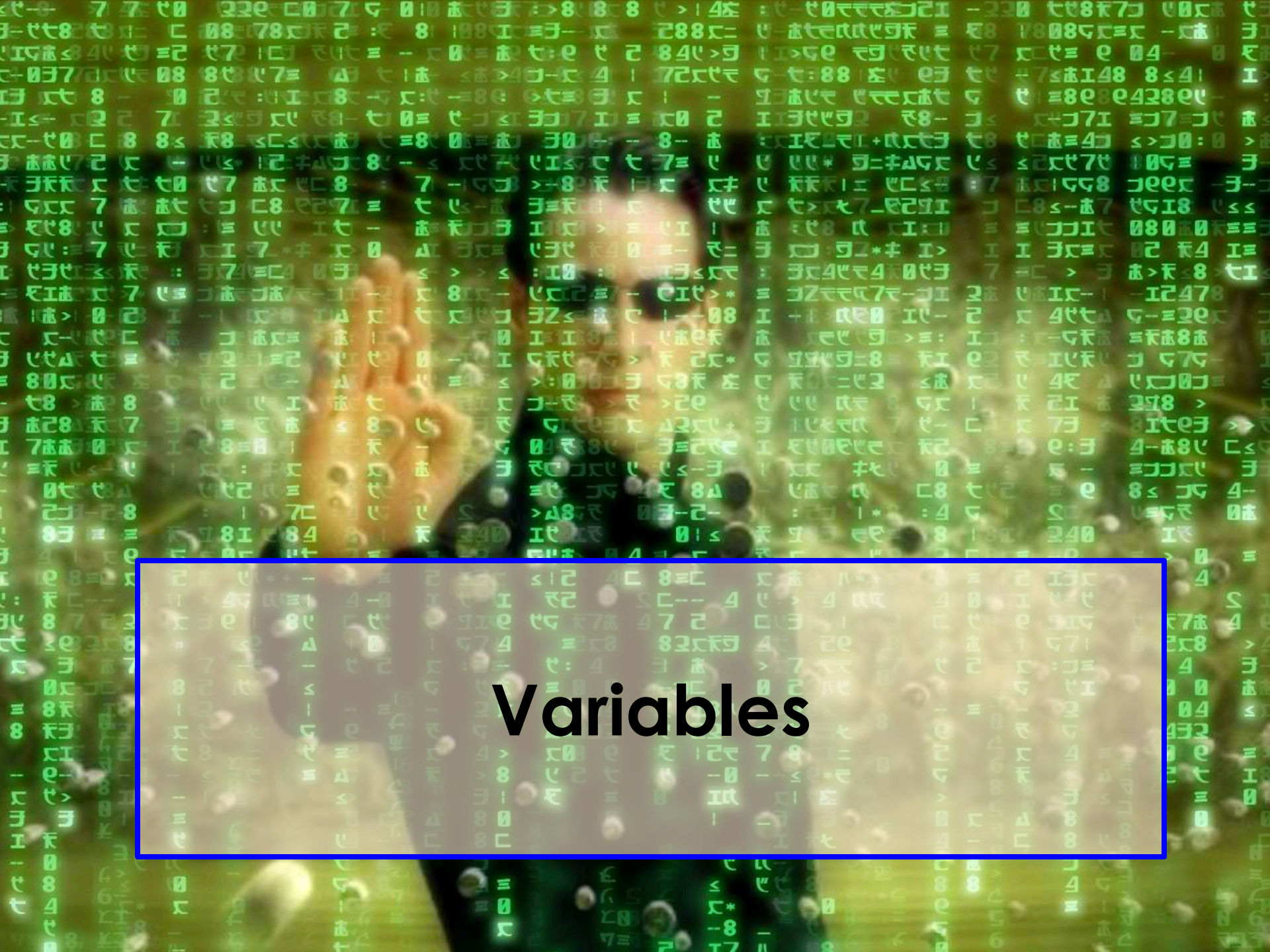
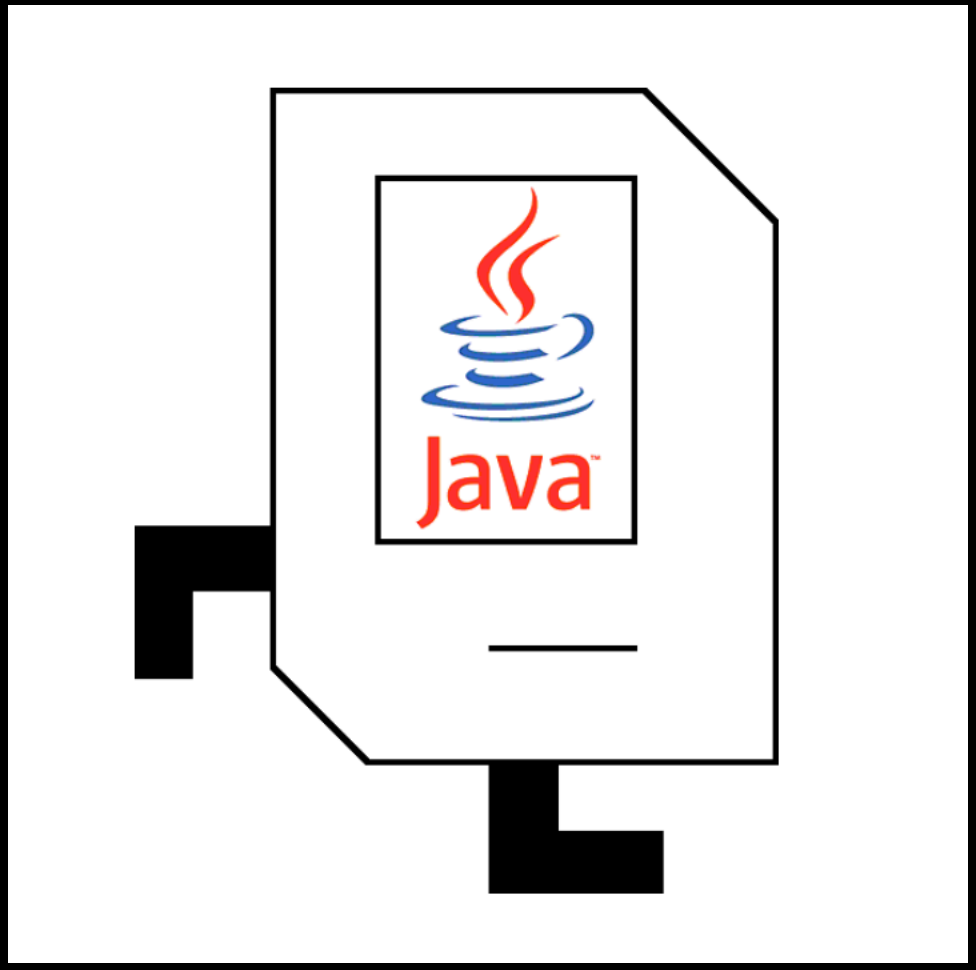

