Making Decisions

Lecture 4

- We know how to do some simple things with data, numbers, etc.
- We can only do so much!
  - We go from the top of our code to the bottom with no exceptions.
  - *This isn't good enough...*
- For one, we can't react to user input.
  - We **can't** write dynamic websites, programs to do our homework for us, or video games!

Motivation

Overview

- See how to build a simple Java program
- Take a look at an algorithm
  - What types of behaviors are required?
  - How can we translate a visual representation to code?
- Learn about the “if” statement
  - How it is used?
  - How are more complicated decisions made?

First Program

- Let's review using some variables to write a simple program.
- We have to start with some boilerplate code:

```
public class MyFirstProgram {
    public static void main(String[] args) {
        // This is a comment!
        // Our code goes here!
    }
}
```

```java
public class MyFirstProgram {
    public static void main(String[] args) {
        int number;
        number = 42;

        int remainder;
        remainder = number % 2;

        boolean truth = remainder == 0;

        // This will print out to the screen:
        System.out.println(truth);
    }
}
```

A Finished Program

- Save program as: `MyFirstProgram.java`
- First, we need to **compile** the program
  - Translates our code into machine language
  - Produces a `MyFirstProgram.class` file
  - Type: `javac MyFirstProgram.java`
- Now, we can run!
  - Type: `java MyFirstProgram`

