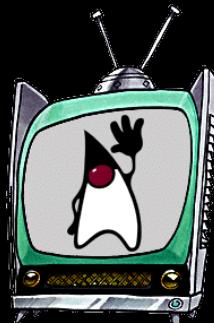


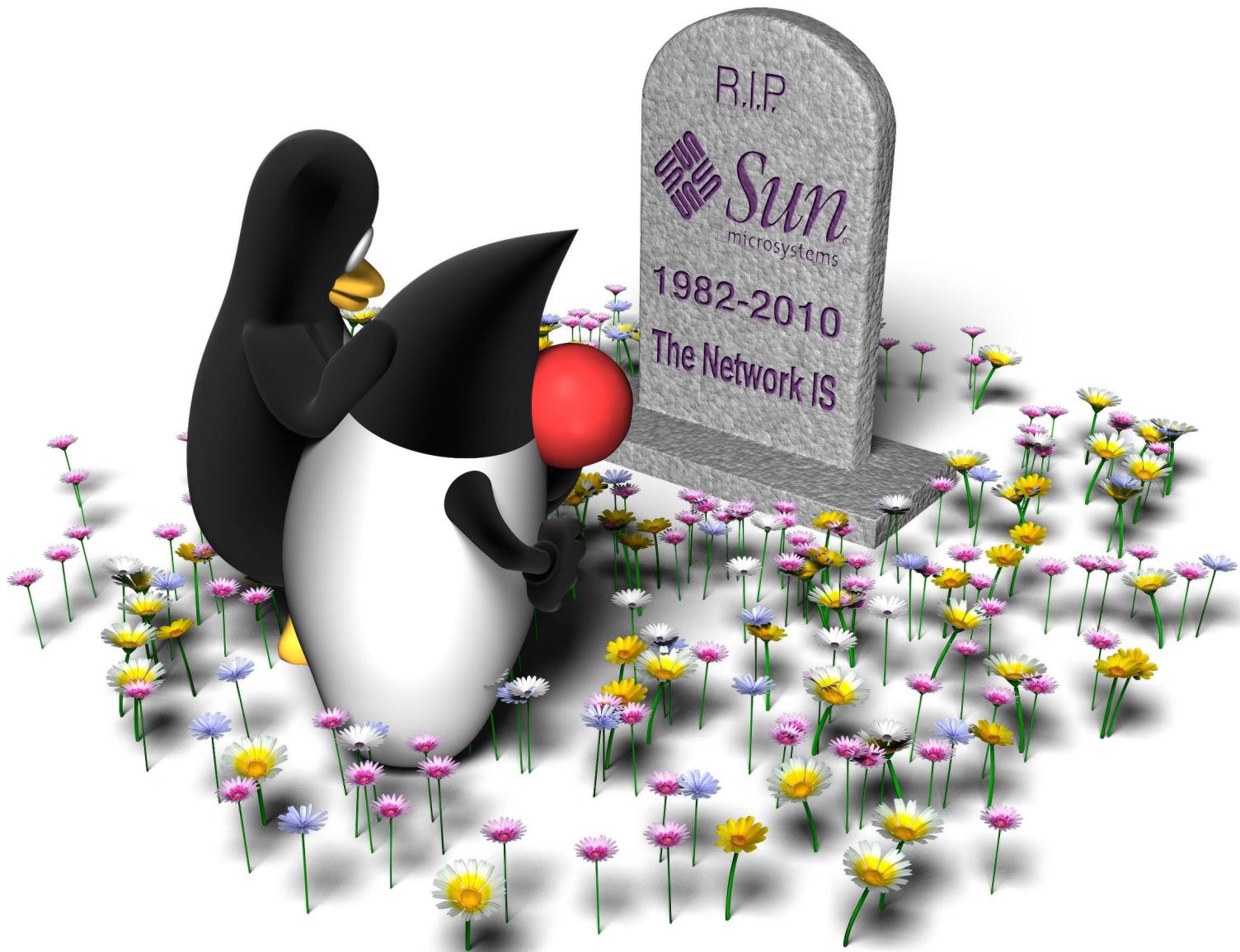
CSC 203 / CPE 203

Project-Based Object-Oriented
Programming and Design
(using Java)



Bill Foote
wffoote@calpoly.edu
(<http://jovial.com>)







