Nested Loops
Chris Piech
CS106A, Stanford University
// Store expressions that evaluate to true/false

boolean x = 1 < 2;    // true
boolean y = 5.0 == 4.0; // false
// 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;
// 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) {
    ...

Please... NO FOOD OR DRINKS

*know your logical precedence
Welcome to the CS106A game show!
Choose a door and win a prize
Door: 2
You chose door 2
You win $
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

```java
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++;
}
```
```java
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

```java
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

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

```java
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++;  
}
```
How would you println “Stanford rocks socks” 100 times
public void run() {
    for(int i = 0; i < 100; i++) {
        println("Stanford rocks socks!");
    }
}
for (int i = 0; i < 100; i++) {
    println("Stanford rocks socks! ");
}
for (int i = 0; i < 3; i++) {
    println("Stanford rocks socks!");
}
For Loop Redux

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

\[
\begin{array}{c|c}
\text{i} & 0 \\
\end{array}
\]

\begin{verbatim}
for(int i = 0; i < 3; i++) {
    println("Stanford rocks socks!");
}
\end{verbatim}

Piech, CS106A, Stanford University
for(int i = 0; i < 3; i++) {
    println("Stanford rocks socks!");
}
For Loop Redux

\[
\begin{array}{c|c}
& i \\
\hline
& 0 \\
\end{array}
\]

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

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

<table>
<thead>
<tr>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

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

![Output of the loop](image)
for (int i = 0; i < 3; i++) {
    println(“Stanford rocks socks!”);
}
For Loop Redux

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

```
Stanford rocks socks
Stanford rocks socks
```
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!”);
}

Stanford rocks socks
Stanford rocks socks
Stanford rocks socks
For Loop Redux

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

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

For Loop Redux

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

For Loop Redux

![Output of the loop](image)
You can use the for loop variable
How would you println the first 100 even numbers?
## Printing Even Numbers

<table>
<thead>
<tr>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>36</td>
</tr>
<tr>
<td>38</td>
</tr>
</tbody>
</table>
Printing Even Numbers

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

\[
\text{for(int } i = 0; i < 3; i++) \{ \\
\quad \text{println}(i \times 2); \\
\}
\]
Printing Even Numbers

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

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

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

```
  i  0
```

Piech, CS106A, Stanford University
Printing Even Numbers

\[
\begin{array}{c|c}
\text{i} & 0 \\
\end{array}
\]

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

Piech, CS106A, Stanford University
Printing Even Numbers

\[
\begin{array}{c|c}
\text{i} & 1 \\
\end{array}
\]

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

Piech, CS106A, Stanford University
Printing Even Numbers

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

For Loop Redux

Piech, CS106A, Stanford University
Printing Even Numbers

\[
\begin{array}{c|c}
\text{i} & 1 \\
\end{array}
\]

\begin{verbatim}
for(int i = 0; i < 3; i++) {
    println(i * 2);
}
\end{verbatim}

![Output of the code]

Piech, CS106A, Stanford University
Printing Even Numbers

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

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

```
0
2
```
Printing Even Numbers

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

0
2
4
Printing Even Numbers

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

\[
\begin{array}{|c|}
\hline
i & 3 \\
\hline
\end{array}
\]

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

Output:

0
2
4
Printing Even Numbers

\[
\text{i} \quad 3
\]

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

For Loop Redux

0
2
4
for(int i = 0; i < 3; i++) {
    println(i * 2);
}
A Variable love story

By Chris
Once upon a time...
X was looking for love!

```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

X was looking for love!
X was looking for love!