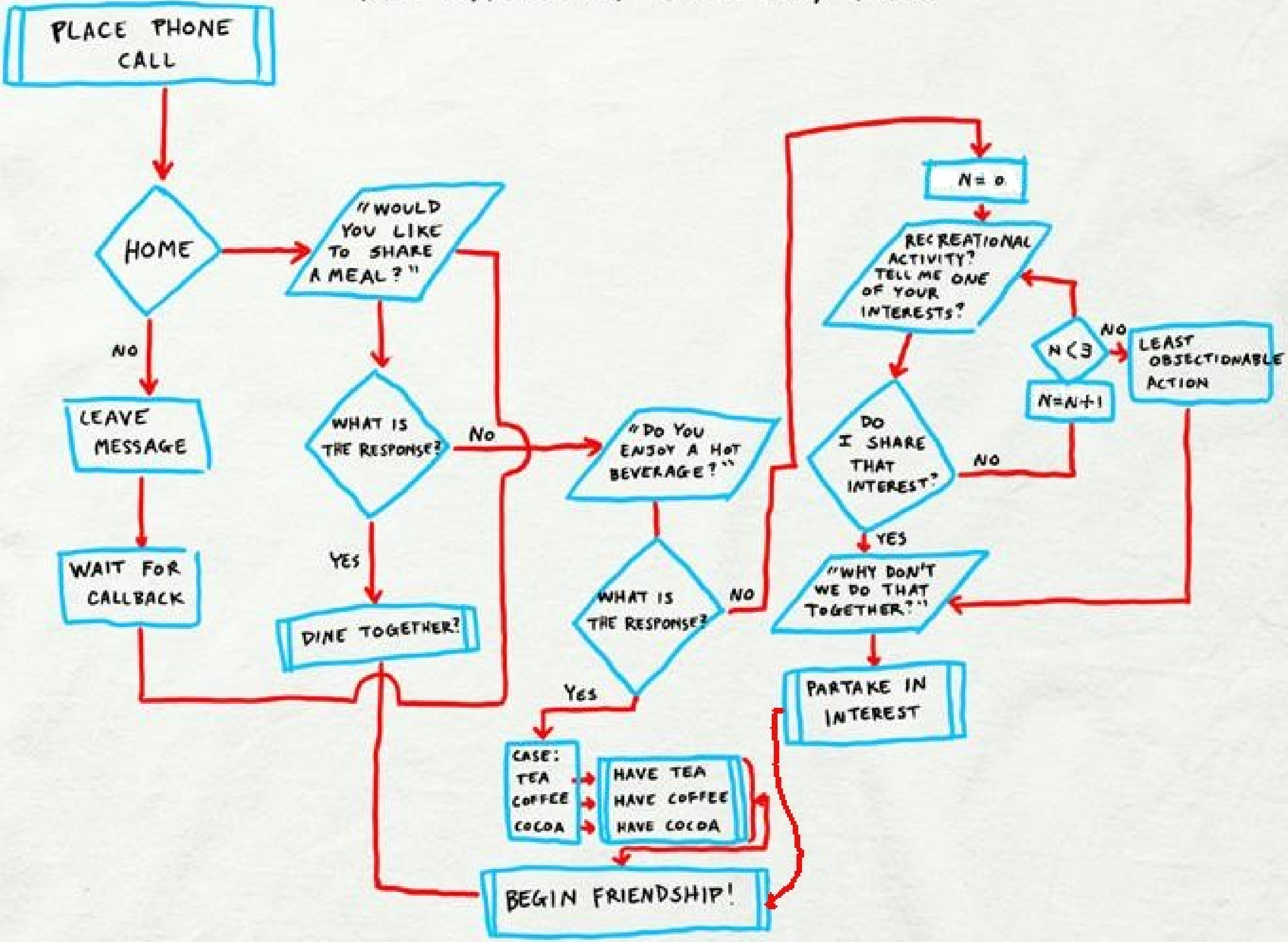
To Execute

Algorithms

- An **algorithm** is a procedure to solve a **problem**.
- A **program** is an solution for a machine based upon an algorithm.
- Example:
  - **Problem**: Sorting midterms by name
  - **Algorithm**: Finding the first... finding the second...
  - **Program**: Write the code!

THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, Ph.D



Breaking it Down

- This *algorithm*, used by geeks to acquire friends, is comprehensive!
  - It contains all elements of computability
  - If we learn how to code each of these, we can code anything!
- These elements include *procedures*, *decisions*, and *repetition*.

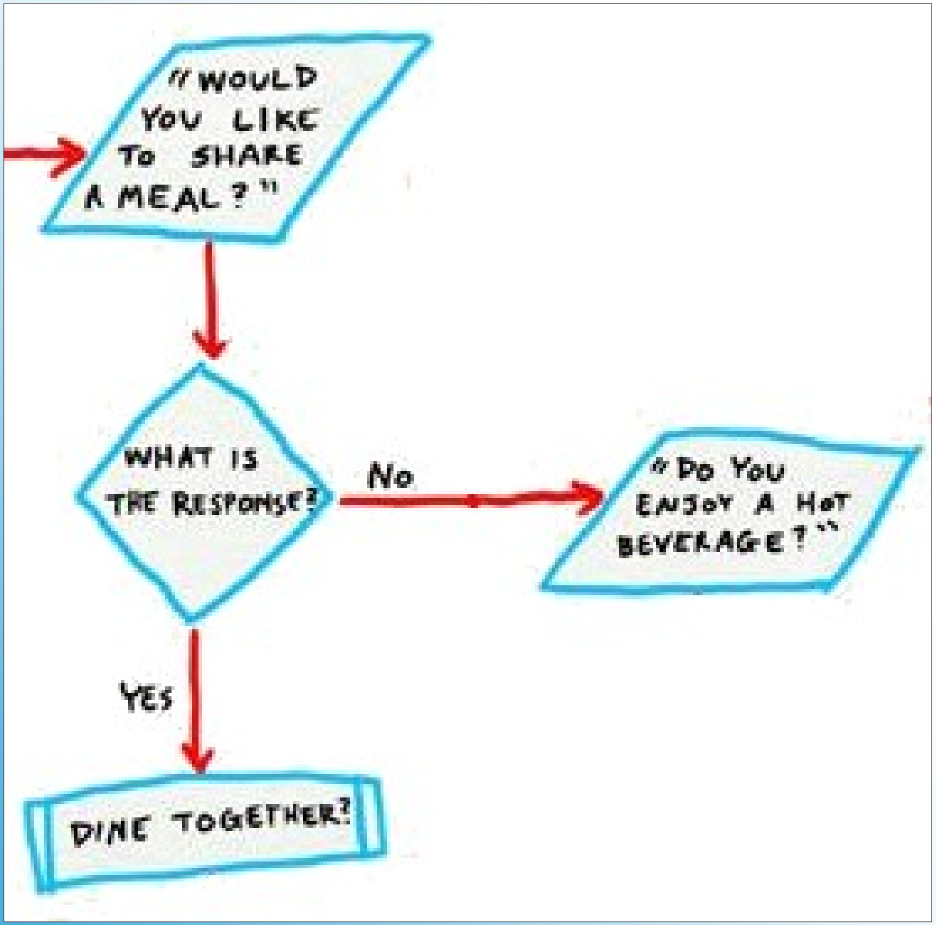
- We looked at this situation in the last lecture
  - Step-by-step instructions

```
int remainder;
remainder = number % 2;

boolean truth = remainder == 0;

// This will print out to the screen:
System.out.println(truth);
```

