

# Crash Course in Java

An Introduction to Object-Oriented  
Programming

# Introducing Java

- Has absolutely nothing to do with Javascript
- Developed in 1995, among most popular languages.
- Syntax derived from C/C++
- Designed to let application developers “write once, run anywhere.”



# Properties of Java

- Strictly-typed
- Object-oriented
- Compiled... kind of.

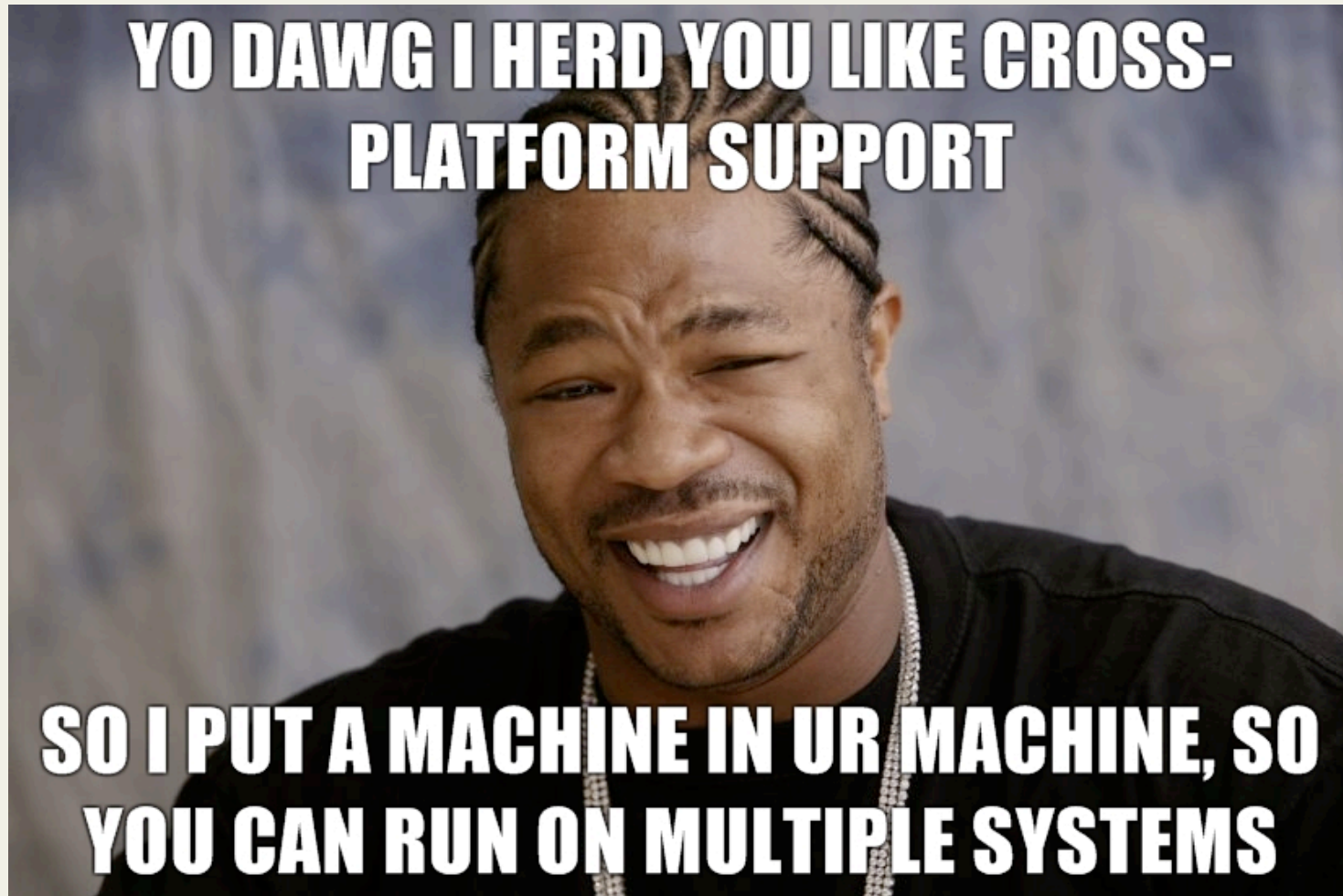


# Java Virtual Machine

- Java code is compiled to Java 'bytecode'.
- Bytecode runs on the Java Virtual Machine, which runs on your computer!



