

Scope and Nesting

Neither Mouthwash, nor submarines, nor bird houses
will enter this discussion

Before we begin...

- You can have (though we haven't so far) variables that hold true or false values called **booleans**.
- A boolean variable can either be assigned the value **true** or **false** directly, or they can be assigned the value of an expression.

```
int myAge = ...;  
boolean iThinkThereforeIAm = true;  
boolean iAmOld = (myAge > 30);
```

Using Boolean Variables

- Boolean variables can make reading conditionals easier!

```
int myAge = ...;  
int height = ...;  
boolean iAmOld = (myAge > 30);  
boolean iAmTall = (height >= 72);
```

```
if(iAmOld && iAmTall)  
{  
}
```

Using Boolean Variables

- There are different correct ways to use boolean variables

```
boolean youAreTired = ...;  
boolean youAreHungry = ..;
```

```
if(youAreTired == true || !youAreHungry)  
{  
    // you should probably take a nap!  
}
```

Scope

- The Major Rule: A variable declared within a set of braces only exists within those braces
- Corollary: Just because a variable is declared at line 20 does not mean that this variable can be used at all lines greater than 20!

```
if(x >= 50)
```

```
{  
    int y = 40;  
    System.out.println(y);  
}
```

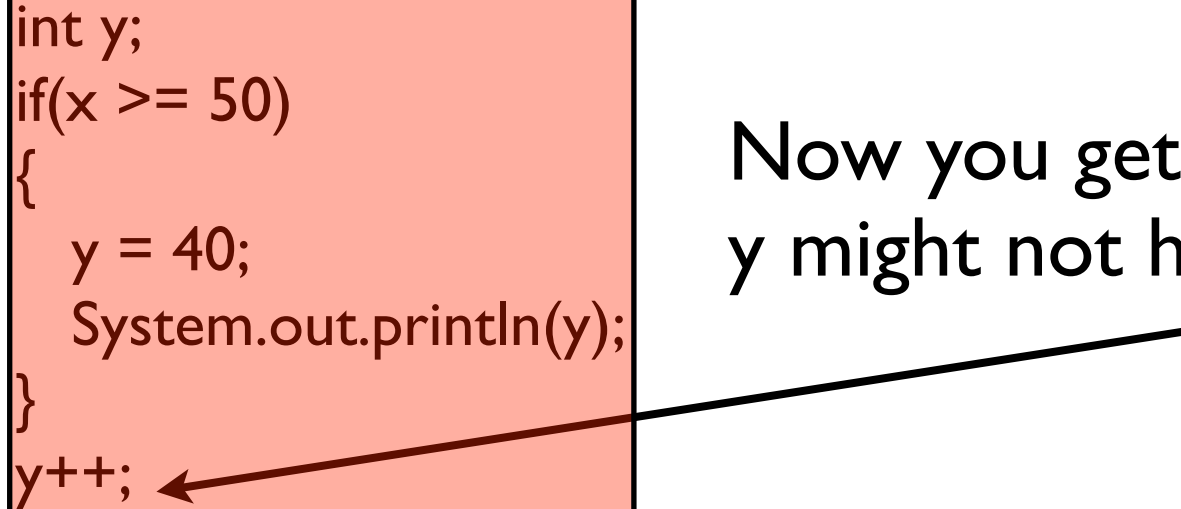
```
y++;
```



y doesn't exist here!

Scope Continued

```
int y;  
if(x >= 50)  
{  
    y = 40;  
    System.out.println(y);  
}  
y++;
```



Now you get an annoying error...
y might not have been initialized!

Imagine what would happen if x was less than 50...

Scope Continued

```
int y = 0;  
if(x >= 50)  
{  
    y = 40;  
    System.out.println(y);  
}  
y++;
```



No more error!

Unassigned Variables

```
int y;  
if(x >= 50)  
{  
    y = 40;  
    System.out.println(y);  
}  
if(x < 50)  
{  
    y = 0;  
}  
y++;
```

Still an error :(

- 1) Java will not look at the code in your parenthesis to see if your logic makes sense
- 2) Java does not know structurally that exactly one of those cases has to occur
- 3) Java does not know that y must be assigned

Unassigned Variables

```
int y;  
if(x >= 50)  
{  
    y = 40;  
    System.out.println(y);  
}  
else if(x < 50)  
{  
    y = 0;  
}  
y++;
```

We structurally are closer, but in Java's mind, it is still possible to miss both of those cases... how to fix?

