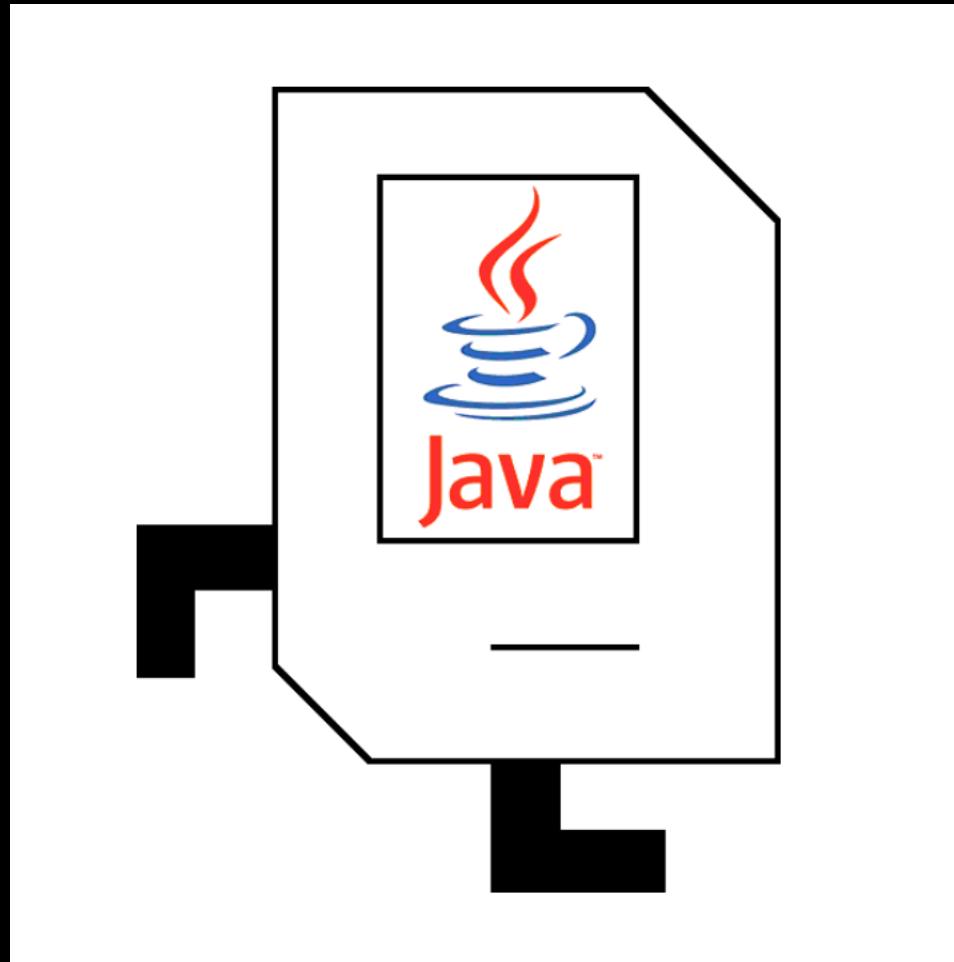


Variables

Review

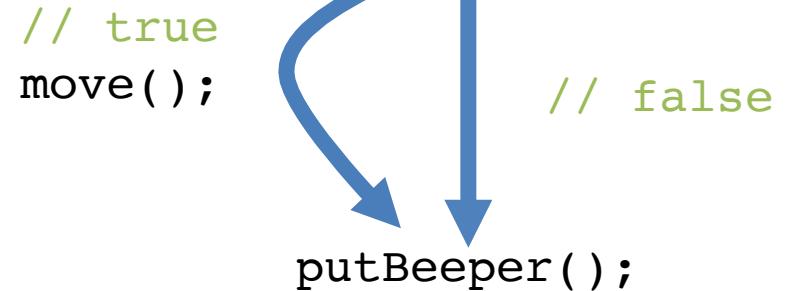


Control Statements

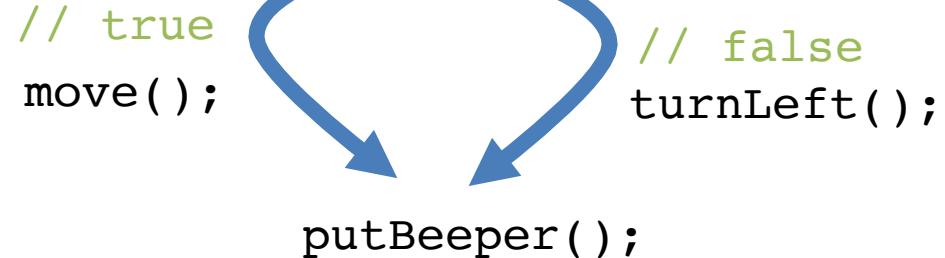
```
private void solaDon() {  
    turnLeft();  
}  
  
for(int i = 0; i < N; i++) {  
    // to repeat N times  
}  
  
while(condition) {  
    // all the code in here repeats  
    // while the condition is true  
}  
  
if(condition) {  
    // do this code if true  
} else {  
    // do this code if false  
}
```

If-else statements

```
if(frontIsClear()) {  
    move();  
}  
putBeeper();
```



```
if(frontIsClear()) {  
    move();  
} else {  
    turnLeft();  
}  
putBeeper();
```



Brackets and Indentation

```
public void run() {
    while (beepersInBag()) {
        findTree();
        addLeavesToTree();
    }
    moveToWall();
}
private void addLeavesToTree() {
    turnLeft();
    climbTree();
        addLeaves();
    descendToGround();
    turnLeft();

private void findTree() {
    moveToWall();
}
```

Brackets and Indentation

```
public void run() {
    while (beepersInBag()) {
        findTree();
        addLeavesToTree();
    }
    moveToWall();
}

private void addLeavesToTree() {
    turnLeft();
    climbTree();
    addLeaves();
    descendToGround();
    turnLeft();
}

private void findTree() {
    moveToWall();
}
```

Brackets and Indentation

```
public void run() {
    while (beepersInBag()) {
        findTree();
        addLeavesToTree();
    }
    moveToWall();
}

private void addLeavesToTree() {
    turnLeft();
    climbTree();
    addLeaves();
    descendToGround();
    turnLeft();
}

private void findTree() {
    moveToWall();
}
```

Brackets and Indentation

```
public void run() {  
    while (beepersInBag()) {  
        findTree();  
        addLeavesToTree();  
    }  
    moveToWall();  
}
```

```
private void addLeavesToTree() {  
    turnLeft();  
    climbTree();  
    addLeaves();  
    descendToGround();  
