How to Succeed in CS103

Thanks to the Fall 2014 CS103 students and staff for their input!

Learning mathematics is, in many ways, like learning how to program computers. You have to learn a lot of new syntax (for loops, while loops, if statements, etc.) along with exactly how the syntax translates into what the program does (things like what variable assignment actually means, or when the condition in a while loop gets checked). After a while, you realize that most of programming isn't about learning new tools as much as putting together the tools that you already have in new and clever ways. The more you program, the more experience you gain and the more mistakes you learn from. At times, you'll get totally stuck and need to ask for help, which really is a normal part of learning how to program. Over time, if you keep programming, you'll get better and better at writing, testing, and debugging your programs.

When you're learning proof-based mathematics for the first time, you'll have many of the same experiences that you'll have encountered when learning to program. You have to learn a lot of new syntax (proof structures, mathematical vocabulary, etc.) along with exactly how the syntax translates into what the proof means (things like what "assume for the sake of contradiction," "without loss of generality," etc. mean). After a while, you realize that most of mathematics isn't about learning new proof techniques as much as putting together the techniques that you already have in new and clever ways. The more practice you get with mathematics, the more experience you gain and the more mistakes you learn from. At times, you'll get totally stuck and need to ask for help, which really is a normal part of learning mathematics. Over time, if you keep doing math, you'll get better and better at solving problems and writing up proofs.

Math is a skill just like any other skill, and you will get a lot better at it if you keep practicing and you're strategic in your approach to learning.

At the end of last fall quarter's offering of CS103, I asked everyone in the class to offer one piece of advice to future CS103 students. I've compiled and organized their answers here in this handout. The quotes you're seeing here are from actual CS103 students. I hope that it gives you some input about how best to approach CS103 and how to have the best possible experience!

Start problem sets early

Start the problem sets early so you have time to think through your proofs.

Start all the PSets several days early! Create an independent understanding!

Start the homework early so that when you become frustrated with a problem, you can take a break for a day or two; the intuition might strike you when you least expect it.

DO NOT LEAVE YOUR PSETS UNTIL THE NIGHT BEFORE!!!

Definitely start the problem sets early. Once you're behind in this class, you're behind forever.

Start your psets early and not the night before

Start problem sets early! They actually all take a completely reasonable amount of time; most of the stress came from attempting to cram the entire problem set in the night before.

Don't rush through problems, they need to stew in your brain.

Spend time wrestling with problems; it gives you a sense of confidence.

This was the most common advice that former CS103 students had to offer. One of the biggest mistakes you can make in this course is putting everything off until the last minute. If you find out at the last minute that you don't actually understand some major concept, you might not be able to get any help. Some of the problems on the problem sets might require some time to work through, and if you start earlier you'll have more time to think over different approaches.

Stay on top of the material

Don't fall behind on material since there's a lot of it

All the material builds on itself, so don't get behind.

Make sure you keep up, and if anything is unclear, work on it right away.

Don't let the material get ahead of you.

The material in CS103 builds on itself. We *strongly* recommend taking some time after each lecture to review what was covered and to play around with the concepts. Sometimes you'll realize that there are certain concepts you're not particularly comfortable with or that just don't make sense. When that happens, don't worry! It's normal. You might just need to get more practice or to get help and advice from someone else.

Attend lectures in person and take notes

Go to lecture and take good notes.

Go to lecture and stay on top of the material

Pay attention in class and your life will be a lot easier!

Do not fall behind in lecture. Go to lectures.

Pay attention in lecture!

Make sure to be always up-to-date with lectures

Stay on top of lectures!

Make sure to stay on top of lectures--digging a lecture hole is stressful and hurts your learning.

Stay on top of lectures because they build on one another.

The lectures in CS103 are pretty packed with information. Even though we'll post the slides online, it's important that you keep up to speed with the lectures – there's a lot presented that isn't in the slides. Take notes on the lectures, even if you're watching them at 2x speed, to make sure you absorb what's going on. Also, we advise against doing a "lecture marathon" and trying to go through a week's worth of lectures in one sitting. Four hours of CS theory is a lot to absorb at once. ☺

Get help when you need it

Discuss with classmates, TAs, and Keith at every step of the process! Ask a lot of questions.

don't be afraid to ask for help

if you get stuck, ask for help

Go to office hours even after you have finished the PSET. It will help you check whether your intuition for concepts is correct and help you gain a deeper understanding of the material.

It is ***completely normal*** to get stuck and need help at some point in this class. When that happens, you have a lot of options available. If you're working in a group, you can ask your teammates for help or advice. If you haven't already done some of the practice problems, you can take a break and do some of those instead. You can also ask questions in office hours or on Piazza if you'd like the staff to weigh in.