Still an error :(

Unassigned Variables

```
int y;  
if(x >= 50)  
{  
    y = 40;  
    System.out.println(y);  
}  
else  
{  
    y = 0;  
}  
y++;
```

We have now structurally determined that `y` has to be assigned before it is incremented.

No more error!

Nesting Conditionals

- Conditional Statements can be **nested** inside of other conditional statements

```
int die1 = ...;  
int die2 = ...;
```

```
if(die1 % 2 == 0 && die2 == 1)  
{  
}  
else if(die1 % 2 == 0 && die2 == 2)  
{  
}
```

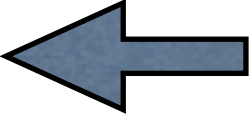
```
int die1 = ...;  
int die2 = ...;
```

```
if(die1 % 2 == 0)  
{  
    if(die2 == 1)  
    {  
    }  
    else if(die2 == 2)  
    {  
    }  
}
```

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



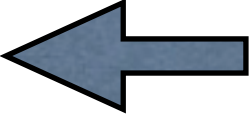
x	y
0	

Console:

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



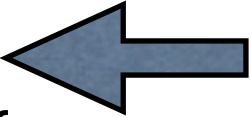
x	y
0	

Console:

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
  int y = 0;
  while(y < 2) {
    y++;
    System.out.print(y + " ");
  }
  System.out.println();
  x++;
}
```



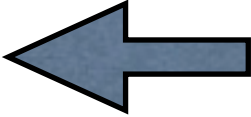
x	y
0	0

Console:

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



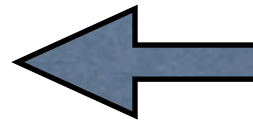
x	y
0	0

Console:

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



x	y
0	1

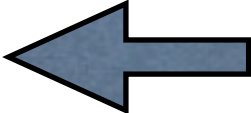
Console:

1

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



x	y
0	1


Console:

1

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



x	y
0	2

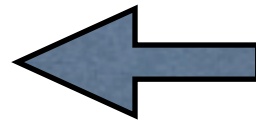
Console:

|

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



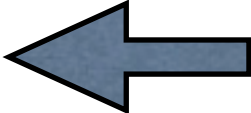
x	y
0	2

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



x	y
0	2

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println(); ←
    x++;
}
```

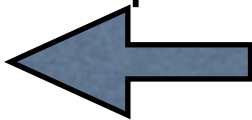
x	y
0	

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



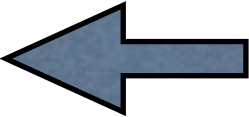
x	y
1	

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



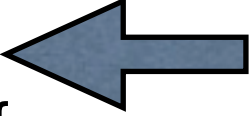
x	y
1	

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
  int y = 0;
  while(y < 2) {
    y++;
    System.out.print(y + " ");
  }
  System.out.println();
  x++;
}
```



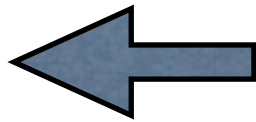
x	y
1	0

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



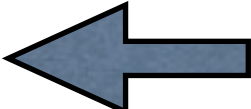
x	y
1	0

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```

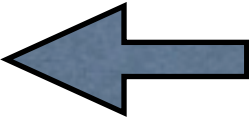


x	y
1	1

Console:
1 2

Nesting Loops!

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " "); 
    }
    System.out.println();
    x++;
}
```

x	y
1	1

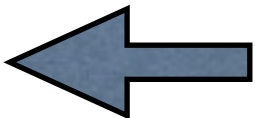
Console:

```
1 2
1
```

Fast Forwarding...

- Loops can be nested as well

```
int x = 0;
while(x < 3) {
    int y = 0;
    while(y < 2) {
        y++;
        System.out.print(y + " ");
    }
    System.out.println();
    x++;
}
```



x	y
3	

Console:

```
| 2
| 2
| 2
```

Scope Example

- See Example Java File on [Website](#)