LEAVE MESSAGE
WAIT FOR CALLBACK

Procedures

- Now, we want to ask questions and react to the answer
- Notice that these are “Yes” or “No” questions
  - (A common trend, eh?)
- We will focus on these today!

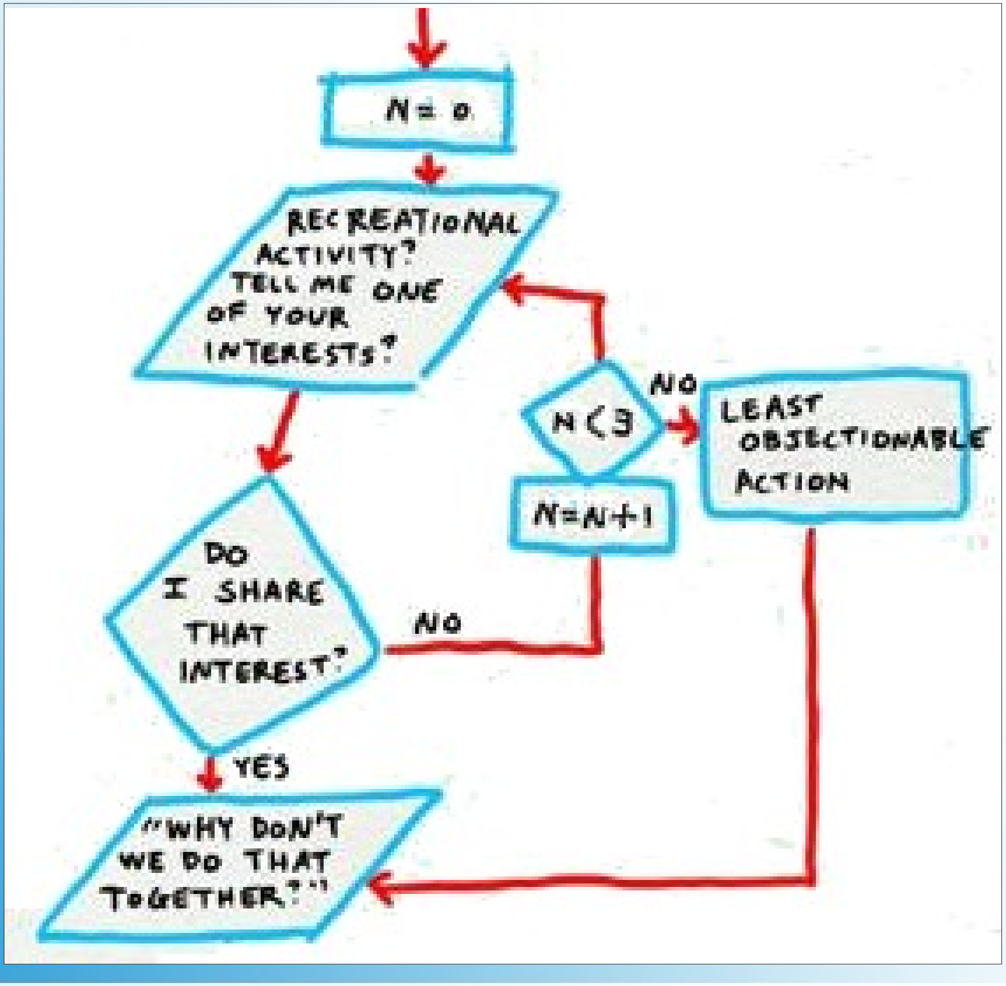

"WOULD YOU LIKE TO SHARE A MEAL?"
WHAT IS THE RESPONSE?
No
"DO YOU ENJOY A HOT BEVERAGE?"
YES
DINE TOGETHER?

