Mathematical Prerequisites

Although CS103 is primarily a math class, this course does not require any higher math as a pre-requisite. The most advanced level of mathematics you'll need for this course is high school algebra. Anything beyond that – trigonometry, complex numbers, calculus, limits, and (especially) graphing functions on a coordinate axis – isn't necessary for the course and won't come up at all over the course of the quarter.

In this handout, we've outlined the particular skills that we'll be assuming over the course of this class. As you'll probably find out pretty quickly in this class, the sort of math that we'll be doing looks very little like the math that you probably did in high school, so even if you are a bit shaky on these concepts, you should be fine in this class. As always, you are welcome – and in fact, encouraged – to come talk to us if you have any concerns about whether you're in the right place. If you would prefer to talk to me (Amy) about this directly, consider this an open invitation. ☺

Skill 1: Multiplying Polynomials

Early on this course (and, surprisingly, not much beyond then), we'll assume that you know how to multiply two polynomials together. For example, in one of our early proofs, we're going to arrive at the equation

$$n^2 = (2k)^2$$

and will simplify the right-hand side to get that

$$n^2 = 4k^2.$$

Similarly, on Friday, we'll end up with the equation

$$n^2 = (2k + 1)^2$$

and will simplify the right-hand side to conclude that

$$n^2 = 4k^2 + 4k + 1.$$

Oddly enough, we won't be factoring many – if any – polynomials over the quarter. It's probably good to be able to recognize that $n^2 + 2n + 1$ happens to be equal to $(n+1)^2$, but I doubt we'll do anything more advanced to that. Also, we certainly won't be dividing polynomials, so don't lose any sleep over that. (Confession time: I have never needed to divide polynomials outside of high school.)

Skill 2: Manipulating Equalities

This skill from high school algebra is the one we'll get the most mileage out of in this class. In case you're curious about what "manipulating equalities" means, we're talking about things like the following. Imagine that you know that

$$2x + 3 = 5$$

and you want to solve for x. To do so, you could subtract three from both sides to get

$$2x = 2$$

and then divide both sides by two to get that

$$x = 1.$$

Sometimes, in CS103, we'll write out a long chain of equalities to show that two numbers or quantities happen to be equal to one another. For example, we might show that $(2n+1)^2 - (2n)^2 = 4n + 1$ by writing out something like this:

$$(2n+1)^2 - (2n)^2 = 4n^2 + 4n + 1 - 4n^2$$
$$= (4n^2 - 4n^2) + 4n + 1$$
$$= 4n + 1.$$

Here, the top line performs one simplification, and each line after that is a simplification from the previous line. The final result is that the left-hand side of the first equality ends up being equal to the right-hand side of the last equality.

Skill 3: Adding, Subtracting, Multiplying, and Dividing Fractions

From time to time, we'll be assuming that you're comfortable manipulating fractions. For example, you might need to add together two fractions whose numerator and denominator are given symbolically, like this:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad}{bd} + \frac{bc}{bd} = \frac{ad + cb}{bd}.$$

Similarly, you may need to multiply two fractions together, like this:

$$\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$$

I've always thought it's unusual that multiplying fractions is cleaner than adding fractions. Interesting how that works out.

Skill 4: Working with Exponents and Roots

Exponents and their properties are one of those areas that often give students trouble in this course, primarily because we end up using them in ways that are a bit different from what you might be used to in high school. There are a few properties of exponents that we'll be using over the course of this class, most of which come up pretty early on (as early as Friday of the first week of class). The first is what happens if you multiply together two powers of some number x:

$$x^a \cdot x^b = x^{a+b}.$$

This property of exponents is fundamental – exponents are often used precisely because they turn multiplication into addition. In fact, some mathematicians *define* what it means to raise a number to a power in terms of the above equation!

One special case of this result that comes up a few times in CS103 is this one:

$$x^{n+1} = x \cdot x^n.$$

For example, $2^{n+1} = 2 \cdot 2^n = 2^n + 2^n$. We'll use this specific fact when we talk about induction around the halfway point of the quarter.

