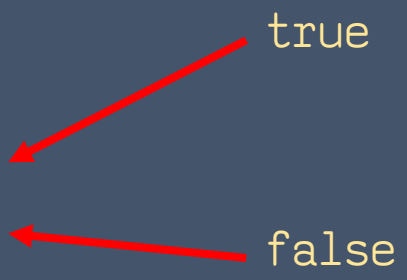
CMPSC 100

Computational Expression

Boolean expressions



- Evaluate “truthiness”

```
boolean isItBreakYet = false;
// So keep working.
```

Boolean expressions

conditional statement

```
boolean isItLunch = hour > 12;
```

```
> false
```

“Relational” operators

Operator	Meaning
==	equal to
!=	not equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

```
int apples = 5;
int oranges = 6;

apples == oranges

> false

apples != oranges

> true

apples <= oranges

> true

apples >= oranges

> false
```

“Relational” operators and `String`

```
String name = “G. Wiz”;
String fullName = “Gator Wizard”;

boolean sameName = name == fullName;
```

“Relational” operators and `String`

```
String name = “G. Wiz”;
String fullName = “Gator Wizard”;

boolean sameName = name.equals(fullName);
boolean sameLen = name.length() == fullName.length();
```

> Both are `false`

“Logical” operators

“inclusive” and

Operator	Description	Example	Result
!	logical NOT	! a	true if a is false and false if a is true
&&	logical AND	a && b	true if a and b are both true and false otherwise
||	logical OR	a || b	true if a or b or both are true and false otherwise

“exclusive” or

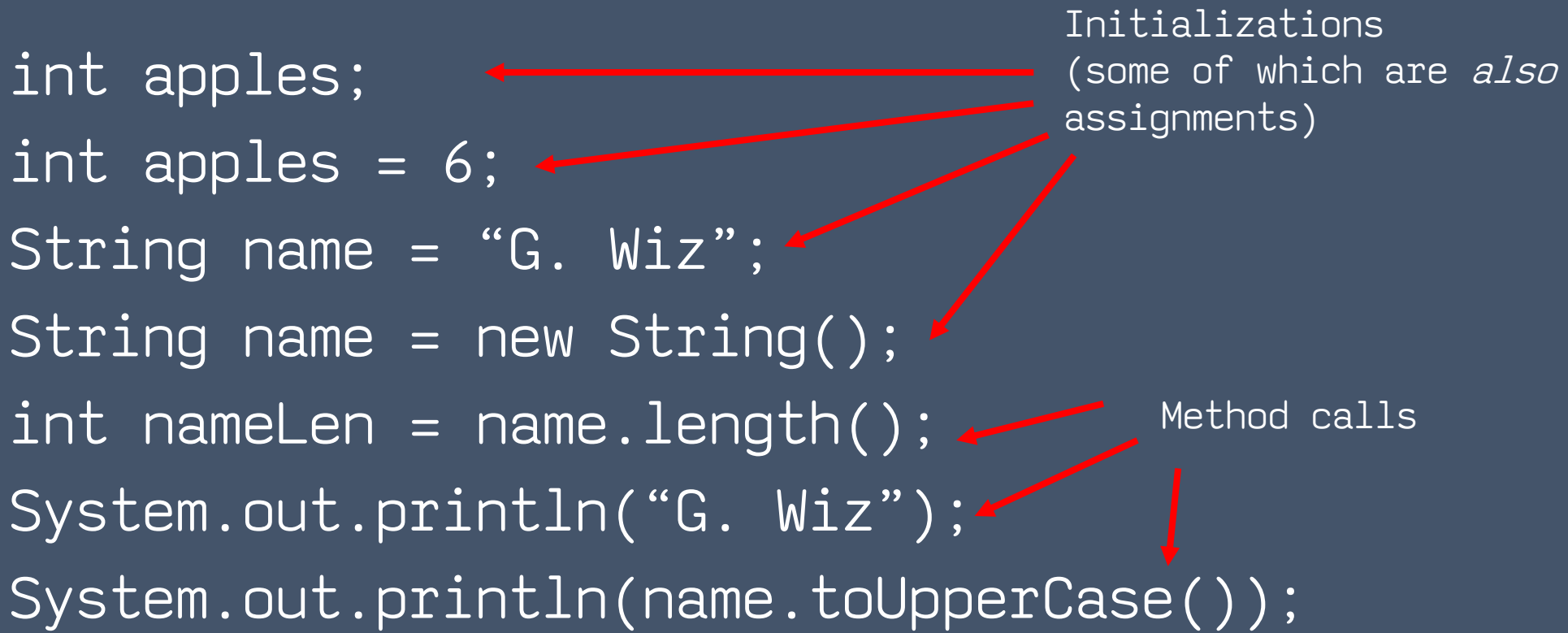
if statements

```
if ({BOOLEAN EXPRESSION}) {
  // Something to do
}
```