Decisions

- Repeating sections of code is also important

- We can repeat while a question responds "Yes"

- We will look at this in a later class.



N = 0
RECREATIONAL ACTIVITY? TELL ME ONE OF YOUR INTERESTS?
N < 3
NO
LEAST OBJECTIONABLE ACTION
N=N+1
DO I SHARE THAT INTEREST?
NO
YES
"WHY DON'T WE DO THAT TOGETHER?"

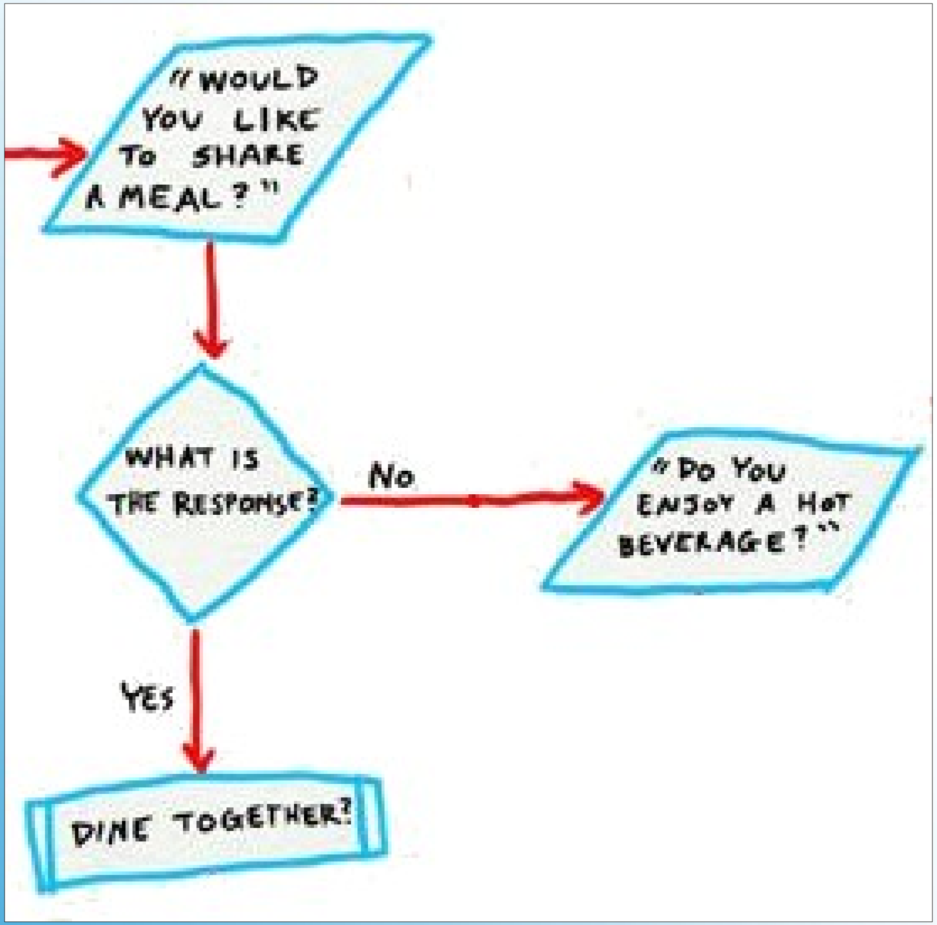
Repetition

Our Focus Today

- Today we will be looking at how we make decisions in our programs.
- We do this by using an “if” statement.
  - It reflects our intuition in the English language: *“if it is sunny, then we will have a barbeque”*

```
if (is_sunny == true) {
    // Have a barbeque!
}
```

```
// Notice the shorthand:
// We can declare and assign!
int number = 3;

if (some boolean expression) {
    // Code when 'true'
    number = number + 5;
}

// The rest of the program...

number = number + 3;
```


"WOULD YOU LIKE TO SHARE A MEAL?"
WHAT IS THE RESPONSE?
No
"DO YOU ENJOY A HOT BEVERAGE?"
YES
DINE TOGETHER!

