Control Flow Revisited

Chris Piech
CS106A, Stanford University

Review

Java

Making a New Variable

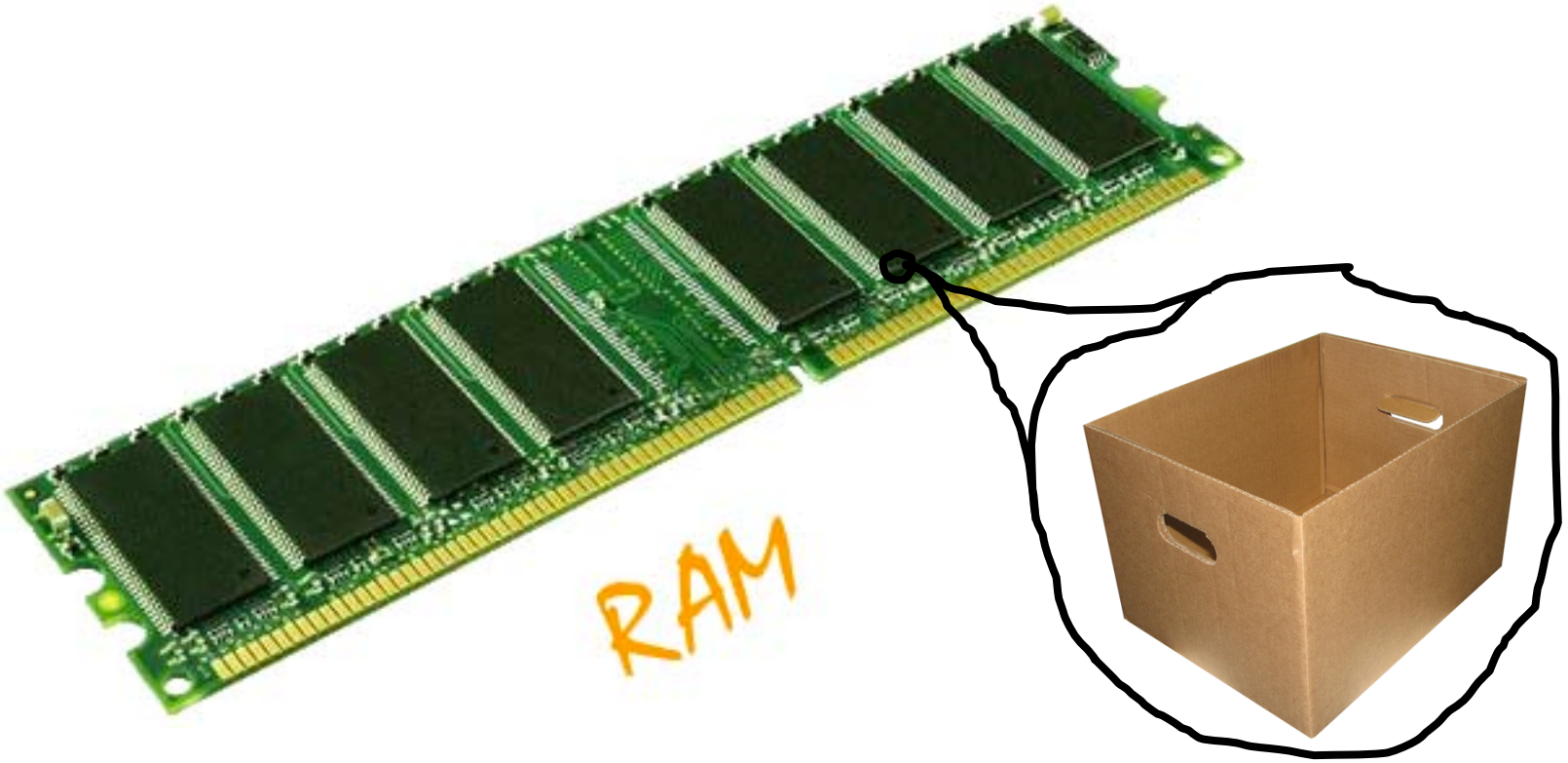
type

name

```
int age = 29;
```

Initial value

Teeny Tiny Boxes



My computer has space for about 2 billion boxes

Create, Modify, Use

```
// Create a variable, of type int
// called age with the value 29.
int age = 29;

// Modify age to be one greater.
age = age + 1;

// Use the value in age (output it)
println("age is: " + age);
```


Types

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

// real values
double fraction = 0.2;

// letters
char letter = 'c';

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

* Why is it called 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`

Example:

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

Review Example: Carbon Dating

Write a program that can turn a measurement of C14 into an estimate of age.

CarbonDating

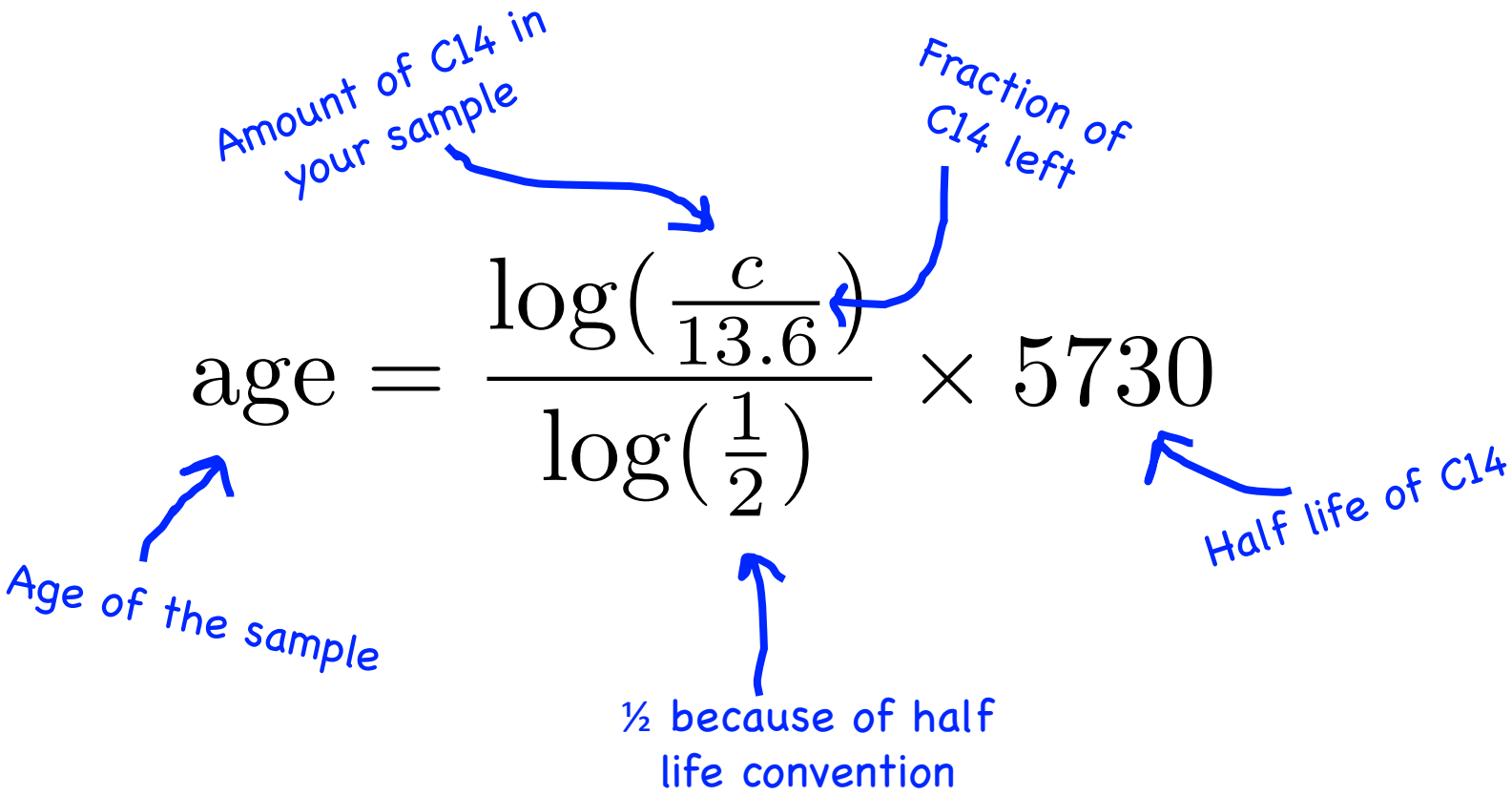
```
Radioactive molecule = C14
Halflife = 5730 years
C14 in living organisms = 13.6 dpm
-----

What is the amount of C14 remaining in your sample: 10.2
Your sample is 2378.0 years old.
```

Example: Carbon Dating

C14 = 1.2 dpm

C14 = 13.6 dpm

Carbon Dating Equation



* Some of these values are constants

** Use the function: `Math.log(` *num* `)`

End Review

Constants

- **constant**: A variable that cannot be changed after it is initialized. Declared at the top of your class, *outside of the run() method*. Can be used anywhere in that class.
- Better style – can easily change their values in your code, and they are easier to read in your code.
- Syntax:

```
private static final type name = value;
```

  - name is usually in *ALL_UPPER_CASE*

  - Examples:

```
private static final int DAYS_IN_WEEK = 7;
private static final double TAX_PERCENT = 0.08;
private static final int SSN = 658234569;
```

Receipt Program Before

```
public class Receipt extends ConsoleProgram {
  public void run() {
    double subtotal = readDouble(”Meal cost? $”);
    double tax = subtotal * 0.08;
    double tip = subtotal * 0.20;
    double total = subtotal + tax + tip;

    println("Tax : $” + tax);
    println("Tip: $” + tip);
    println(”Total: $" + total);
  }
}
```


Receipt Program After

```
public class Receipt extends ConsoleProgram {
  private static final double TAX_RATE = 0.08;
  private static final double TIP_RATE = 0.2;

  public void run() {
    double subtotal = readDouble("Meal cost? $");
    double tax = subtotal * TAX_RATE;
    double tip = subtotal * TIP_RATE;
    double total = subtotal + tax + tip;

    println("Tax : $" + tax);
    println("Tip: $" + tip);
    println("Total: $" + total);
  }
}
```