Variables

# Review



# Control Statements

method

```
private void solaDon() {
    // three turnLeft()'s
}
```

for-loop

```
for(int i = 0; i < N; i++) {
    // to repeat N times
}
```

while-loop

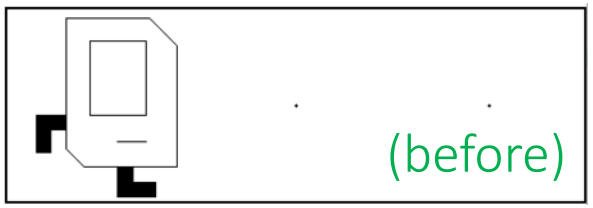
```
while(condition) {
    // all the code in here repeats
    // while the condition is true
}
```

if-else statements

```
if(condition) {
    // do this code if true
} else {
    // do this code if false
}
```

# If-else statements

What do these two code snippets do?

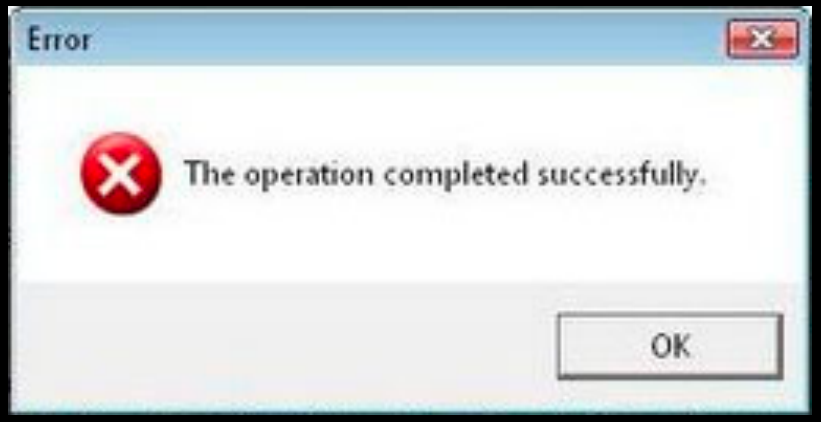



✅

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


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


# Errors




What kind of bugs did you find in your code?

🤔

# Semicolons and Curly Braces { }

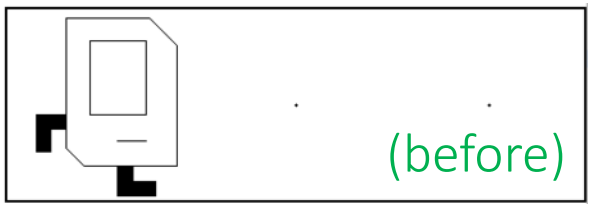
;

This line is a command.

{ … }

These lines are grouped.

What do these code snippets do?




```
while (frontIsClear()) {
   move();
}
```

(done)

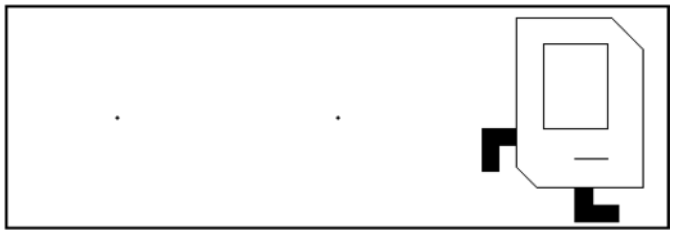
```
while (frontIsClear()); {
   move();
}
```

(running)

# Semicolons ; and Curly Braces { }

```
while (frontIsClear()) {
   move();
}
```

```
while (frontIsClear())
{
   move();
}
```



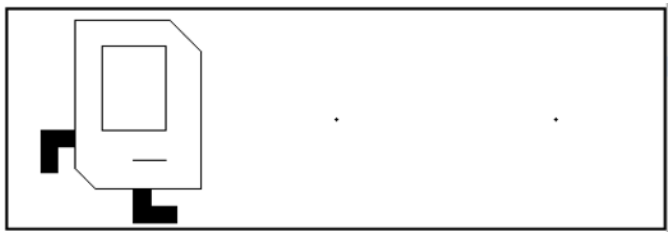
(done)

```
while (frontIsClear()); {
   move();
}
```

```
while (frontIsClear())
{
}
{
   move();
}
```

"do nothing"

We never reach here!



(running)

# Make It a Habit

```
for (int i = 0; i < N; i++) {
  move();
}
```

No semicolon between ( ) and { }

```
while (frontIsClear()) {
  move();
}
```

No semicolon between ( ) and { }

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

No semicolon between ( ) and { }

```
move();
```

A command

Questions?

# Asking for Help

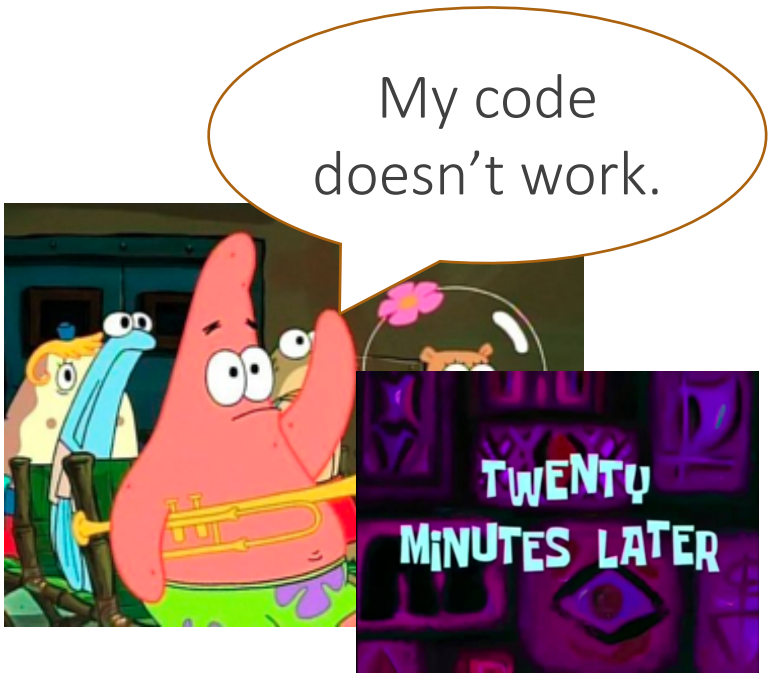
We love helping…

…but we love it when you help us help you.

