



# Variables

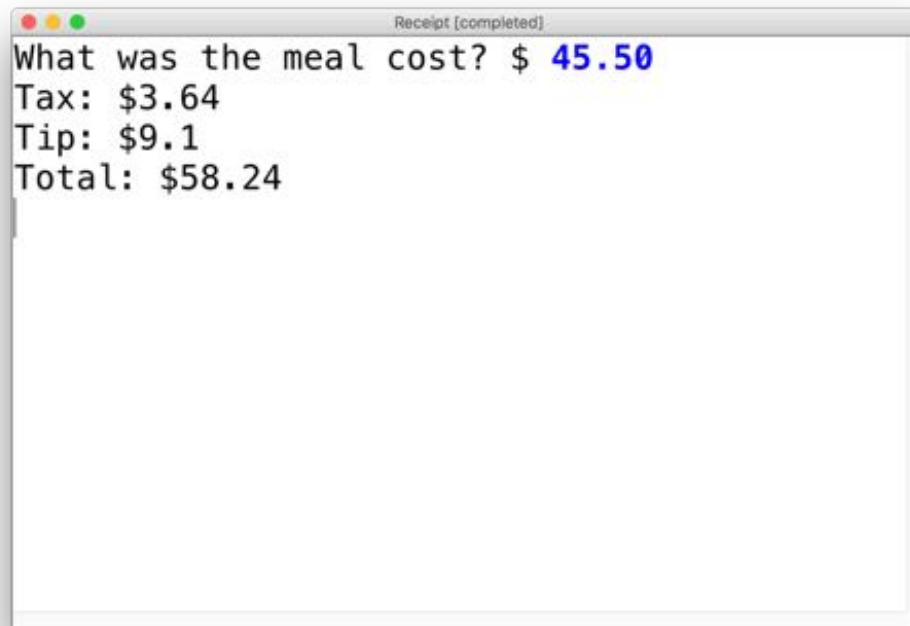
Chris Piech

CS106A, Stanford University

# New Ability

Write a program that calculates the tax, tip and total bill for us at a restaurant.

The program should ask the user for the subtotal, and then calculate and print out the tax, tip and total.



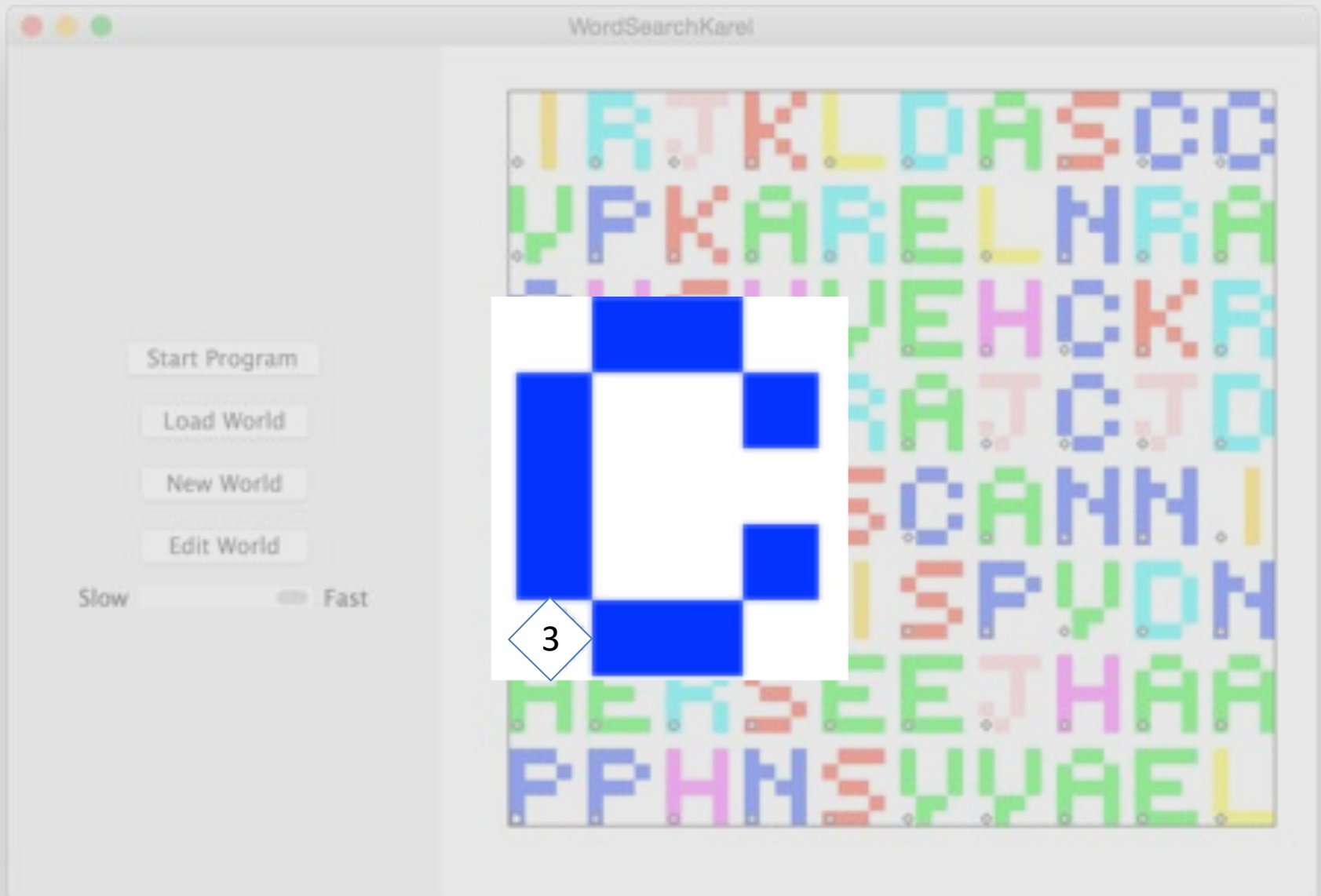
```
Receipt [completed]
What was the meal cost? $ 45.50
Tax: $3.64
Tip: $9.1
Total: $58.24
```



# Review: Decomposition

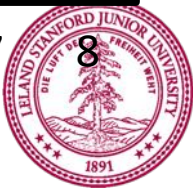
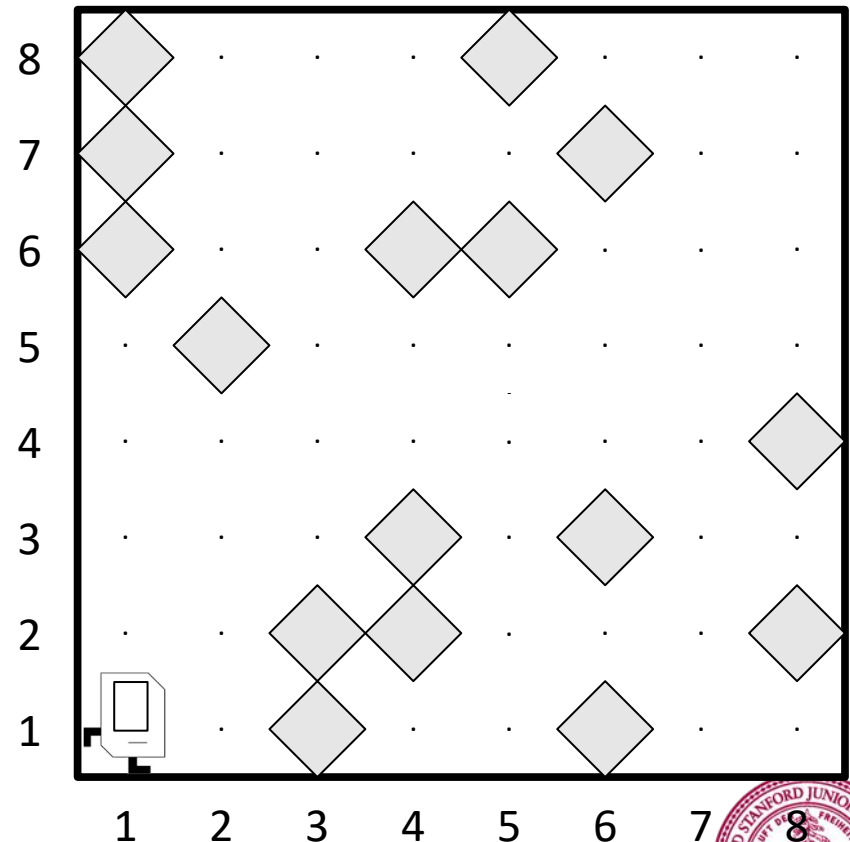
1. Each method solves one "problem"
2. Methods should have good names
3. Comment each of your methods
4. Length of methods should be  $< 15$  lines
5. Methods should ideally be generalizable