WALT DISNEY STUDIOS
HOME ENTERTAINMENT

Google

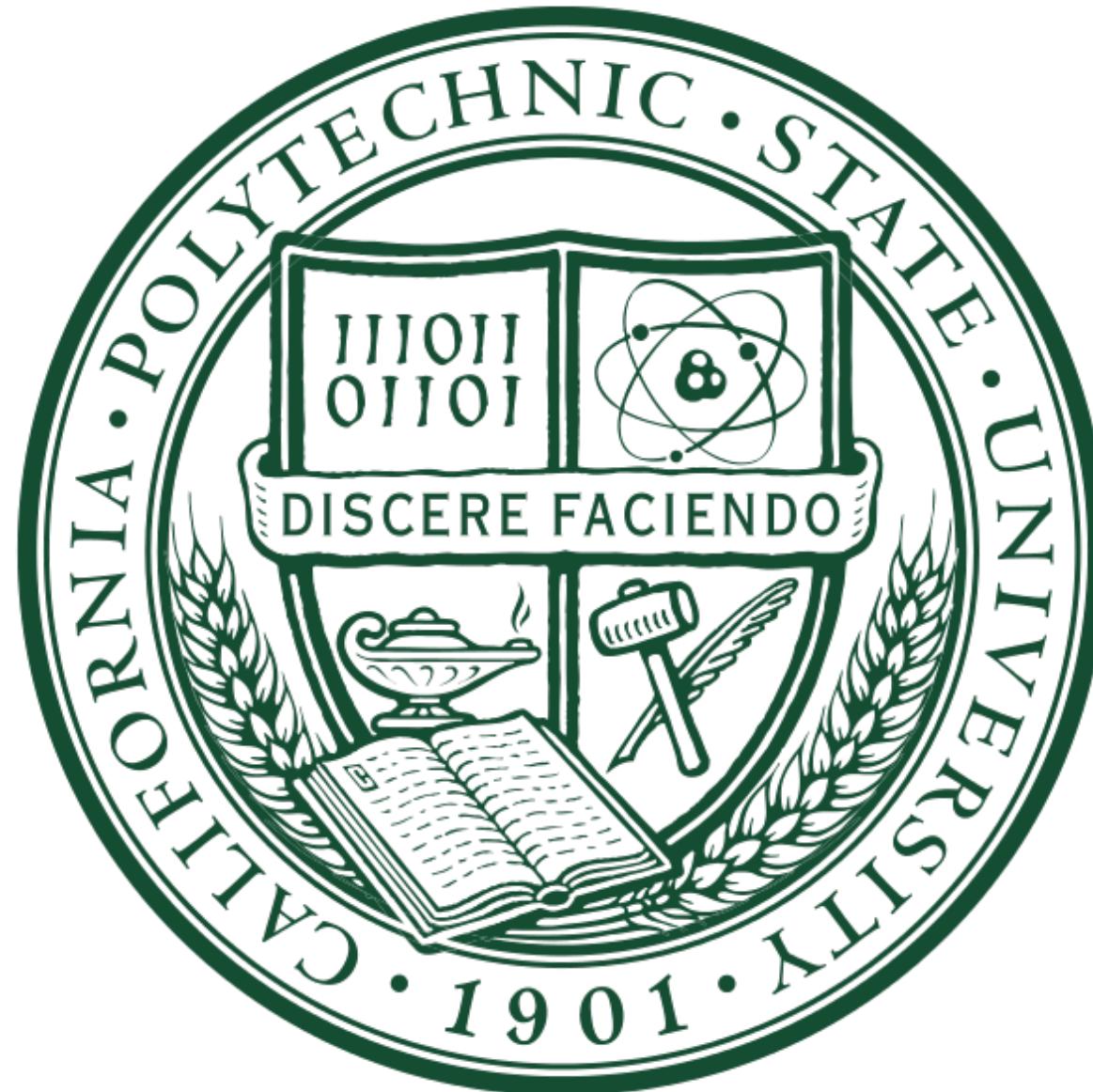
SAMSUNG





M. Soote





Class Objectives

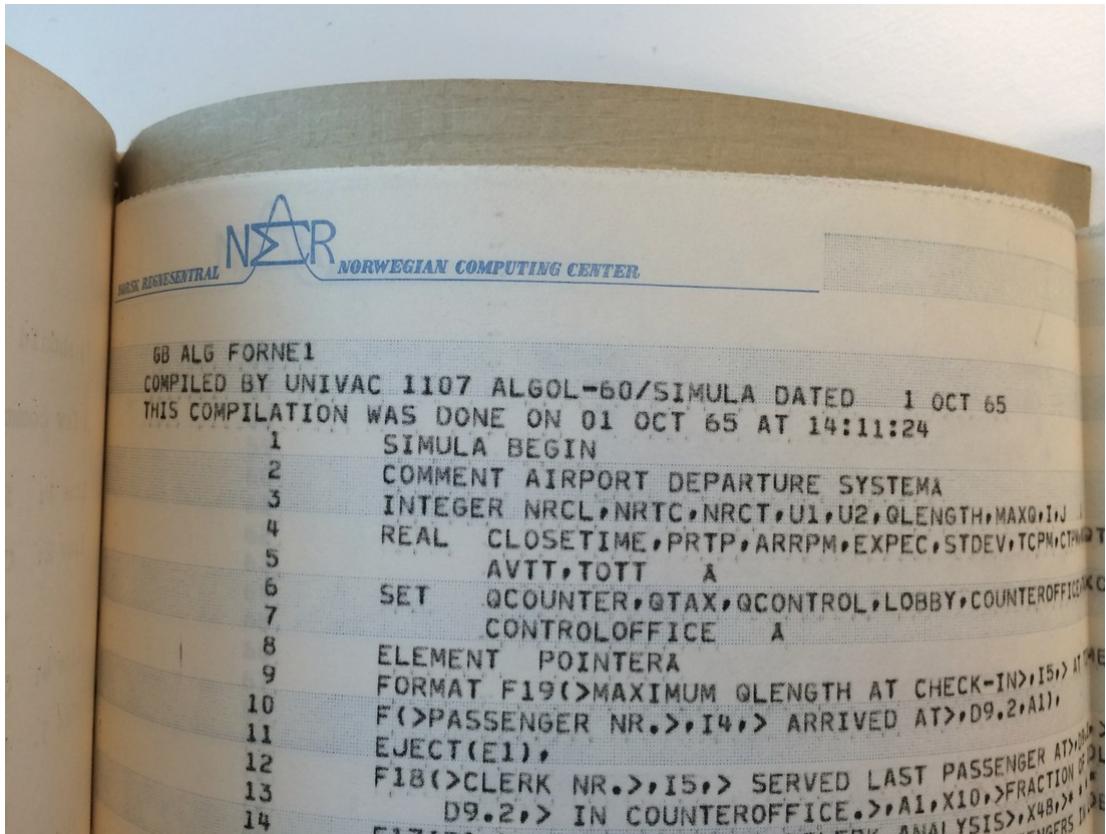


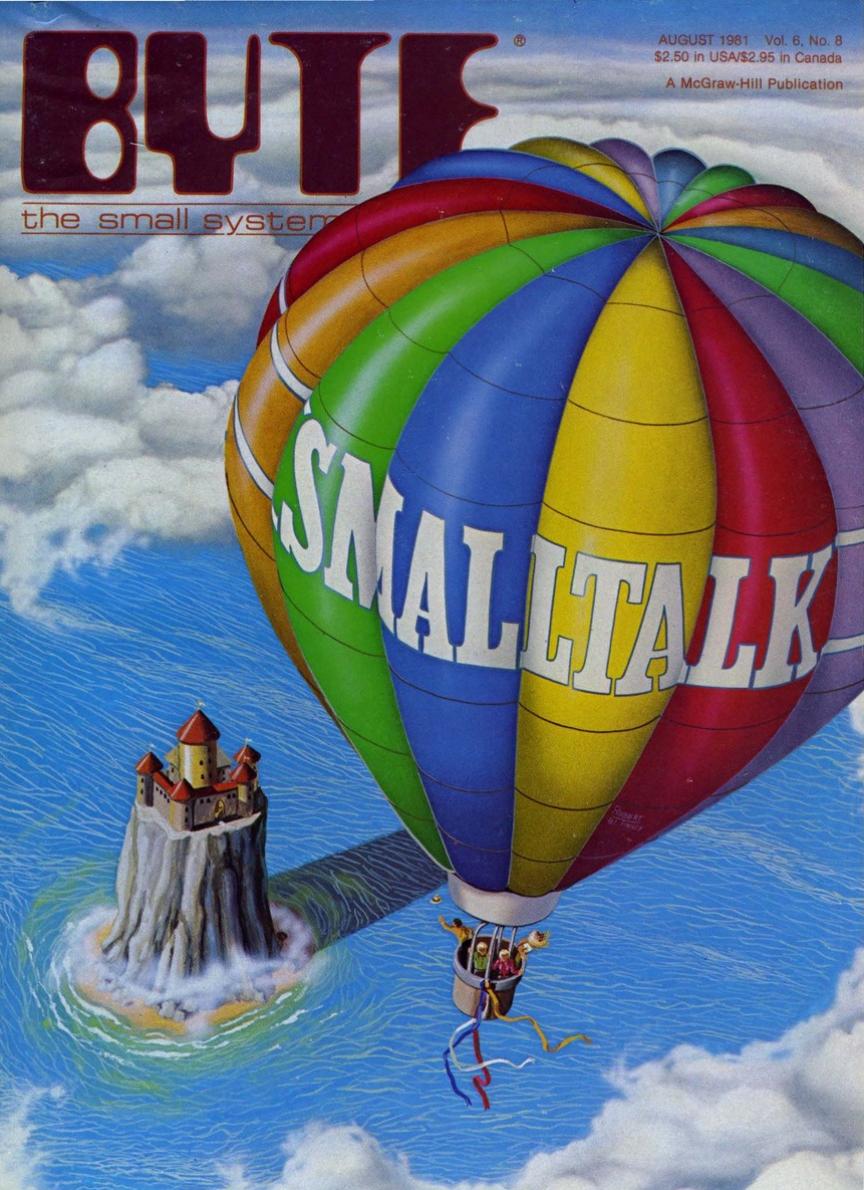