# Eclipse Is Actually Your Friend

BanishWinterFixed.java

```
1  import stanford.karel.*;
2
3  public class BanishWinterFixed extends SuperKarel {
4        public void run() {
5            while(beepersInBag()) {
6                moveToTree();
7
8
9            while(frontIsBlocked()) {
10               turnLeft();
11               move();
12               turnRight();
13               }
14               placeLeaves();
15               turnRight();
16               moveToWall();
17
18               while(frontIsClear()) {
19                     move();
20
21               turnLeft();
22           }
23
24 Syntax error, insert "}" to complete Block
25
26
27           public void moveToTree() {
28             moveToWall();
29           }
30
31           public void moveToWall() {
32               while(frontIsClear()) {
33                   move();
34               }
35           }
```



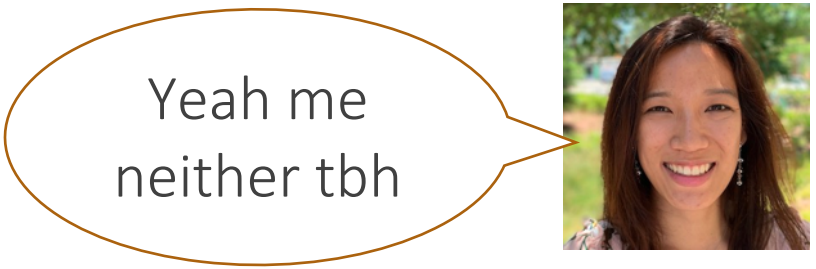

Syntax error, insert "}" to complete Block

# Style Is Also Your Friend

BanishWinterFixed.java

```
1  import stanford.karel.*;
2
3  public class BanishWinterFixed extends SuperKarel {
4        public void run() {
5            while(beepersInBag()) {
6                moveToTree();
7
8
9            while(frontIsBlocked()) {
10               turnLeft();
11               move();
12               turnRight();
13               }
14               placeLeaves();
15               turnRight();
16               moveToWall();
17
18               while(frontIsClear()) {
19                     move();
20
21               turnLeft();
22           }
23
24           moveToWall();
25           }
26           }
27
28           public void moveToTree() {
29             moveToWall();
30           }
31
32           public void moveToWall() {
33               while(frontIsClear()) {
34                   move();
35               }
36           }
```



# You Understand Your Code Best

## Section Leaders are GREAT at…

- Clarifying logic and strategy
- Debugging
- Making you feel at peace

## You are just as good as Section Leaders at…

- Reading (most) Eclipse errors
- Fixing brackets { } and indentation (tab)

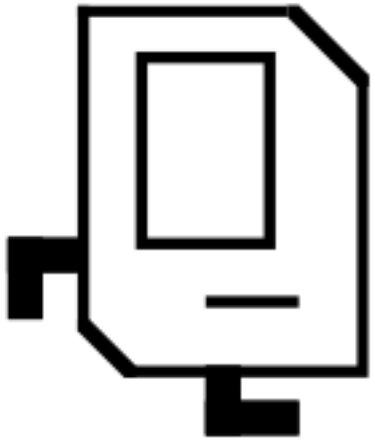
## You are BETTER than Section Leaders at…

- Explaining your own code
- Coding your own program

Programming takes practice.

Computers execute code,
but humans read code.

# See You Later!



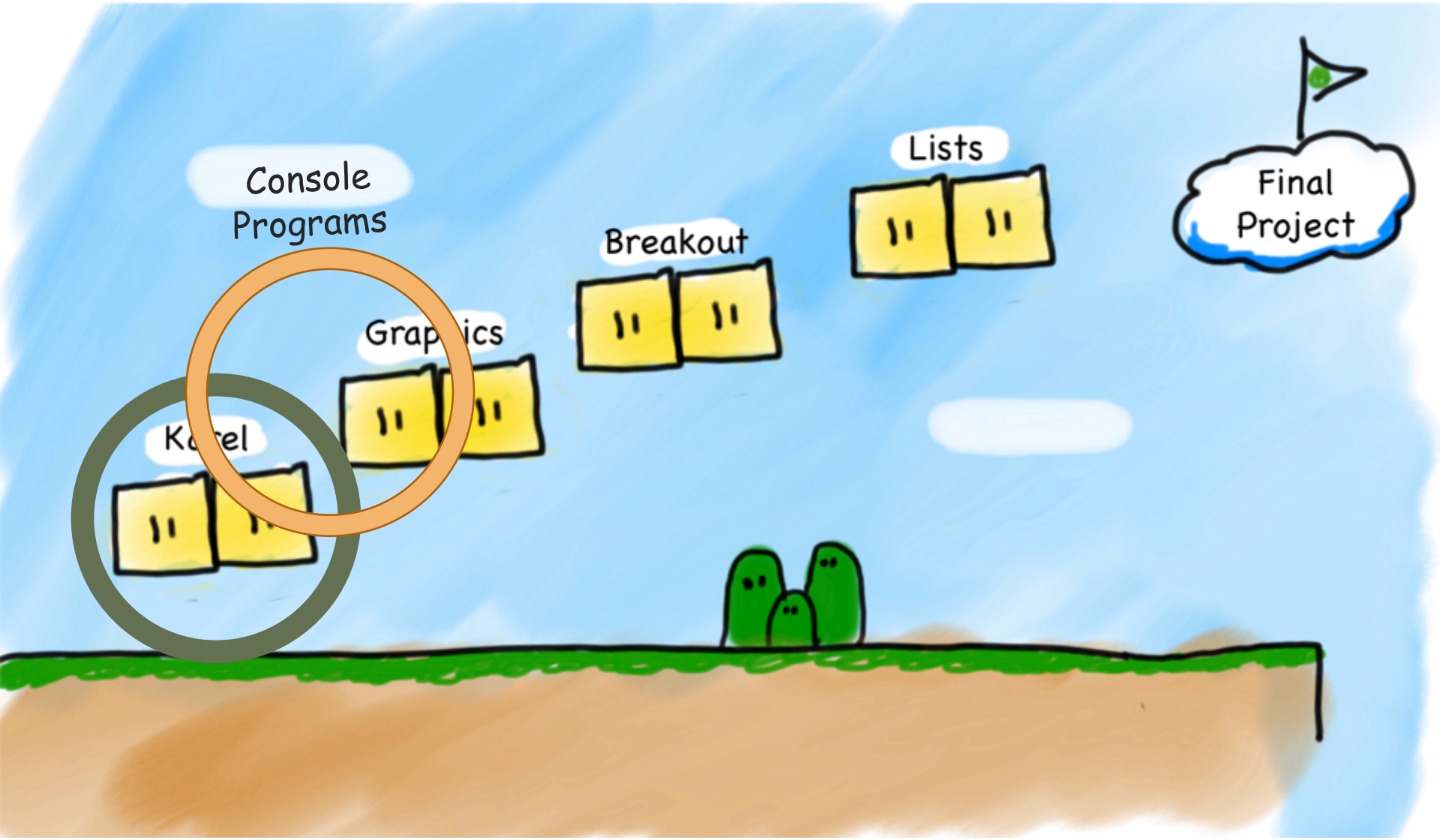
I will miss you.