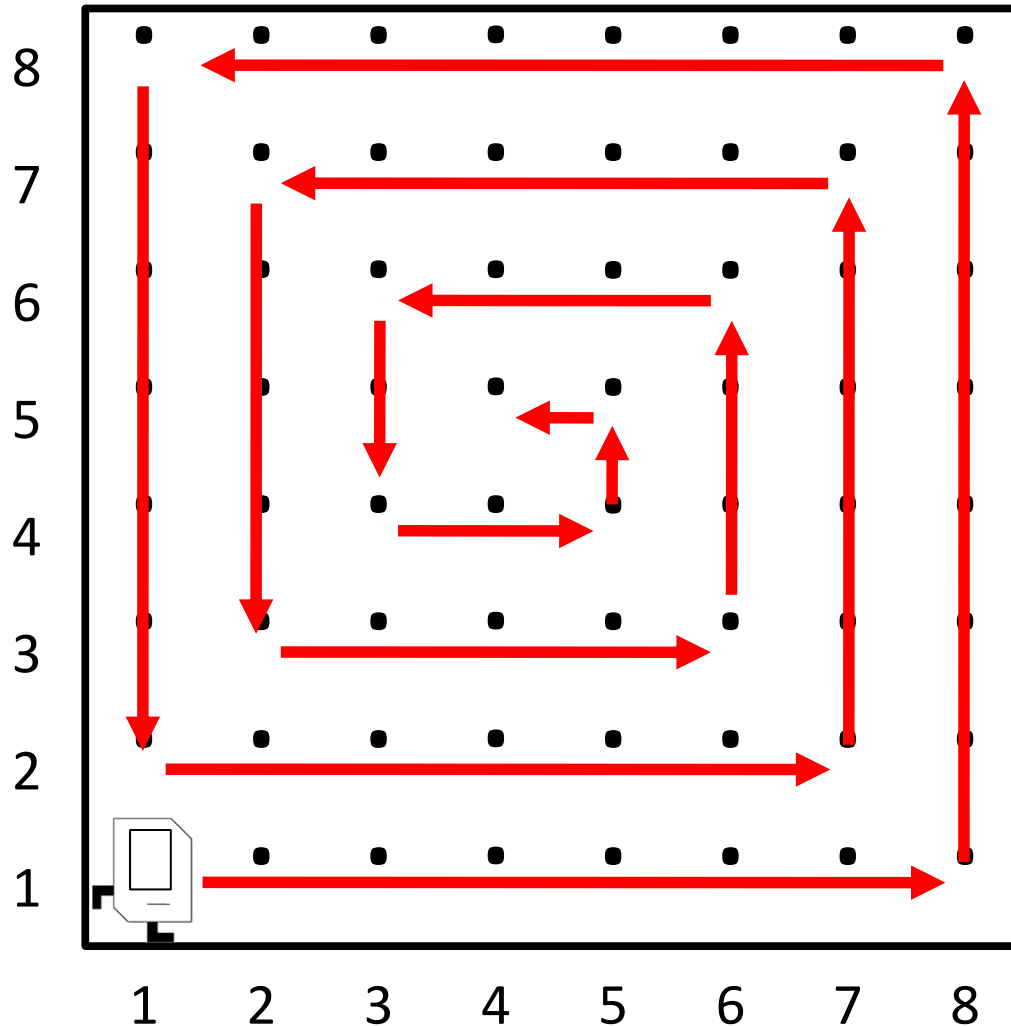


# Rhoomba Karel

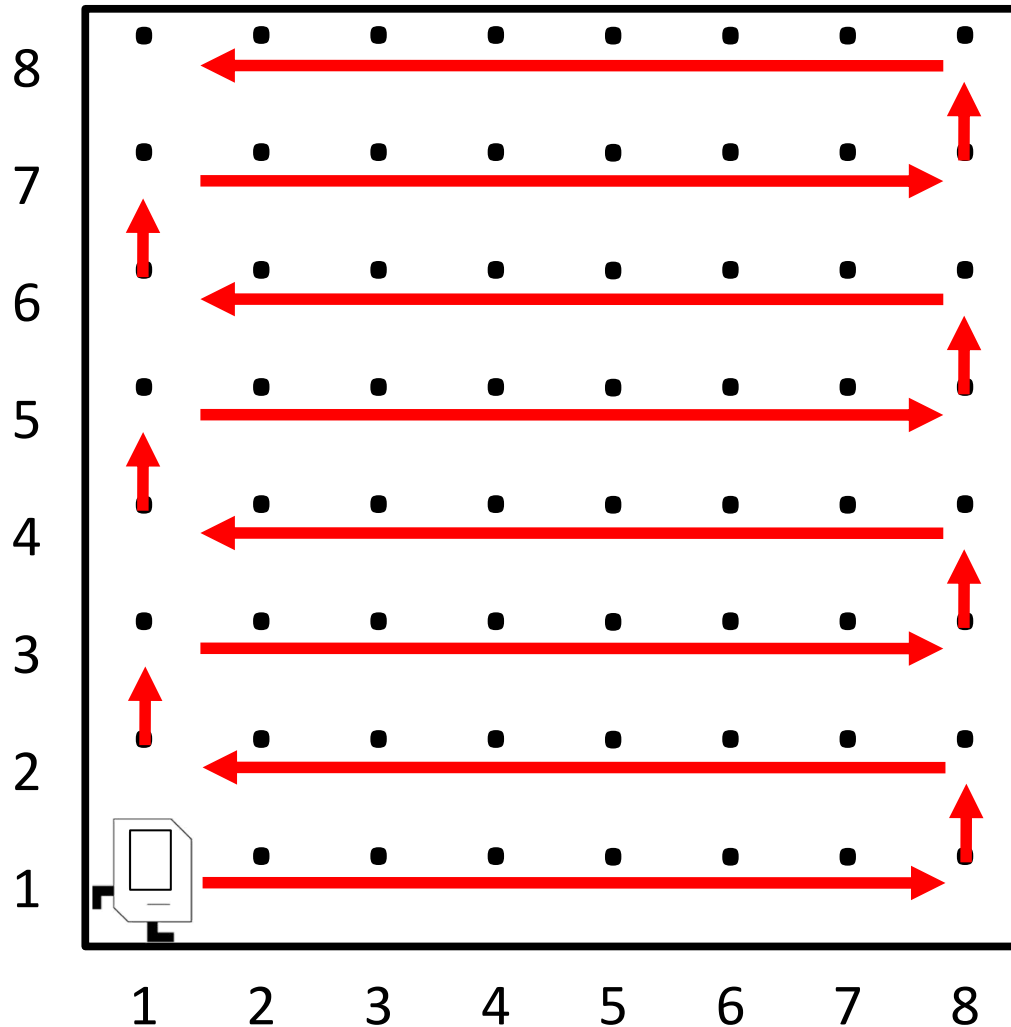
- Write a **Roomba** Karel that sweeps the entire world of all beepers.
  - Karel starts at (1,1) facing East.
  - The world is rectangular, and some squares contain beepers.
  - There are no interior walls.
  - When the program is done, the world should contain 0 beepers.
  - Karel's ending location does not matter.
- How should we approach this tricky problem?