    turnLeft();  
}
```

```
private void findTree() {  
    moveToWall();  
}
```

Brackets and Indentation

```
public void run() {  
    while (beepersInBag()) {  
        findTree();  
        addLeavesToTree();  
    }  
    moveToWall();  
}
```

```
private void addLeavesToTree() {  
    turnLeft();  
    climbTree();  
    addLeaves();  
    descendToGround();  
    turnLeft();  
}
```

```
private void findTree() {  
    moveToWall();  
}
```

Brackets and Indentation

```
public void run() {  
    while (beepersInBag()) {  
        findTree();  
        addLeavesToTree();  
    }  
    moveToWall();  
}  
  
private void addLeavesToTree() {  
    turnLeft();  
    climbTree();  
    addLeaves();  
    descendToGround();  
    turnLeft();  
}  
  
private void findTree() {  
    moveToWall();  
}
```

Brackets and Indentation

```
public void run() {  
    while (beepersInBag()) {  
        findTree();  
        addLeavesToTree();  
    }  
    moveToWall();  
}
```

```
private void addLeavesToTree() {  
    turnLeft();  
    climbTree();  
    addLeaves();  
    descendToGround();  
    turnLeft();  
}
```

```
private void findTree() {  
    moveToWall();  
}
```

Decomposition - CollectNewspaper

Good

```
public void run(){
    exitHouse();
    pickUpPaper();
    returnHome();
}

private void exitHouse() {
    move();
    move();
    turnRight();
    move();
    turnLeft();
    move();
}

private void pickUpPaper() {
    pickBeeper();
}

private void returnHome() {
    turnAround();
    move();
    move();
    move();
    turnRight();
    move();
}
```

Bad

```
public void run(){
    move();
    move();
    turnLeft();
    turnLeft();
    turnLeft();
    move();
    turnLeft();
    move();
    pickBeeper();
    turnLeft();
    turnLeft();
    move();
    move();
    move();
    turnLeft();
    turnLeft();
    turnLeft();
    move();
}
```

Use methods to decompose your code.

