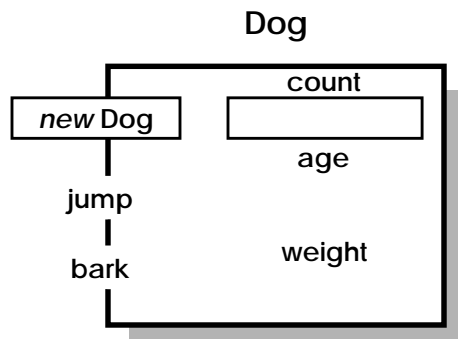
27.) One programmer is implementing an instantiable class that another programmer will use. Which one of the following aspects about the public methods of the instantiable class does *not* need to be known by both programmers.

- A. the methods' names
- B. how the methods are implemented
- C. the number, types, and order of each method's parameters
- D. what the methods do
- E. what the methods return

28.) Which one of the following describes the most common task of a constructor?

- A. Initialize the data members for the instance of the class being created.
- B. Test all of the class's methods.
- C. Call the garbage collector to free up unused memory.
- D. Display a message informing the user that a new instance of the class has been created.
- E. Do nothing since default constructors are automatically provided.

29.) Consider the following class diagram:



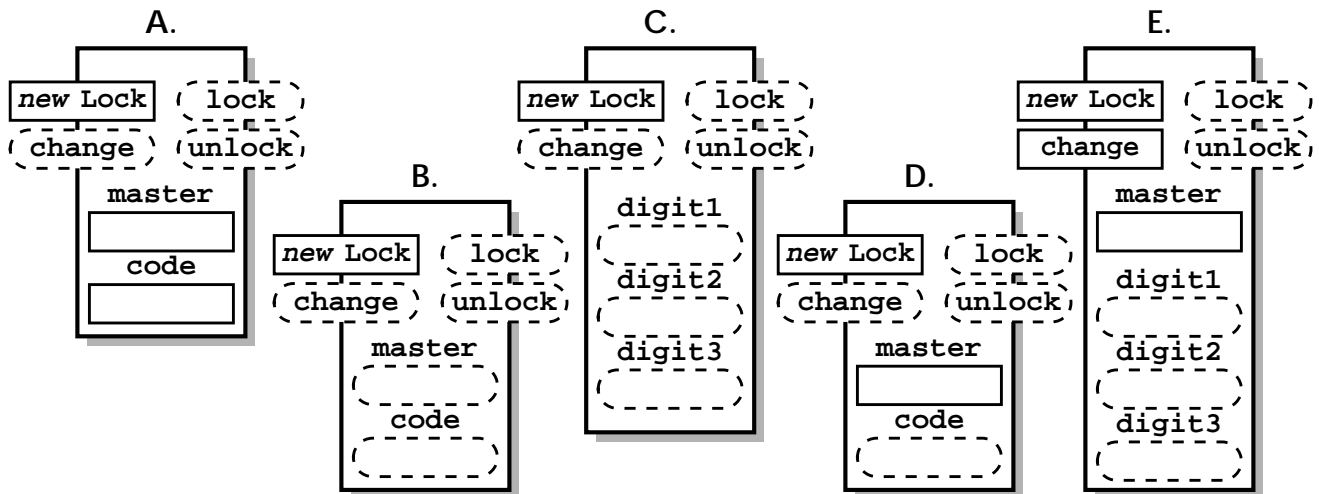
Which one of the following is *not* always true about all of the instances of this class?

- A. All instances have the instance variables `age` and `weight`.
- B. All instances have the same values for instance variables `age` and `weight`.
- C. All instances share the class variable `count`.
- D. All instances have the same value for the class variable `count`.
- E. All instances have the instance methods `jump` and `bark`.

30.) Consider the following description:

The parental lock for a TV is a three digit number that is set by parents, and is used to block their children from watching locked channels. The code number must be entered to lock and unlock channels. This code is also entered to change the lock to a new code. There is also a master code, which can't be changed, that is the same for all TV parental locks. It is used to unlock all of the channels if the secret code is forgotten.

Which one of the following class diagrams is best designed to meet the description above?



31.) A data type with a larger range of values is said to have a higher precision. Which one of the following correctly orders the numeric data types from higher to lower precision?

- A. double, float, long, int
- B. double, long, float, int
- C. float, double, int, long
- D. int, float, long, double
- E. int, long, float, double

32.) Which one is *not* a reason for making the data members of a class private?