What is a Programming Language?

Object-Oriented Programming

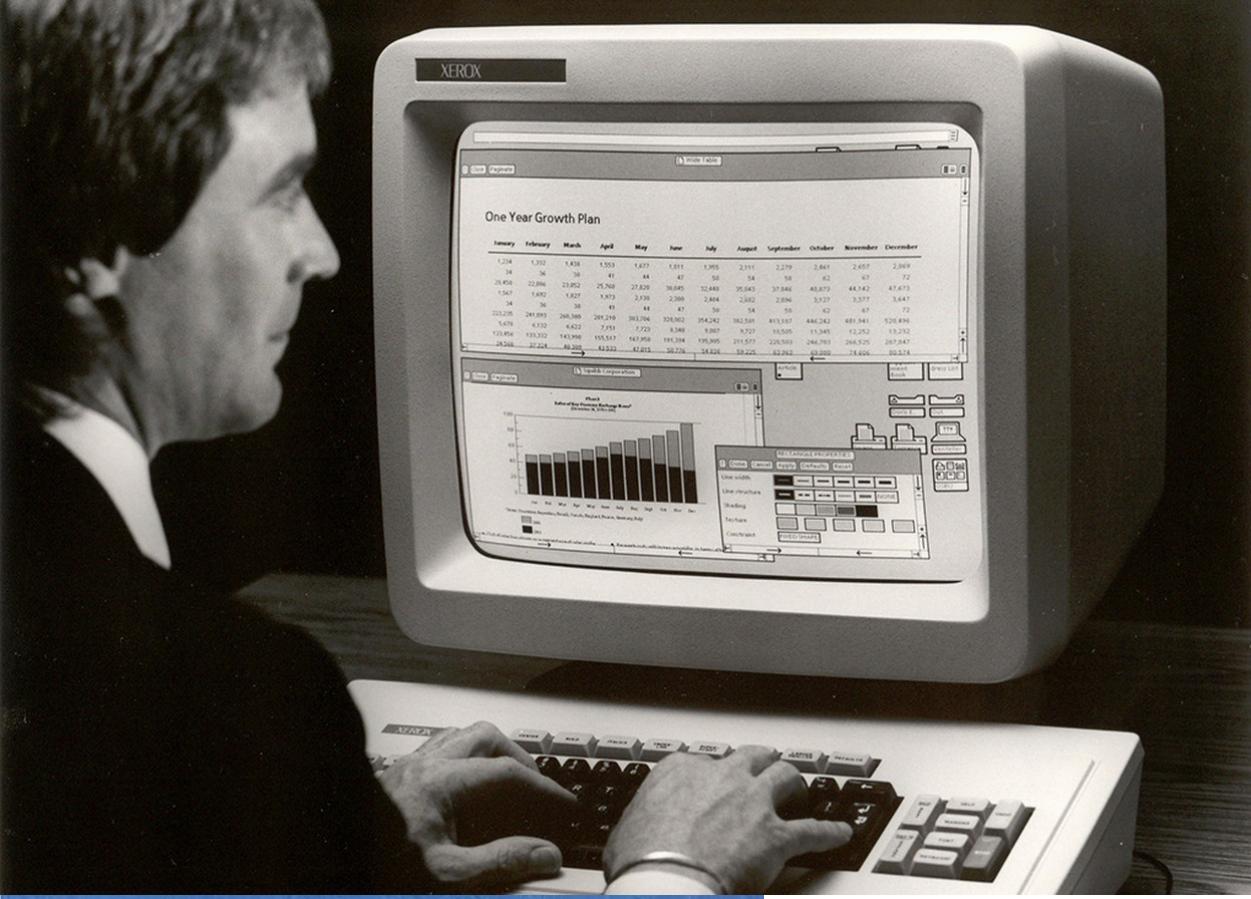


Simula '67
University of Oslo
First OO program:
“Airport Departure System”