Good or Bad?

Good

```
// moves karel one step
private void git() {
    move();
}

// rotates karel
private void solaDon() {
    turnLeft();
}
```

Bad

```
private void sey1() {
    move();
}
private void sey2() {
    turnLeft();
}
private void sey3() {
    putBeeper();
}
```

Use meaningful names
for your methods.

Match your curly braces

```
1 import stanford.karel.*;
2
3 public class Place100 extends SuperKarel {
4
5     public void run() {
6         move();
7         for(int i = 0; i < 100; i++) {
8             putBeeper();
9         }
10        move();
11
12 }
```

Coding Style

??

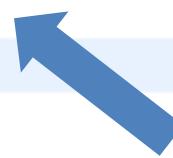


```
1 import stanford.karel.*;
2
3 public class Place100 extends SuperKarel {
4
5     public void run() {
6         move();
7         for(int i = 0; i < 100; i++) {
8             putBeeper();
9         }
10        move();
11    }
12}
```



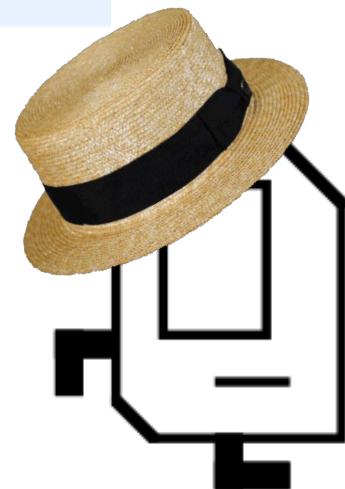
??

??



Coding Style

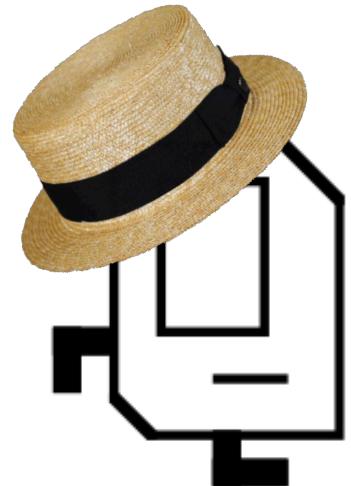
```
1 import stanford.karel.*;
2
3 public class Place100 extends SuperKarel {
4
5     public void run() {
6         move();
7         for(int i = 0; i < 100; i++) {
8             putBeeper();
9         }
10        move();
11    }
12}
```



Humans Read Code, Too

Decompose by writing
meaningful methods

Indent and match your
curly braces



Programming takes practice.

See You Later!



I will miss you.

Enjoy Java!

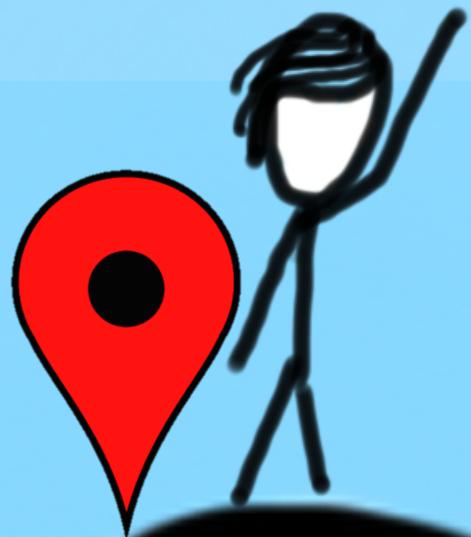
Call me maybe?