Binary Operators

+	Addition	*	Multiplication
–	Subtraction	/	Division
		%	Remainder

Today is your day, tio

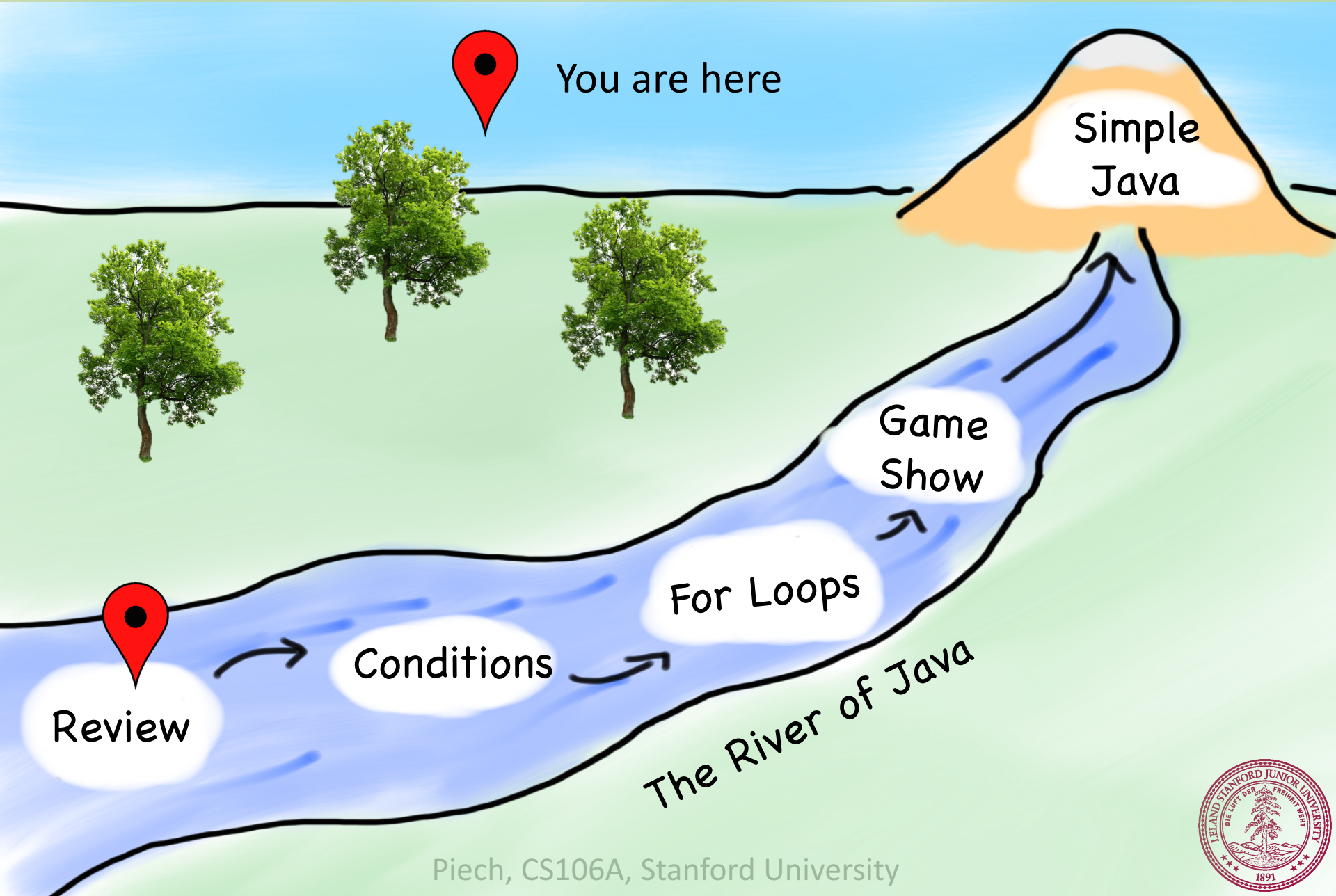
Remainder

```
// an example of the % operator
println(17 % 4);

// reads a number from the user
int num = readInt("?: ");

// stores the ones digit
int onesDigit = num % 10;

// equal to 1 if num is odd,
// 0 if num is even.
int isOdd = num % 2;
```

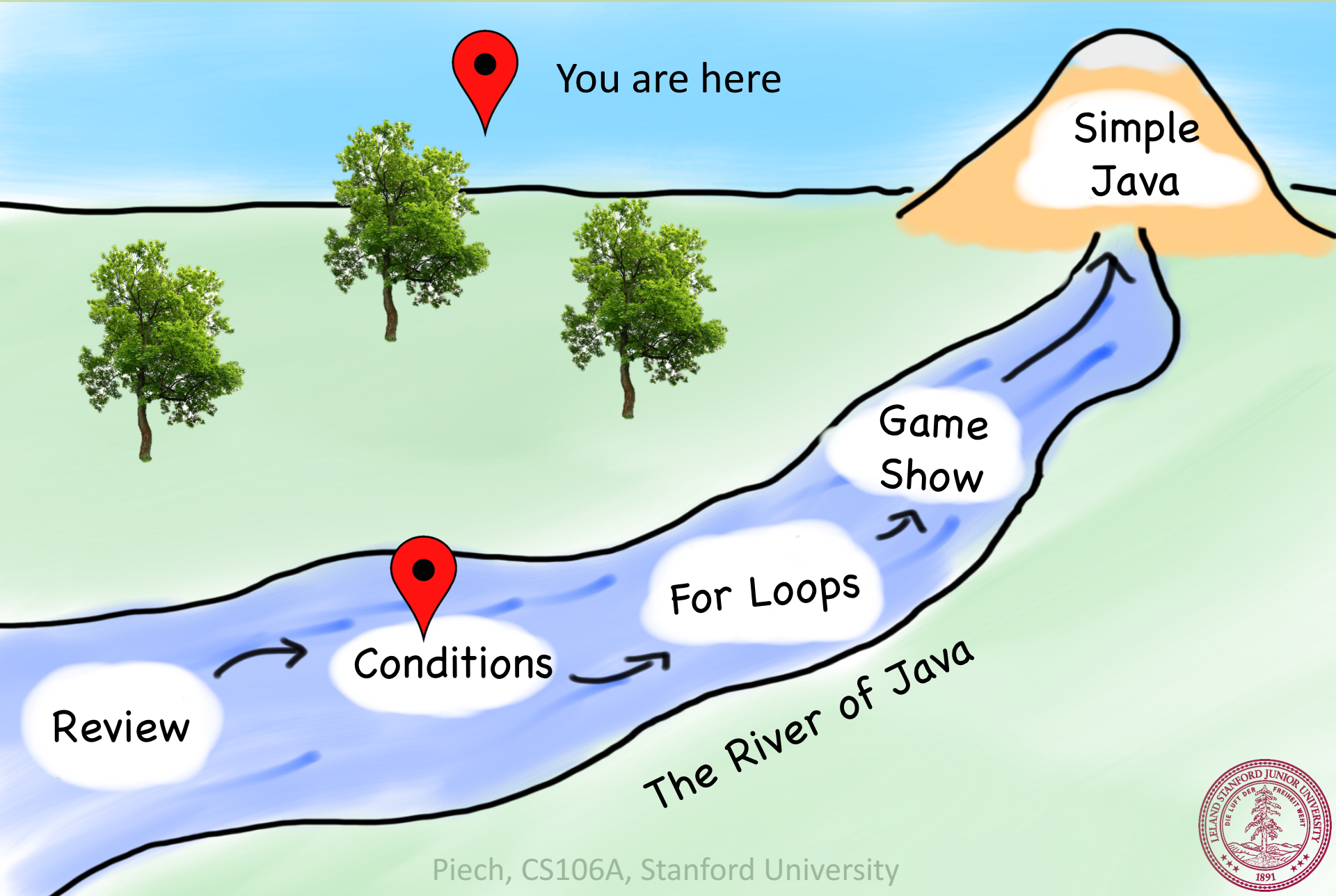

Today's Goal

1. Be able to use For / While / If in Java

Today's Route



Today's Route



While Loop in Karel

```
while(frontIsClear()) {
    body
}
```

```
if(beepersPresent()) {
    body
}
```


While Loop Redux

```
while(condition) {
    body
}
```

```
if(condition) {
    body
}
```

The condition should be a “boolean” which is either **`true`** or **`false`**

What Does This Do?

```
// read the amount of C14 from the user
double amount = readDouble("Amount of C14 in your sample: ");

// use the half life formula to calculate the age
double fractionLeft = amountLeft / LIVING_C14;
double age = Math.log(fractionLeft) / Math.log(0.5) * HALF_LIFE;
println("Your sample is " + age + " years old.");
```

* It calculates the age of a C14 sample

What Does This Do?

Before repeating the body, check if this statement evaluates to true

```
while(true) {
    // read the amount of C14 from the user
    double amount = readDouble("Amount of C14 in your sample: ");

    // use the half life formula to calculate the age
    double fractionLeft = amountLeft / LIVING_C14;
    double age = Math.log(fractionLeft) / Math.log(0.5) * HALF_LIFE;
    println("Your sample is " + age + " years old.");

    // add an extra line between queries
    println("");
}
```

* It repeatedly calculates the age of a C14 sample

Booleans

$$1 < 2$$


LELAND STANFORD JUNIOR UNIVERSITY
DIE LUFT DER FREIHEIT WEHT
1891

Booleans

$1 < 2$

`true`

Comparison Operators

Operator	Meaning	Example	Value
==	equals	`1 + 1 == 2`	`true`
!=	does not equal	`3.2 != 2.5`	`true`
<	less than	`10 < 5`	`false`
>	greater than	`10 > 5`	`true`
<=	less than or equal to	`126 <= 100`	`false`
>=	greater than or equal to	`5.0 >= 5.0`	`true`

* All have equal precedence

Comparison Operators

Operator	Meaning	Example	Value
==	equals	`1 + 1 == 2`	`true`
!=	does not equal	`3.2 != 2.5`	`true`
<	less than	`10 < 5`	`false`
>	greater than	`10 > 5`	`true`
<=	less than or equal to	`126 <= 100`	`false`
>=	greater than or equal to	`5.0 >= 5.0`	`true`

* All have equal precedence

Comparison Operators

```
if (1 < 2) {
   println("1 is less than 2!");
}
```

```
int num = readInt("Enter a number: ");
if (num == 0) {
   println("That number is 0!");
} else {
   println("That number is not 0.");
}
```


If Else Revisited

```
int num = readInt("Enter a number: ");
if (num == 0) {
   println("Your number is 0 ");
} else {
   if (num > 0) {
      println("Your number is positive");
   } else {
      println("Your number is negative");
   }
}
```


Else If

```
int num = readInt("Enter a number: ");
if (num == 0) {
   println("Your number is 0 ");
} else if (num > 0) {
   println("Your number is positive");
} else {
   println("Your number is negative");
}
```


Example: Sentinel Loops

- **sentinel**: A value that signals the end of user input.
  - **sentinel loop**: Repeats until a sentinel value is seen.