Enjoy Java!

Call me maybe?

# Java

# Our First Step



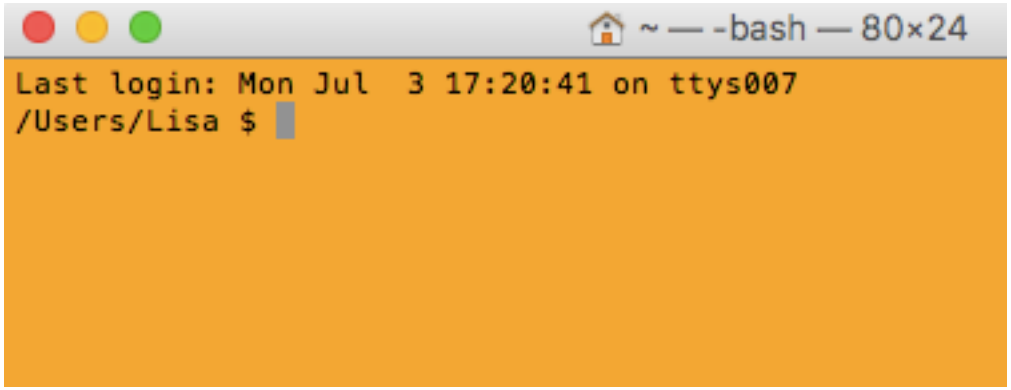
# Today's Goals

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



# First Console Program: Hello World

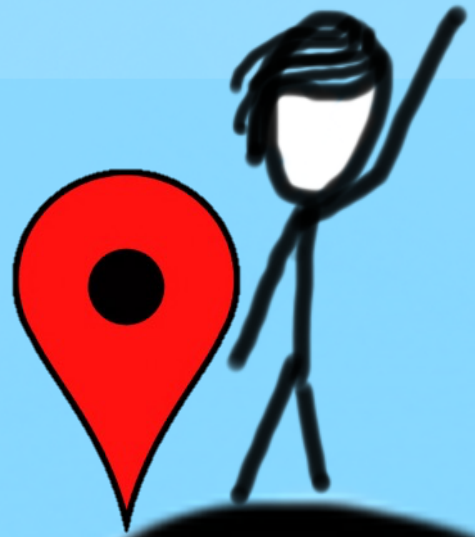
```
import acm.program.*;

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

# In Pop Culture

# 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?

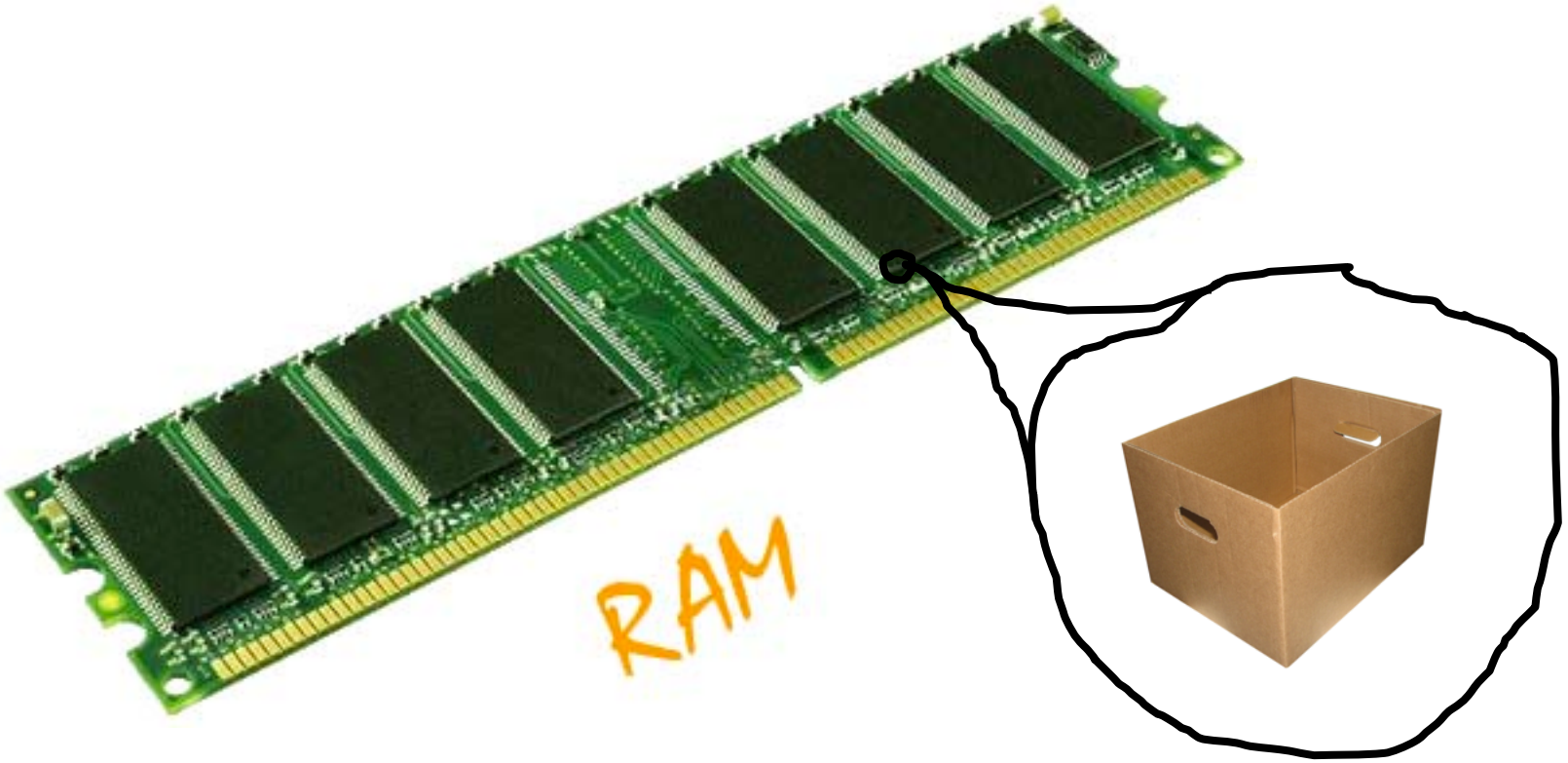


# What is a variable?

[suspense]

# Variables are Like Boxes

# Teeny Tiny Boxes

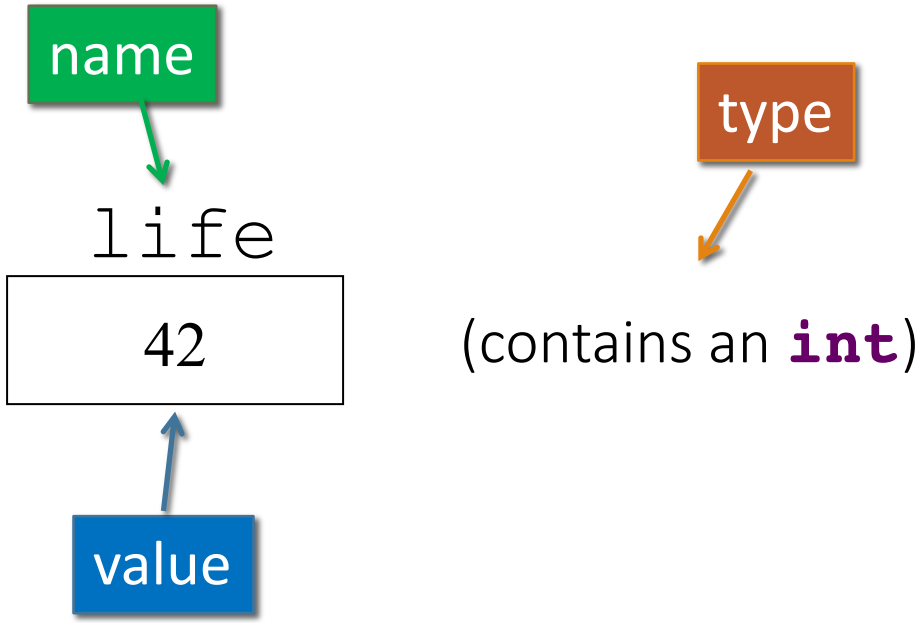



My computer has space for about 64 trillion boxes

# Variables are Like Boxes

type name value