Java



Today's Goals

1. How do I write a console program?
2. What are variables and how do I use them?
3. How do I get user input in a console program?



Console Program



Takes text input

Prints text output

A screenshot of a Mac OS X terminal window. The window title bar shows a red, yellow, and green button, followed by the path '~ — -bash — 80x24'. The main window area contains the text 'Last login: Mon Jul 3 17:20:41 on ttys007' and '/Users/Lisa \$'. A blue cursor bar is visible at the bottom right of the window.

First Console Program: Hello World

```
import acm.program.*;

public class HelloProgram extends ConsoleProgram {
    public void run() {
        println("hello, world");
    }
}
```



In Pop Culture

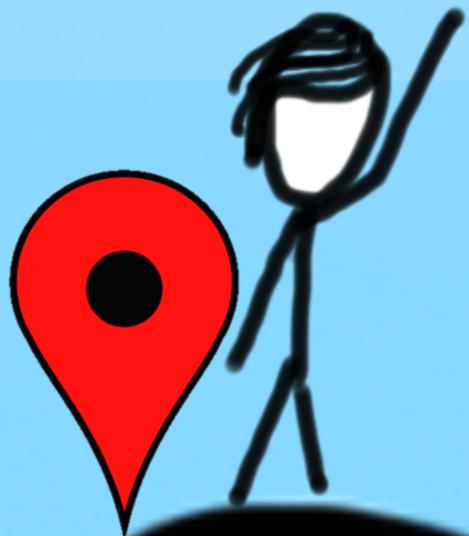


You had me at
"Hello, world"

Today's Goals



1. How do I write a console program?
2. What are variables and how do I use them?
3. How do I get user input in a console program?



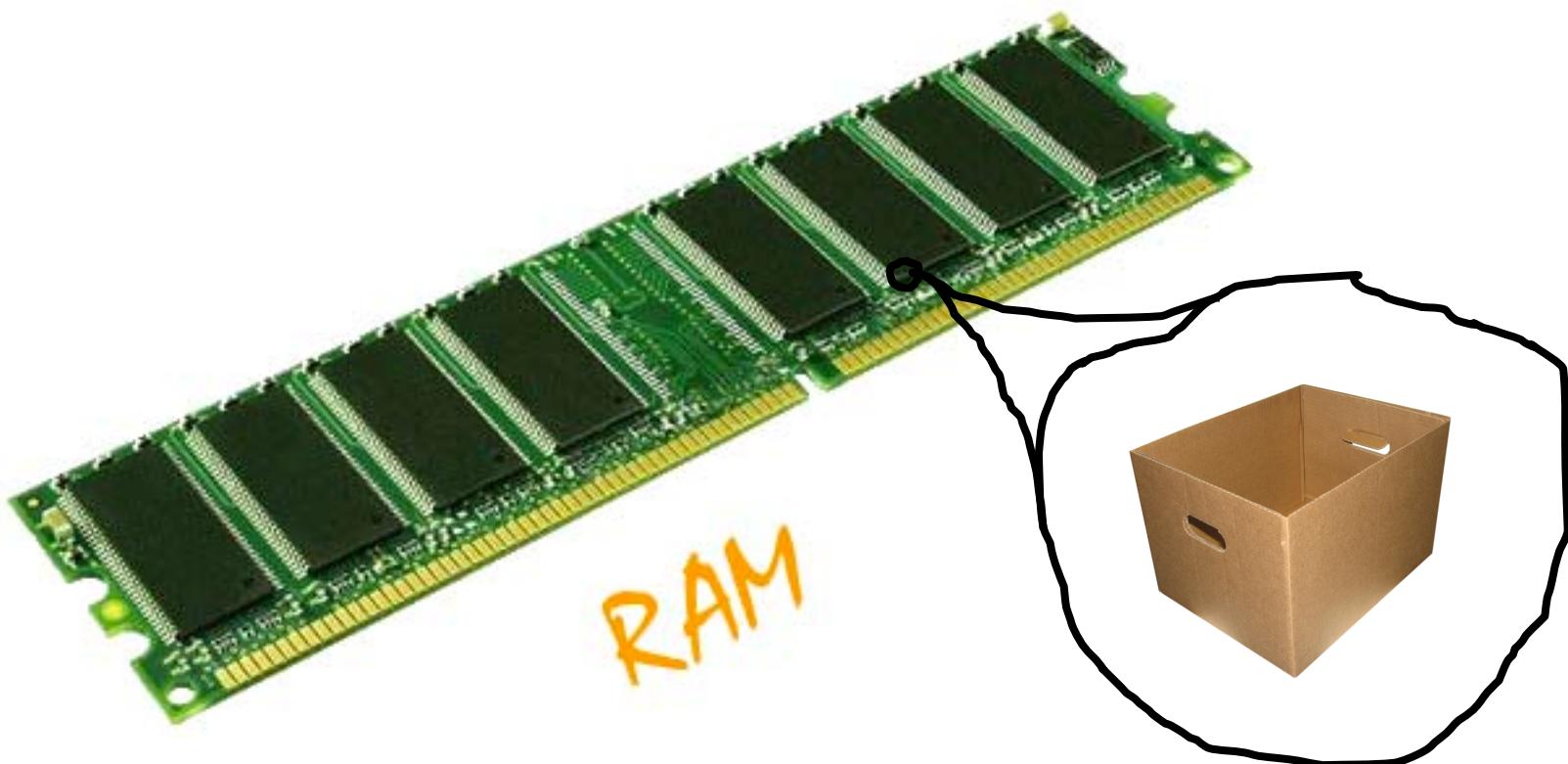
What is a variable?