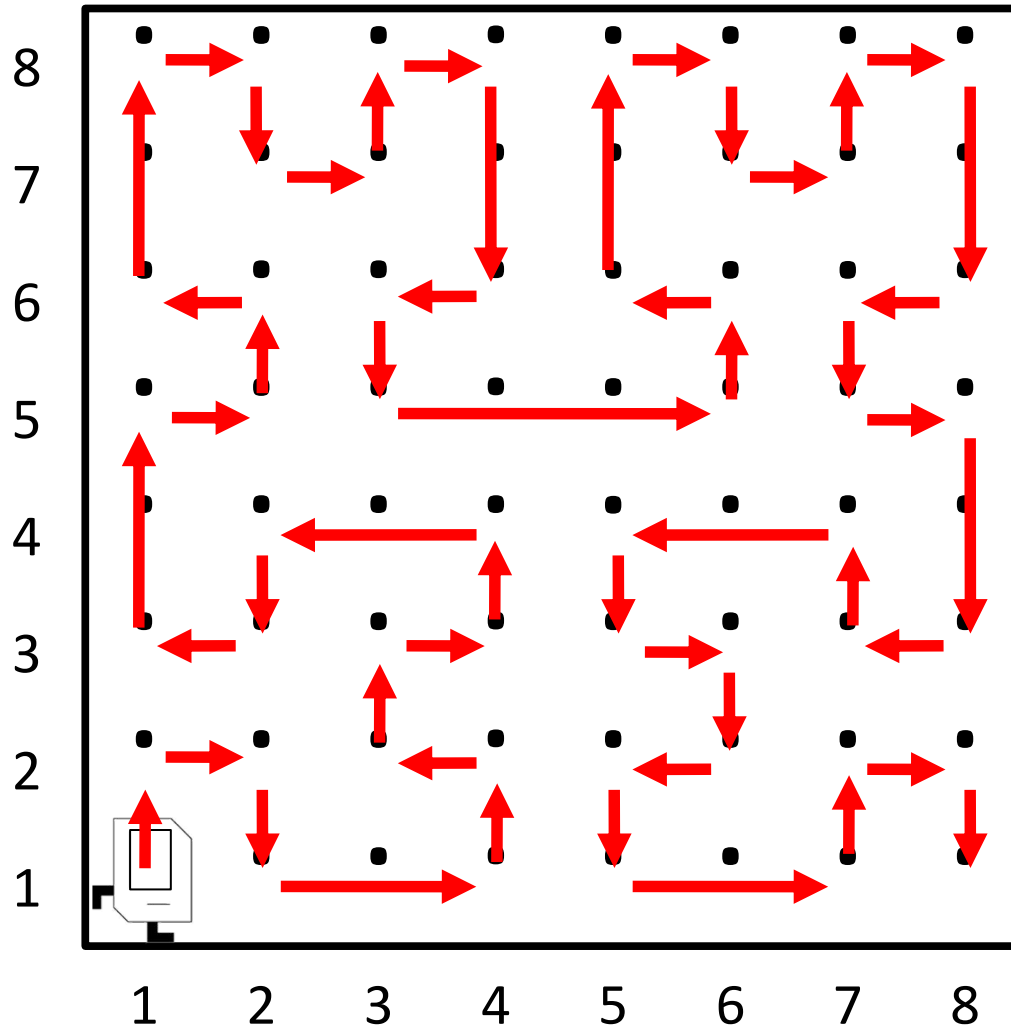
# Possible Algorithm 1



# Possible Algorithm 2

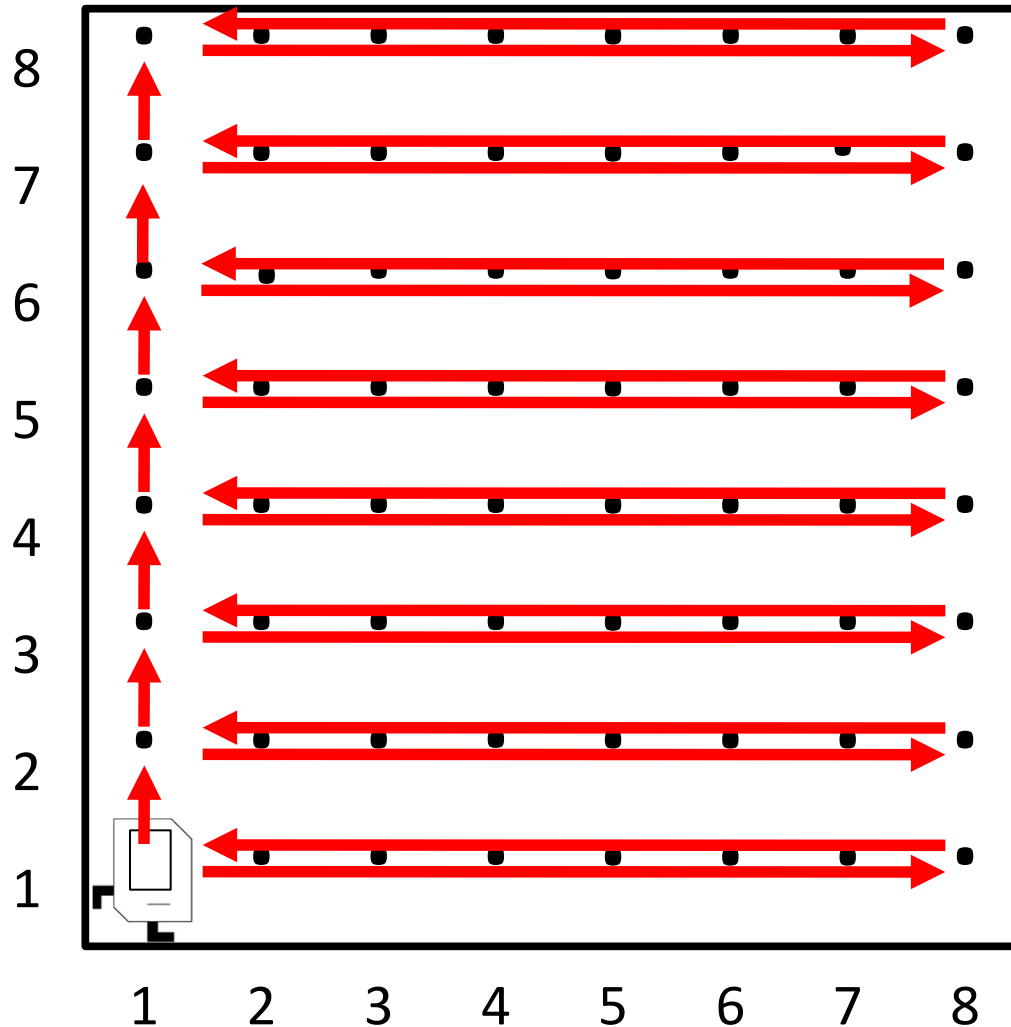


# Possible Algorithm 3

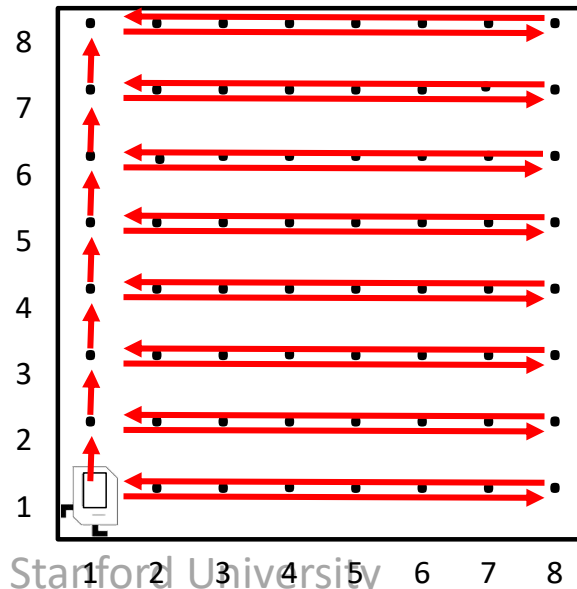
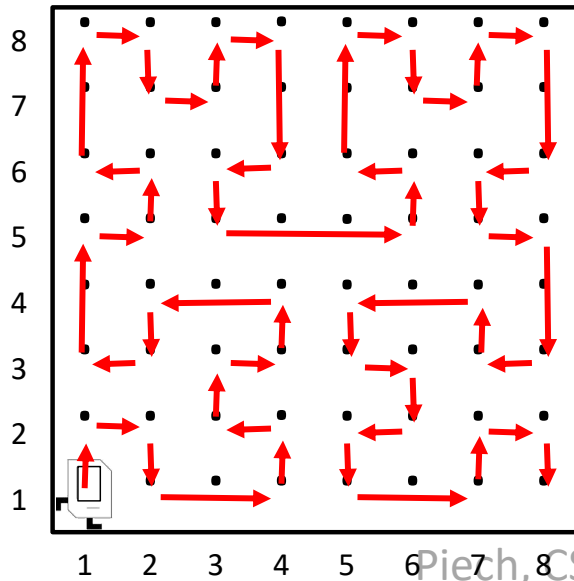
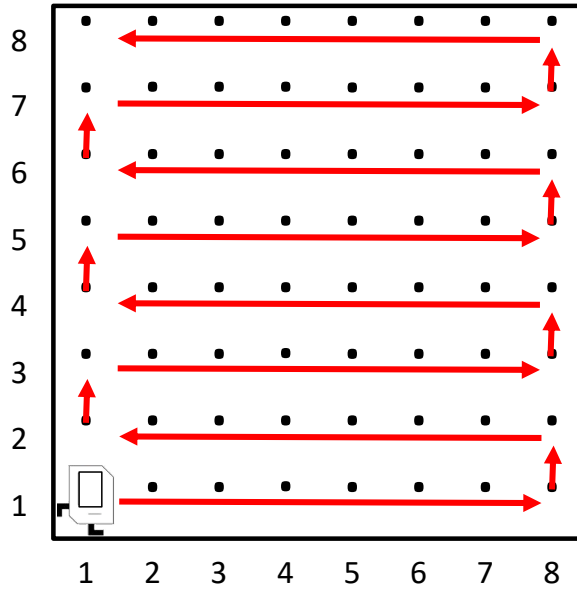
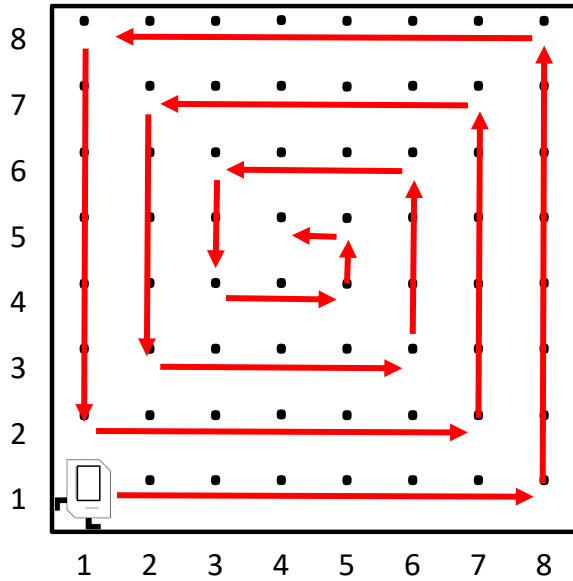




# Possible Algorithm 4



# Decision 2018



# See You Later!



I will miss you.

Enjoy Java!

See you on the  
midterm 🥲.