Example:

```
double temperature = 100.0;
if (temperature >= 100) {
  System.out.println(“It’s hot!”);
}
```

Detour: statements

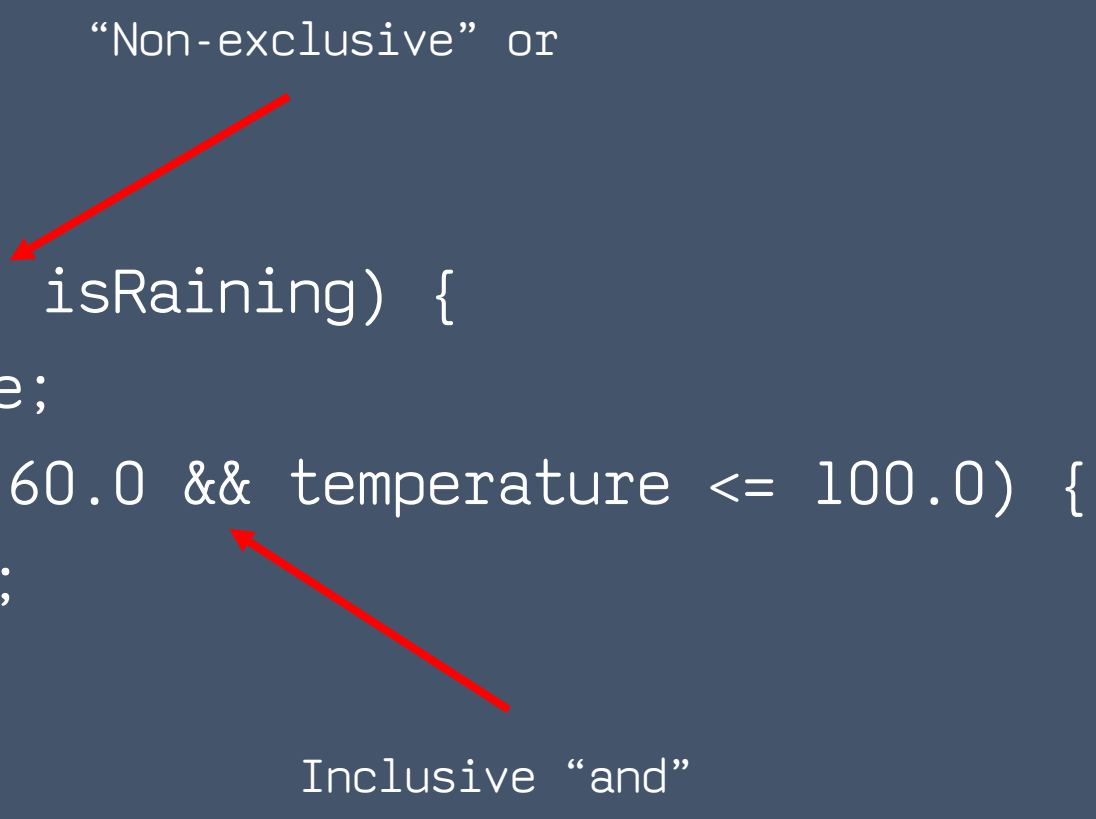



```
int apples;
int apples = 6;
String name = “G. Wiz”;
String name = new String();
int nameLen = name.length();
System.out.println(“G. Wiz”);
System.out.println(name.toUpperCase());
```

Detour: types of statements

```
if (temperature >= 100) {
  System.out.println(“It’s hot!”);
}
```

if statements




```
if (temperature > 100.0 || isRaining) {
      me.setComfort = false;
} else if (temperature >= 60.0 && temperature <= 100.0) {
      me.setComfort = true;
}
```

if statements

```
if (temperature > 100.0 && isRaining) {
      isHot = true;
      isHumid = true;
} else if(temperature > 100.0) {
      isHot = true;
} else if (temperature < 60.0 && isRaining) {
      isDreary = true;
      professor.setPreference(isDreary);
}
```

if statements

```
if (temperature >= 80 && temperature <= 110.0) {
        isWarm = true;
        professor.setPreference(isWarm);
} else if (temperature >= 45 && temperature <= 80) {
        System.out.println("Everyone else probably likes this weather.");
        // It's really just OK
} else {
        isCold = true;
}
```