- Example: Write a program that prompts the user for numbers until the user types -1, then output the sum of the numbers.
  - In this case, -1 is the sentinel value.

```
Type a number: 10
Type a number: 20
Type a number: 30
Type a number: -1
Sum is 60
```


Example: Sentinel Loops

```
// fencepost problem!
// ask for number - post
// add number to sum - fence


int sum = 0;
int num = readInt("Enter a number: ");
while (num != -1) {
   sum += num;
   num = readInt("Enter a number: ");
}
println("Sum is " + sum);
```


Example: Sentinel Loops

```
// Solution #2
// harder to see loop end condition here

int sum = 0;
while (true) {
   int num = readInt("Enter a number: ");
   if (num == -1) {
      break; // immediately exits loop
   }
   sum += num;
}
println("Sum is " + sum);
```


Guess My Number

GuessMyNumber

```
I am thinking of a number between 0 and 99...
Enter a guess: 50
Your guess is too high

Enter a new number: 25
Your guess is too low

Enter a new number: 40
Your guess is too low

Enter a new number: 45
Your guess is too low

Enter a new number: 48
Congrats! The number was: 48
```

Guess My Number

```
int secretNumber = SECRET;
println("I am thinking of a number between 0 and 99...");
int guess = readInt("Enter a guess: ");
// true if guess is not equal to secret number
while(guess != secretNumber) {
    // true if guess is less than secret number
    if(guess < secretNumber) {
        println("Your guess is too low");
    } else {
        println("Your guess is too high");
    }
    println(""); // an empty line
    guess = readInt("Enter a new number: ");
}
println("Congrats! The number was: " + secretNumber);
```

Logical Operators

In order of precedence:

Operator	Description	Example	Result
!	not	`!(2 == 3)`	`true`
&&	and	`(2 == 3) && (-1 < 5)`	`false`
||	or	`(2 == 3) || (-1 < 5)`	`true`

Cannot "chain" tests as in algebra; use && or || instead

```
// assume x is 15
2 <= x <= 10
true   <= 10
Error!
```

```
// correct version
2 <= x && x <= 10
true   && false
false
```

Precedence Madness

Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
```


Precedence Madness

Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
5 * 7 >= 3 + 5 * 6 && !false
```


Precedence Madness

Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
5 * 7 >= 3 + 5 * 6 && !false
5 * 7 >= 3 + 5 * 6 && true
```


Precedence Madness

Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
5 * 7 >= 3 + 5 * 6 && !false
5 * 7 >= 3 + 5 * 6 && true
35    >= 3 + 30 && true
```


Precedence Madness

Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
5 * 7 >= 3 + 5 * 6 && !false
5 * 7 >= 3 + 5 * 6 && true
35    >= 3 + 30 && true
35    >= 33 && true
```


Precedence Madness

Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
5 * 7 >= 3 + 5 * 6 && !false
5 * 7 >= 3 + 5 * 6 && true
35    >= 3 + 30 && true
35    >= 33 && true
true && true
```


Precedence Madness

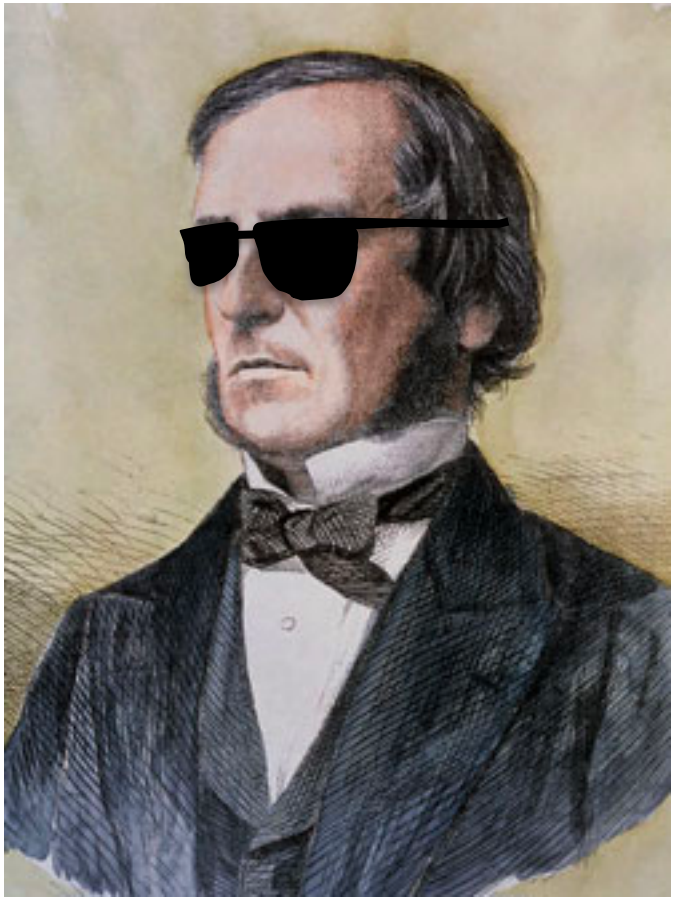
Precedence:

! > arithmetic > comparison > logical

```
5 * 7 >= 3 + 5 * (7 - 1) && !false
5 * 7 >= 3 + 5 * 6 && !false
5 * 7 >= 3 + 5 * 6 && true
35    >= 3 + 30 && true
35    >= 33 && true
true && true
true
```

Never write code like this ☺

George Boole



English Mathematician 1815 – 1864

Boole died of being too cool

Boolean Variables

```
// Store expressions that evaluate to true/false
boolean x = 1 < 2;     // true
boolean y = 5.0 == 4.0;   // false
```


Boolean Variables

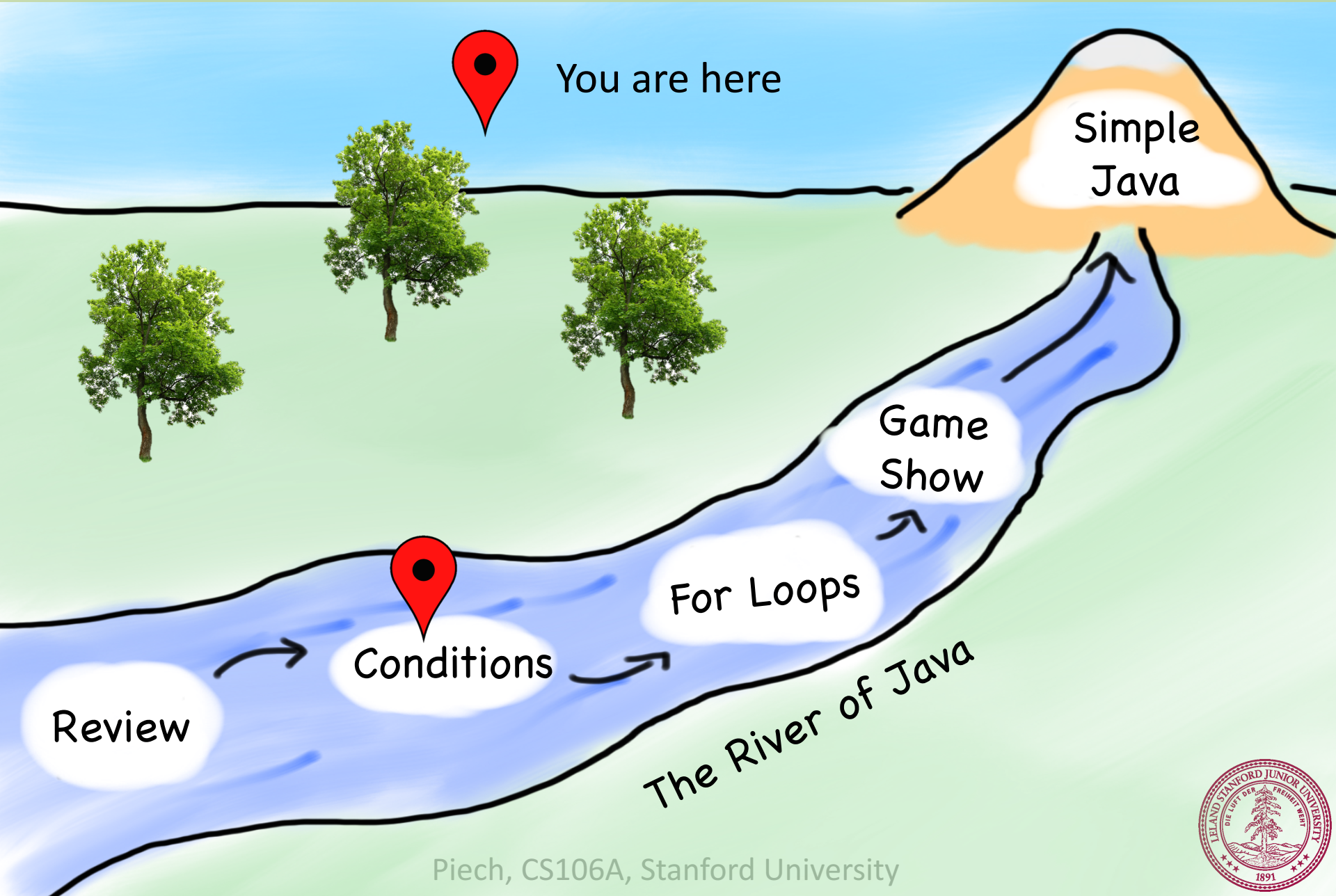
```
// Store expressions that evaluate to true/false
boolean x = 1 < 2;     // true
boolean y = 5.0 == 4.0;   // false

// Directly set to true/false
boolean isFamilyVisiting = true;
boolean isRaining = false;
```


Boolean Variables

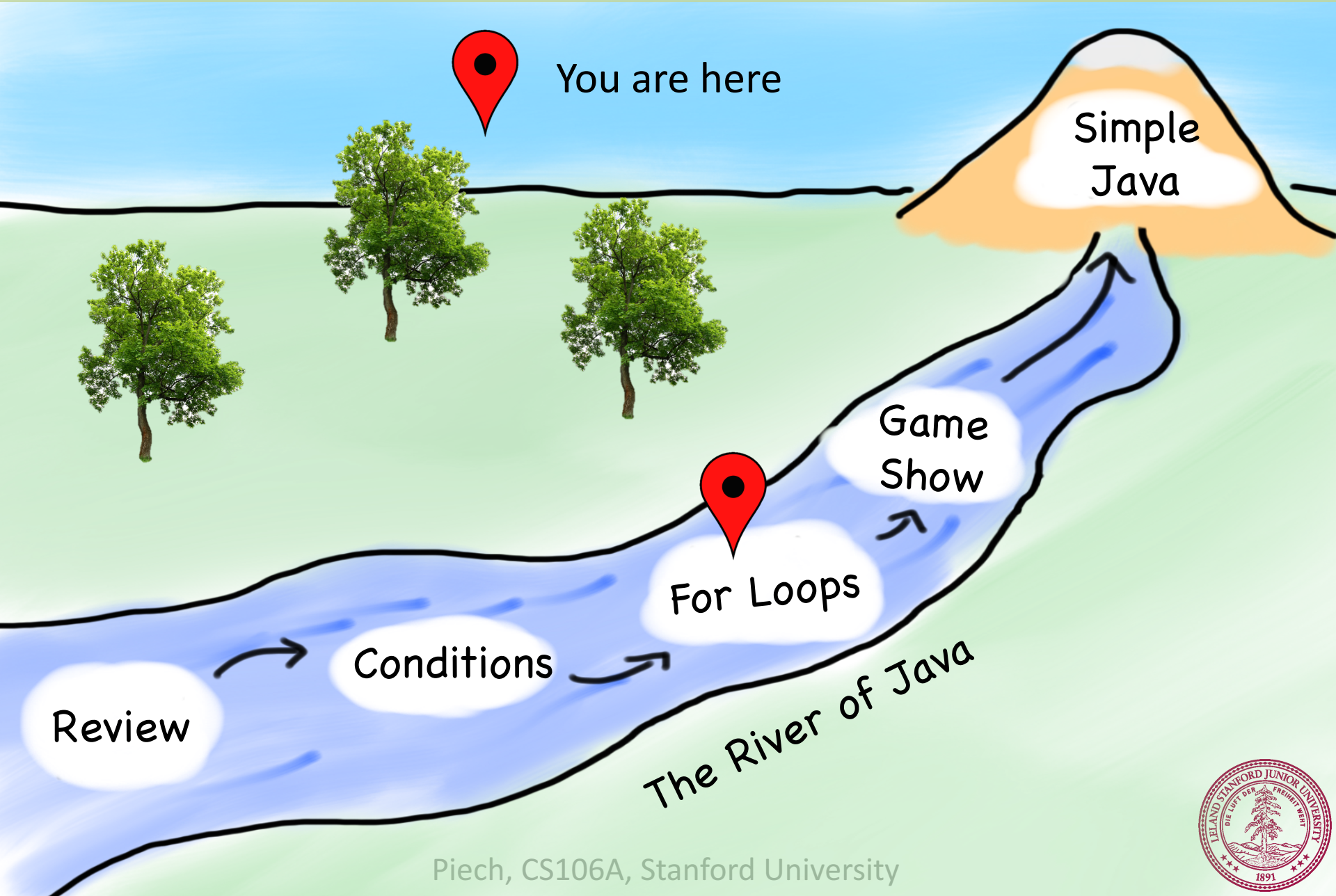
```
// Store expressions that evaluate to true/false
boolean x = 1 < 2;     // true
boolean y = 5.0 == 4.0;   // false

// Directly set to true/false
boolean isFamilyVisiting = true;
boolean isRaining = false;

// Ask the user a true/false (yes/no) question
boolean playAgain = readBoolean("Play again?", "y", "n");
if (playAgain) {
...
```

