# Class 39 - Where to Go From Here

## Schedule

You have until **11:59pm tonight** to submit materials for qualifying for a new belt exam (unless you have already arranged for more time with me).

I will have my last scheduled office hours on **Tuesday (May 3), 3:30-4:30pm**. After that, I will have meetings by appointment, but I will be traveling May 5 through May 16, so not able to meet in person or accept any paper drop-off assignments during this time. To submit your work after May 3, you should send a PDF file that is a readable scan of your work. Please check the PDF is readable before sending it.

## Course Feedback

On the course pledge you signed:

> **I will provide useful feedback.** I realize that this is a new and experimental course, and it is important that I let the course staff know what they need to improve the course. I will not wait until the end of the course to make the course staff aware of any problems. I will provide feedback either anonymously or by contacting the course staff directly. **I will fill out all requested surveys honestly and thoroughly.**

Please honor this.

In addition to the University's course survey, I have **my own survey** (http://xplorecs/finalsurvey) that focuses on things I especially want feedback on in designing future courses. As an added incentive to do the survey, after submitting it you will receive the "magic words".

I am particularly interested in feedback on "tae kwon do" grading. The grading system used in this class was designed to emphasize learning, reduce student stress, and provide sufficient motivation for students to succeed in the class. I am not sure if it did this successfully, and I think there was a lot of confusion about how it worked. I would much appreciate your thoughts on this, and any ideas you have for how this might or might not work in a larger class.

## Where to Go From Here

If you'd like to see a summary of the main themes in the course, and some interviews with UVA alumni CS graduate students at Berkeley, watch Lesson 7 of the cs101 course.

If you would like to go into more depth on theoretical computer science and its mathematical foundations, take *cs2102: Discrete Math*. If you want to take another class from me, I'm teaching one of the sections of cs2102 in the Fall (the one listed now as "Staff", Tuesday/Thursday at 11am). If not, you should take David Edwards' section (Tuesday/Thursday at 2pm).

If you are stressed out during finals week, I recommend borrowing a three-year-old and going to Jump. If you can't find a three-year-old to borrow, just pretend to be one yourself.

If you like taking on-line courses and want to build on what you've learned in this class to learn how to make scalable, interactive web applications, you should take Steve Huffman's Web Development: How to Build a Blog course.

If you like taking on-line courses and want to get more deeply into how interpreters work and more advanced ways of using Python, you should take Westley Weimer's Programming Languages: Building a Web Browser course.

If you are interested in research in computer security or you have your own idea for and interesting project, talk to me about opportunities in my research group (see https://www.jeffersonswheel.org). I also have funding to support students working on developing educational materials targeted to high school students over the summer.

If you would like to understand better how music, art, and logic use recursive definitions, read Douglas Hofstadter's fascinating and compelling book, *Gödel, Escher, Bach: An Eternal Golden Braid*. If you prefer comic books, try *Logicomix: An Epic Search for Truth* by Apostolos Doxiadis and Christos Papadimitriou.

If you would like more personal advice on what to do next, feel free to contact me. I'm always happy to take any cs1120 graduates out for lunch or coffee to hear about what you're doing, and offer any advice I might have.

> *I think that it's extraordinarily important that we in computer science keep fun in computing. When it started out, it was an awful lot of fun. Of course, the paying customer got shafted every now and then, and after a while we began to take their complaints seriously. We began to feel as if we really were responsible for the successful, error-free perfect use of these machines. I don't think we are. I think we're responsible for stretching them, setting them off in new directions, and keeping fun in the house. I hope the field of computer science never loses its sense of fun. Above all, I hope we don't become missionaries. Don't feel as if you're Bible salesmen. The world has too many of those already. What you know about computing other people will learn. Don't feel as if the key to successful computing is only in your hands. What's in your hands, I think and hope, is intelligence: the ability to see the machine as more than when you were first led up to it, that you can make it more.*
> Alan Perlis, quoted in Abelson & Sussman, *Structure and Interpretation of Computer Programs.*

Although cs1120 has striven to be consistent with Perlis' spirit, when you fly in an airplane, put your money in a bank, get LASIK eye surgery, or live near a nuclear power plant, you would be worried if the people who programmed those things agreed with Alan Perlis that their job was not to make them "error-free". If you want to learn how to make robust programs, take *CS2110: Software Development Methods* in the fall. This courses focus on engineering, not computer science. That means the course is mostly about ideas and methods for building programs that behave reliably under constraints of cost and time (how long does it take to get the program working, and how expensive will it be to change it). The ideas and techniques you learn from cs2110 will enable you to think about design more clearly (whether of software, or something else), and will lead you to build more useful and exciting programs than you would without them.