Intro to COM S: Week 1

> Programming is like sex: one mistake, a lifetime of maintenance – Michael Sinz

Preface

This series of articles is a learning path guide notes, and will not go into the specific knowledge content, please read the provided materials carefully in order, all you need to learn is in the materials. Hope you enjoy :)

Reading suggestions

You don't have to learn all the tools below, just look back when you available.

<Section one, tools that you need to know at beginning>

VPN:

As we all know, China has magic to keep us from accessing some contents, so we need to learn to defeat magic with magic.

So what is VPN? In short, a VPN is a tool that allows us to break network blockages, in terms of terminology, A **virtual private network** (**VPN**) extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network.[https://en.wikipedia.org/wiki/Virtual_private_network] From this definition, Chinese internet is a huge **PRIVATE NETWORK**.

Suggestion tools:

Clash: this a famous tool to cross the GFW(Great Fire Wall), it support almost all the devices, and easy to use. Here is the download link below:

Github Repo, Mac version of clash, Windows version of clash, Android version of clash

If you are ios users, I suggest to use shadowrocket.

I highly recommend reading the readme file of the repo to learn how to use it, but of course, considering the knowledge level of beginners, there is also a Chinese tutorial here: for clash, for shdowroacket.

However, only tool is not enough, you also need a server to connect in order to cross the GFW.

If you want to buy a vpn service, probably you could look at Gacloud, BoomCloud, 泡芙云.

Please pay attention to identify the authenticity of these service providers, or ask friends to recommend a reliable one, all VPN service providers have the risk of debt escape.

Git

Git is the most powerful tool for version control, while version control is important to SDE work. So at the beginning, it is needed to learn how to use Git.

Here is the simple guide of Git.

You are required to submit summary by pull request on Github, I suggest to use command line to commit.

Linux

Linux is the most normal operating system for COM S students. However, learning how to use Linux is not that easy. So I don't think you have to use Linux at beginning. However, Linux is more convenient than Windows. **Window is the worst system ever.**

Here is some basic Linux knowledge. I suggest you use Linux as your main operating system if you can. Terminal is better than GUI(Graph user interface).

Suggest Linux system: Ubuntu, Manjaro, WSL(Windows Subsystem for Linux)

Install guide: Official guide of Ubuntu, WSL, Ubuntu parallel install guide on Windows, Manjaro

IDE

An **integrated development environment** (**IDE**) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger. Some IDEs, such as NetBeans and Eclipse, contain the necessary compiler, interpreter, or both; others, such as SharpDevelop and Lazarus, do not.

The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a version control system or various tools to simplify the construction of a graphical user interface (GUI) are integrated. Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram for use in object-oriented software development.

IDE suggestions:

For Java: IntellJ IDEA

For C/C++: CLion, DevCpp(Why DevCpp? I think DevCpp is beginner friendly (At least I used this for about 3 years))

However, I have to admit that VScode is the most convenient and powerful IDE for every language. But in my prespective, to configure VScode is complex for beginners. I just like Jetbrains' IDE :)

Here is some resources that might helpful for you to handle on an IDE:

Official IDEA guide, Third party IDEA guide, Official CLion guide, Third party CLion guide, DevCpp guide

Also, Simple VScode guide

Tips: Jetbrians IDE are free for students. If you have an edu mail, just reg here and then you can get ALL FREE IDE!

If you don’t have tho, just use community version is ok.

Learn the principles of CS

At the beginning, I want to share some self-experience about learning principles.

Search ability: There is a famous joke that said most programmers are face to GOOGLE coding, that’s true tho. So develop the ability of how to search useful info is important. As you can see, although I provide many guides for you to read and learn, there are still amount of unknown info that need you to search and know. The first step to develop the search ability is that **make your question a question**. Check this repo to learn how to ask a question. Second step is that use English to search, it is important that English resources are more abundant than Chinese resources. **Don’t believe that English is useless for learning program, that’s bullshit.** Thrid step is that read official docs in a first order, don’t always trust the knowledge from others’ mouth instead of Official docs.

Do more: Always start trying than just thinking. Some students told me that they are worried about whether they can learn CS well. However, how can you know that even if you havn’t started to do that? So always remeber that try it yourself and then ask.

Develop yourself: I think this is the most important principle for CS. Here I want to share my own story. I was started learning CS for about 7 years ago. At that time, I also don’t know how to do well like your guys, then I found that the special ability of myself is that develop myself. I learned Android by developing lua apps on Android, flash my devices quiet often to try different systems. Everytime when I meet a problem, I will try to solve it myself at first, no matter use resources from google, from other docs. Aftter solve the problems, I will think the reason of this problem, that is, **develop yourself from problems**.

So please stay curious to explore what you don't know.

<Section two, beginning courses>

We will begin at UCB CS61B, I suggest that no matter how far you have studied about CS, it is also useful to re-learn data structure. Please stay humble.

For first week, we will learn some basic java knowledge and finish the labs. The lecture you have to read is below:

Intro, Hello World Java

A Little Programming

Values and Containers

Simple Pointer Manipulation

Arrays

Labs:

Lab 1: Intro to Java & Git

Homeworks:

HW0

HW1

Tips: you don't need to submit your hw on github, just write a simple summary.

Please remeber to finish the summary at the end of this week:)

Here is a website that might useful for you to learn: `csdiy.wiki`