# Java

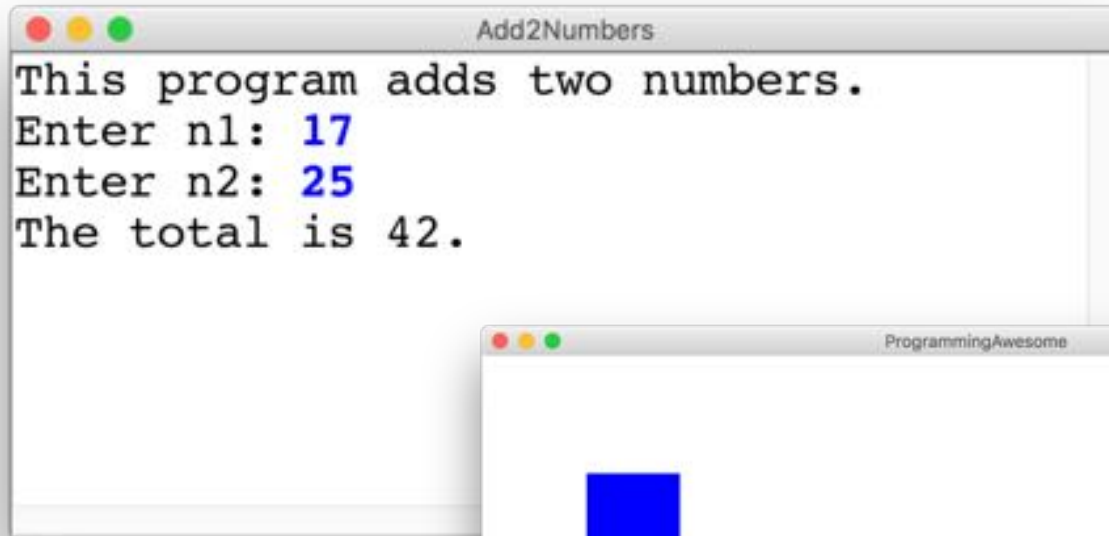


# Today's Goal

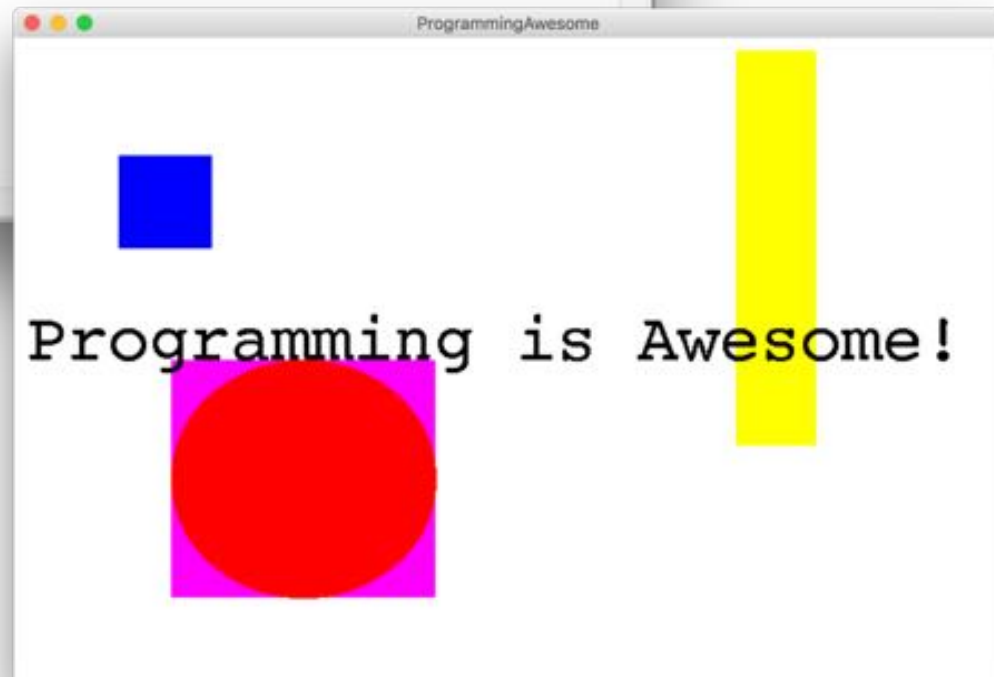
1. How do you make a box?
2. How do you see what is in a box?
3. What can you put in a box?
4. How do you change what is in a box?



# Two Example Programs



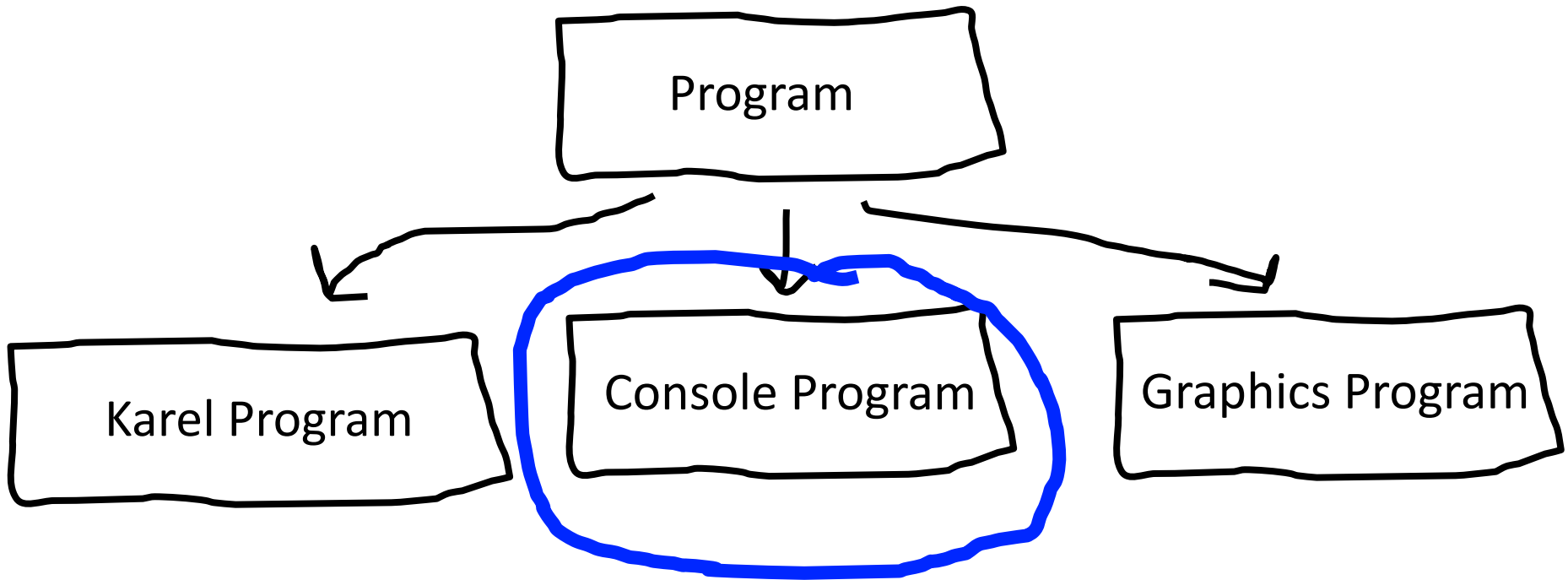
```
Add2Numbers
This program adds two numbers.
Enter n1: 17
Enter n2: 25
The total is 42.
```



```
ProgrammingAwesome
Programming is Awesome!
```



# Types of Programs



# First Console Program: Hello World

```
import acm.program.*;

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





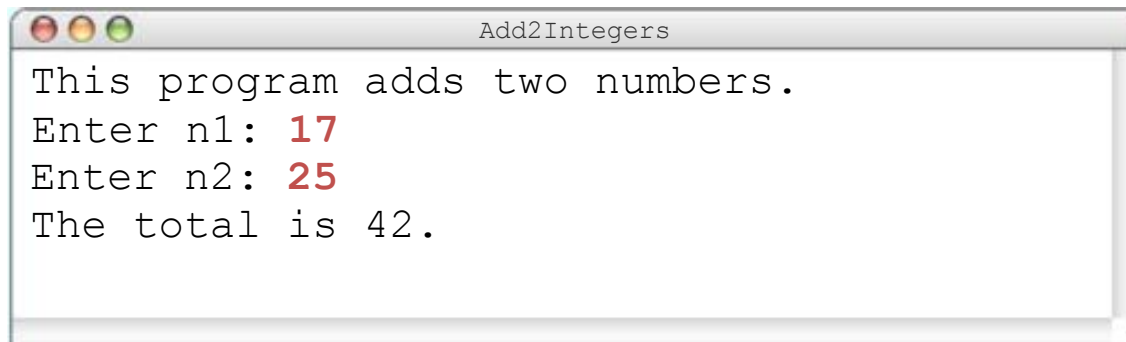
# In Pop Culture



# Add2Integers

