We're going to talk about propositions and logical operations in this little clip, and let's begin then with a discussion of propositions. So to a mathematician and, in particular in this class, we're going to use the word proposition to refer to something that is either true or false. An example would be there are five regular solids. This happens to be true. In some terms we even prove it. It implies, for example, that if you wanted to place, let's say, 100 fixed-position satellites around the earth in a uniform way, you can't do it because there isn't any 100 vertex regular solid. The biggest one is 20 vertices.

OK. Well, if I change it to six, guess what? The assertion there are six regular solids is false. So that's a simple-minded example of two well-defined propositions, one of which is true, one of which is false. Propositions don't have to be true always. What are some non-examples? Well, wake up is not a proposition, because it's an imperative. It's not true or false, and where am I is a question. It's not true or false. And it's 3:00 PM is not a proposition, because it's true or false at any given moment, but whether or not it's true or false depends on the time, and that's a complication we don't want to get into. The idea is, a proposition is some fixed assertion that's either true forever or not true forever.

Now, one of the reasons why mathematicians bring up this abstraction of propositions and the operations on them that we're about to see is that ordinary language tends to be ambiguous and that, of course, will cause problems in mathematical reasoning just as it would in a program.

One of the most ambiguous of the phrases in English that connects propositions is Or. So let's look at this example. Greeks carry swords or javelins, and if I was transcribing this into precise math notation, I could say G for Greeks implies S for swords or J for javelin, so this is an assertion that if you're Greek, then you carry a sword or a javelin. Greeks implies sword or javelin. Really, I should say Greek soldiers, but let that be implicit.

That's how we're going to translate this sentence into just using these operators to paraphrase what's going on. The problem is, what does Or mean? And it turns out that for javelins and swords it's true even if a Greek carries both a sword and a javelin. This is an inclusive Or. A Greek soldier would carry both a sword and a javelin because, in fact, a javelin is a good long-distance weapon, and a sword is good for defending yourself close in, and you certainly want to have both, especially after you've thrown your javelin, and you don't have anything left but the sword.

Now, there's many standard notations for these logical connectives that build up larger propositions out of component propositions, so one of the things is this V symbol, or disjunction symbol, is used by logicians often instead of Or, and this arrow means implies or sometimes a double bar arrow also means implies, but we're not going to get into that. I'm not going ask you to memorize these symbols. We'll just stick to the words which don't take up that much more room.

Let's look at another example. Greeks carry bronze or copper swords. Syntactically this has the same structure as the previous phrase, but we're going to translate it differently, and the reason is that we mean here that a Greek soldier is not going to carry both a bronze and copper sword. Why is that? Well, bronze swords are just way better than copper swords. They'll slice right through copper. They're much harder, and it's not worth the weight to carry this inferior copper sword when you have a much better one.

So this time we mean the Greeks carry exactly one of a bronze or a copper sword. You'd carry a copper sword if you didn't have access to a bronze one. And so now we translate that into Greek implies B for bronze or C for copper, but this time we use the X Or. X Or means that one of them is true exactly but not both and not neither. Again, there's this plus sign notation for X Or, because as we'll see it acts a little bit like adding numbers by 2 where 1 plus 1 is 0.

So let's be more precise about the two definitions of Or and X Or and how they work, and the assertion is that if I think of P and Q as placeholders for propositions that are either true or false, then the composite proposition P or Q is true if, and only if, P is true or Q is true or both are true, and I could express this assertion in English this, if and only if, by giving you a so-called truth table, where in these two columns or all these rows I've enumerated all the possible pairs of values of P and Q.

So P and Q might both be true, that P might be true and Q false, P false, Q true, and both of them are false. And for each of those possible combinations of the truth values of Q and P, I tell you the truth value of P or Q, and the notable thing is this last row where I'm telling you that the only way that P or Q is false is if both P and Q are false.

For X Or, as we said, the P or X Or Q is true if, and only if, exactly one of P and Q is true, so if I was expressing that as a truth table, we'd see that where there's TT is false, and where there's FF it's false, because it's not the case in either of those two rows that exactly one is true, but the middle row is exactly one is true. And so P X or Q is true. So this truth table is just a precise way of defining how X Or acts on truth values.

There's another connective, And, which works even more straightforwardly. The value of P and Q is true if, and only if, both P and Q are true, and there's this truth table. Again, the salient row is that it's true only if, and only if, both P and Q are true. Another crucial logical operation is the negation operation, or Not. So Not P just flips the truth value of P. If P is true, then not P is false. If not P is true, then P is false, and there it's very trivial truth table, trivial because there's only two values to be concerned about. When P is true, not P is false. When P is false, not P is true.

One of the places that this notion of combining basic propositions to using logical operations to build up more complicated composite proposition is in programming. Here's a typical kind of phrase that comes from Java. Java uses this double vertical bar to mean Or, inclusive by the way, and double ampersand to mean And, so in Java this is a piece of legitimate Java code that's doing a test to evaluate this expression, if X is greater than 0 or X is less than or equal to 0 and Y is greater than 100, then if that test comes out to be true, then you do a bunch of code that follows the test up to some delimiter that tells you where to stop. And if it's false, you just skip all that stuff and go on.

So we use sort of Boolean expressions or logical expressions like this in a very standard way in most programming languages. The other place where these operations come up is in digital logic, and the digital circuit designers have their own notation, which I'll just mention, but we're not, again, impose on you but you should be aware. One notation they use that we'll use as well because it's so economical is that not X can be abbreviated by writing a bar over the X.

More generally, not a complicated expression can be abbreviated by writing a bar over the complicated expression. I just saved some space, and so we'll use it, but not the following. In digital logic, the idea is that you're talking about circuits where the only distinction of the signal that's coming in electrically is whether it's a high voltage or a low voltage, with high voltage is denoted by 1, and low voltage denoted by 0, and the way that digital logic behaves is that the 1 corresponds to true at the 0 corresponds to false. Then the And operation is simply multiplication, because the 1 or 0 times 1 or 0 is 1 only when both of them are 1 for ordinary multiplication, which is exactly the way And behaves when 1 means true and 0 means false.

Unfortunately, the digital designers use Plus when they mean Or. They do not mean that 1 plus 1 is 2. They mean that 1 plus 1 is 1, and just a thing to watch out four. That's part of the reason we're not going to use this notation. Let's just stick to our ordinary word notation that we give on the right.