AUGUST 1981 Vol. 6, No. 8
\$2.50 in USA/\$2.95 in Canada
A McGraw-Hill Publication



AUGUST 1981 Vol. 6, No. 8
\$2.50 in USA/\$2.95 in Canada
A McGraw-Hill Publication

Xerox PARC
1973

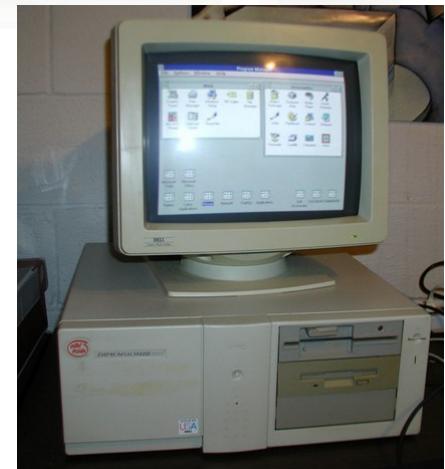


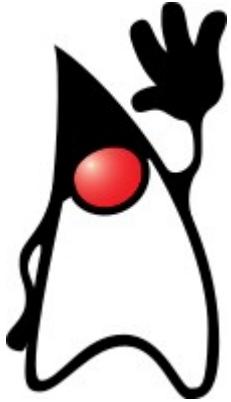
Early 80's: Unix Workstations running X/10,
Apple Macintosh
Microsoft Windows 1.0

1990: Microsoft Windows 3.0

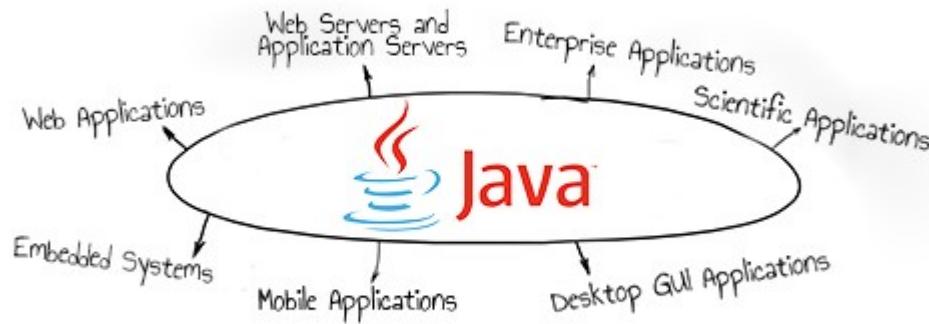


C
C++





1995: Java



- “Smalltalk in C++’s clothing”
- Find errors early, e.g. static typing
- “Write once, run anywhere”

STOP THE TECHNOLOGY MADNESS.

This Java logo was first used with Java 1.0. The Java logo is a registered trademark of Sun Microsystems, Inc. All rights reserved. Sun and Java are trademarks of Sun Microsystems, Inc. and that is why the Java logo™ includes the registered trademark symbol. The right to grant Java the name of an item and the right to use the Java logo™ is held by Sun Microsystems, Inc.



**GIZMOS OF
THE WORLD
UNITE.**

This Java logo is used in the world to identify what is known as technology. It is a way of life that is used to make a living profession. With computing, Java is a technology, and it brings together people all over the world. This logo is used to make the world a better place. It is a reminder to the world that we are connected to each other, and that we are all in this together.



A close-up portrait of James Gosling, an elderly man with white hair and a full white beard, wearing round-rimmed glasses and a light blue striped shirt.

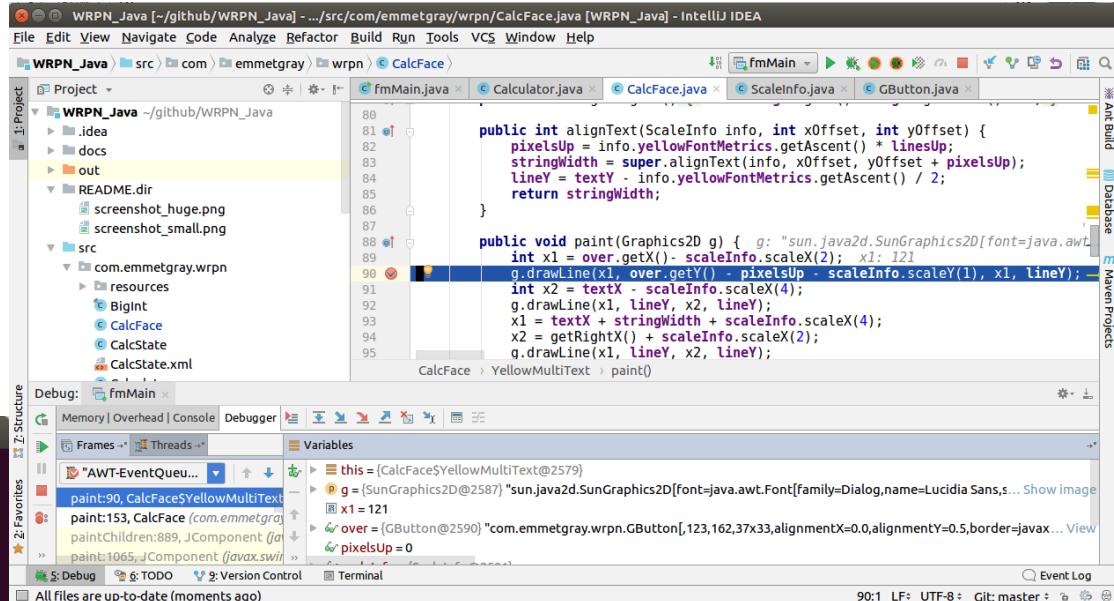
Java is C++ without the guns, knives,
and clubs