[suspense]

Variables are Like Boxes



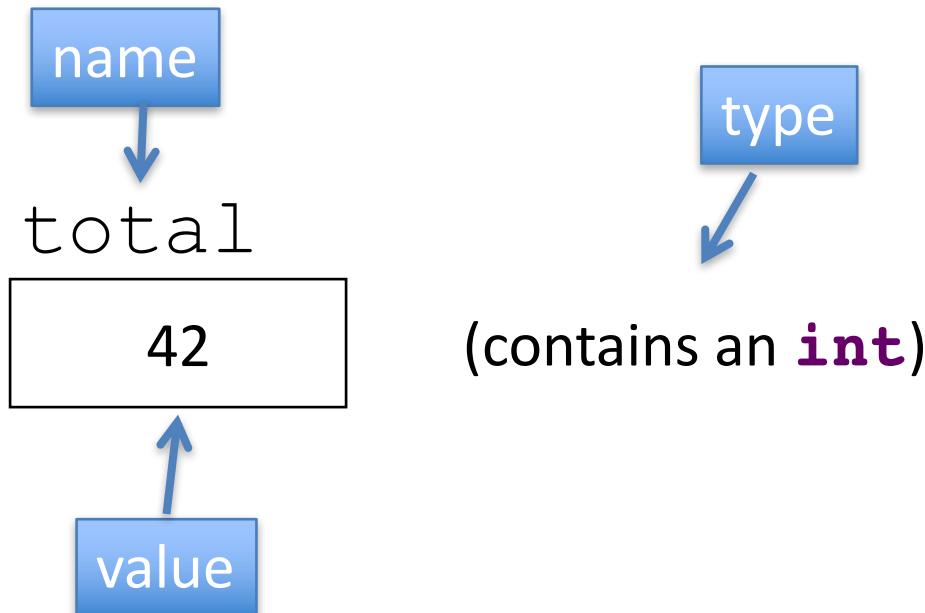
Teeny Tiny Boxes



My computer has space for about 64 billion boxes

Variables are Like Boxes

```
int total = 42;
```



Types

```
// integer values  
int num = 5;
```

```
// real values  
double fraction = 0.2;
```

```
// letters  
char letter = 'c';
```

```
// true or false  
boolean isLove = true;
```

Double: How Much Do I Weigh?



* Answers could be real valued numbers

Int: How Many Children Do I Have?