*know your logical precedence

Today's Route



Today's Route

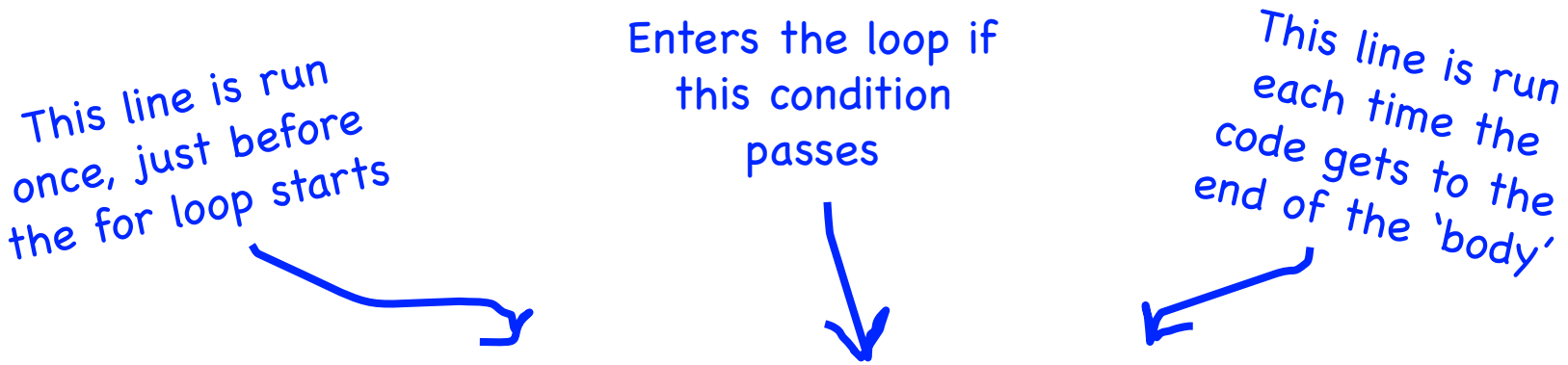


How would you println “Stanford rocks socks” 100 times

For Loop Redux

```
public void run() {
   for(int i = 0; i < 100; i++) {
      println("Stanford rocks socks!");
   }
}
```

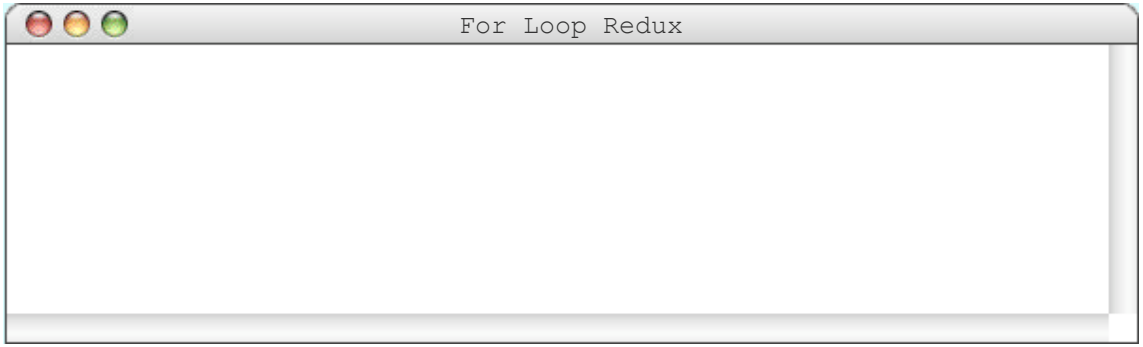

For Loop Redux




```
for(int i = 0; i < 100; i++) {
   println("Stanford rocks socks!");
}
```