— *James Gosling* —

Logistics – Building and Running Programs

IDE or command-line?
GUI, or Terminal?

```
billf@Zathras:~/github/WRPN_Java/src/com/emmetgray/wrpn
44K -rw-rw-r-- 1 billf billf 43K Mar  1 2014 Egray_small.png
4.0K -rw-rw-r-- 1 billf billf 1.5K Feb 29 2012 WRPN_ico.png
16K -rw-rw-r-- 1 billf billf 15K Feb 29 2012 WRPN_small.png
billf@Zathras:~/github/WRPN_Java/src/com/emmetgray/wrpn/resources$ cd ..
billf@Zathras:~/github/WRPN_Java/src/com/emmetgray/wrpn$ ll
total 432K
 68K -rw-rw-r-- 1 billf billf 65K May  4 2012 BigInteger.java
8.0K -rw-rw-r-- 1 billf billf 5.0K Mar 26 17:57 CalcFace.java
24K -rw-rw-r-- 1 billf billf 24K Mar 20 14:45 CalcState.java
4.0K -rw-rw-r-- 1 billf billf 3.5K Mar 20 14:43 CalcState.xml
152K -rw-rw-r-- 1 billf billf 149K Mar 22 15:03 calculator.java
4.0K -rw-rw-r-- 1 billf billf 2.1K Mar 20 14:45 CStack.java
4.0K -rw-rw-r-- 1 billf billf 3.3K Mar 20 14:46 DisplayPacket.java
12K -rw-rw-r-- 1 billf billf 11K Jul 22 2015 fmAbout.form
12K -rw-rw-r-- 1 billf billf 12K Jan  6 2016 fmAbout.java
4.0K -rw-rw-r-- 1 billf billf 3.0K Mar 11 2012 fmBackPanel.form
8.0K -rw-rw-r-- 1 billf billf 4.9K Mar 11 2012 fmBackPanel.java
108K -rw-rw-r-- 1 billf billf 107K Mar 25 09:56 fmMain.java
8.0K -rw-rw-r-- 1 billf billf 6.5K Mar 26 09:21 GButton.java
4.0K -rw-rw-r-- 1 billf billf 1.2K Mar 20 14:46 Register.java
4.0K drwxrwxr-x 2 billf billf 4.0K Mar 21 23:16 resources/
4.0K -rw-rw-r-- 1 billf billf 1.6K Mar 23 13:04 ScaleInfo.java
4.0K -rw-rw-r-- 1 billf billf 599 Mar 20 14:43 WRPNconfig.xml
billf@Zathras:~/github/WRPN_Java/src/com/emmetgray/wrpn$ vi CalcFace.java
```





A few words about class organization

Logistics

USE BOTH!

- Command-line:
 - <http://java.sun.com/>, download “Java SE 9.x” (8 or 10 are OK, too).
- IDE: I’ll be using IntelliJ IDEA
 - <https://www.jetbrains.com/idea/>, free student license. The free community version is fine for this class, too.

Class Organization

- Take written notes!
- Call me! (e-mail works too)
- Github
 - Do the setup exercise: <https://classroom.github.com/a/Lft9B6xp>
 - Make sure you get notified when /general changes!
 - Set up a “watch” – upper-right corner

Code Conventions

- Tabs set to 8
 - violators will spend eternity in The Bad Place
- Indent level set to 4
- Other conventions in general/code_conventions.txt



Settings

Code Style

Editor

General

Auto Import

Appearance

Code Folding

Code Style

Java

ActionScript

CFML

CoffeeScript

CSS

Gherkin

Groovy

GSP

Haml

HTML

JavaScript

JSON

JSP

Editor > Code Style > Java

Scheme: Default (1) IDE

Set from...

Tabs and Indents

Spaces

Wrapping and Braces

Blank Lines

JavaDoc

Imports

Use tab character

Smart tabs

Tab size:

8

Indent:

4

Continuation indent:

8

Keep indents on empty lines

Label indent:

0

Absolute label indent

Do not indent top level class members

Use indents relative to expression start

```
public class Foo {  
    public int[] X = new int[]{1,  
  
    public void foo(boolean a, in  
        label1:  
            do {  
                try {  
                    if (x > 0) {  
                        int someVaria  
                        int anotherVa  
                    } else if (x < 0)  
                        int someVaria  
                        someVariable.  
                    } else {  
                        label2:  
                            for (int i =  
                                }  
                            switch (a) {  
                                case 0:  
                                    doCase0()  
                                    break;  
                                default:  
                                    doDefault
```

OK

Cancel

Apply

Help

Logistics

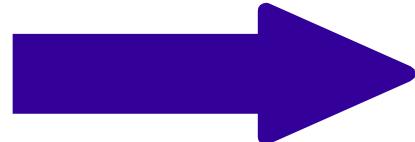
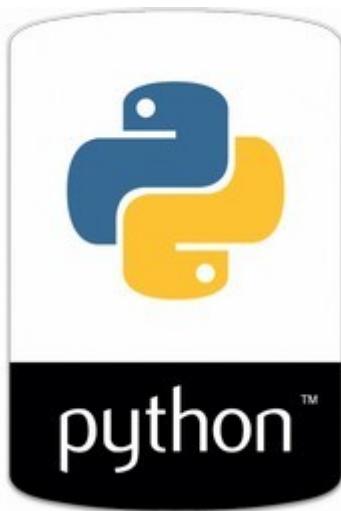
- Revision Control System
 - github
 - <https://classroom.github.com/a/QEhsICLP>

Command-Line set up

- When you download Java:
 - JavaSE. This is also called the JDK, and the Java SDK.
 - Version 8, 9 or 10 is fine, if you already have it installed
- You may need to set these environment variables:
 - JAVA_HOME <place where JDK was installed>
 - PATH \$PATH:<place where JDK was installed>/bin
(Windows spells \$PATH as "%PATH%")
 - You might need to set JDK_HOME and JRE_HOME too, but I don't think this is necessary.
- “Folder” and “directory” mean the same thing.



From Python to Java...



