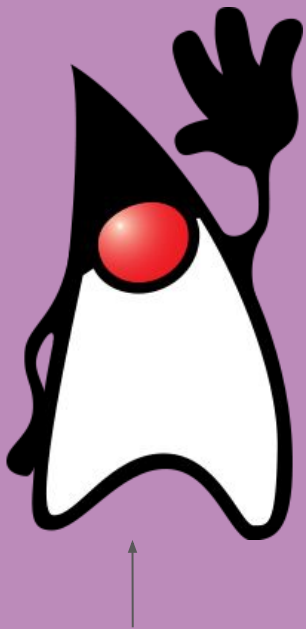




# CMPSC 201: PROGRAMMING LANGUAGES

# An introduction to Java



That's Duke, the mascot.

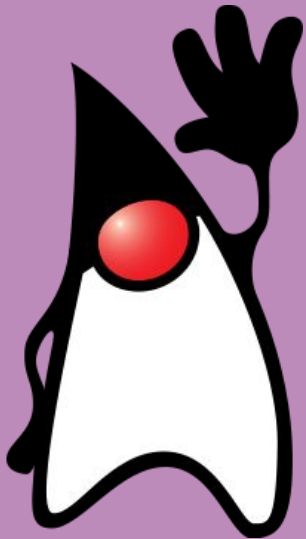
Created in 1991 by Sun Microsystems as an alternative to C++

Developed to enable programs to "run anywhere" once compiled

Necessitated the creation of a special Java system architecture (the JVM, Java Virtual Machine)...

...but a Java program compiled anywhere runs anywhere (that has the Java "runtime")

# An introduction to Java



Developed in 1991 by Sun Microsystems as an alternative to C++

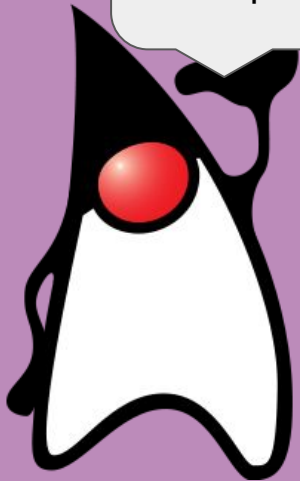
Defined as an “Object-Oriented Language”



```
class {  
    property;  
    method {  
        ...  
    }  
}
```

# What does Object-Oriented mean, anyway?

I'm a wizard.  
I can cast  
spells.

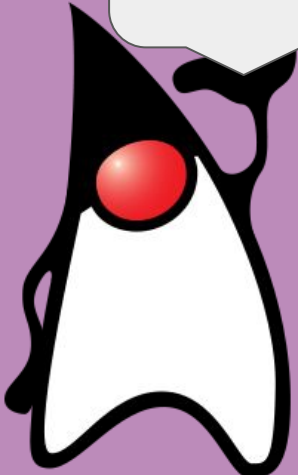


## Wizard.java

Contains the definition of a wizard: says cool stuff, casts spells.

- Method to cast spells
- Method to say things in magical languages with authoritative tone

# What does Object-Oriented mean, anyway?



(Object) (Method)  
Wizard . sayStuff()



## Wizard.java

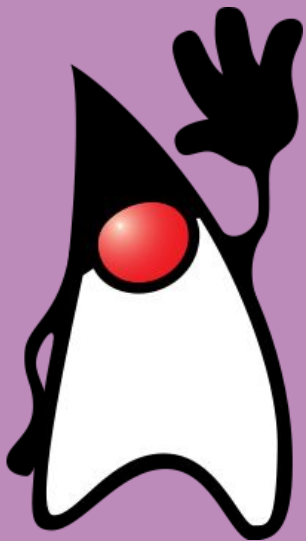
class: Wizard

property: name

methods: castSpell()

sayStuff()

# An introduction to Java



Is “high level” (i.e. more human readable).

Consider:

```
mov r0, #1
```

```
mov r1, #2
```

```
add r0, r0, r1
```

```
int a = 1;
```

```
int b = 2;
```

```
a += b;
```



# Hello, World!

Name of the object

(Java classes must be named the same as the file containing them)

Name of the program

```
package com.golf.one;
```

```
public class Main {
```

Method:

- Name
- Return type
- Visibility

```
    public static void main(String[] args) {
```

```
        System.out.println("Hello, World!");
```

```
    }
```

```
}
```

Functionality



Golfing tip #1: Whitespace doesn't actually matter outside of statements.

# Hello, World!

Other classes can  
"see" this method



Returns...nothing



```
public static void main(String[] args)
```

Programs can call  
this without  
creating new  
instances of a  
"Main" object



Required parameter: a  
string array called  
"args"