# Java Virtual Machine



# Java Virtual Machine

- JVMs are written for all kinds of devices.
- “Java powers set-top boxes, printers, Web cams, games, car navigation systems, lottery terminals, medical devices, parking payment stations, and more.” ([java.com/en/about](http://java.com/en/about))

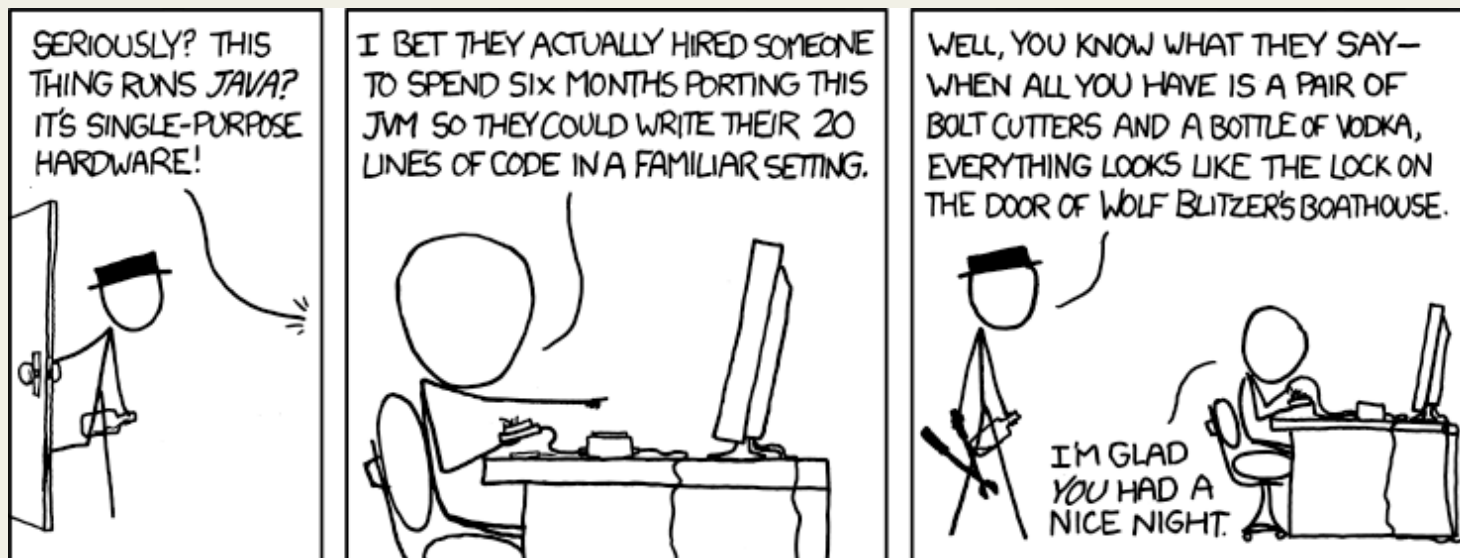


# Java Virtual Machine

- JVMs are written for all kinds of devices.
- “Java powers set-top boxes, printers, Web cams, games, car navigation systems, lottery terminals, medical devices, parking payment stations, and more.” ([java.com/en/about](http://java.com/en/about))



# Java Virtual Machine



XKCD – “Golden Hammer”





# Advantages & Limitations

- + Supported across multiple platforms.
- + No need to manage memory directly
  - 'Garbage collection' included in Java
- + Expansive library of useful classes/objects
- Considerable overhead, efficiency costs.



# Hello, Java!



**Hello.java:**

```
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Hello, Java!");
    }
}
```



public static void what?



# public static void what?

- Java is an 'Object-oriented' programming language.
- Programs in Java are 'classes', which have:
  - fields
  - Methods
- So programs are things which can have 'variables' and 'functions'! That's not so bad...



# public static void what?

- 'public' and 'static' are special properties of things in a class.
- We'll come back to this when we discuss classes/objects more in-depth!
- For now...

**DON'T PANIC**



# Compiling & Running

## Compiling Java Bytecode

- `javac Hello.java`
  - Compile `Hello.java` to create `Hello.class`

## Run in JVM

- `java Hello`
  - Having compiled, execute bytecode in `Hello.class`



# Using Classes

To use some method 'method1' which is contained in class 'class0'...

- Include the appropriate package
  - *import ... class0;*
- Call the method!
  - *class0.method1(parameters);*

Let's look at some examples...



# Scanner



*“What does the Scanner see?”*





# Scanner

- A Scanner sees input from a stream, often `System.in` (this is just `stdin`!)
- To use Scanner, we use the Scanner class to create a Scanner object... (remember creating a `'new Array()'` in Javascript?).