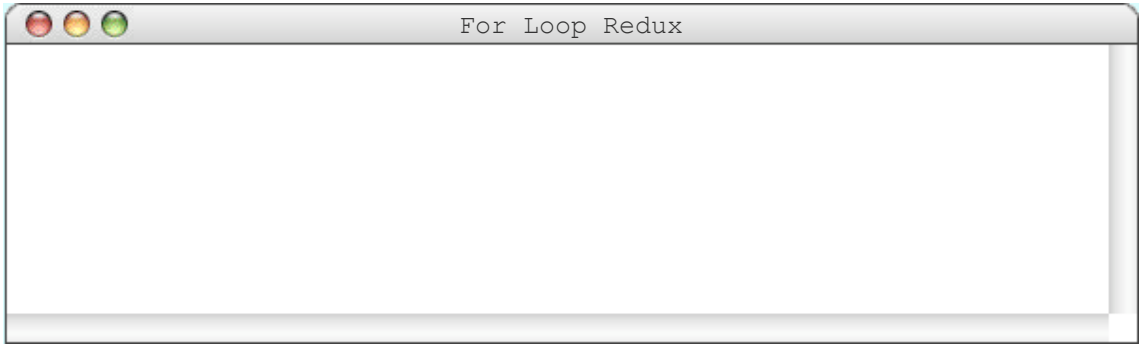
For Loop Redux

```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

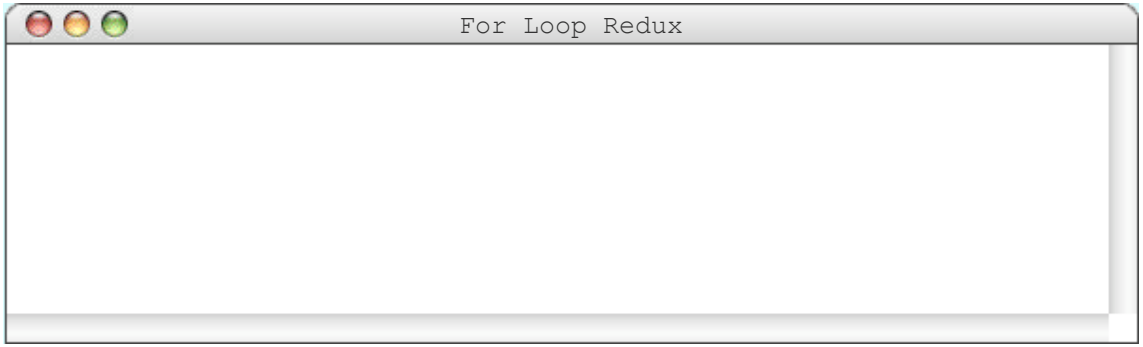
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i	0

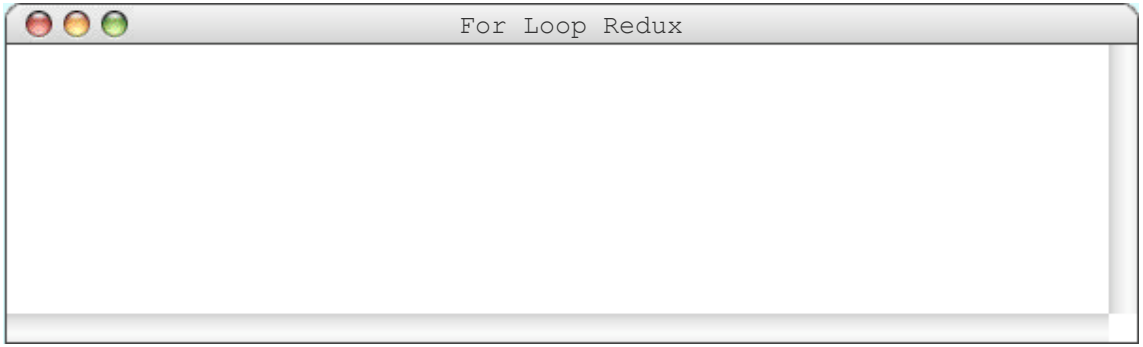
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i	0

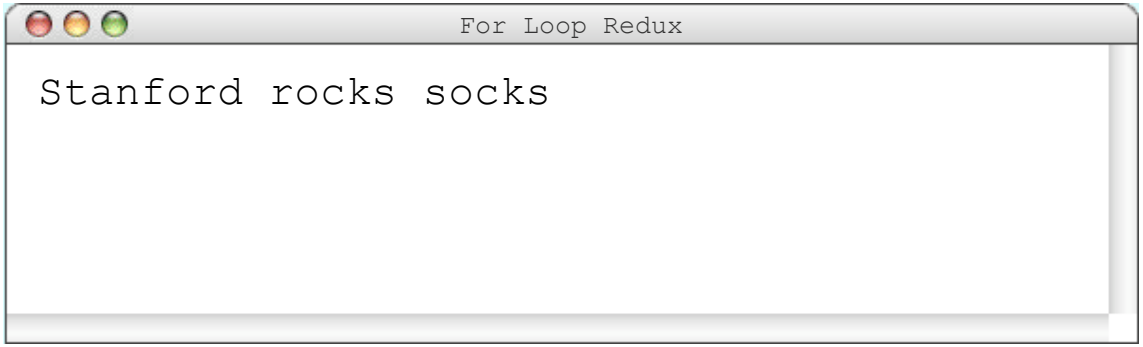
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i	0

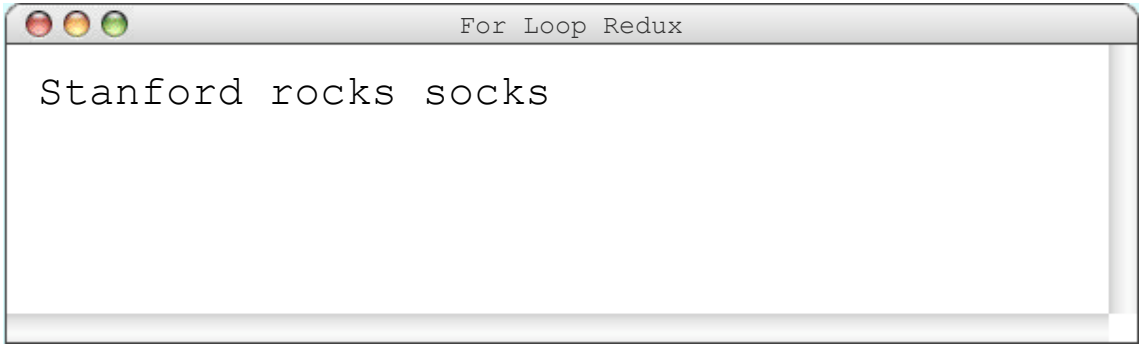
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i 1

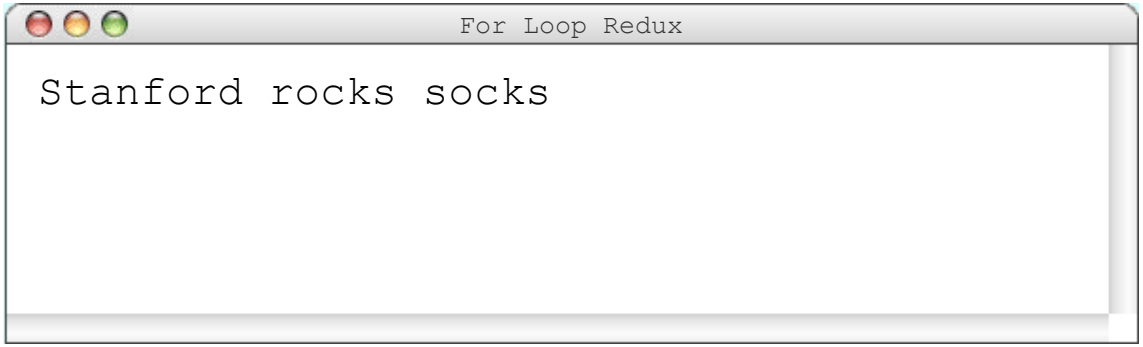
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i | 1

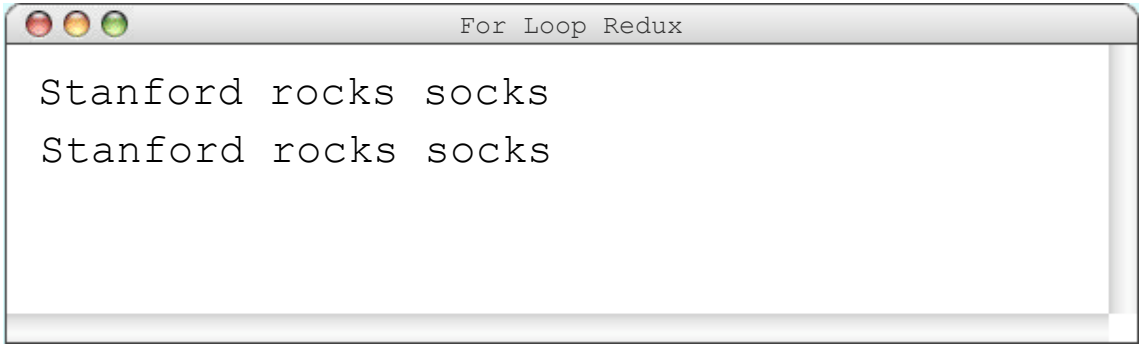
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i | 1

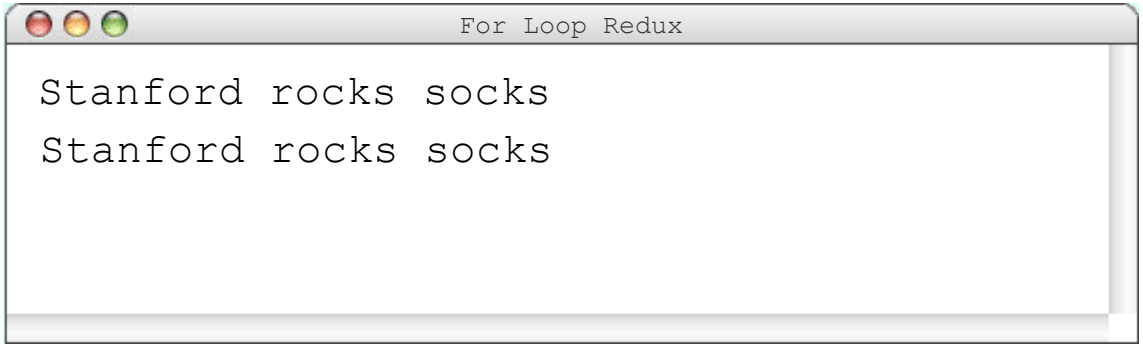
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i 2

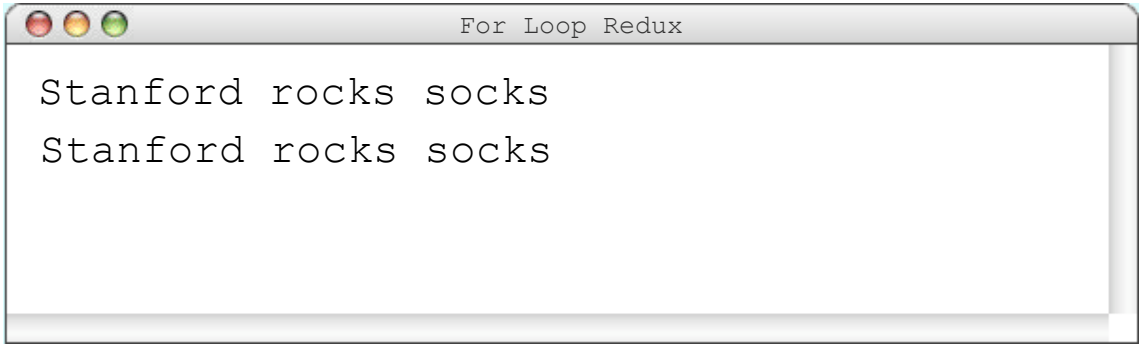
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i	2

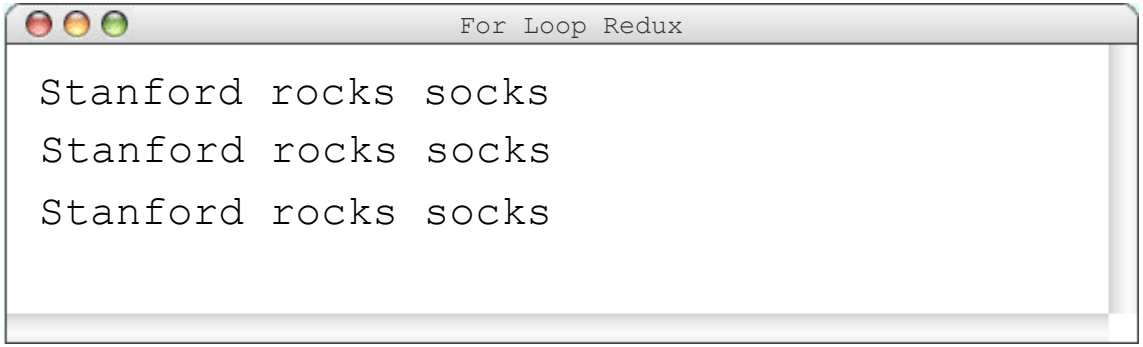
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i | 2

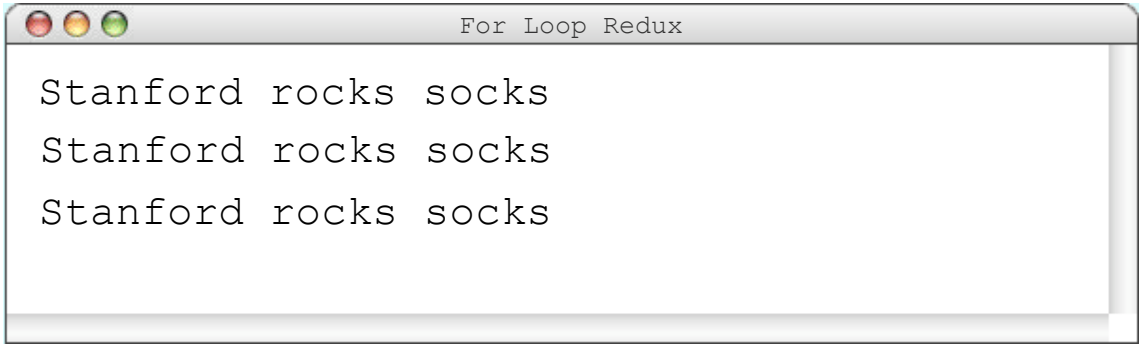
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

i 3

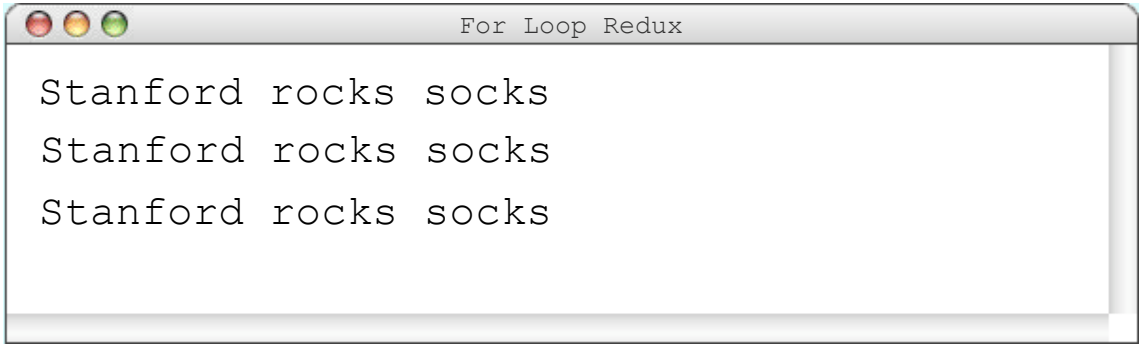
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

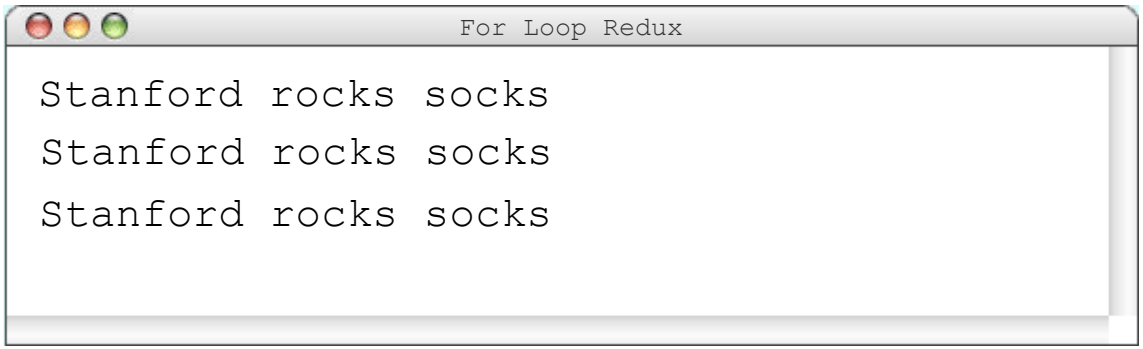
i	3