Another property is what happens if you raise one number to a power, then raise that to the power of another number. Specifically:

$$(x^a)^b \;=\; x^{ab} \;=\; (x^b)^a.$$

You can see where this rule comes from in the case where a and b are positive integers by using the earlier rule about what happens when you multiply together two powers. For example, $(x^a)^b$ means "x^a multiplied by itself b times," so we'd expect that

$$\begin{aligned}
(x^a)^b \;&=\; x^a \cdot x^a \cdot \ldots \cdot x^a &&(b \text{ times})\\
&=\; x^{a + a + \ldots + a} &&(b \text{ times})\\
&=\; x^{ab}.
\end{aligned}$$

That last step happens because if you add a to itself b times, you get ab, that is, a times b. Remember how earlier we said that you could derive things just using the fact that exponents turn multiplication into addition? This is one of the cases where we do that.

The last key property we'll need is the connection between roots and exponents. Let's think about what it would mean to raise something to a fractional power. For example, what is the number $x^{1/2}$? Well, using our previous rule, we know that

$$(x^{1/2})^2 \;=\; x^1 \;=\; x.$$

So, in other words, whatever number $x^{1/2}$ is, it has the property that, when you multiply it by itself, you get x. We have another name for that number – it's the square root of x! So this means that

$$x^{1/2} \;=\; \sqrt{x}$$

and, more generally, that

$$x^{\frac{1}{a}} \;=\; \sqrt[a]{x}.$$

Skill 5: Categorizing Types of Numbers

If you have experience programming in a typed language like C, C++, or Java, you're probably comfortable with the idea that when working with numbers, it's important to know what kind of number you're dealing with. You use `int`s to hold numbers that are integers (something you'd use for counting) and `double`s to store real numbers (something you'd use for measuring). The idea that there are different types of numbers that you'd use in different circumstances is fundamental in mathematics, and so throughout the quarter we'll be sure to be specific about what kinds of numbers we're using.

There are a zillion flavors of numbers in math, but we'll only need to use a few of them over the course of the quarter. The numbers we'll use the most extensively throughout the quarter – and which seem to get used more than any other type in all of computer science – are the ***natural numbers***. These are the numbers 0, 1, 2, 3, 4, 5, …, etc. They're the whole numbers (no fractions or decimals allowed) starting at 0 and continuing on toward infinity. However, we don't consider ∞ to be a natural number.

The next class of numbers we'll be using are the ***integers***. They're basically things you can store in an `int`: …, -3, -2, -1, 0, 1, 2, 3, …, etc., except that there's no limit to how large or small the numbers can be. Notice that every natural number is an integer, but the reverse isn't true: -1 is an integer, but it's not a natural number. As with the natural numbers, we don't consider ∞ or $-\infty$ to be integers.

The last "fundamental" class of numbers we'll deal with are the ***real numbers***. These are all the numbers that exist on the number line: 0, 1, ½, π, -3, $^{-137}/_{42}$, $-e^2$, etc. Every integer is a real number, but not the other way around. Continuing our trend from before, we don't consider ∞ or $-\infty$ real numbers.

You may have encountered complex numbers in your high school math classes, but we won't be talking about them this quarter. That doesn't mean that they're not useful in computer science, though. Surprisingly, they have some neat applications in computer graphics!

Skill 6: Using Closure Properties

There's a very good chance that you know what a "closure property" is even if you have never heard the name before. Consider this section a way of learning new fancy terms you can throw around at cocktail parties, as in, "I study closure properties of the natural numbers." By the time you're done with CS103, you'll have seen a bunch more of these in more advanced contexts, so you can sound even cooler: "I study closure properties of the regular languages, such as closure under Kleene star." Yeah. You'll actually know that that means. ☺

Let's imagine you add two natural numbers together, like 4 + 5, 9 + 13, or 0 + 137. Regardless of what two natural numbers you add together, you're guaranteed that you'll get back a natural number as a result. The mathematical term for this is to say that "the natural numbers are ***closed under addition***." In other words, if you start with two natural numbers and add them together, you're guaranteed to get back a natural number.

