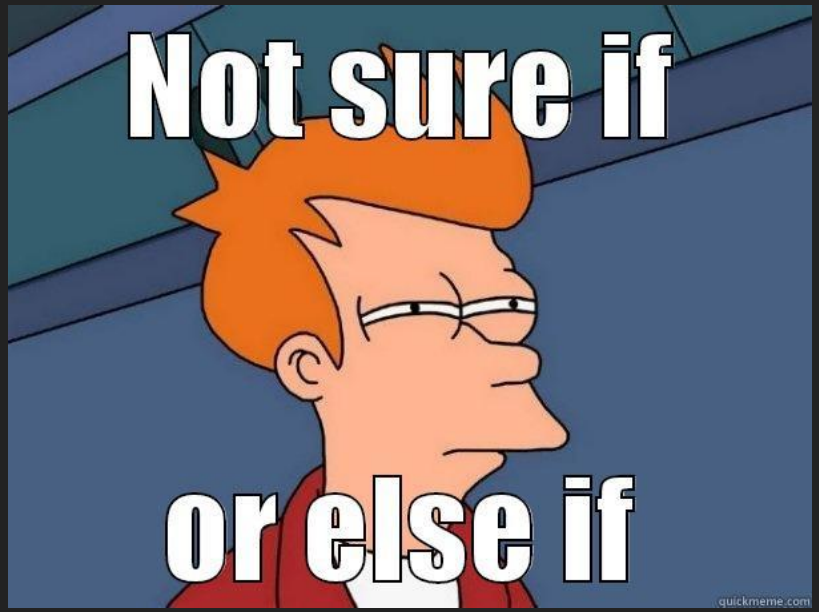
if - else



if

The if statement is used for control flow. It lets you determine when code is executed (or skipped).

```
if (what's in here is true) {
    // Then this happens
}
```

```
if (room == 200) {
    System.out.println("You are in the correct room!");
}
```

if

Notice == is used for equals and != is used for not equals.

```
if (room == 200) {
    System.out.println("You are in the correct room!");
}

if (room != 200) {
    System.out.println("You are in the wrong room!");
}
```

if

Some of the comparison operators you might use:

```
==          Equals

!=          Does not equal

>           Greater than

<           Less than

>=          Greater than or equal to

<=          Less than or equal to
```

if String.equals

Be careful! When comparing Strings, use .equals. Do not use ==

```
String name = "CS";

if (name.equals("CS")) {
    System.out.println("Computer Science");
}


String name1 = "Professor Carmine";
String name2 = "Professor X";

if (name1.equals(name2)) {
    System.out.println("This would explain so much!");
}
```

if else

Given a condition, you can execute some code (or execute something else).

```
if (room == 200) {
    System.out.println("You are in the correct room!");
}

else {
    System.out.println("You are in the wrong room!");
}
```

if else if

You can combine multiple statements using else if.

```
if (department.equals("CS")) {
    System.out.println("Computer Science");
}

else if (department.equals("CIT")) {
    System.out.println("Computer Information Technology");
}

else {
    System.out.println("Unknown department.");
}
```

Nested if

You can put if statements inside of other if statements.

```
if (department.equals("CS")) {

    if (courseNumber == 121) {
        System.out.println("Computer Programming I");
    }

    else if (courseNumber == 122) {
        System.out.println("Computer Programming II");
    }
}

else if (department.equals("CIT")) {
    ...
}
```

Working with Ranges of Numbers

Ranges of Numbers

You can work with a range of numbers by combining if, else if, and else.

```
if (number > 0) {
    // Do this
}

else if (number < 0) {
    // Do this
}

else {
    // Do this
}
```

Ranges of Numbers

indexOf returned the index of a string in a string or -1 if not found.

```
if (email.indexOf("@") >= 0) {
    System.out.println("Looks Good!");
}

else {
    System.out.println("You are missing the @ in your email.");
}
```

Ranges of Numbers

Here is a more complex example of working with a range.

```
if (grade >= 94) {
    letterGrade = "A";
}
else if (grade >= 90) {
    letterGrade = "A-";
}
else if (grade >= 87) {
    letterGrade = "B+";
}
// And so on...
else {
    letterGrade = "F";
}
```

Grade Ranges

100–94%	A	79–77%	C+
93–90%	A-	76–73%	C
89–87%	B+	72–70%	C-
86–83%	B	69–60%	D
82–80%	B-	59–0%	F

Let’s Code

Don’t Forget!

