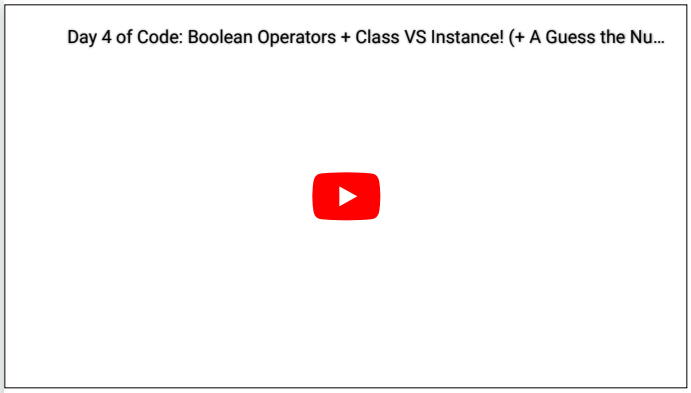
Day 4: Class vs. Instance



Terms you'll find helpful in completing today's challenge are outlined below, along with sample Java code (where appropriate).

Class

A blueprint defining the charactaristics and behaviors of an object of that class type. Class names should be written in CamelCase, starting with a capital letter.

```
class MyClass{
    ...
}
```

Each class has two types of variables: class variables and instance variables; class variables point to the same (static) variable across all instances of a class, and instance variables have distinct values that vary from instance to instance.

Class Constructor

Creates an instance of a class (i.e.: calling the Dog constructor creates an instance of Dog). A class can have one or more constructors that build different versions of the same type of object. A constructor with no parameters is called a default constructor; it creates an object with default initial values specified by the programmer. A constructor that takes one or more parameters (i.e.: values in parentheses) is called a parameterized constructor. Many languages allow you to have multiple constructors, provided that each constructor takes different types of parameters; these are called overloaded constructors.

```
class Dog{ // class name
    static String unnamed = "I need a name!"; // class variable
    int weight; // instance variable
    String name; // instance variable
    String coatColor; // instance variable

    Dog(){ // default constructor
        this.weight = 0;
        this.name = unnamed;
        this.coatColor = "none";
    }
    Dog(int weight, String color){ // parameterized constructor
        // initialize instance variables
        this.weight = weight; // assign parameter's value to instance variable
        this.name = unnamed;
        this.coatColor = color;
    }
    Dog(String dogName, String color){ // overloaded parameterized constructor
        // initialize instance variables
        this.weight = 0;
        this.name = dogName;
        this.coatColor = color;
    }
}
```

Method

A sort of named procedure associated with a class that performs a predefined action. In the sample code below, returnType will either

TUTORIAL DETAILS

Tutorial By

AllisonP

Video By

blondiebytes

View Practice Challenge

NEED HELP?

View discussions

View editorial

View top submissions

be a data type or ***void*** if no value need be returned. Like a constructor, a method can have **0** or more parameters.

```
returnType methodName(parameterOne, ..., parameterN){
    ...
    return variableOfReturnType; // no return statement if void
}
```

Most classes will have methods called getters and setters that get (return) or set the values of its instance variables. Standard getter/setter syntax:

```
class MyClass{
    dataType instanceVariable;
    ...
    void setInstanceVariable(int value){
        this.instanceVariable = value;
    }
    dataType getInstanceVariable(){
        return instanceVariable;
    }
}
```

Structuring code this way is a means of managing how the instance variable is accessed and/or modified.

Parameter

A parenthetical variable in a function or constructor declaration (e.g.: in `int methodOne(int x)`, the parameter is `int x`).

Argument

The actual value of a parameter (e.g.: in `methodOne(5)`, the argument passed as variable `x` is `5`).

Additional Language Resources

Python Class and Instance Variables

View Practice Challenge