Declaring and initializing variables



```
n = 42
name = "Zaphod Beeblebrox"
m = 99.44
```



```
int n = 42;
String name = "Zaphod Beeblebrox";
float m = 99.44f;
```

Declaring and initializing variables



```
n = 42
name = "Zaphod Beeblebrox"
m = 99.44
```



```
int n = 42;
String name = "Zaphod Beeblebrox";
float m = 99.44f;
```

Declaring Variables



```
int n;  
String name;  
float m;
```

Setting variables to a new value



```
n = 42  
name = "Zaphod Beeblebrox"  
m = 99.44
```



```
n = 42;  
name = "Zaphod Beeblebrox";  
m = 99.44f;
```

Declaring and initializing variables



```
n = 42
name = "Zaphod Beeblebrox"
m = 99.44
```

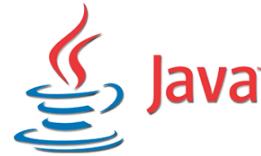


```
int n = 42;
String name = "Zaphod Beeblebrox";
float m = 99.44f;
```

Lists and Arrays



```
ppap = [ 'Pen', 'Pineapple',  
        'Apple', 'Pen' ]
```



```
String[] arr = {  
    "Pen", "Pineapple",  
    "Apple", "Pen"  
};  
  
List<String> list  
= java.util.Arrays.asList(  
    "Pen", "Pineapple",  
    "Apple", "Pen");
```

Lists and Arrays



```
ppap = [ 'Pen', 'Pineapple',  
        'Apple', 'Pen' ]
```



```
String[] arr = new String[4];  
  
arr[0] = "Pen";  
arr[1] = "Pineapple";  
arr[2] = "Apple";  
arr[3] = "Pen";
```

for loops



```
# Iterate 10 times  
  
for n in range(1,11)  
    print('n=' , n)
```



```
// Iterate 10 times  
  
for (int n = 1; n <= 10; n++) {  
    System.out.println("n=" + n);  
}
```

for loops



```
for s in ppap:  
    print(s, end=' ')  
print()
```



```
for (String s : ppapArray) {  
    System.out.print(s + " ");  
}  
System.out.println();  
  
for (String s : ppapList) {  
    System.out.print(s + " ");  
}  
System.out.println();
```

for loops



```
for i in range(0, len(ppap)):
    print(s[i], end=' ')
print()
```



```
for (int n = 0; n < arr.length; n++)
{
    System.out.print(arr[n] + " ");
}
System.out.println();

for (int n=0; n < list.size(); n++) {
    System.out.print(list.get(n)
                    + " ");
}
System.out.println();
```

Start-up code



```
if __name__ == "__main__":
    print("Hello World")
```



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

Running a Program



```
python3 foo.py
```



```
javac MyClass.java
```

```
java MyClass
```

Defining and calling a function



```
def shout(s):
    print(s + '!')
    return 7

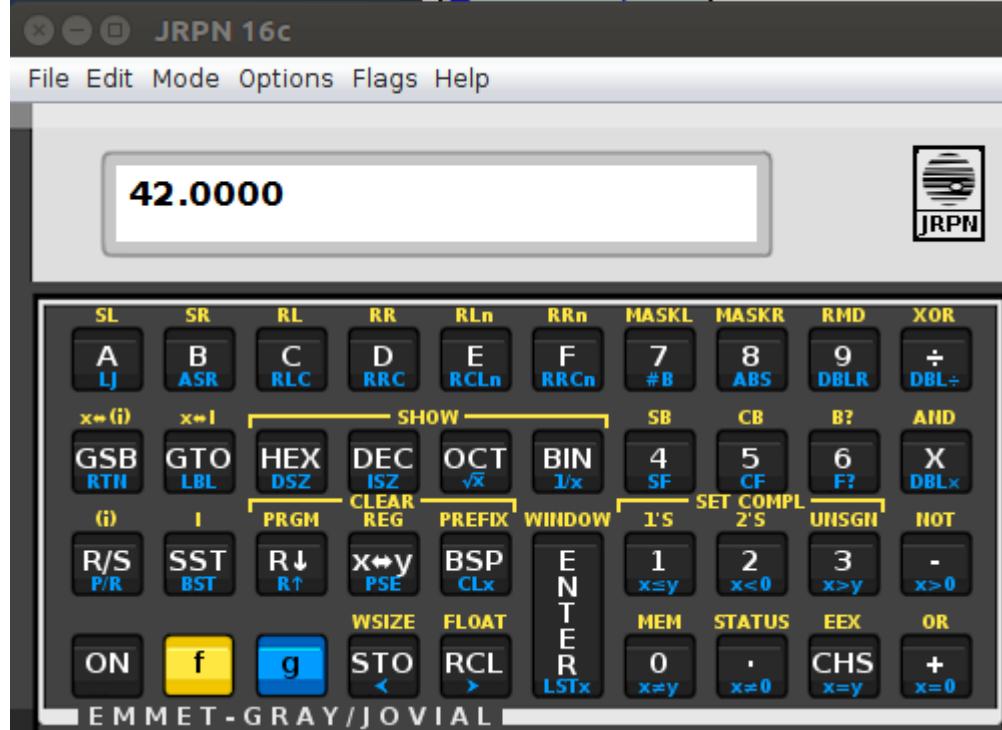
x=shout('Huzzah')
print(x)
```