If Statement

```
int number = 3;
int remainder = number % 2;

if (remainder == 0) {
    // Code when number is even
    number = number + 1;
}

// The rest of the program...
// What do we know about the value in “number” ?
```

- Try to draw the diagram with this small program

If Statement

- What if we want to do one thing in one case and a different thing in another?

If Statement

- What if we want to do one thing in one case and a different thing in another?
- We could write two if statements:

```
if (remainder == 0) {
    // Code when number is even
}

// Do the opposite condition
if (remainder != 0) {
    // Code when number is odd
}
```

If Statement

Else Statement

- This is very common
  - Languages typically provide a way to do this
  - In Java, this is the *else statement*

```
if (remainder == 0) {
    // Code when number is even
}
else {
    // Code when number is odd
}
```

```
int number = 3;
int remainder = number % 2;

if (remainder == 0) {
    // Code when number is even
    System.out.println(number + “ is even”);
}
else {
    // Code when number is odd
    System.out.println(number + “ is odd”);
}
```

Else Statement

Else Statement

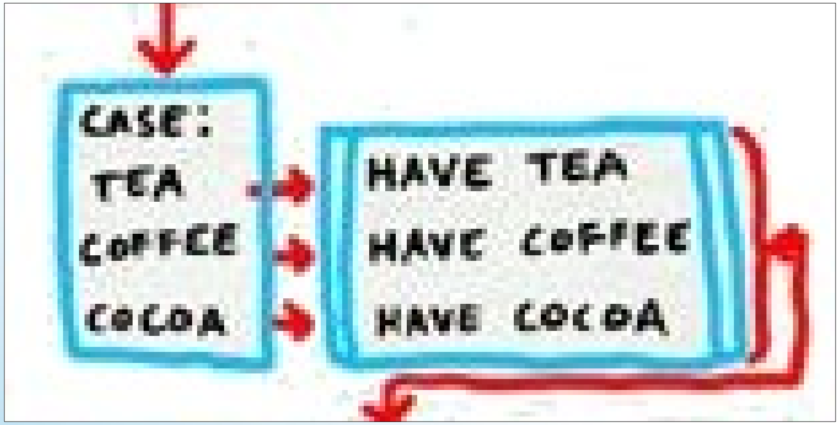
```
if (remainder == 0) {
    // Code when number is even
}
else {
    // Code when number is odd
}
```

- These situations are *mutually exclusive*
  - Only one section of code is executed

More Complex Decisions

- You could have many if statements

```
// 'case' is a String variable
if (case.equals(“tea”)) {
    // Code when input is “tea”
}
if (case.equals(“coffee”)) {
    // Code when input is “coffee”
}
if (case.equals(“cocoa”)) {
    // Code when input is “cocoa”
}
```



- Again, this is common, and is *built-in* in Java

```
if (case.equals(“tea”)) {
    // Code when input is “tea”
}
else if (case.equals(“coffee”)) {
    // Code when input is “coffee”
}
else {
    // Code when input is something else
}
```

More Complex Decisions

- How about a silly example
- A program that decides when we can eat an apple (when we have some)
    - Like an inventory program at a grocery store

```
if (apples > 0) {
    // Eat an apple
}
```

Example

- Alright. Obviously, we should do something when we do not have any:

```
if (apples > 0) {
    // Eat an apple
}
else {
    // Go to store
    // and buy more
}
```

Example

- We can also be explicit and give another condition using an “else if” statement:

```
if (apples > 0) {
    // Eat an apple
}
else if (apples == 0) {
    // Go to store
    // and buy more
}
```

Example

- In that case, an else statement is optional...
  - it sometimes doesn't make sense to have one

```
if (apples > 0) {
    // Eat an apple
}
else if (apples == 0) {
    // Go to store and buy more
}
else {
    // Question sanity!
}
```

THANK YOU, MARIO!
BUT OUR PRINCESS IS IN ANOTHER CASTLE!
INSANITY
Doing the same thing over and over and expecting a different result

Insanity: “Doing the same thing over and over and expecting a different result” (Albert Einstein)

Example