```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



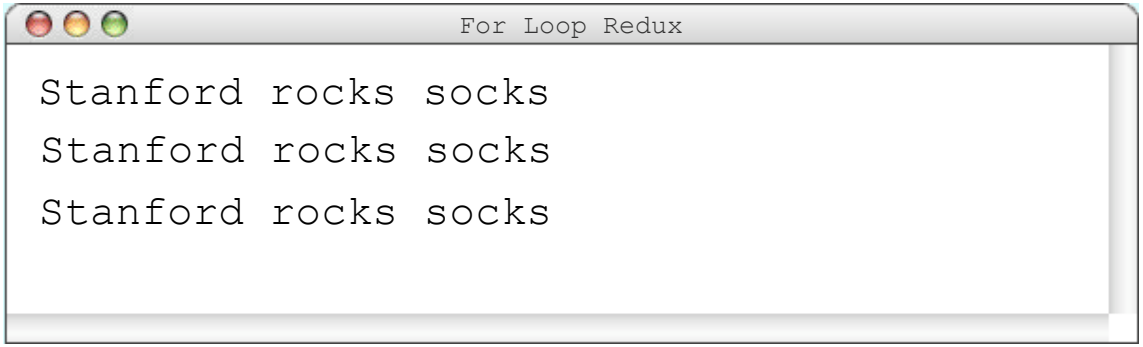
For Loop Redux

```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



For Loop Redux

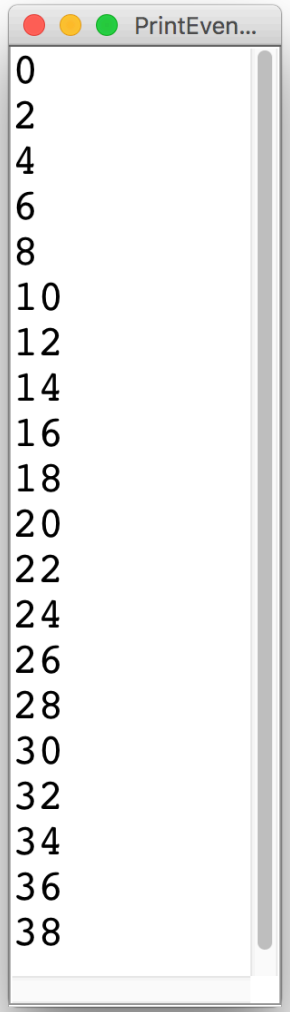
```
for(int i = 0; i < 3; i++) {
   println("Stanford rocks socks!");
}
```



You can use the for loop variable

How would you println the first 100 even numbers?

Printing Even Numbers

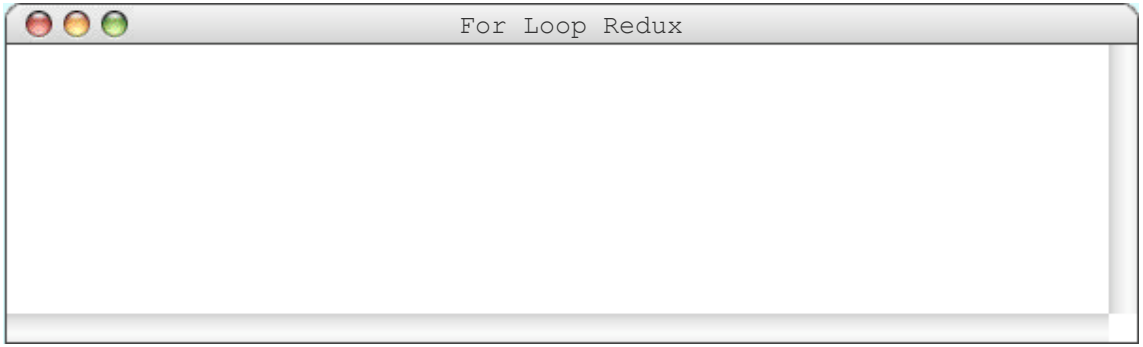


Printing Even Numbers

```
for(int i = 0; i < NUM_NUMS; i++) {
   println(i * 2);
}
```

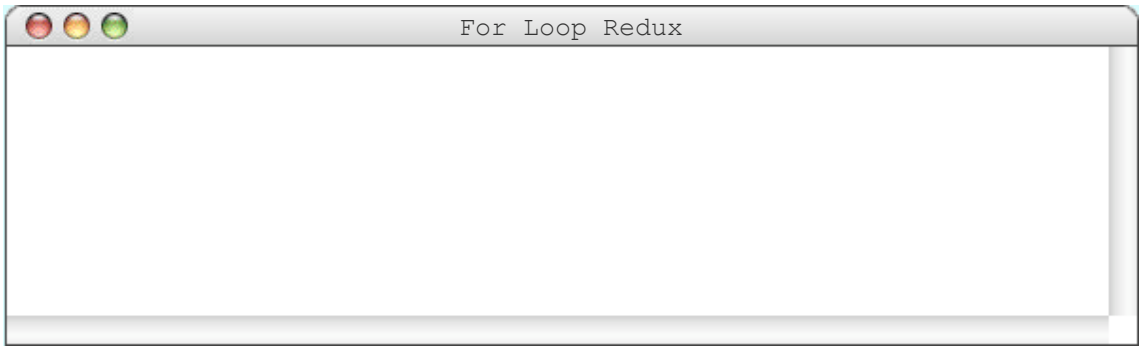

Printing Even Numbers

```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

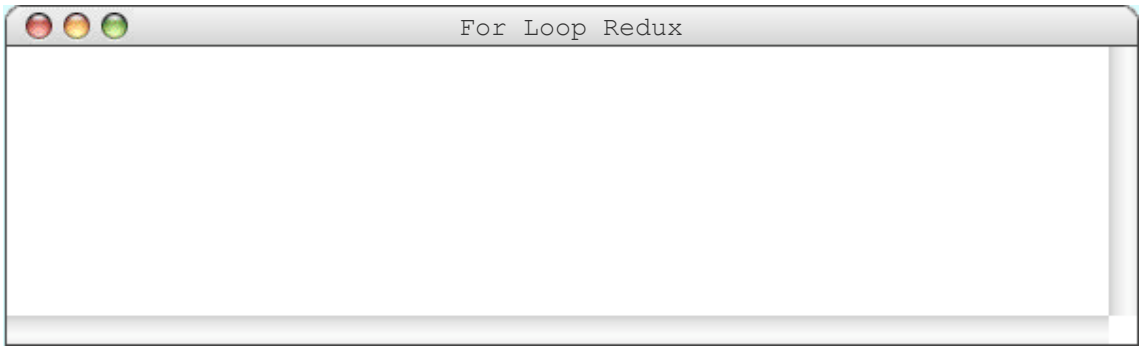
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	0

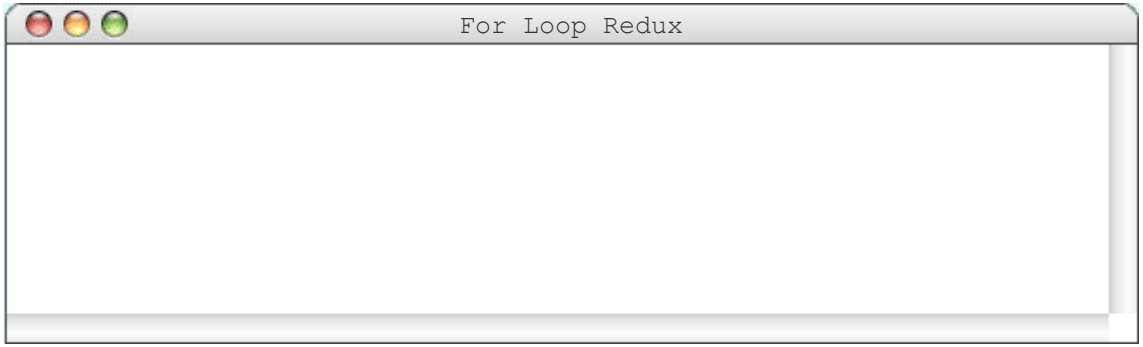
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	0

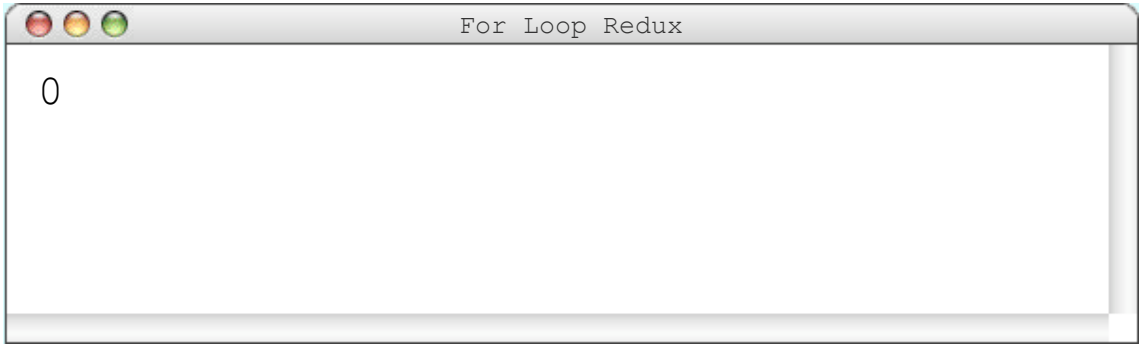
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	0

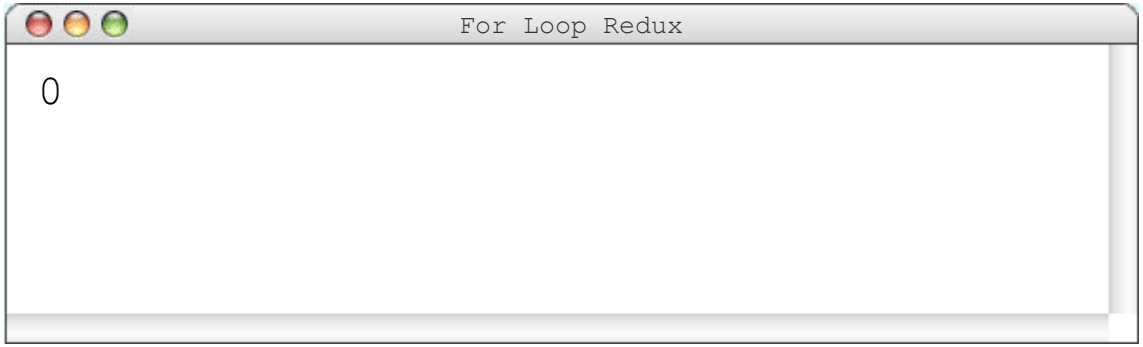
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i 1

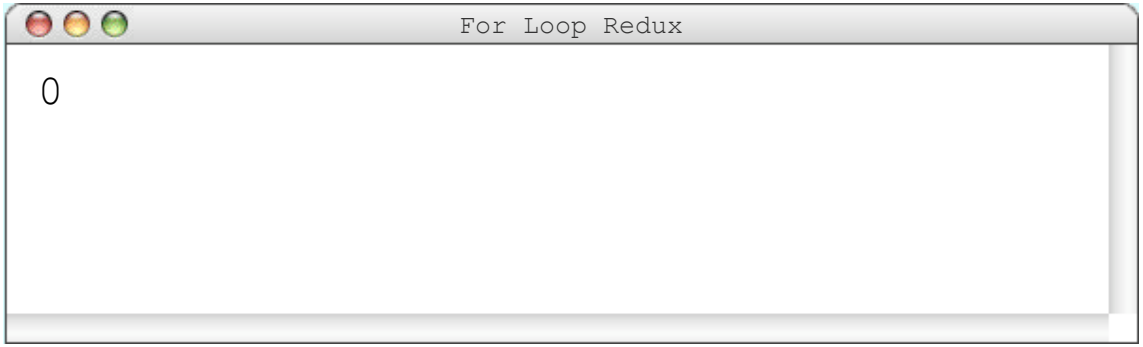
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	1