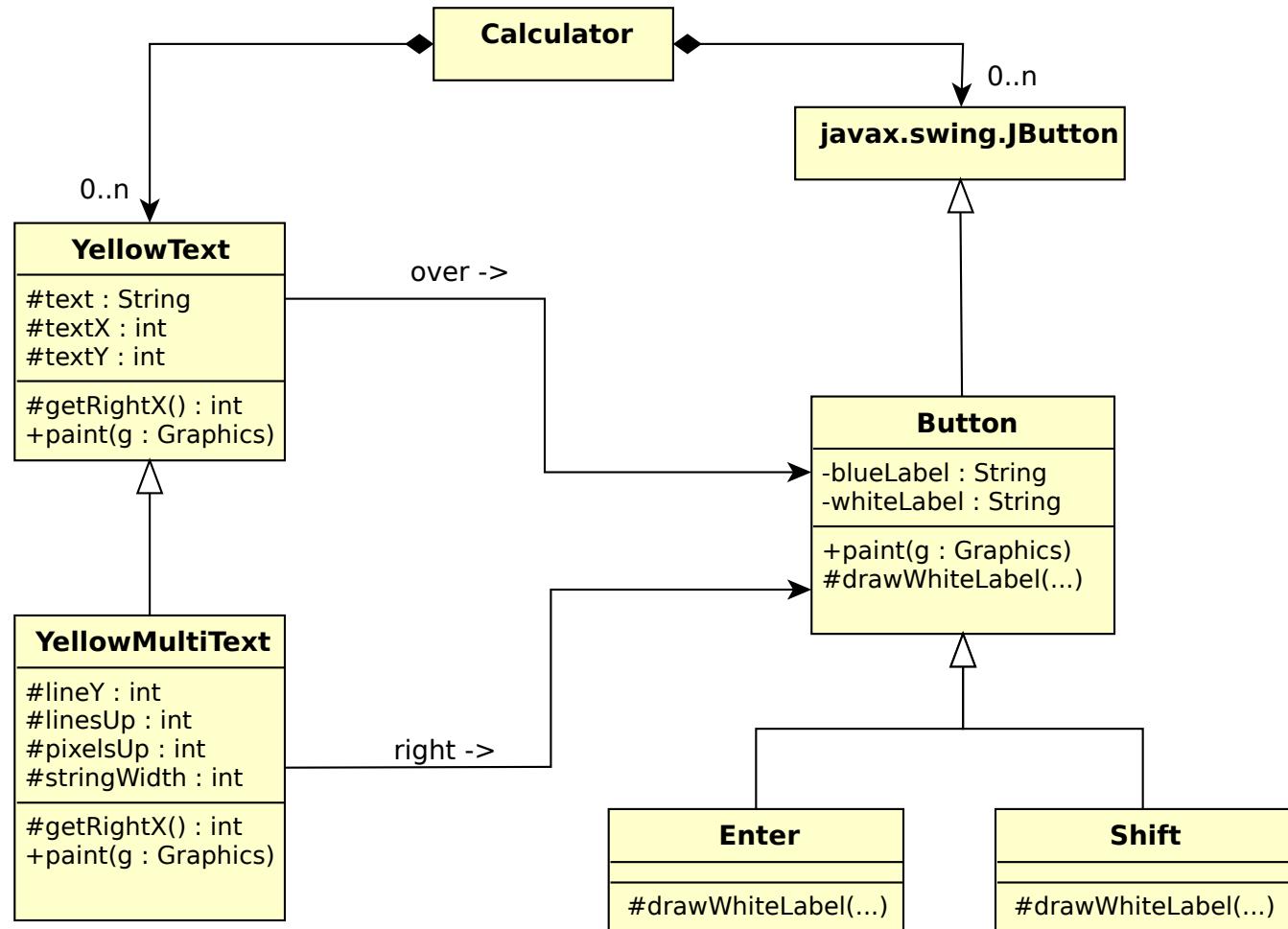


```
public class MyClass {
    public static int shout(String s)
    {
        System.out.println(s+"!");
        return 7;
    }
    public static void
    main(String[] args) {
        int x = shout("Huzzah");
        System.out.println(x);
    }
}
```

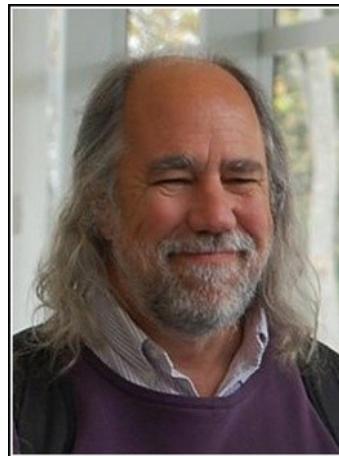
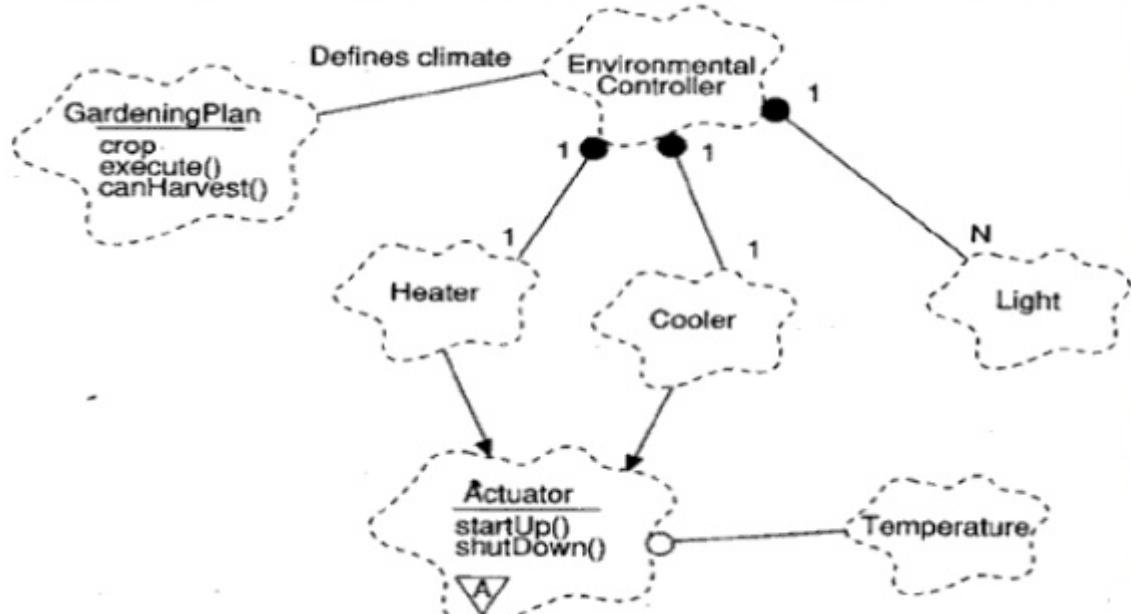
OOP:

An *Object-Oriented* Calculator!