```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

5

X was looking for love!
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

x was definitely looking for love

X was looking for love!
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

And met y

Hi, I'm y

Piech, CS106A, Stanford University
“Wow!”
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);

Wow
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);

We have so much in common
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);

We both have value 5!
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

Maybe one day we can...
```java
int x = 5;
if (lookingForLove()) {
    int y = 5;
}
println(x + y);
```

And met y together?
int \ x = 5;  
if(lookingForLove()) {
    int \ y = 5;
}
println(x + y);

They got along
It was a beautiful match...
But then tragedy struck.
Tragedy Struck

```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

```
\( x \) = 5
\( y \) = 5
```

Piech, CS106A, Stanford University
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);

Tragedy Struck
Nooooooooooooooooooooo!
You see...
When a program exits a code block...

```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

5

When a program exits a code block...
All variables declared inside that block..

```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

Piech, CS106A, Stanford University
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

Get deleted from memory!
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
It gets deleted from memory here

```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```

![Diagram showing the deletion of y from memory]
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
}
println(x + y);
```
Error. Undefined variable y.

println(x + y);
The End
Sad times 😞
public void run(){
    double v = 8;
    if(condition) {
        v = 4;
        ... some code
    }
    ... some other code
}
public void run(){
    double v = 8;
    if(condition) {
        v = 4;
        ... some code
    }
    ... some other code
}
public void run(){
    double v = 8;
    if(condition) {
        v = 4;
        ... some code
    }
    ... some other code
}
public void run(){
    double v = 8;
    if(condition) {
        v = 4;
        ... some code
    }
    ... some other code
}

This is the *inner most* code block in which it was declared....
public void run() {
    double v = 8;
    if (condition) {
        v = 4;
        ... some code
    }
    ... some other code
}
public void run(){
    double v = 8;
    if(condition) {
        v = 4;
        ... some code
    }
    ... some other code
}

It dies here (at the end of its code block)
public void run() {
    double v = 8;
    if (condition) {
        v = 4;
        ... some code
    }
    ... some other code
}

It dies here (at the end of its code block)
Example 2

```java
public void run(){
    ... some code
    if(condition) {
        int w = 4;
        ... some code
    }
    ... some other code
}
```

This is the scope of w
public void run() {
    ...
    if (condition) {
        int w = 4;
        ...
    }
    ...
    ...
}
A Variable love story

Chapter 2
The programmer fixed her bug
int x = 5;
if(lookingForLove()) {
    int y = 5;
    println(x + y);
}

x was looking for love!
int x = 5;
if(lookingForLove()) {
    int y = 5;
    println(x + y);
}

x was definitely looking for love...
```java
int x = 5;
if(lookingForLove()) {
    int y = 5;
    println(x + y);
}
```

```
x
```

```
y
```

x met y

Piech, CS106A, Stanford University
Since they were both “in scope”

```
int x = 5;
if(lookingForLove()) {
    int y = 5;
    println(x + y);
}
```

5
---
5

x
---
5

y
The story had a happy ending!
The **scope** of a variable refers to the section of code where a variable can be accessed.

- **Scope starts** where the variable is declared.
- **Scope ends** at the termination of the inner-most code block in which the variable was defined.

- **A code block** is a chunk of code between `{ }` brackets.
Back to our regularly scheduled program...
Types of Programs

- Program
  - Karel Program
  - Console Program
  - Graphics Program
Graphics Programs
**GRect**

GRect is a variable type that stores a rectangle.

As an example, the following `run` method displays a blue square.

```java
public void run() {
   GRect rect = new GRect(200, 200);
   rect.setFilled(true);
   rect.setColor(Color.BLUE);
   add(rect, 50, 50);
}
```
GRect is a variable type that stores a rectangle.

As an example, the following `run` method displays a blue square

```java
public void run() {
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
GRect is a variable type that stores a rectangle.

As an example, the following `run` method displays a blue square.

```java
public void run() {
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
GRect is a variable type that stores a rectangle.

As an example, the following **run** method displays a blue square

```java
public void run() {
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
**GRect**

GRect is a variable type that stores a rectangle.

As an example, the following `run` method displays a blue square

```java
public void run() {
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
Graphics Coordinates

0,0

40,20
120,40
40,120

getWidth();

getHeight();
Goal
Milestone 1
Milestone 2