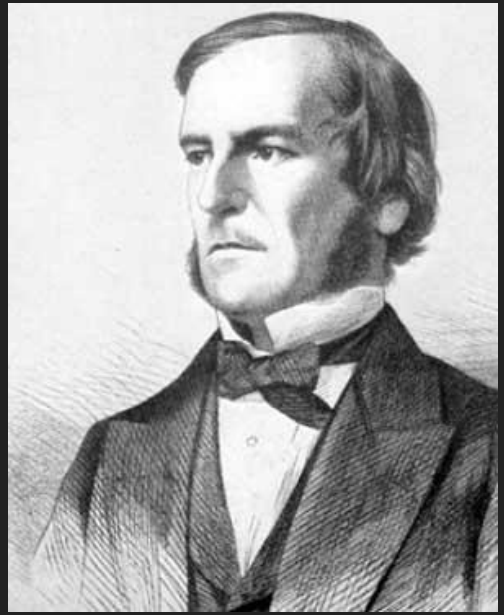
Check the syllabus / schedule for reading assignments and due dates!

Logic

boolean

A boolean value can be either true or false. We saw a hint of this when writing if statements.

You will notice a relation to when we discussed bits which were either a value of 1 or 0.



George Boole, 1860

boolean

There is a data type called boolean which is used for storing values of true or false. Notice that true and false are reserved words in Java.

```
boolean isProfessor = true;

boolean drinksCoffee = false;

boolean wasLeftShark = false;
```

Logical Operators

There are 3 logical operators we will be working with mostly:
and, or, not

You may have encountered these in the real world such as:

Prerequisites:
CS101 or CS102
CS101 and CS102

If it's Tuesday or Thursday and close to Noon. Go to CS Class.

If I am not late, then walk, otherwise take the subway.

Truth Table (and)

Truth tables let you know the results of logical operators.

Truth table for "and"

p	q	p AND q
T	T	**T**
T	F	**F**
F	T	**F**
F	F	**F**

If I have class and I am late then take an Uber.

Truth Table (or)

Truth tables let you know the results of logical operators.

Truth table for "or"

p	q	p OR q
T	T	**T**
T	F	**T**
F	T	**T**
F	F	**F**

If it is raining or snowing then wear boots.

Truth Table (not)

Truth tables let you know the results of logical operators.

Truth table for "not"

p	NOT p
T	**F**
F	**T**

If you are not ready then study!

Java Logical Operators

Java uses the following symbols for logical operators.

`&&`	`Logical and`	`if (isSunny && areHungry)`
`||`	`Logical or`	`if (x == 5 || x == 10)`
`!`	`Logical not`	`if (!isReady)`

Be careful!

For logic (notice and or are doubled): && ||

Using just one such as & or | is "mathematical and"

Logical Operators

Logical operators are great to use when working with ranges of numbers.

```
if (value >= 1 && value <= 40)
{
    System.out.println("Gryffindor");
}

else if (value >= 41 && value <= 70)
{
    System.out.println("Slytherin");
}
```

Logical Operators

Don’t forget we use equals with Strings.

```
if (name.equals("Iron-Man") || name.equals("The Hulk"))
{
    System.out.println("The Avengers");
}

else if (name.equals("Wonder Woman") || name.equals("Batman"))
{
    System.out.println("The Justice League");
}
```

Order of Logical Operators

Similar to the PEMDAS rule, logical operators have a precedence rule.

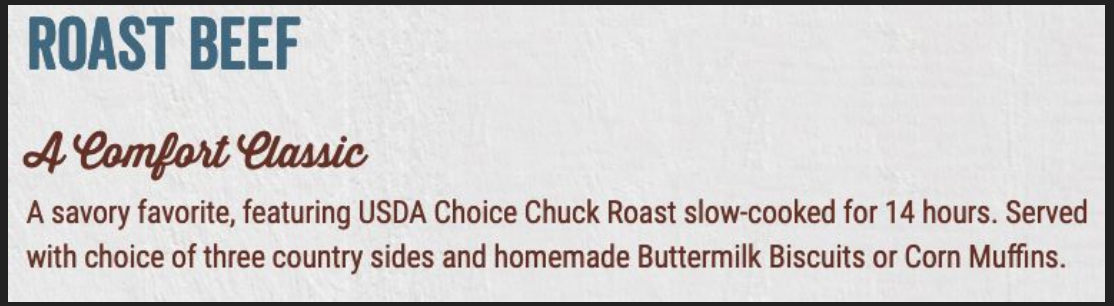
```
()          Parenthesis

!           Logical Not

&&          Logical And

||          Logical Or
```

Knowing this order seems like a good thing to know.

Logical Operators

After this lesson you will never look at a menu the same again.

Your choice of fries and salad or mashed potatoes and green beans.
Your choice of (fries and salad) or (mashed potatoes and green beans).



It says: (three sides and Buttermilk biscuits) or (Corn Muffins)
They mean: (three sides) and (homemade Buttermilk biscuits or Corn Muffins)

To be or not to be

To be or not to be

Being Computer Scientists, we can solve this with boolean logic!

```
to_be || !to_be =    ??
```

To be or not to be

It is always true!

```
to_be || !to_be =    ??


if to_be is true:
    true || !true
    true || false
        true

if to_be is false:
    false || !false
    false || true
        true
```

Let’s Code

Don’t Forget!

Check the syllabus / schedule for reading assignments and due dates!