```
int life = 42;
```



# Types

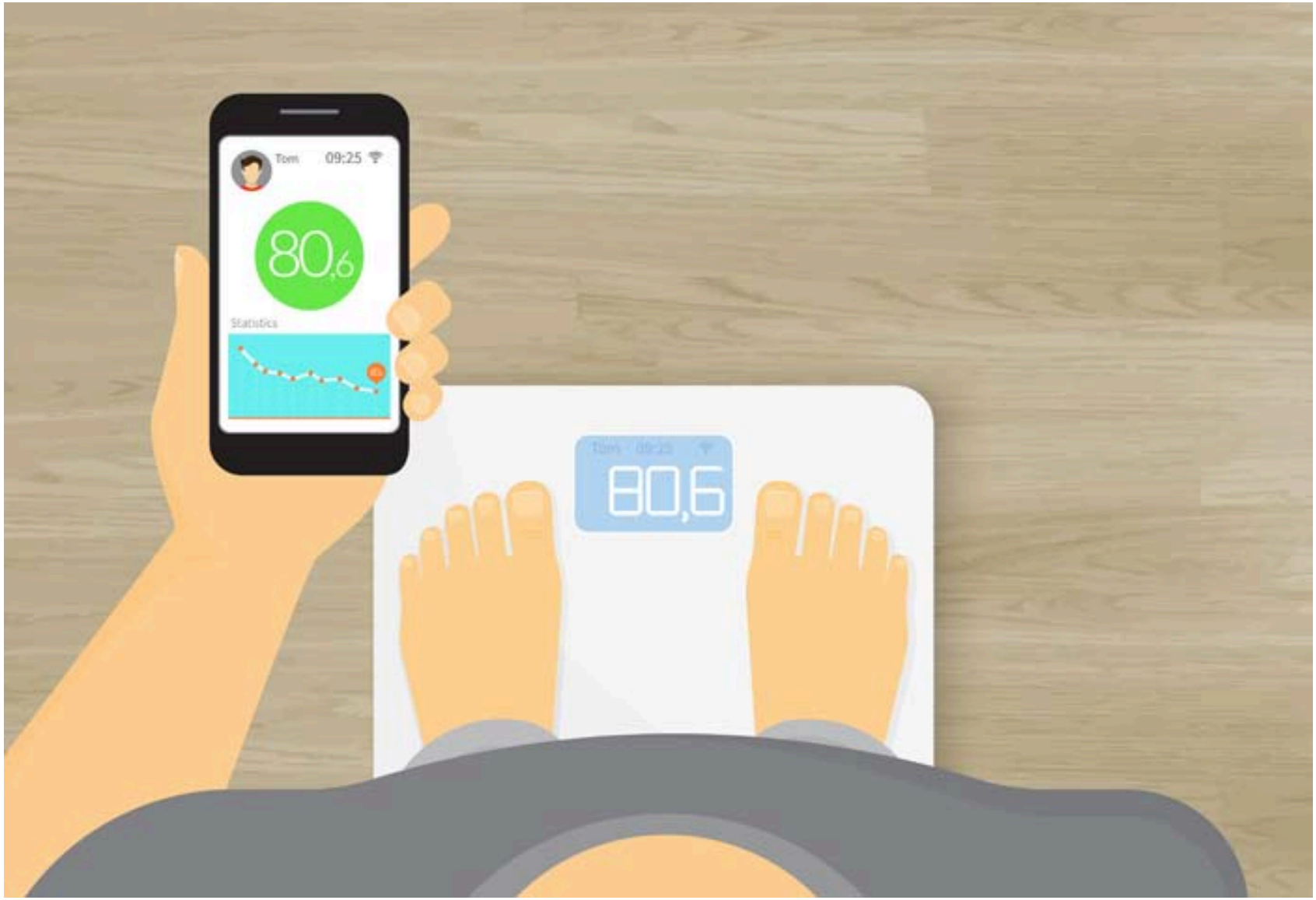
```
// integer values
int num = 5;

// 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?