```
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 + ".");  
    }  
}
```

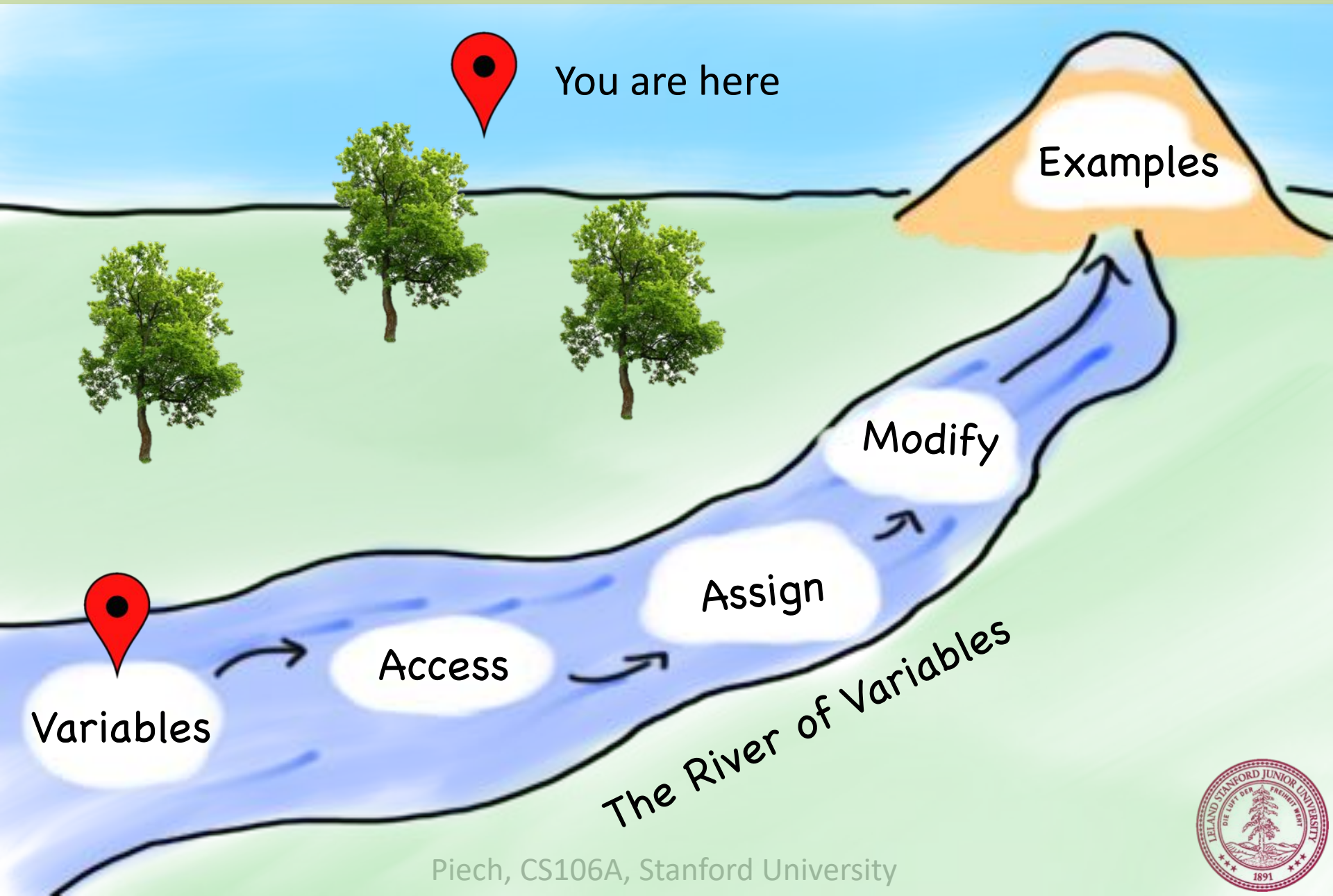
n1	n2	total
17	25	42



```
Add2Integers  
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25  
The total is 42.
```



# Today's Route



Programs are *control flow* and *variables*

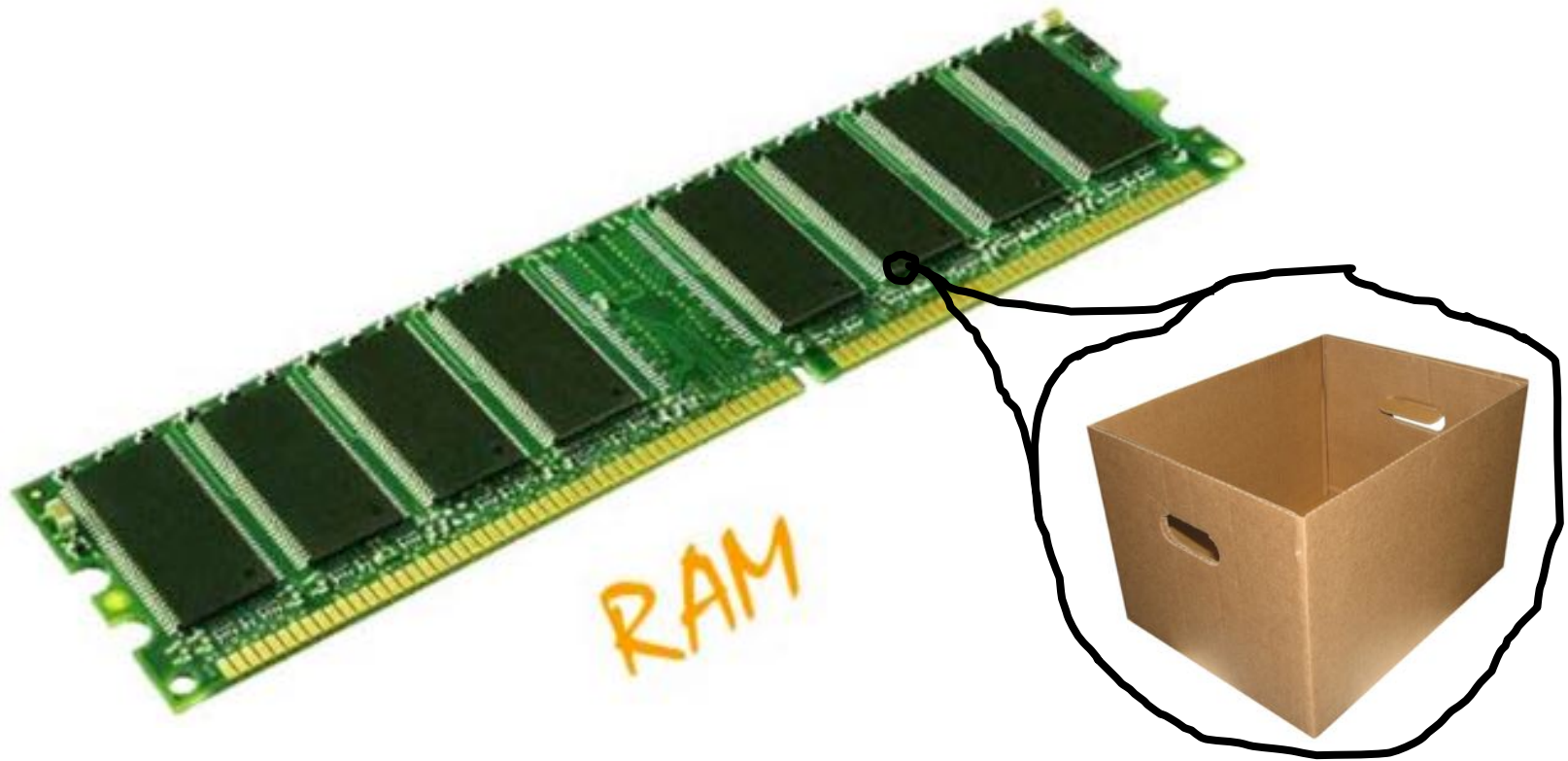
What is a variable?

[suspense]

# Variables are Like Boxes



# Teeny Tiny Boxes



My computer has space for about 2 billion boxes



# Making a New Variable

```
int age = 29;
```



# Making a New Variable

type



**int**

name



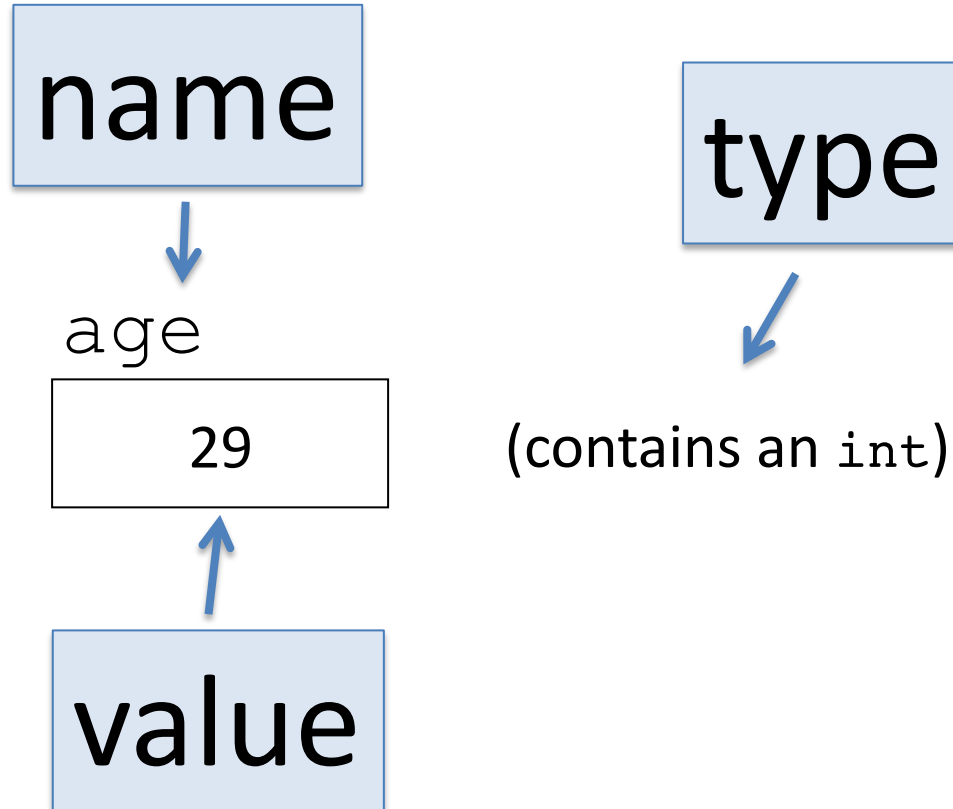
age = 29;



Initial value



# Three Properties



# Resulting Type of Binary Expression



When a line starts with a variable *type*, it is creating a new variable ...aka a box.

Example:

