What’s Next? Keep on Learning

Computing is about Power:

I want to come back to why learning about computing and learning to program is important. Most of you are computer science majors and will likely work at industry or academia as a programmer or a computer scientists. Some of you are from other majors and will equally likely be using computers in your profession. Learning to talk to computers (programming) is thus important for your careers whatever career you choose or land upon.

But there is more to computing than helping you in your jobs. Computing skills is a literacy skill similar to writing and reading. Computers are everywhere and will remain to be everywhere for the imaginable future. Our lives are more and more influenced by computing. Whether you are eligible to rent an apartment or to get a credit card, whether your resume is shortlisted for an interview, or what news is presented to you and what is hidden from you are all dictated by computing systems! Computing is shaping the way we think and what we see is of value.

Even though computing may seem neutral, there are human decisions behind it. You got a brief glimpse of it when you wrote conditional statements (if, if-else) and the fields you considered to be important when you created objects. All computing systems you see around you (e.g., Facebook, Google, Microsoft Word, Instagram, etc.) are created by making lots and lots of similar decisions. Those decisions are made by humans -- by programmers. So, computing systems are never neutral and have several limitations. This is true about Artificial Intelligence (AI) and Machine Learning (ML), which you may have heard so much these days. They are also computing systems and are vulnerable to the same weakness and limitations as any other computing systems. Developing computing skills enables us to notice these limitations and have the power to mitigate the harm that it may cause to us and to those we care about. We will have some power; it will not remain --- and should not remain --- only with the programmers working in a tech company.

No matter where you go (whether you work at a Big Tech company, academia, non-profits, or anywhere else), having computing skills means you will have the power to make the world better. I hope you see this class as the place where you could build a foundation further your computing skills. I hope you continue taking CS courses at SFSU (and elsewhere) and become a lifelong learner. In addition to the courses at SFSU, I want to encourage you to keep learning by using resources available on the Internet for free. Have confidence that this course has prepared you enough to be able to learn any programming language that you would like to.

How Do I Learn More About Java?[1]

The textbook (by Liang) is a valuable reference to keep going back to. We have covered upto Chapter 12 in our class. The book goes in greater depth in those chapters and there are additional chapters that are valuable to learn.

- You can enroll in CSC 220 where you will learn about data structures.
- To revise the concept that you have learned in this class:
    - There are some online resources like Codecademy and LearnJavaOnline
    - SoloLearn has a well-structured course
- If you want to make a mobile app, a good resource is Sofia, a Java library for making phone applications.
- An okay guide to read to start learning about Java is the Official Sun Guide, by the people who made Java. Don't be surprised if a lot of the stuff in there is confusing. So much about programming only starts making sense when you start making bigger and more complicated programs - then things will suddenly "click". It can take a few years of programming before some things start "clicking", but the best way to speed up that process is to keep coding!

I keep hearing about Python. How Do I learn Python?

Python is a very popular language, because it is easy to use but also very powerful. You can do amazing things in Python with very little prior knowledge.

- Try Codecademy, which is a good introduction. Codecademy, in general, is a good resource to develop a broad range of computing skills.
- A good interactive textbook is "How to Think like a Computer Scientist".
- If you want help from people, check out the subreddit for learning Python.
- If you're having a hard time understanding what happens when you run your program, you can try playing around with the Online Python Tutor or our python-to-blocks editor.
- One of the "Big Ideas" in computer science is using external libraries that other people have made, so that you're not reinventing the wheel. Learning how to use libraries is a great way to make your programming better, fast! Here are some cool libraries for doing interesting stuff.
    - You can make webapps in Python using Anvil.
    - A web application, using Bottle (or Flask, but that's more complicated). Fair warning, making a web application is medium difficulty, but getting it on the internet can be very difficult.
    - A desktop application, using TKinter (but there are other options too that look better!)
    - A mobile application, using Kivy
    - A game, using Pygame (or Spyral)
    - You might be the kind of person who likes math (Numpy), science (Scipy), and making cool graphs (matplotlib). Python is really great for that stuff too!
    - There are tons of other libraries, take a look through them sometime!

[1] Some of these resources were collected by CT@VT team.

This Python and Java stuff is too much, too fast!

If all this feels overwhelming to you, don't worry about it. Here are some other good options to try. These may not feel like “real programming” but trust me, they are very powerful ways to learn about computing. Remember, programming is not about text coding, it is about formulating ideas in a way that a computer understands.

- Blockly Games: Some really fun activities to learn how to program. These are drag and drop blocks which you use to program.
- Scratch: Another block-based programming system that will help you do a lot of things like create interactive stories and make fun games. There are tons of Scratch projects avaialble online that you can use for inspiration or reuse and build upon it.
- Alice: Like Scratch, only for moving 3D characters!
- AppInventor: Block-based programming for making mobile applications!
- Thunkable: A wonderful easy-to-use mobile application development system. You can publish the applications to Google Store and Apple Store.
- Game Maker: Block-based OR regular code writing.
- Unity: A system for making games, especially 3D ones!

I'm lost! Help!

- Google is always your friend. A recent survey of professional programmers found that almost 100% of them constantly use google and StackOverflow during their day-to-day job.
- Stack Overflow is a great site for getting answers to problems. You'll never, ever learn all there is to learn in this subject. It's necessary to rely on help from others!
- There's a subreddit for learning how to program.

What are some things I can do to be part of the programming community?

- Get a Github account and store your projects there. It's hard to learn how to use Github, but it's a very impressive thing to do. If you tell a professional programmer that you have a Github account with code on it, they'll be pleased!
- Start reading "Hacker News" and "r/programming". These are the major news sources of stuff about the programming world. A lot of it will be over your head though but that’s ok. They get over my head too! Computing is very very vast!
- As you start making stuff, create a website for yourself (an "eportfolio"). Although there are many options, I recommend Google Sites, which is very easy to get started.
- Try to go to Grace Hopper! It is an amazing community of diverse computer programmers and scientists. Computing is for everyone and Grace Hopper embodied that belief. This is an amazing experience, and there's often funding for students. Try to organize a trip with classmates, get a teacher interested.
- Code, code, code. I can't emphasize this enough -- you'll learn by coding. Make something that gets you excited. You can do so much with Computer Science, there's almost no limit!

Can I have another overview?

- A very popular news article was published that really explains "What is Coding?" I don't think you could ask for a more fun and comprehensive look from an article.
- Learn about technology, ethics, and society. Remember, you are not only becoming a programmer. You are a human being in a deeply connected world. It behooves us to learn holistically about our world which means going beyond programming. I highly recommend beginning by reading Critically Conscious Computing and following it with readings on ethics and books on technology and society (e.g., Automating Inequality, Race after Technology, Weapons of Math Destruction, etc.).