However, if you *subtract* two natural numbers, you are not guaranteed to get back a natural number. Sometimes you will (5 − 3 = 2, and 137 − 137 = 0), but sometimes you won't (3 − 5 = -2, and -2 is not a natural number because it's negative). Since the difference of two natural numbers is not guaranteed to also be a natural number, the natural numbers are *not* closed under subtraction.

What about multiplication and division? Well, if you multiply two natural numbers (6 × 5, 3 × 19, 137 × 1, 0 × 5, etc.), you're guaranteed to get back a natural number. That means that the natural numbers are ***closed under multiplication***. However, dividing two natural numbers doesn't guarantee that you get back a natural number. Sometimes you do ($^6/_3$ = 2), but sometimes you don't ($^{137}/_{42}$ isn't a natural number), so we say that the natural numbers are *not* closed under division.

To recap:

- The natural numbers are closed under addition and multiplication.

- The natural numbers are not closed under subtraction or division.

What about the integers? Like the natural numbers, they're closed under addition and multiplication: adding or multiplying two integers always gives you back an integer. Like the natural numbers, they're not closed under division ($^{137}/_{42}$ is not an integer). However, unlike the natural numbers, the integers are **_closed under subtraction_**: the difference of any two integers is an integer. (Fun fact: mathematicians often define the integers as "what you get when you start with the natural numbers and do the least awful thing to them that makes them closed under subtraction." I mean, it's more rigorous than that, but that's the basic idea.)

What about the real numbers? They're closed under all four of the basic arithmetic operations: the sum, product, quotient, and difference of any two real numbers is also a real number. (We're assuming that you aren't dividing by zero, which just isn't possible. Trust us. We tried it, and it didn't work.)

Here's a quick table recapping these facts:

	+	−	×	÷
Natural Numbers	Yes	_**No**_	Yes	_**No**_
Integers	Yes	Yes	Yes	_**No**_
Real Numbers	Yes	Yes	Yes	Yes

The closure properties of the natural numbers, integers, and real numbers are usually considered so fundamental that they're not really talked about in proofs even though they often play a major role in them. If you look closely at the proofs from Wednesday and Friday's lectures, you'll see that these closure properties are referenced implicitly.

Skill 7: Manipulating Inequalities

Mathematical inequalities come up all the time in computer science. We use them to precisely say things like "this algorithm will take at most n^2 time steps to complete" or "you'll need at least 2^n bits of memory to solve this problem exactly." If you continue onward in computer science, you'll get a lot of practice manipulating inequalities.

In CS103, we're not going to spend that much time working with inequalities, though there are a few places where they'll come up at the start of the quarter. There's one particular set of rules for inequalities that is sometimes covered in high-school algebra and sometimes skipped, and that's what happens when you work with inequalities involving integers. For example, consider the inequality

$$2 < x < 3.$$

This says that x is strictly greater than 2 and strictly less than 3. If x is allowed to be a real number, then there are lots of values of x that satisfy this inequality, such as 2.5, 2.1, or 2.71828. However, if x is an *integer* or a *natural number*, then there are *no* choices of x that make this true, since there aren't any integers between 2 and 3.

It's worth looking into this in a little bit more detail. Let's imagine that n is a natural number. Then the inequality

$$2 < n$$

is *exactly the same* as the inequality

$$3 \leq n.$$

Why is that? Well, since 2 is strictly less than n, then n can't be two or less, so it has to be three or more. More generally, let's imagine that m and n are integers or natural numbers. In that case, the inequality

$$m < n$$

is exactly the same as the inequality $m + 1 \leq n$.

This comes up every now and then in this course, so I figured I'd mention it here.

Some Concluding Remarks

I've taught CS103 to about 3,000 students and have never met anyone who was not appropriately mathematically prepared for the course. I've encountered many students who didn't feel *confident* with their understanding of mathematics or who were a bit *out of practice* with it, but I've never turned someone away from CS103 due to a lack of mathematical background. If anything, the students I've recommended not take CS103 usually have *too strong* of a math background!

If you're concerned about whether or not you're prepared for CS103, I strongly recommend checking out our CS103A companion course, which meets for two hours a week and goes over extra practice problems with the material. It's a great way to practice, to meet students who you might end up working with on the problem sets, and to ask questions in a low-stress environment.