```
int myBox = 5;
```



# Core Types

```
// integer values
```

```
int num = 5;
```

```
// real values
```

```
double fraction = 0.2;
```

```
// true or false
```

```
boolean isLove = true;
```

\* Why is it called a double?



# Core Types

```
// letters
```

```
char letter = 'c';
```

```
// "strings" of letters
```

```
String phrase = "Spring rocks!";
```



# Core Types

```
// integer values
```

```
int num = 5;
```

```
// real values
```

```
double fraction = 0.2;
```

```
// true or false
```

```
boolean isLove = true;
```

\* Why is it called a double?



# 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



# Lots of Boxes

```
public void run() {  
  
    // integer values  
    int age = 29;  
  
    // real values  
    double weight = 180.2;  
  
}
```



Can you access the value in a variable  
(aka box)?

# Outputting Variable Value

```
// creates a variable called  
// age with the value 29.  
int age = 29;  
  
// puts the value of the age  
// variable on the screen.  
println(age);
```

\* Fun fact. Chris turns 30 on April 25th



# Fancy Output

```
// creates a variable called
// age with the value 29.
int age = 29;

// puts the following on the
// screen
// age is: <value>
println("age is: " + age);
```

\* Fun fact. Chris turns 30 on April 25th

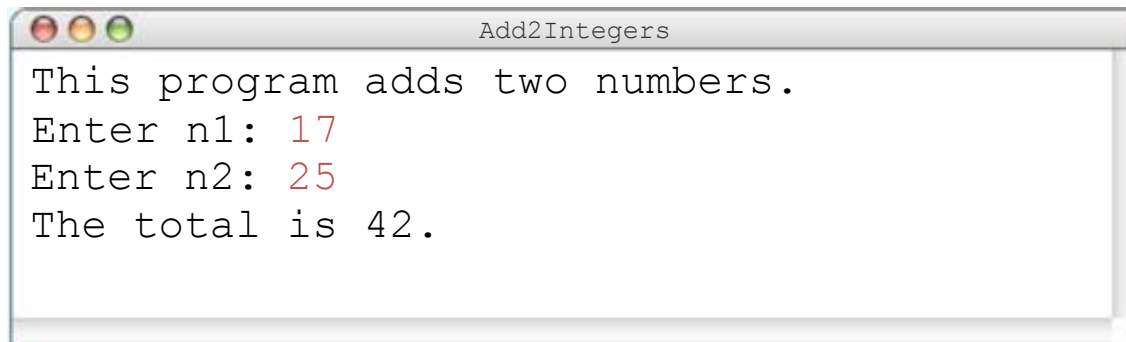


What data can you put in a variable  
(aka box)?

# Add2Integers

```
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 + ".");  
    }  
}
```

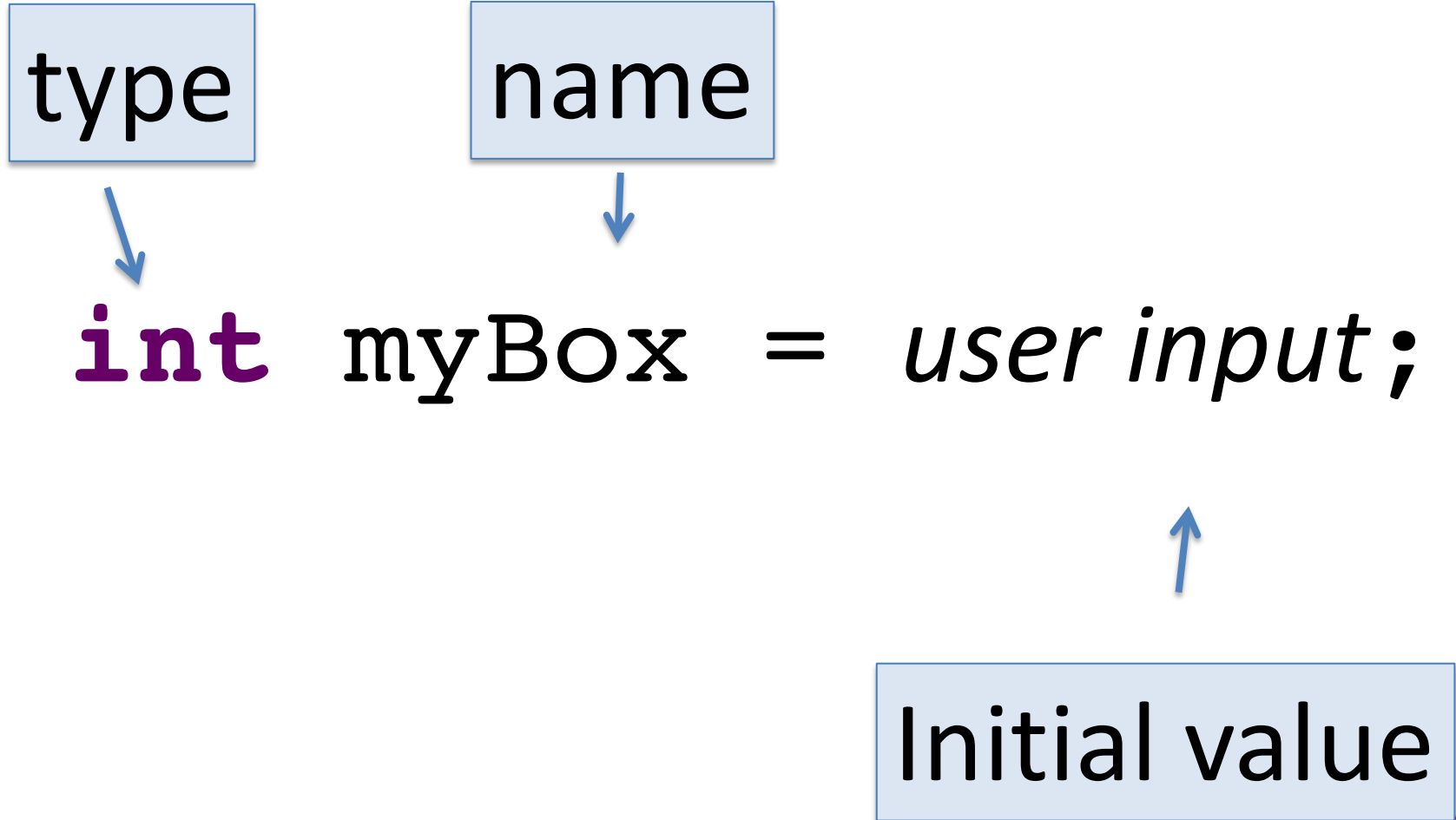
n1	n2	total
17	25	42



```
Add2Integers  
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25  
The total is 42.
```

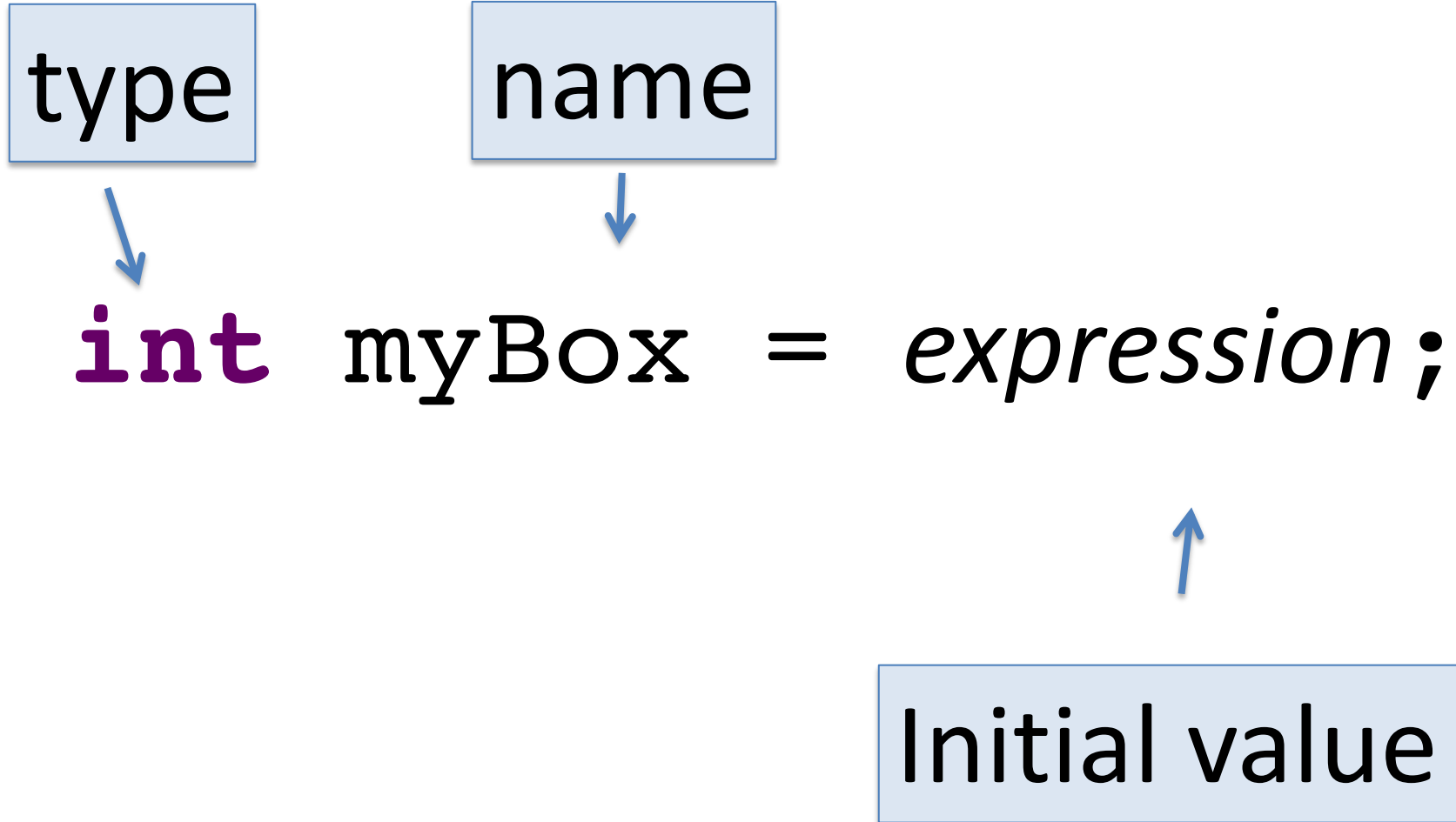


# Making a New Variable





# Making a New Variable



# Values from User Input