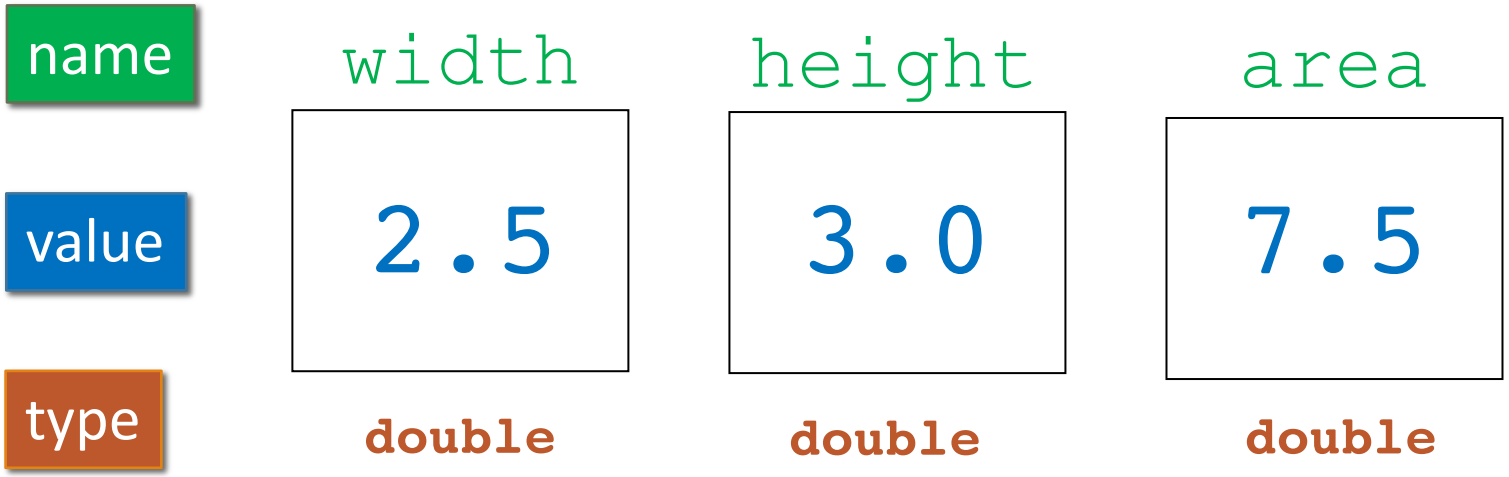
* It is weird to say something like 1.7

# Binary Operators

| | | | |
|---|---|---|---|
| + | Addition | * | Multiplication |
| – | Subtraction | / | Division |
| | | % | Remainder |

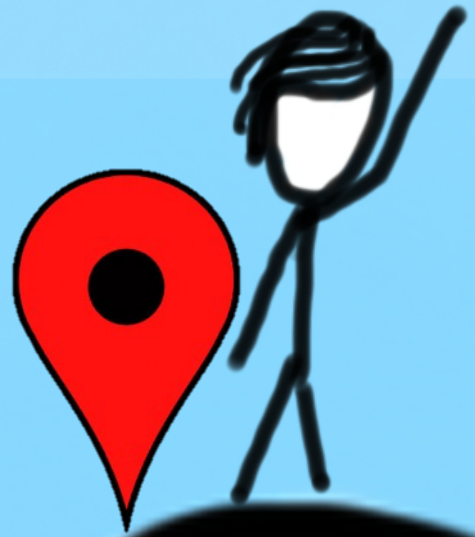
# Binary Operators

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



# 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

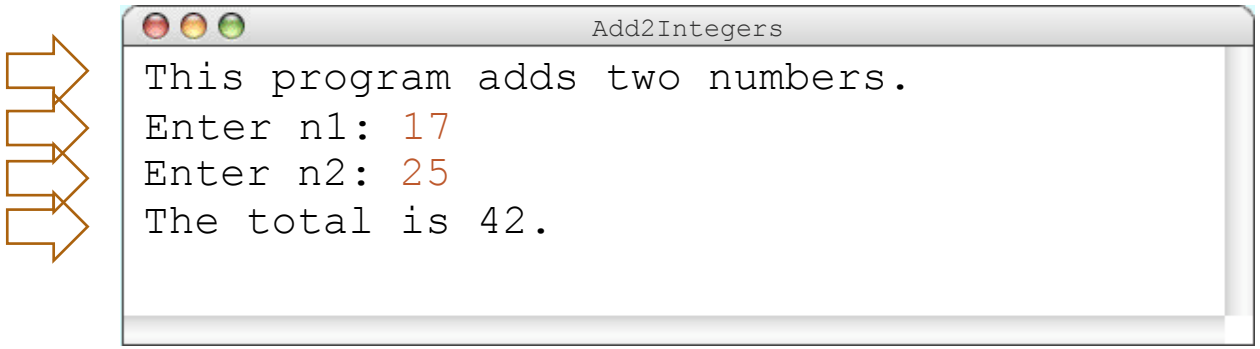
```
int a = readInt("Give me an int!");

double b = readDouble("And a double");
```

# Add2Integers

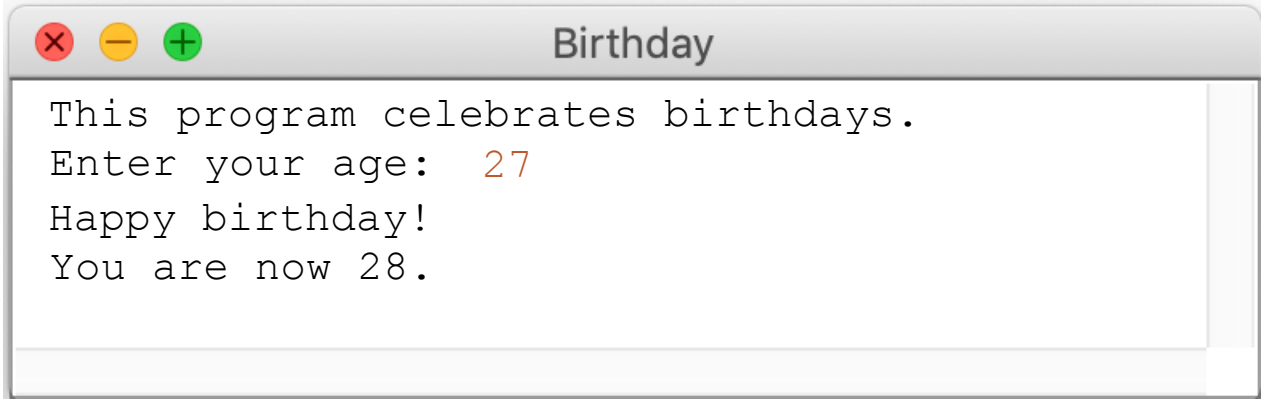
```
public class Add2Integers extends ConsoleProgram {
   public void run() {
      println("This program adds two numbers.");
      int n1 = readInt("Enter n1: ");
      int n2 = readInt("Enter n2: ");
      int total = n1 + n2;
      println("The total is " + total + ".");
   }
}
```

| n1 | n2 | total |
|---|---|---|
| 17 | 25 | 42 |



Questions?