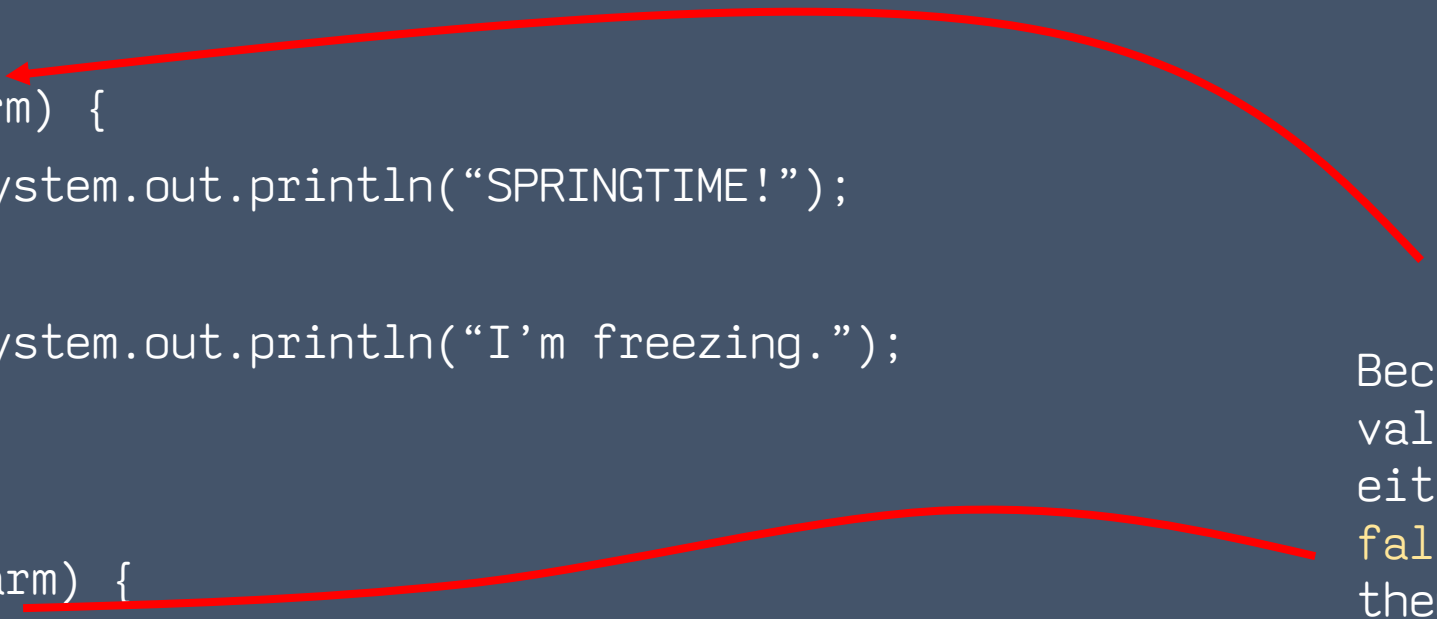
This represents a kind of "catch-all" condition. If no criteria are met above, this "trap" will trigger.

if statements

```
boolean isWarm = temperature > 80;

if (isWarm) {
        System.out.println(“SPRINGTIME!”);
} else {
        System.out.println(“I’m freezing.”);
}

if (!isWarm) {
        System.out.println(“I’m freezing.”);
} else {
        System.out.println(“SPRINGTIME!”);
}
```



Because a boolean value evaluates to either true or false, we can test them directly.

As usual, (activity) story time.

The police have captured two legendary catnappers and are interrogating them in separate rooms, so they can't communicate with each other. Authorities accuse them of trying to catnap Ulysses! The police offer each the following deal:

(Don't worry, they were caught before serious hard could come to him)

- If Alice snitches on Bob, Alice goes free and Bob spends 3 years in jail.
- Similarly, if Bob snitches on Alice, Bob goes free and Alice spends 3 years in jail.
- If neither of them snitch on each other, then they both spend 1 year in jail.
- If they both snitch on each other, then they both spend 2 years in jail.

- The question presumes that freedom is preferable, though that choice may not always be the best one.
- To gain freedom, someone has to defect.
  - The question, then, becomes any one of four:
    - Does Alice defect?
    - Does Bob defect?
    - Do neither of them defect?
    - Do both defect?
- We can form a kind of truth table from this data.

What logical operator best describes this relationship?

&& (“and”!)

Alice defects	Bob defects	Sentences
T	F	Alice: 0, Bob: 3
F	T	Alice: 3, Bob: 0
T	T	Alice: 2, Bob: 2
F	F	Alice 1: Bob: 1 (and they catnap again!)

Lab grading criteria

Several grading criteria use things called “regular expressions” to account for all various outcomes. In this lab, these are:

`!=|![a-z]`

`!=` relational operator or `!` logical operator

`Alice defects!|Alice stays quiet.; Bob defects!|Bob stays quiet.`

Either of these phrases

`Alice\sreceives\s[0-3]\syear\(s\)\.`

The phrase “Alice receives 0-3 year(s).”

`Bob\sreceives\s[0-3]\syear\(s\)\.`

The phrase “Bob receives 0-3 year(s).”