```
// Prompts user for a whole number. Stores  
// result in a variable (aka a box)  
int kids = readInt("How many children?");
```

```
// Prompts user for a decimal number. Stores  
// result in a variable (aka a box)  
double tip = readDouble("Tip? $");
```

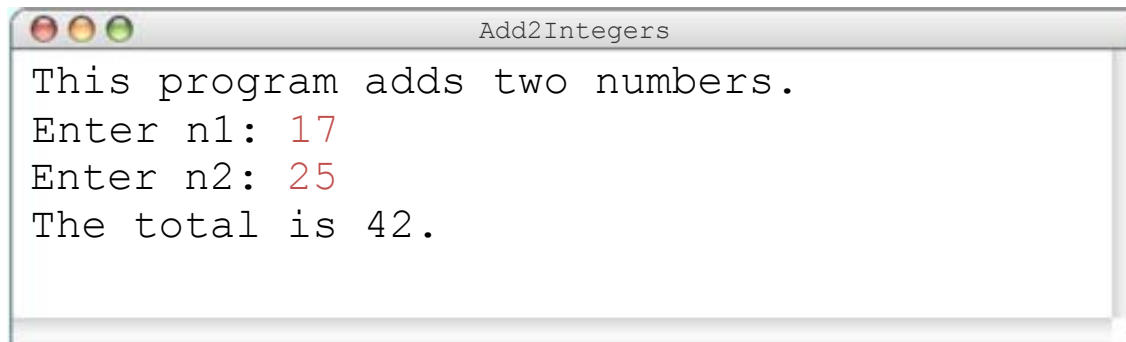
```
// Haven't you ever wondered, who was the first  
// person to eat a carrot??  
boolean edible = readBoolean("Subject alive?")
```



# Add2Integers

```
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



```
Add2Integers  
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25  
The total is 42.
```



# Binary Operators

+ Addition

− Subtraction

\* Multiplication

/ Division

% Remainder

See you another day, tio.



Learn by examples

# Order of Operation

// Mult before addition first!

```
int result = 4 + 2 * 3; // 10
```

// Left to right!

```
int sum = 1 + 2 + (3 * 4); // 15
```

Priority	Operator	Tie breaker
Highest	()	Left to right
Middle	* /	Left to right
Lowest	+ -	Left to right



# What do you think this does?

```
// creates a variable called  
// success rate  
double successRate = 1 / 2;
```



# AHHHHHHH!!!!!!

```
// creates a variable called  
// success rate  
double successRate = 1 / 2;
```

0.0





# Resulting Type of Binary Expression



All *binary operators* result in a value (like a temporary variable) which ***has a type***. You need to know what type that will be.

Most important example:

**`int`** / **`int`** results in an **`int`**

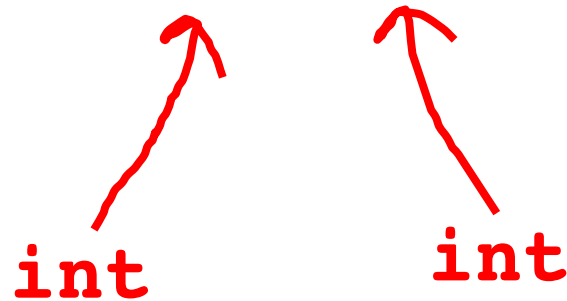


# AHHHHHHH!!!!!!

```
// creates a variable called  
// success rate
```

```
double successRate = 1 / 2;
```

**int**                      **int**



# Resulting Type

**int** / **int** results in an **int**

**double** \* **double** results in a **double**

**int** + **double** results in a **double**



# Resulting Type of Binary Expression



All *binary operators* result in a value (like a temporary variable) which *has a type*. The general rule is: operations always return the *most expressive* type:

Expressive hierarchy:

**boolean** < **char** < **int** < **double** < **String**

Example:

**int** / **double** results in a **double**



Even more examples...

# Pitfalls of Integer Division

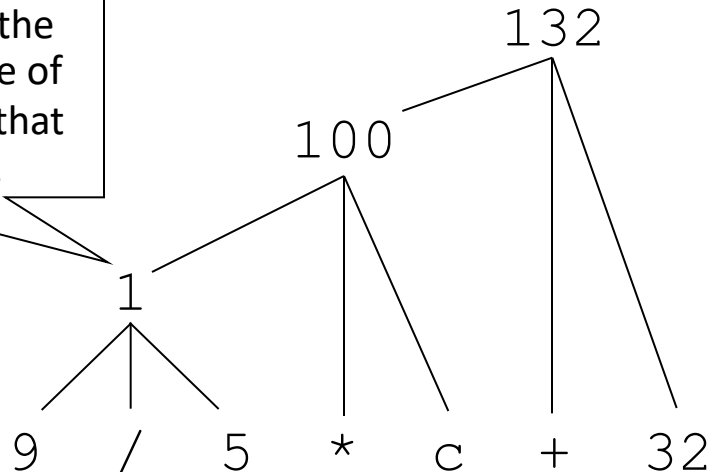
Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
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`.

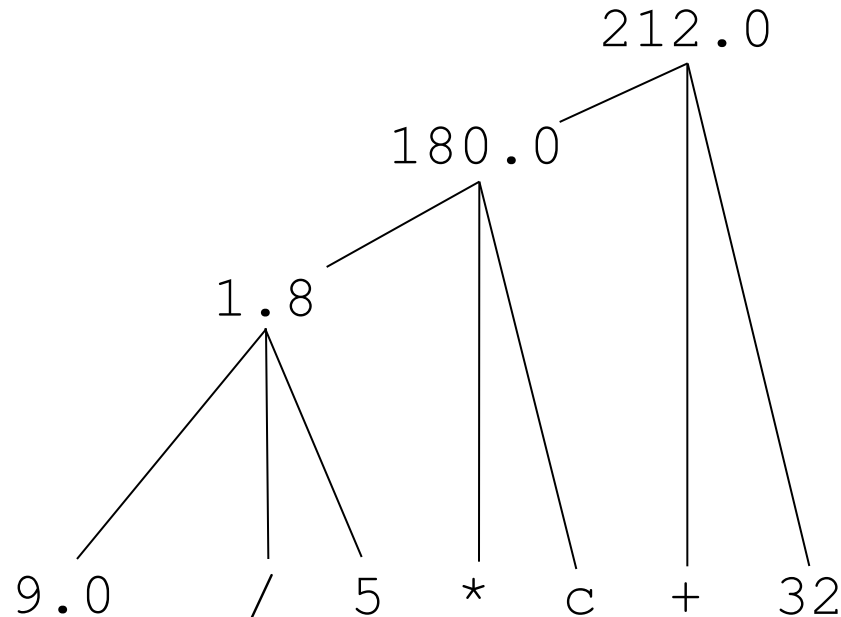


# Pitfalls of Integer Division

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:



# Practice

- $5 + 3 / 2 - 4$

// 2

- $15 / 2.0 + 6$

// 13.5





Can you change the value in a variable  
(aka box)?