# Hello, World!

Braces "scope"  
levels of the  
program

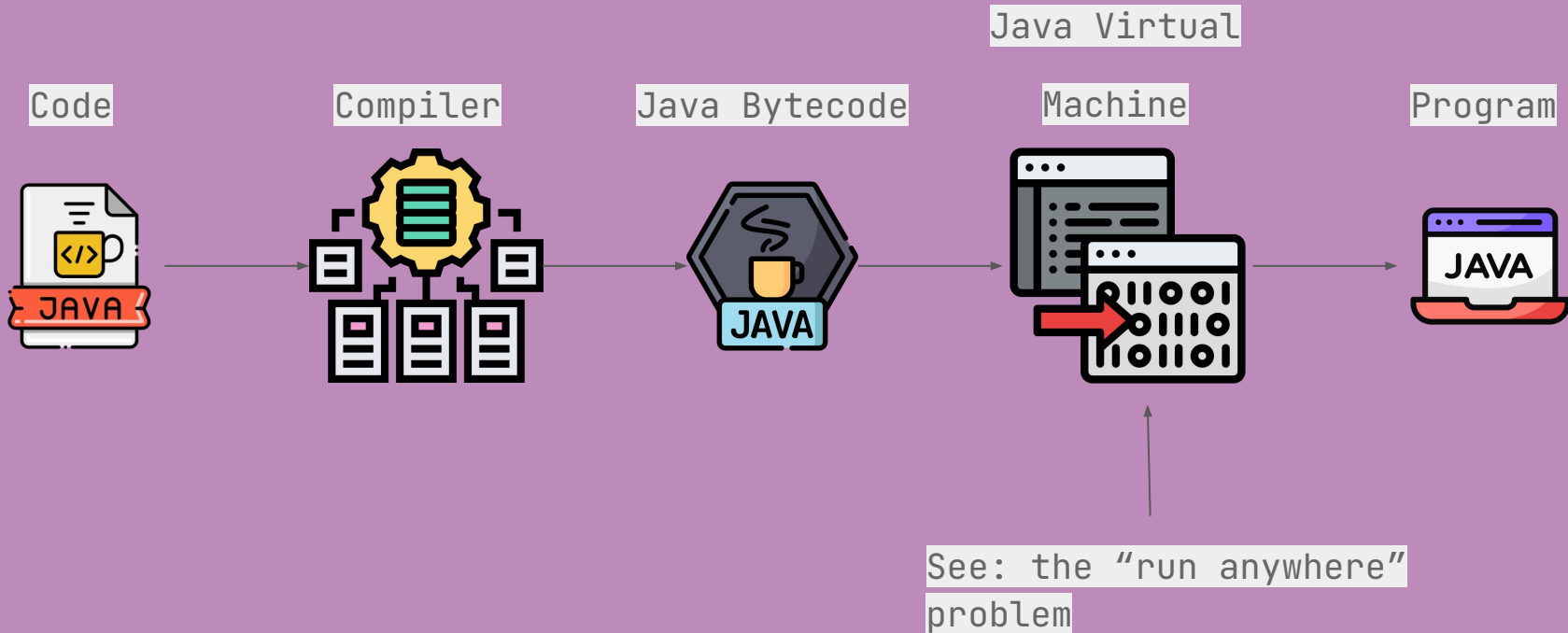


```
public static void main(String[] args) {  
    System.out.println("Hello, World!");  
}
```

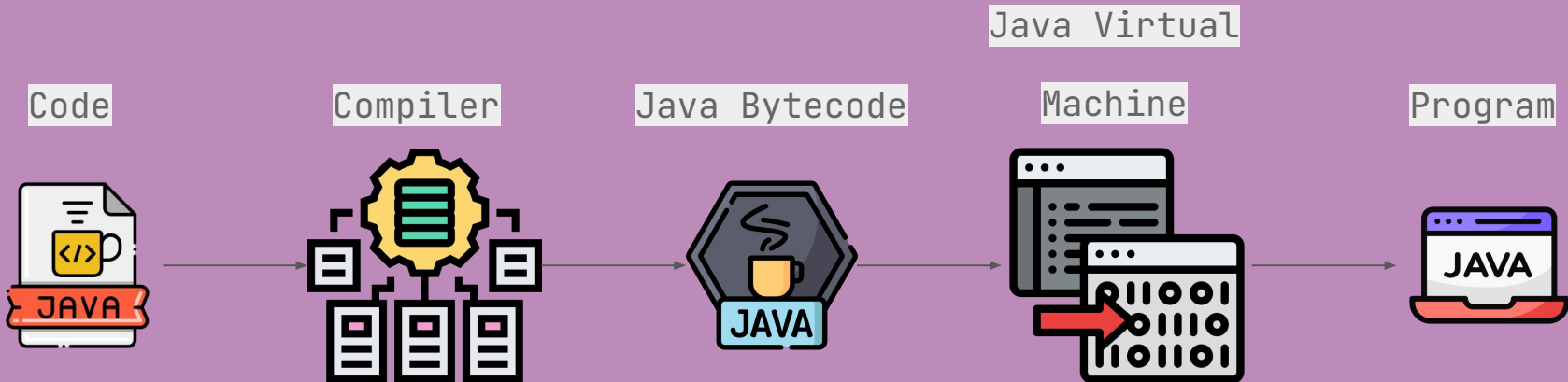
Semicolons end  
statements



# Compiling, running, testing



# Using VSCode and Maven



Apache  
**Maven**

Write  
code

compile

exec:java

# Maven lifecycle

clean

Deletes contents of target/

Cleans up build files

compile

Compiles package into runnable form

Results as ".class" files

exec:java

Runs program

We need to add this as a "Favorite"

test

Runs tests defined in src/test

Our grader uses test to evaluate



Today is  
a good day  
to golf.