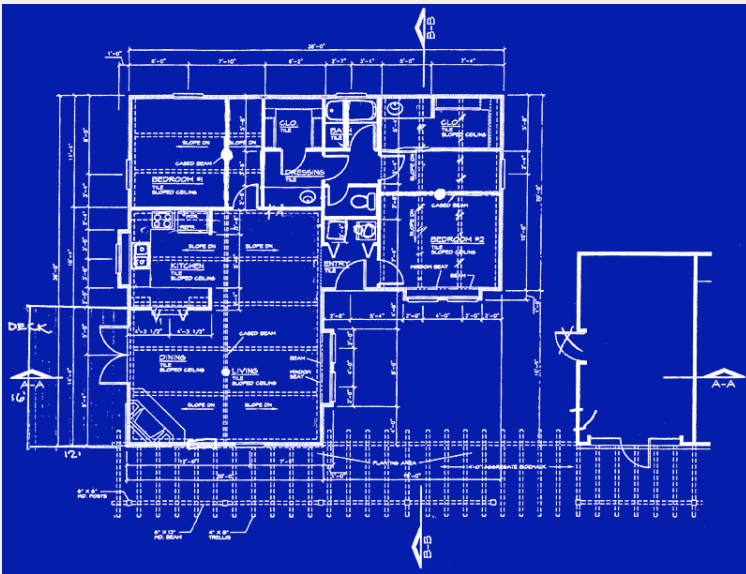


# Classes & Objects

- Class – description of an object. Describes a structure containing methods and fields.
- Object – an instantiated class. The realization of the object which was described by a class.



# Classes & Objects



Class

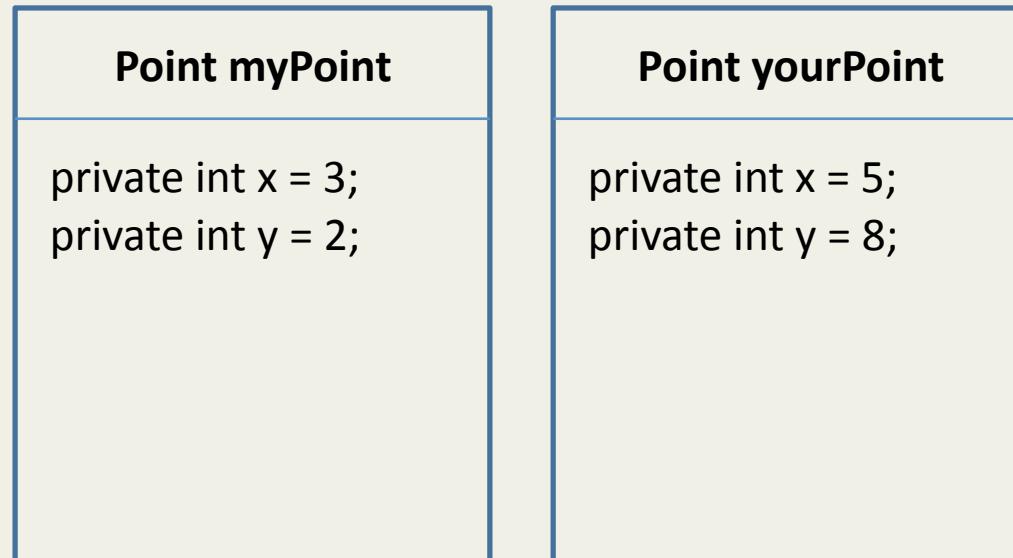


Object



# Classes & Objects

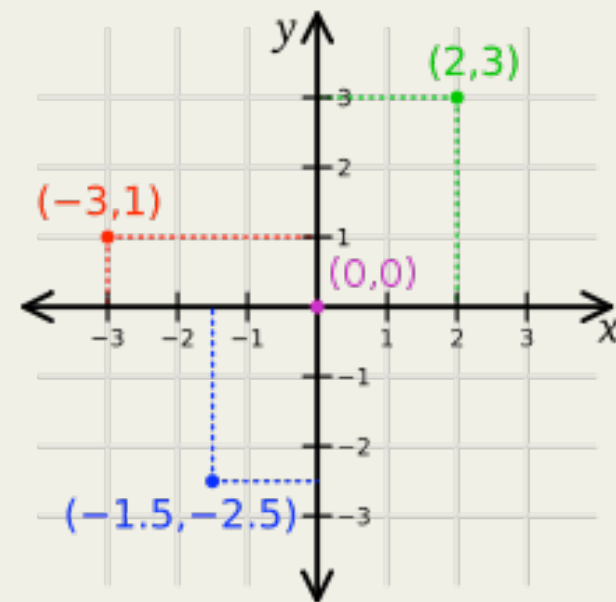
- A particular objects field are its own!
- Changing a field of one object does not affect others. It gets its own copy.



# Making Our Own Objects

- Let's start simple...
  - How about a 2D Cartesian point?

- Points:
  - have an 'x' value
  - have a 'y' value
  - are a certain distance from the origin (0,0)



# Making Our Own Objects

**Constructor Method** – shares the name of the class; immediately called when an object is created. Object ‘sets itself up’.

**Public** – accessible from outside of class.

**Private** – not accessible from outside of class.



# Making Our Own Objects



Now for something a little  
more sophisticated...



# More Complete Programs

- May make use of multiple classes you've created in same directory.
- Functionality of a program may be broken up into many classes which may be re-used in other programs!





# Hangman

- Project consists of two files:
  - HangmanGame: main class, handles interface, instantiates board.
  - HangmanBoard: contains fields and methods relating to the state of the 'hangman game board', including the letters guessed, the secret word, etc.



# Final Thoughts and Questions