* It is weird to say something like 1.7

Binary Operators

+	Addition	*	Multiplication
-	Subtraction	/	Division
		%	Remainder

Binary Operators

```
double width = 2.5; // meters  
double height = 3.0;  
double area = width * height;
```

name	width	height	area
value	2.5	3.0	7.5
type	double	double	double

Today's Goals

- ✓ 1. How do I write a console program?
- ✓ 2. What are variables and how do I use them?
- 3. How do I get user input in a console program?



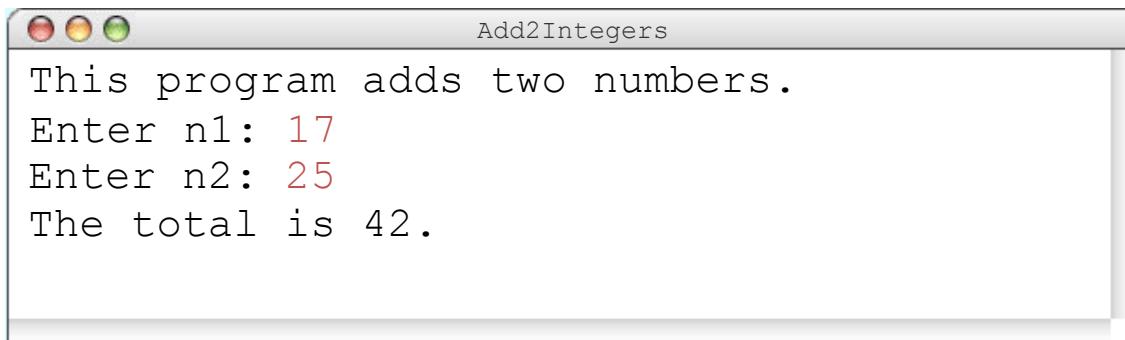
User Input

```
int a = readInt("Give me an int!");  
  
double b = readDouble("And a double");
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	42

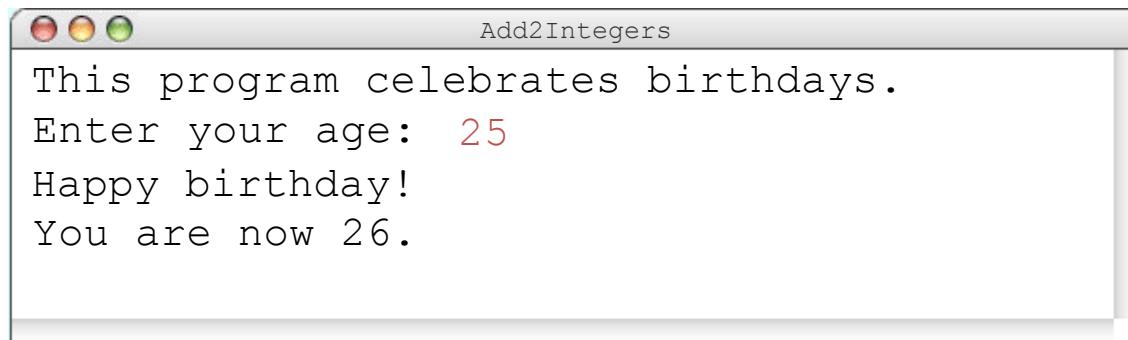


Birthday