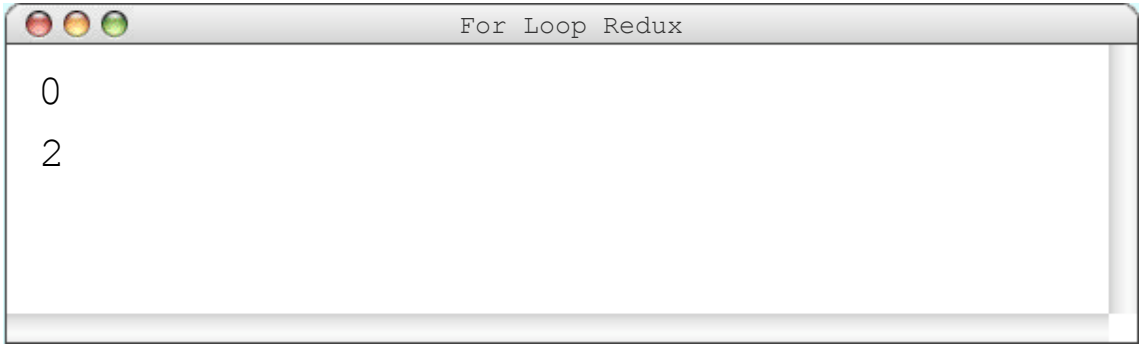
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i | 1

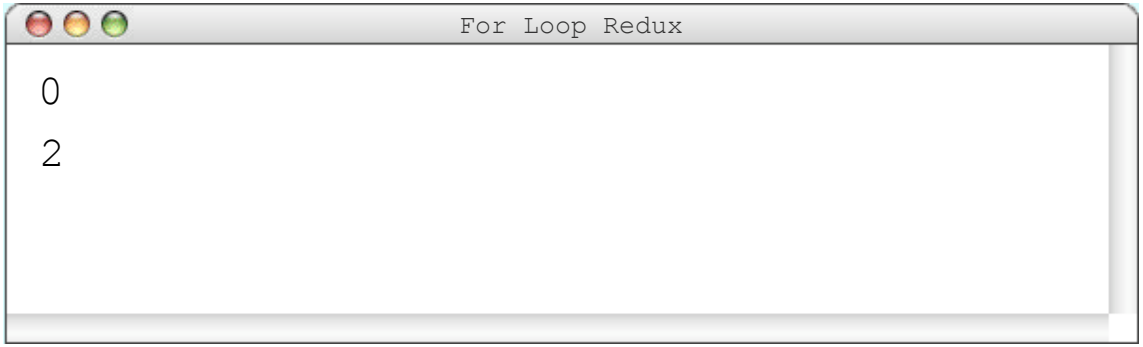
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i 2

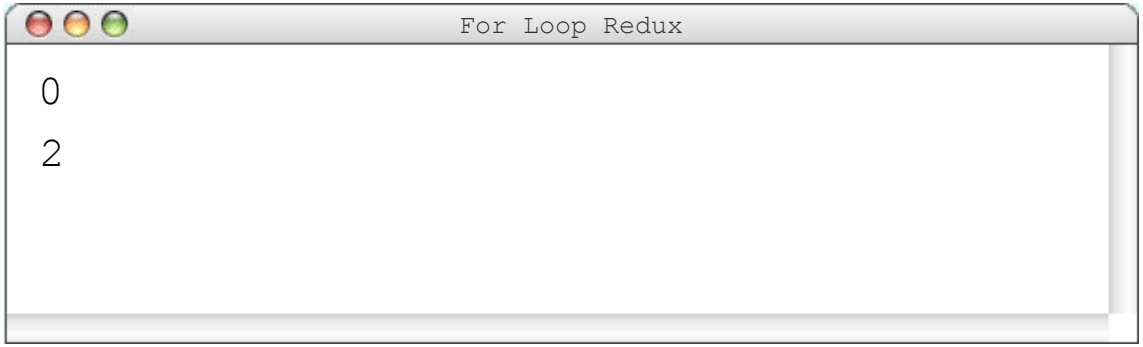
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	2

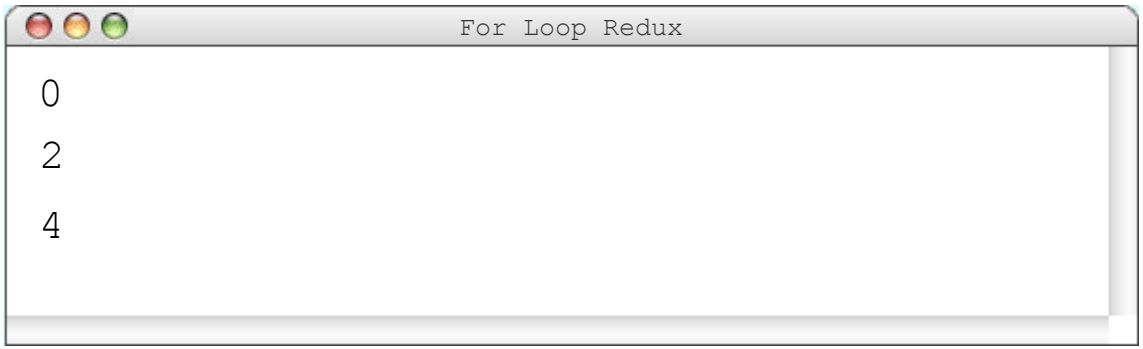
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	2

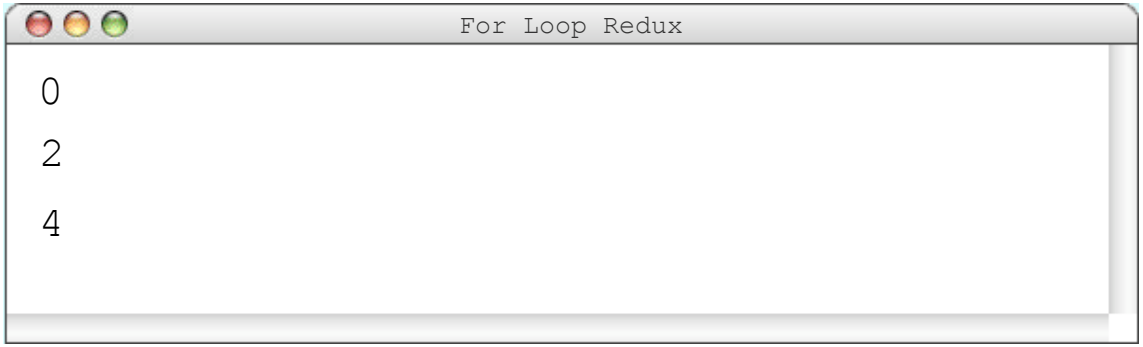
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	3

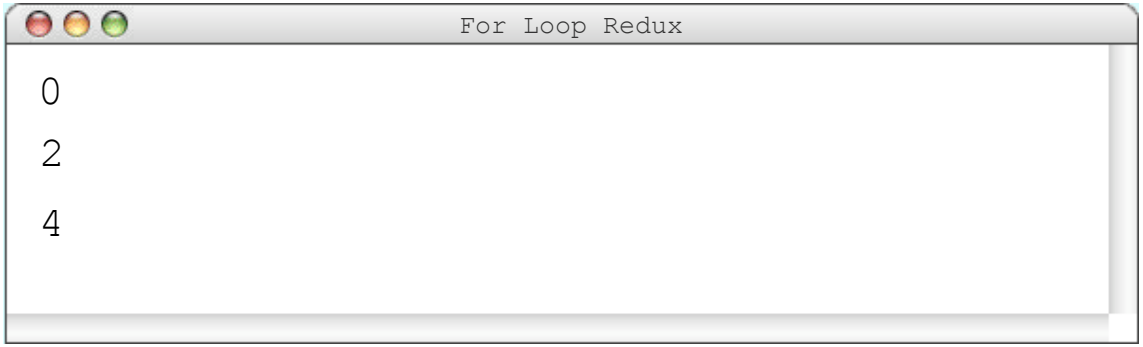
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

i	3

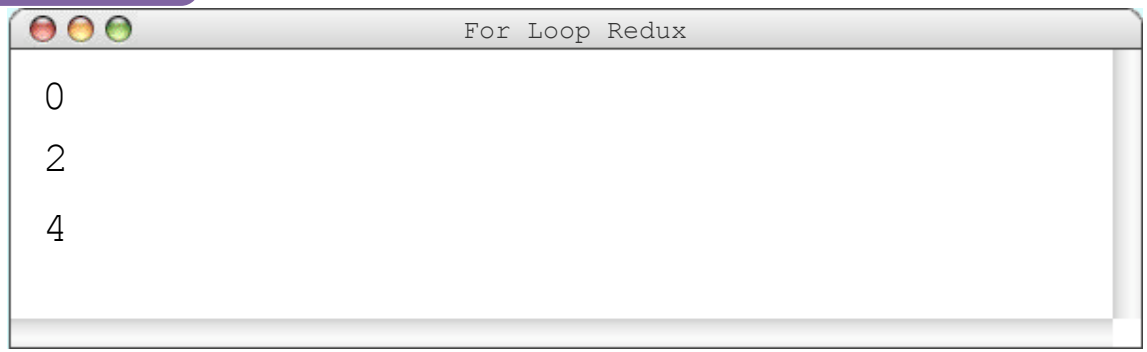
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