Do all the problems on the problem sets, even if you're working in a team

> *Do ALL the work in the homeworks, then compare results with the rest of your group and fix problems -- splitting the problems into thirds and trusting your groupmates to solve them leaves you with a shaky understanding and worse grades.*
>
> *Having each member of the group work through each problem separately before discussing worked very well for my group.*
>
> *Do every problem in the Psets even if you are distributing the work among group members.*
>
> *Do every problem in the PSETs even if you're working in a group so that everyone is familiar with the way of thinking required for different problems.*
>
> *Be careful in working with groups, make sure you pull your weight and understand everything.*
>
> *Find a group, even if you all do your own psets, to talk about the problems with.*
>
> *Don't rely on others for psets. Struggling through them helps a lot with building an intuition.*
>
> *Do every single problem on the pset (don't just divide it up) and give yourself at least two days, because sometimes you really just need to sleep on a problem.*
>
> *Do homework's individually before meeting as a group*
>
> *Do homework by yourself before collaborating - it gives you better understanding*
>
> *Really spend time discussing the PSet and any topics in class with your team members. It's good for your soul.*

From experience, the students who did best in this course were the ones who knew how to work well in a team. We *strongly advise* against splitting up the problems in a problem set across different teammates – you'll end up getting less practice with the material and spending less time discussing important ideas with others.

There are many advantages to having everyone in the group solve every problem. If everyone does the problems individually, then everyone in the team can come together and talk about their solutions. You might find that your approach is not the same as your teammates', which could mean that you've found an alternate solution or that one of your solutions has a flaw in it. Taking the time to figure out which of these cases you're in will give extra practice reading proofs. Plus, having everyone write up their own proof gives everyone in the team a chance to read over everyone else's solutions, which will help you catch typos or logical errors. Oh, and speaking of proofreading your answers...

Proofread your problem sets before submitting

> *Re-read your problem sets before you submit them, you'll catch a lot of silly mistakes that way*
>
> *Make sure you read and re-read EVERY PROOF you write, even it you think it's perfect.*
>
> *Make sure you actually understand everything in detail. Also, pay very close attention to wording!*
>
> *Pay a lot of attention to the wording of the proofs. Copy terms until you feel comfortable. Don't be like me and go cowboy with your proof writing style!*
>
> *Make sure your partners check your answers*

In programming courses, you can test your software before you submit it. You usually have several levels of defense – the compiler will check that you don't have any syntax errors, and testing can reveal bugs that you might not have noticed. In math, there aren't (that many) automated tools to check your answers. Before you submit your answers, be sure to proofread them and make sure that they're clear, easy to read, correct, and make sense. If you're working in a team, this should be pretty easy: just hand your proofs to a teammate and ask them for their input. If your teammate has trouble understanding what you're saying, talk over your reasoning with them and see if you can clear up the confusion. Then, based on your conversation, rewrite your proof so that it's cleaner and easier to understand.

If you're working individually, proofreading is still possible but is a bit trickier. We recommend writing a proof, sitting on it for a day, then reviewing your answer a day later to make sure that you can still understand it. If so, great! If not, consider cleaning up your answer – after all, if you wrote it and have trouble understanding it, what do you think will happen when the TAs read over it? ☺

Get lots of practice

Practice! Do EVERY problem on your own, and then do them again. It helps so much.

Practice makes perfect. The extra problems are very helpful in solidifying your grasp on concepts.

Math is not intrinsically hard, the more you do, the better an intuition you build for it, and that's what separates a mathematicians from others.

DO AS MANY PRACTICE PROBLEMS AS POSSIBLE and try to think of everything as a cohesive, big picture.

DO TONS OF PROBLEMS.

Each week, we'll release a problem set that you'll submit for a grade. We'll also release some extra practice problems you can use to play around with the material. If you're shaky on a concept, try working through some of the practice problems first. We'll release solutions so that you can check your answers, and you're welcome to ask questions on our solutions.

Also, remember that CS103A exists. If you'd like to get some more practice in a structured environment (and meet other cool people you can work on problem sets with), consider enrolling!

And finally – some general advice from Fall 2014

come in with an open mind and eagerness to learn

Work. Your. Butt. Off. From the start. You can do it.

Let go of the conception that CS is all about writing code and learn to appreciate the deeper logical thinking behind it.

If you are stuck on a problem give your mind a break because you can easily lose your calm if you just keep trying to solve one problem for hours without end.

One good way to study is to go through the theorems from lecture, cover up the proofs, and write them up yourself.

Don't worry if the problem sets are taking you a long time--they are difficult, but if you do them, you'll feel it in your brains.

Best of luck in CS103 this quarter!