```
public class Birthday extends ConsoleProgram {  
    public void run() {  
        println("This program celebrates birthdays.");  
        int age = readInt("Enter your age: ");  
        age = age + 1;  
        println("Happy birthday!");  
        println("You are now " + age + ".");  
    }  
}
```

age

25

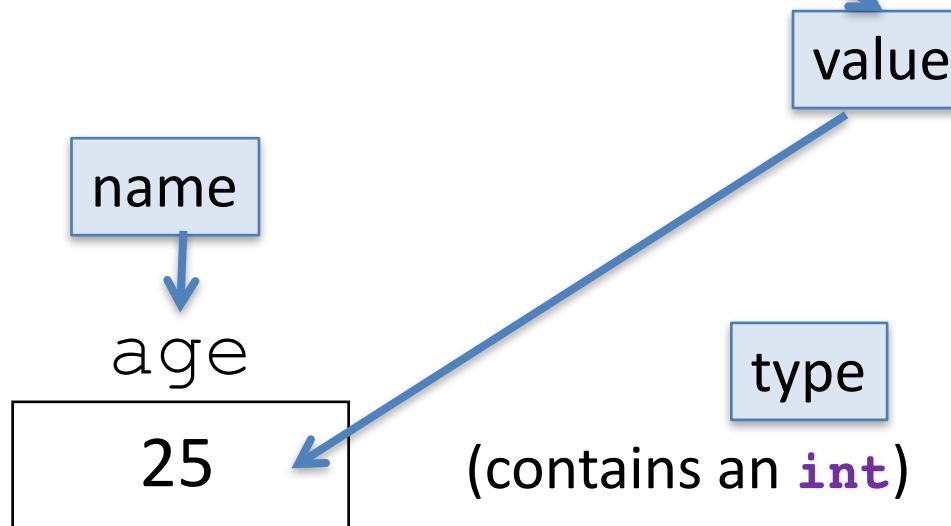


Birthday

```
int age = readInt("Enter your age: " );  
age = age + 1;
```

Birthday

```
int age = readInt("Enter your age: ");
```



Birthday

```
int age = readInt("Enter your age: ");
```

```
age = age + 1;
```



value

age

25

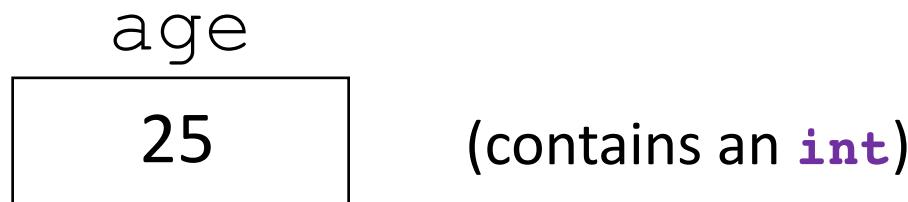
(contains an **int**)

Birthday

```
int age = readInt("Enter your age: ");
```

```
age = age + 1;
```

$$25 + 1 = 26$$



Birthday

```
int age = readInt("Enter your age: ");
```

```
age = age + 1;
```



26

age



(contains an **int**)

Birthday

```
int age = readInt("Enter your age: ");
```

```
age = age + 1;
```



26

age



(contains an **int**)

What do you think this does?

```
println(1 / 2);
```

AHHHHHHHH!!!!!!

```
println(1 / 2);
```

Resulting Type

int + int results in an **int**

double + double results in a **double**

int + double results in a **double**

* The general rule is: operations always return the most expressive type

Pitfalls of Integer Division

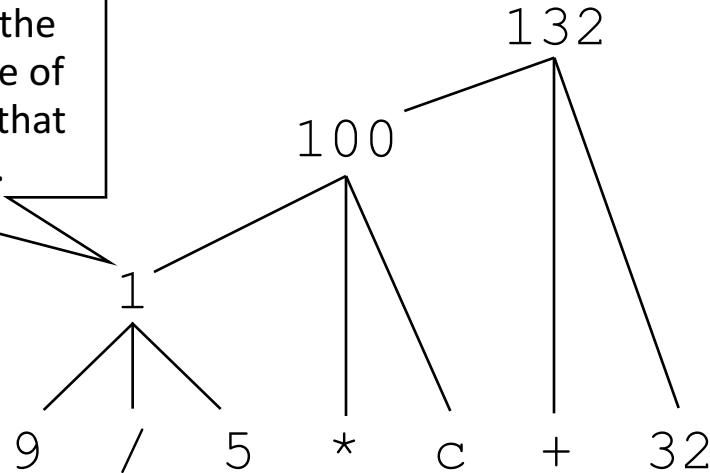
Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;  
double f = 9 / 5 * c + 32;
```



The computation consists of evaluating the following expression:

The problem arises from the fact that both 9 and 5 are of type `int`, which means that the result is also an `int`.