Printing Even Numbers

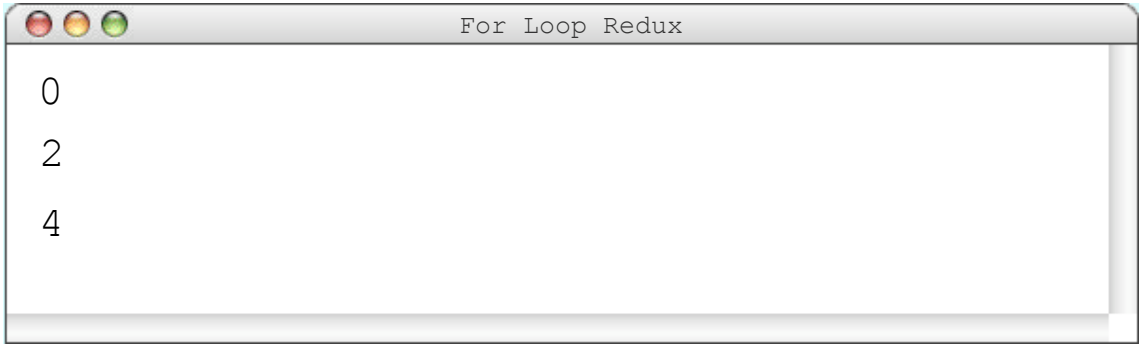
i	3

```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```

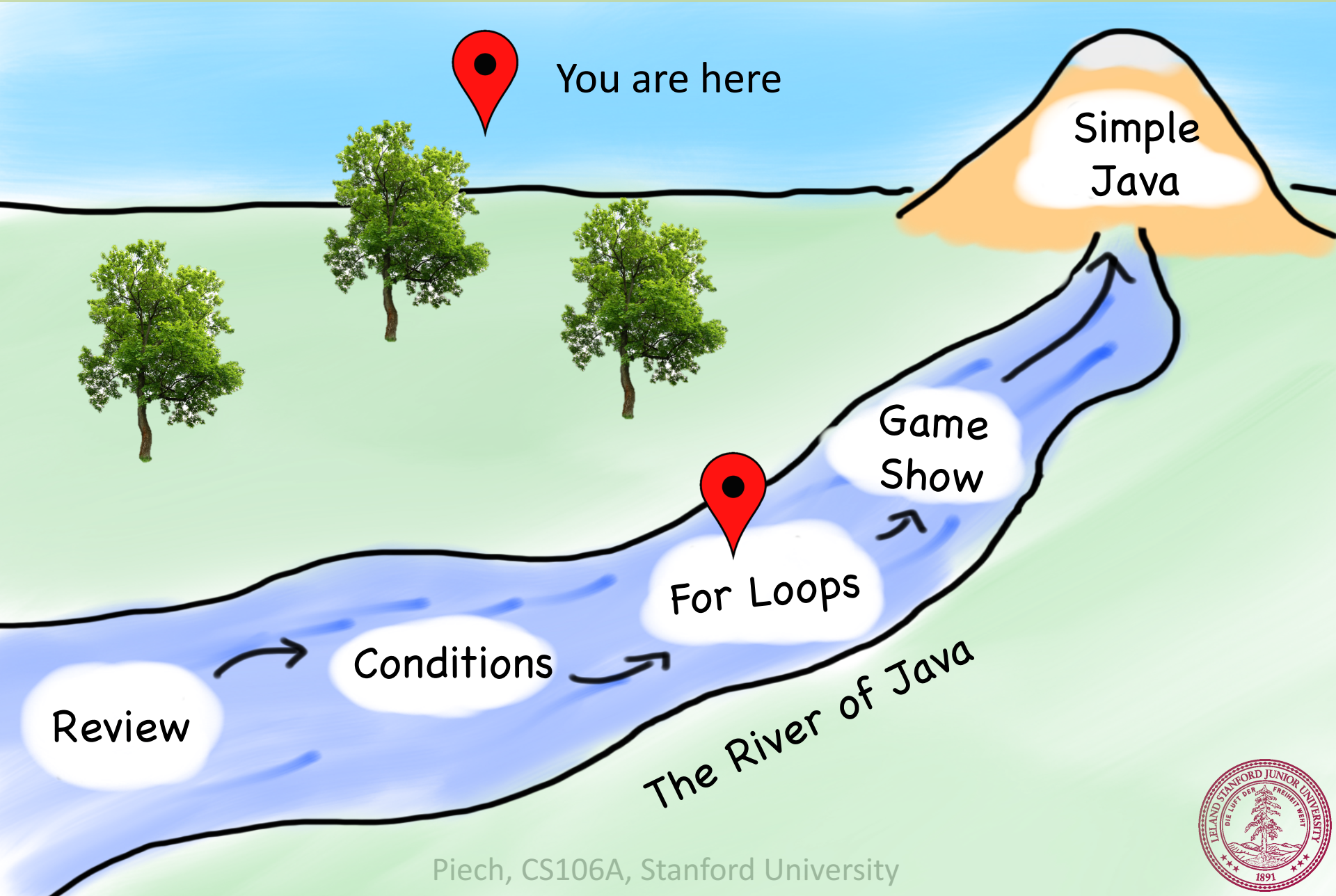


Printing Even Numbers

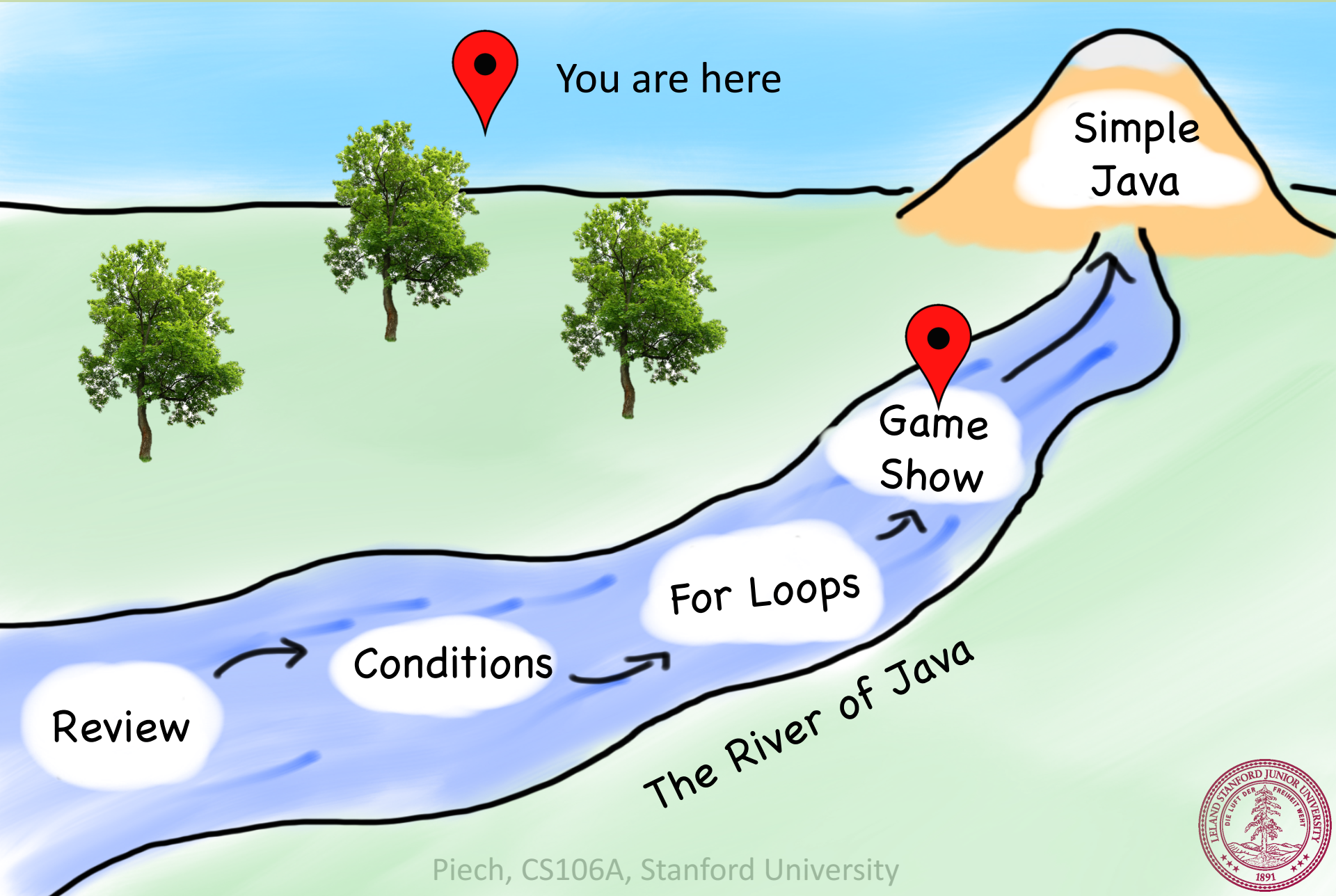
```
for(int i = 0; i < 3; i++) {
   println(i * 2);
}
```



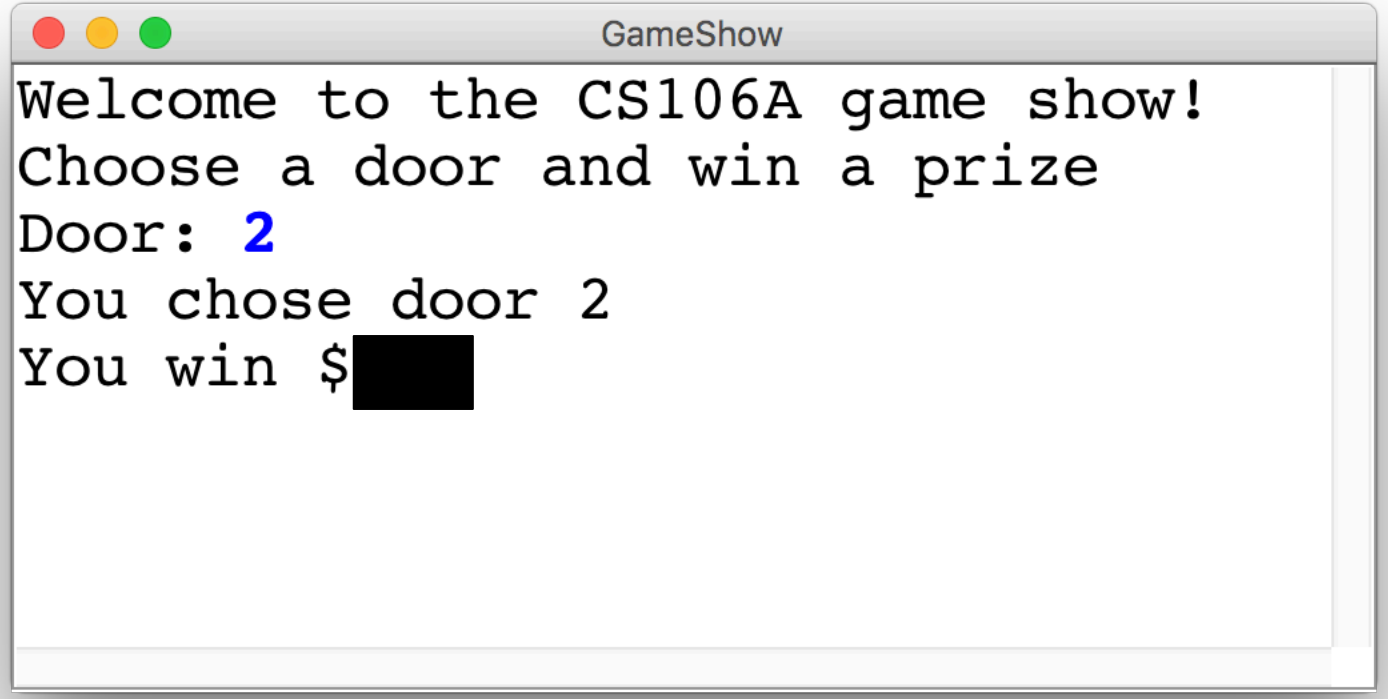
Today's Route



Today's Route



Game Show


GameShow
Welcome to the CS106A game show!
Choose a door and win a prize
Door: 2
You chose door 2
You win $

Choose a Door

```
int door = readInt("Door: ");
// while the input is invalid
while(door < 1 || door > 3) {
   // tell the user the input was invalid
   println("Invalid door!");
   // ask for a new input
   door = readInt("Door: ");
}
```

`||`	or
`&&`	and

The Door Logic

```
int prize = 4;
if(door == 1) {
   prize = 2 + 9 / 10 * 100;
} else if(door == 2) {
   boolean locked = prize % 2 != 0;
   if(!locked) {
      prize += 6;
   }
} else if(door == 3) {
   prize++;
}
```


The Door Logic

```
int prize = 4;
if(door == 1) {
   prize = 2 + 9 / 10 * 100;
} else if(door == 2) {
   boolean locked = prize % 2 != 0;
   if(!locked) {
      prize += 6;
   }
} else if(door == 3) {
   prize++;
}
```


The Door Logic

```
int prize = 4;
if(door == 1) {
   prize = 2 + 9 / 10 * 100;
} else if(door == 2) {
   boolean locked = prize % 2 != 0;
   if(!locked) {
      prize += 6;
   }
} else if(door == 3) {
   prize++;
}
```


The Door Logic

```
int prize = 4;
if(door == 1) {
   prize = 2 + 9 / 10 * 100;
} else if(door == 2) {
   boolean locked = prize % 2 != 0;
   if(!locked) {
      prize += 6;
   }
} else if(door == 3) {
   prize++;
}
```


The Door Logic

```
int prize = 4;
if(door == 1) {
   prize = 2 + 9 / 10 * 100;
} else if(door == 2) {
   boolean locked = prize % 2 != 0;
   if(!locked) {
      prize += 6;
   }
} else if(door == 3) {
   prize++;
}
```


Today's Goal

1. Be able to use For / While / If in Java