# Birthday


Birthday
This program celebrates birthdays.
Enter your age:  27
Happy birthday!
You are now 28.

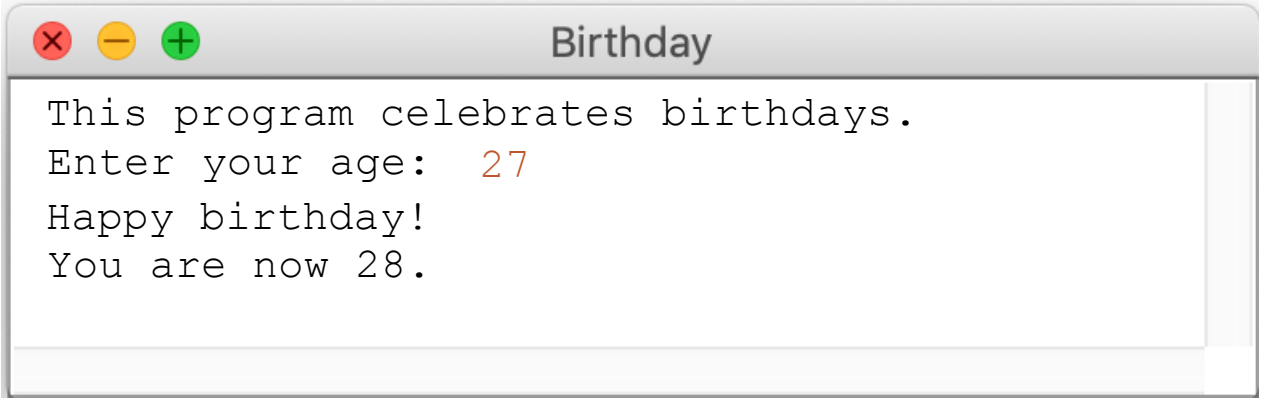
# Birthday

```
public class Birthday extends ConsoleProgram {
    public void run() {
        println("This program celebrates birthdays.");

        // creates a new int variable age
        ??????????? = readInt("Enter your age: ");

        // increments the age variable by one
        ???????????

        println("Happy birthday!");
        println("You are now " + age + ".");
    }
}
```



Let’s try it!

# Birthday

(1)

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

(1) Get a new **int** box.

# Birthday

(1) (2)

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

(1) Get a new **`int`** variable.

(2) The variable's name is **`age`**.

age

# Birthday

(1) (2) (3)

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

(1) Get a new **int** box.

(2) The variable's name is **age**.

(3) Evaluate the right-hand side.

age

# Birthday

(1) (2) (4) (3)

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

age

27

(1) Get a new **int** box.

(2) The variable's name is **age**.

(3) Evaluate the right-hand side.

(4) Set the value of **age**'s to the right-hand side.

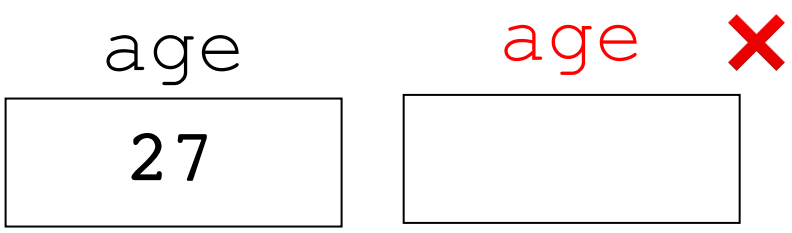
# Incorrect Birthday

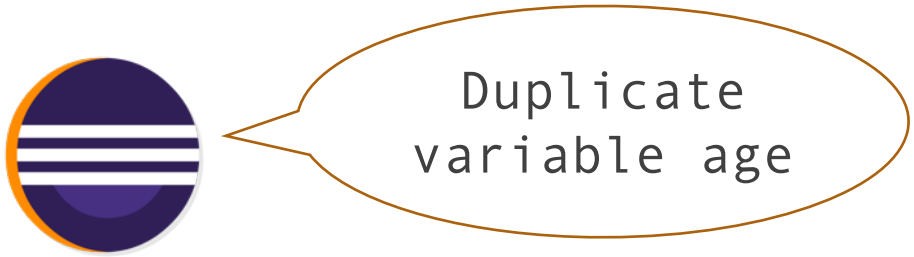
```
int age = readInt("Enter your age: ");

  (1)   (2)
int age = age + 1;
```

(1) Get a new **int** box.

(2) The variable's name is **age**.



# Birthday

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

(1) (3) (2)

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

(1) Get the variable named **age**.

(2) Evaluate the right-hand side. 28

(3) Set the value of **age**'s to the right-hand side.

age

| 28 |
|---|

Questions?