# Modifying a Variable

```
// creates a variable called  
// age with the value 29.
```

```
int age = 29;
```

```
// this puts a new value in the box  
age = 30;
```

```
// In what world does this make sense?  
// Welcome to Java  
age = age + 2;
```



# Compiler Errors

1. A variable can't be used until it is assigned a value.

```
int x;  
println(x);    // Error: x has no value
```

2. You may not declare the same variable twice.

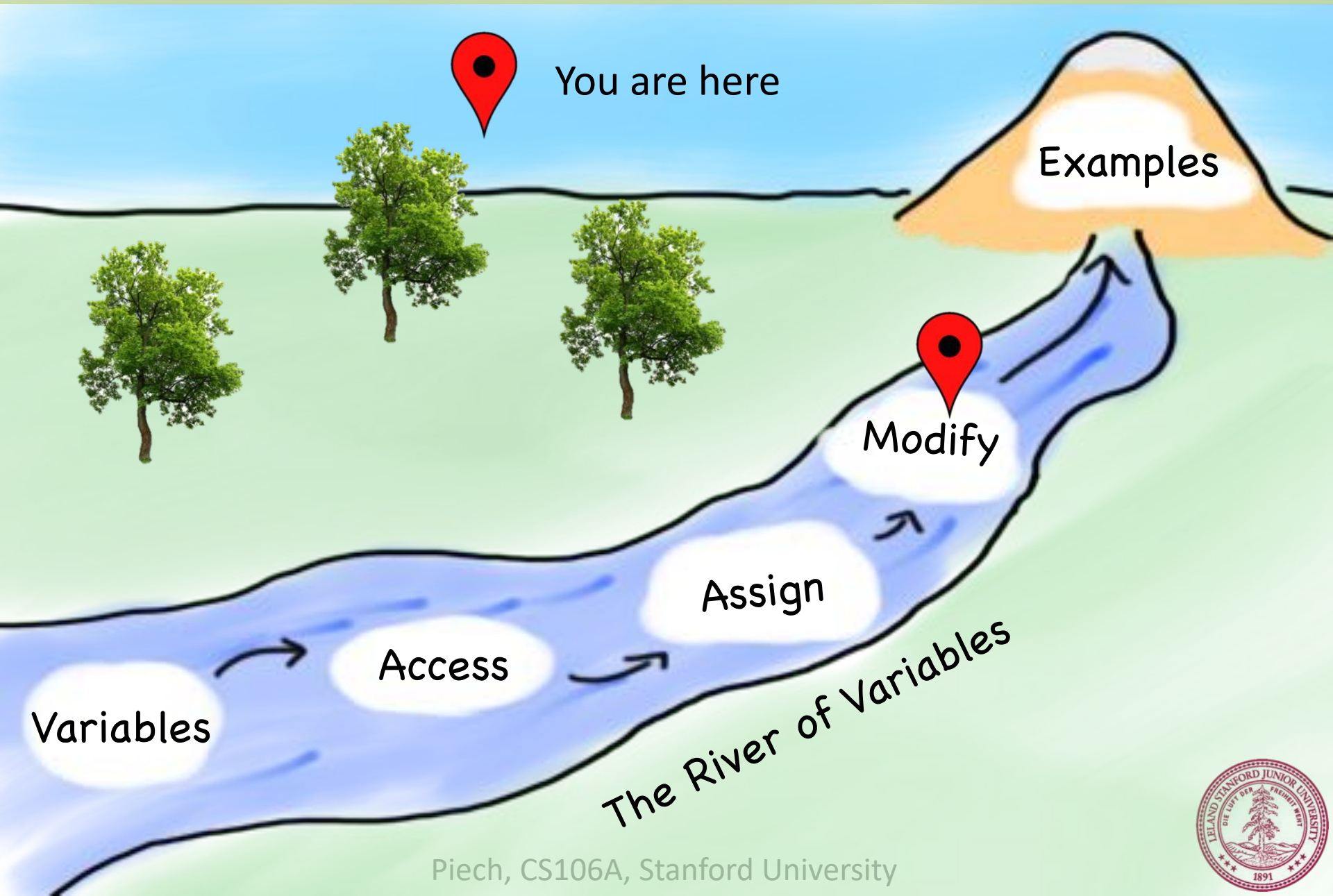
```
int y = 3;  
int y = 5;    // Error: y already exists
```

3. You may not use a variable until it is declared.

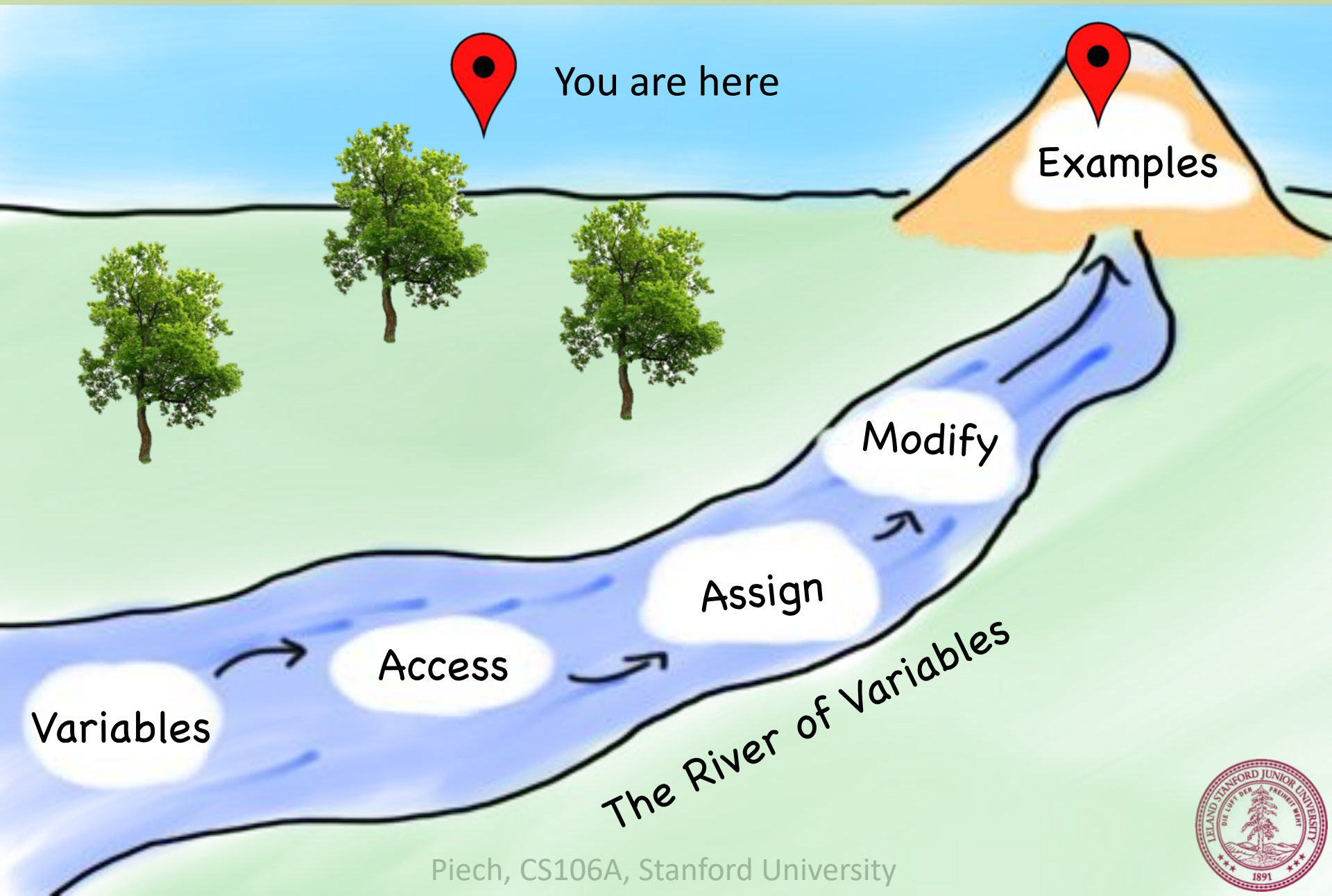
```
z = 10;       // Error: z cannot be resolved
```



# Today's Route



# Today's Route



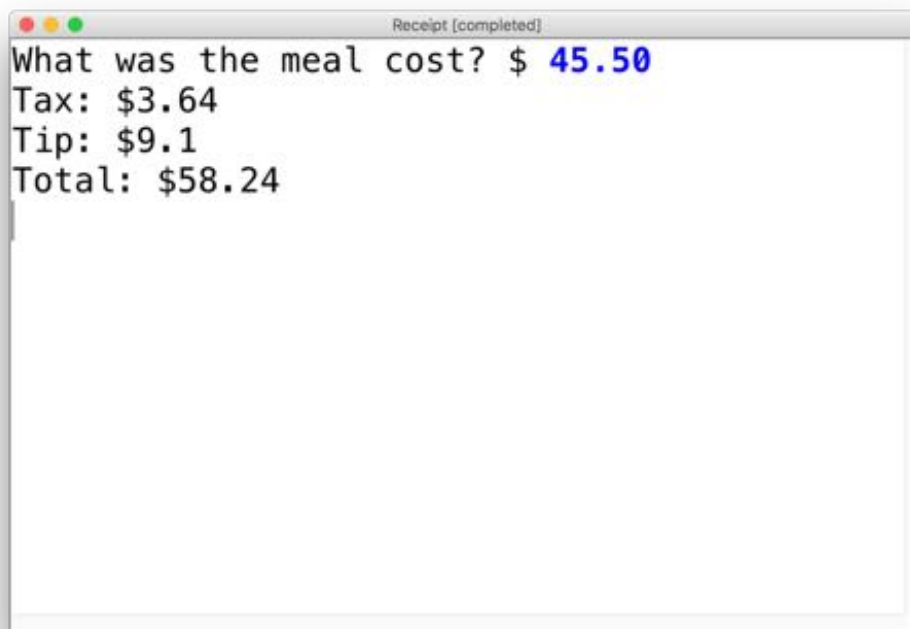
# Today's Goal

1. How do you make a box?
2. How do you see what is in a box?
3. What can you put in a box?
4. How do you change what is in a box?



# Practice 1: Receipt Program

- Let's write a ConsoleProgram that calculates the tax, tip and total bill for us at a restaurant.
- The program should ask the user for the subtotal, and then calculate and print out the tax, tip and total.



```
Receipt [completed]
What was the meal cost? $ 45.50
Tax: $3.64
Tip: $9.1
Total: $58.24
```



# If there is time “Remaining”