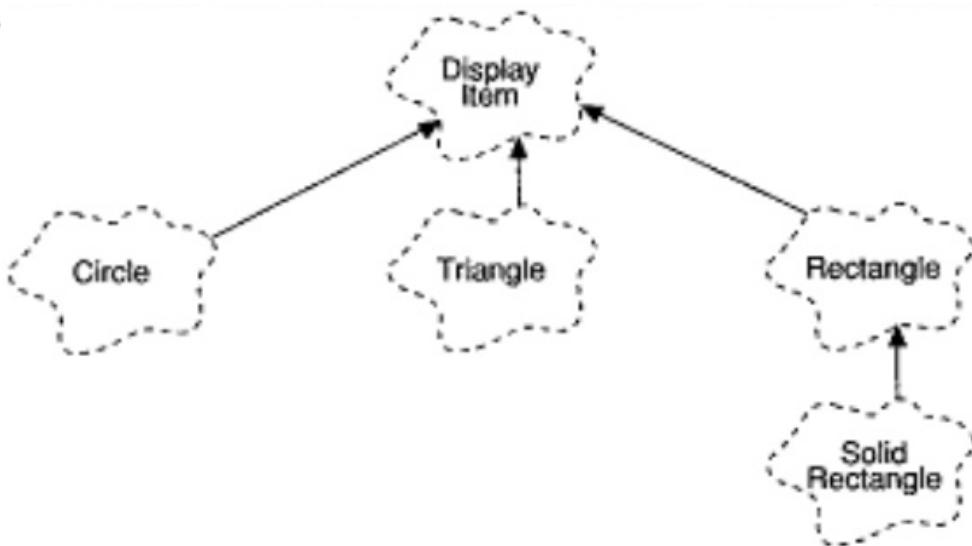
**UNIFIED
MODELING
LANGUAGE**™



The function of good software is to make the complex appear to be simple.

— Grady Booch —

AZ QUOTES

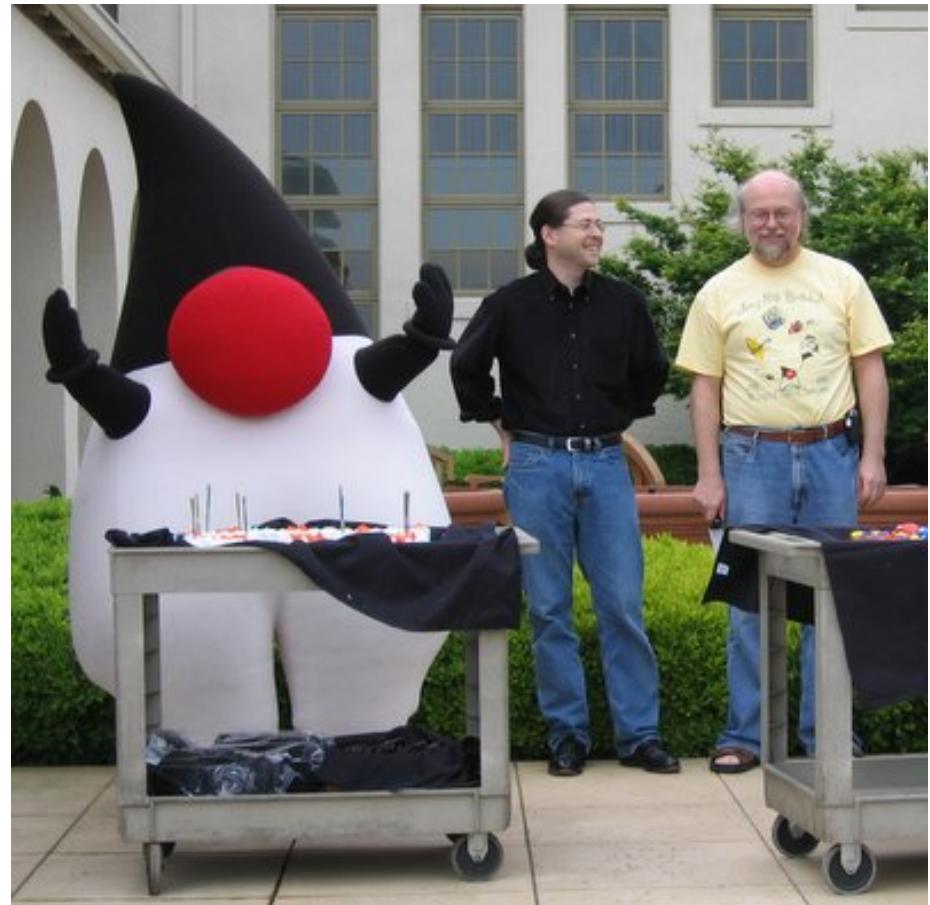


GRIN and bear it

- A real-world design document that uses UML:

[http://jovial.com/hdcookbook_repo/javadocs/grin/javame/com/hdc
ookbook/grin/doc-files/index.html](http://jovial.com/hdcookbook_repo/javadocs/grin/javame/com/hdc
ookbook/grin/doc-files/index.html)

A bit more about Java...



Non-object data types

byte	byte b = 0x7f;	b = (byte) 0xff;
short	short s = 32767;	
int	42	
long	27L	
char	'c', '\t', '\n'	
boolean	true, false	
float	3.7f	
double	42.0	

Constants

```
public final static int MY_HAPPY_CONSTANT = 777;  
public final static String MY_NAME = "Krusty";
```

Other stuff...

- Check out section 3.5 of the book, “Operators”
- Garbage collection
- if/else, while, do/while, switch, break
 - (section 3.8)
- final

Objects, like Strings

```
String s = null;  
System.out.println(s);  
s = "hello";  
System.out.println(s);
```

Objects, like Strings

```
String s = null;  
System.out.println(s);  
s = "hello";  
System.out.println(s);
```

“411” is an object. Is 411 an object?

Objects, like Strings

```
String s = null;  
System.out.println(s);  
s = "hello";  
System.out.println(s);
```

```
billf@Zathras:~/tmp/tmp$ java foo  
hello  
null
```

`==` vs. `.equals()`

```
3 == 3  
3 == 47
```

```
"hello" == "hello"  
String s = getH() + "ello";      // getH returns "h"  
"hello" == s  
"hello".equals(s)  
s.equals("hello")
```

```
"null".equals(null)  
s = null  
s.equals("null")  
s.equals(null)
```