# What do you think this does?

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

# AHHHHHHH!!!!!!

```
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

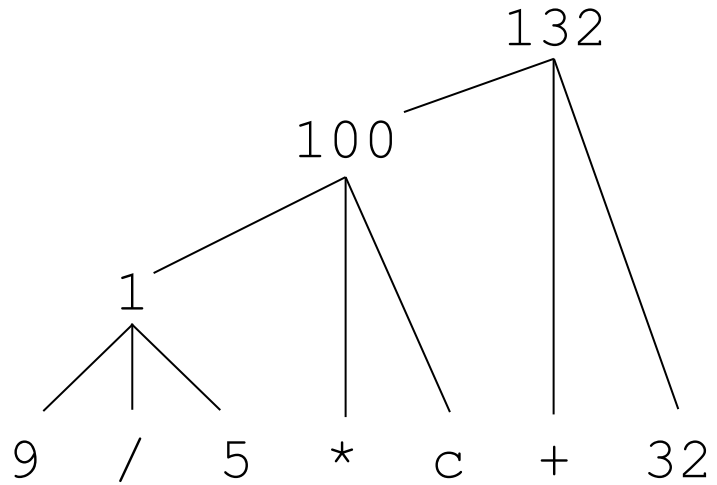
# Pitfalls of Integer Division

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`.

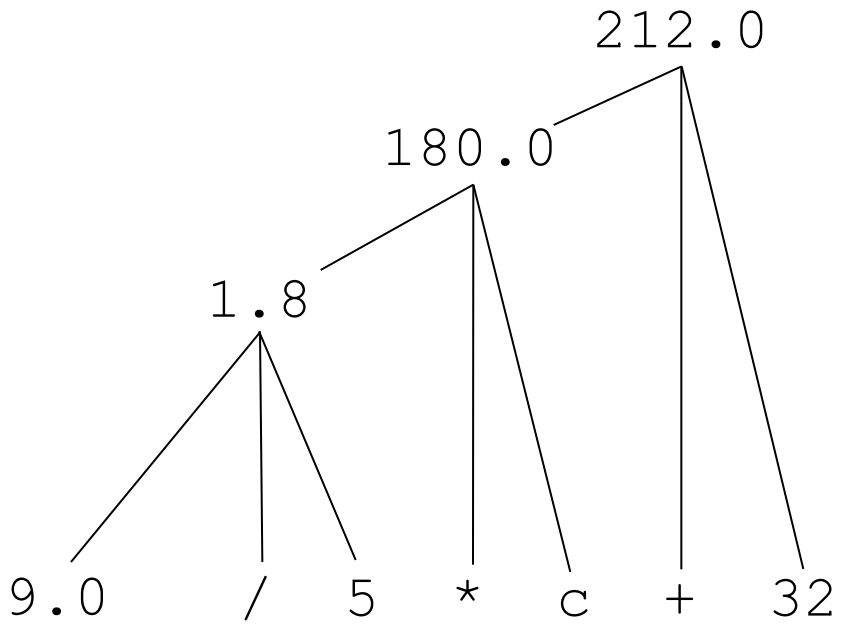


# Pitfalls of Integer Division

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 |

# Equal or Equals equals?

## Set variable

"equals"

=

```
double c = 100;
```

```
c = 25;
```

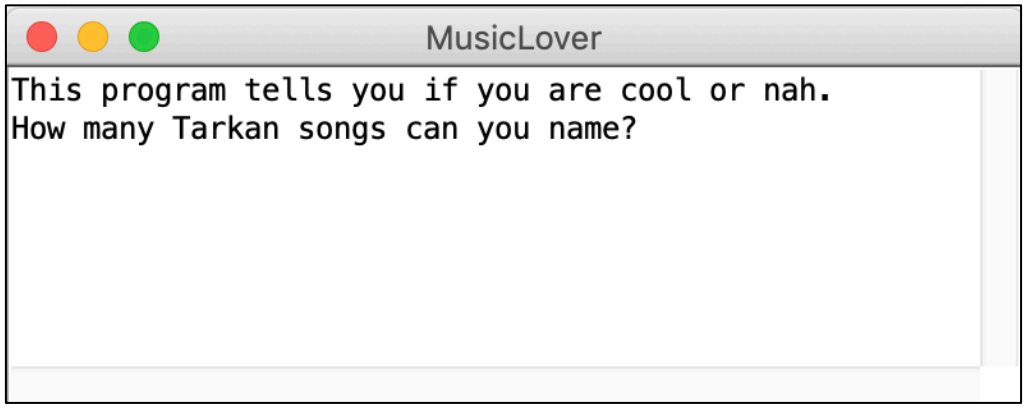
## Equivalent

"equals equals"

==

```
if (c == 100) {
    println("Hi!");
}
```

# Music Lover




User enters an `int`:

Exactly 1

More than 3

Anything else

Print out:

```
You only know Yolla!
```

```
Okay, you're pretty cool.
```

```
I don't know anymore.
```

# Demo

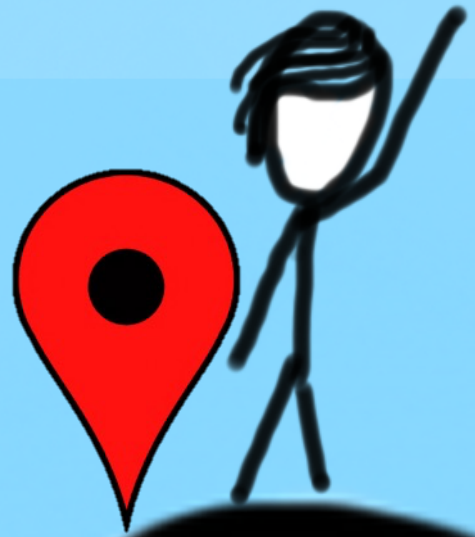
# Music Lover

```
public void run() {
    println("This program tells you if you are cool or nah.");
    int numberOfSongs = readInt("How many Tarkan songs can you name?");

    if (numberOfSongs == 1) {
        println("You only know Yolla!");
    }
    else if (numberOfSongs > 3) {
        println("Okay, you are pretty cool.");
    } else {
        println("I don't know anymore.");
    }
}
```

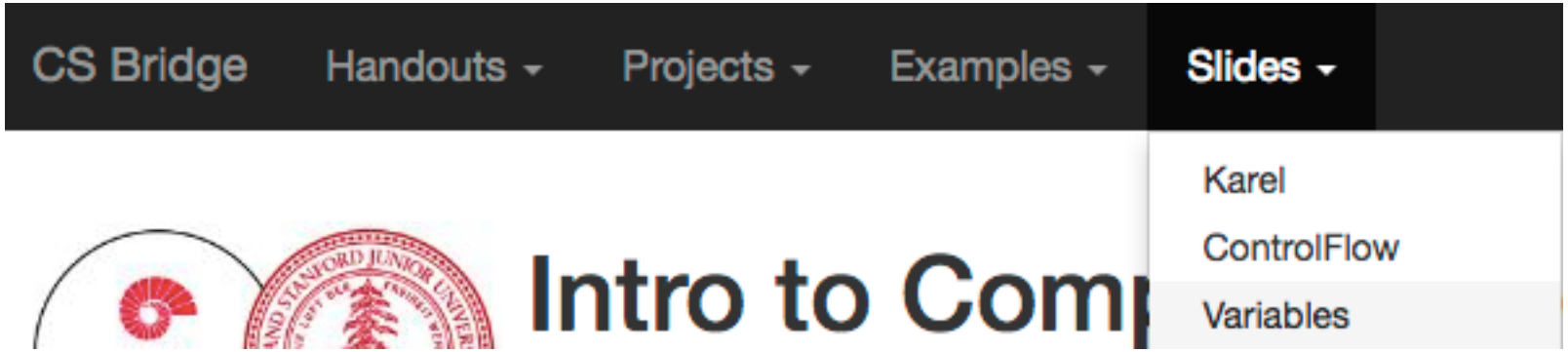
# Today's Goals

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


CS Bridge
Handouts
Projects
Examples
Slides
Karel
ControlFlow
Variables
Intro to Comp