Data members are made private ...

- A. ... to ensure the integrity of the class.
- B. ... to hide the details of a class's implementation.
- C. ... to protect objects from being put into invalid states.
- D. ... so the compiler can generate more efficient code for the class's methods.
- E. ... so their types can be changed without requiring changes to the code that uses the class.

33.) Which one of the following statements about passing arguments to methods is *not* true?

- A. An argument is matched to the parameter with the same name.
- B. The pass-by-value scheme is like assignment in that the argument's value is assigned to the parameter.
- C. An argument and its corresponding parameter are aliases (i.e. reference the same object) when the argument is an object variable.
- D. If a method definition lists no parameters, its corresponding messages cannot contain any arguments.
- E. Changes made to a parameter will not affect its corresponding argument's value.

**34.)** Consider the following complete implementation of the `Location` class:

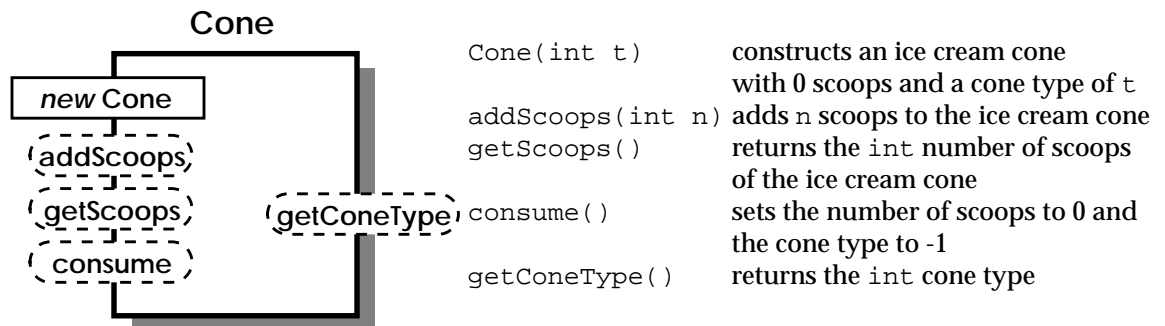
```
class Location {
    private int x, y;
    public Location (int initX, int initY) {
        x = initX;
        y = initY;
    }
    public void setX (int newX) {
        x = newX;
    }
    public void setY (int newY) {
        y = newY;
    }
    public int getX () {
        return x;
    }
    public int getY () {
        return y;
    }
}
```

Assume `point` is a `Location` object in the main method. Which one of the following code fragments from the main method in the `Exam` class correctly moves `point` from `x, y` to `x+1, y+1`?

- A. `Location(point.getX() + 1, point.getY() + 1);`
- B. `point.getX() = point.getX() + 1;`  
`point.getY() = point.getY() + 1;`
- C. `point.y = point.y + 1;`  
`point.x = point.x + 1;`
- D. `point.setX(point.x + 1);`  
`point.setY(point.y + 1);`
- E. `int moveX = point.getX() + 1;`  
`point.setX(moveX);`  
`int moveY = point.getY() + 1;`  
`point.setY(moveY);`

## Part III Written Answers

- 1.) **Write the code for the instantiable class `Cone`** that represents an ice cream cone. The following information is provided. *If something isn't specified do something reasonable.*



- 2.) Using your class from question 1, write a code fragment that creates two cones one having a cone type of 2 and the other 3. Add 11 scoops to one cone and 22 to the other. Finally consume the cone with 22 scoops.

For each of the following code fragments, draw a memory diagram. If a variable's value is replaced, put a line through the old value rather than erasing it. If memory is destroyed or would be reclaimed by garbage collection, cross it out too.

3.)

```
int x, y = 11;  
x = y;  
y = 22;
```

- 4.) Assume the `Quiz` class has a single data member named `count` that is initialized with the argument passed to the constructor.

```
Quiz q = new Quiz(11);  
Quiz p;  
p = q;  
q = new Quiz(22);  
p = q;
```

- 5.) Assume the code on the left is in the main method, the code on the right is in the `Game` class, and the `Die` class is the same as the one shown in lecture.

```
Game g = new Game();  
int x = 121;  
Die die = new Die(11,7);  
g.start(x, die);
```

```
public void start (int n, Die d) {  
    int x = 5  
    n = x;  
    d.setTop(n);  
    d = new Die(6, 1);  
}
```