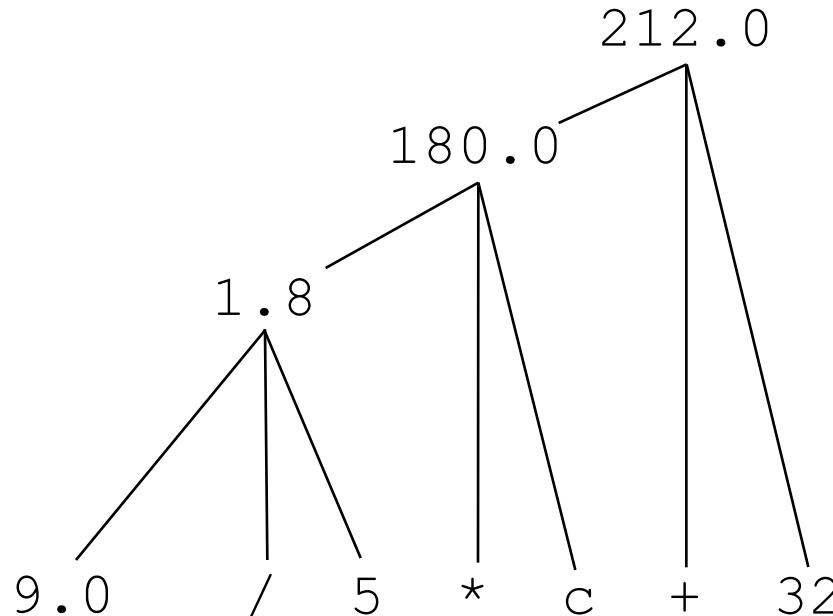


Pitfalls of Integer Division

You can fix this problem by converting the fraction to a double, either by inserting decimal points or by using a type cast:

```
double c = 100;  
double f = 9.0 / 5 * c + 32;
```

The computation now looks like this:



Conditions

Conditions

<	Less Than	==	Equal To
>	Greater Than	>=	More or Equal
		<=	Less or Equal

Demo

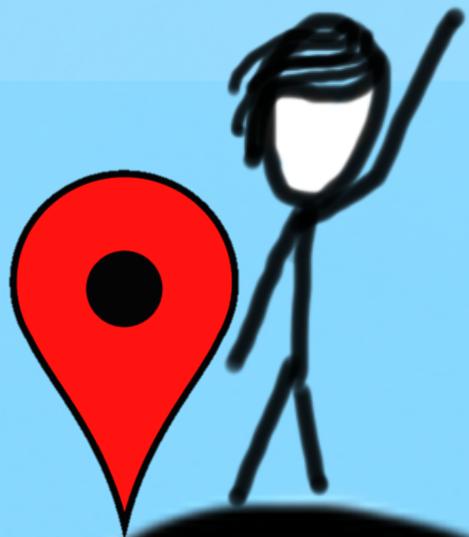
Conditions

```
public void run() {  
    println("This program tells you if you are cool or not.");  
    int numberOfSongs = readInt("How many Tarkan songs can you name? ");  
  
    if (numberOfSongs == 1) {  
        println("You only know Yolla!");  
    }  
  
    if (numberOfSongs > 3) {  
        println("Okay, you are pretty cool.");  
    } else {  
        println("Wow, you aren't very cool.");  
    }  
}
```

Today's Goals



1. How do I write a console program?
2. What are variables and how do I use them?
3. How do I get user input in a console program?



Sandcastles



Website

The screenshot shows a dark-themed website interface. At the top, there is a navigation bar with the following items: "CS Bridge", "Handouts ▾", "Projects ▾", "Examples ▾", "Slides ▾", and a dropdown menu for "Slides" containing the options "Karel", "ControlFlow", and "Variables". Below the navigation bar, there are two circular icons: one red icon on the left and the Stanford University seal on the right. To the right of the seal, the text "Intro to Comp" is visible, likely part of a larger title. A large blue arrow points upwards from the bottom right towards the "Variables" option in the dropdown menu.

- CS Bridge
- Handouts ▾
- Projects ▾
- Examples ▾
- Slides ▾

- Karel
- ControlFlow
- Variables

Intro to Comp