Development Environment

Missing session lecture 1

To write c++ programs…

- You need to
  - Write c++ code
  - Compile it (so that your computer can understand the code)
  - Execute the compiled program
- You need appropriate tools!
  - there exists various options!!

IDE??

- Integrated Development Environment
- Supports almost every step to develop program!!
  - Coding
  - Debugging
  - Compiling & Releasing…
- Extremely Convenient
  - only thing you need to do : write code and press compile button

IDE??

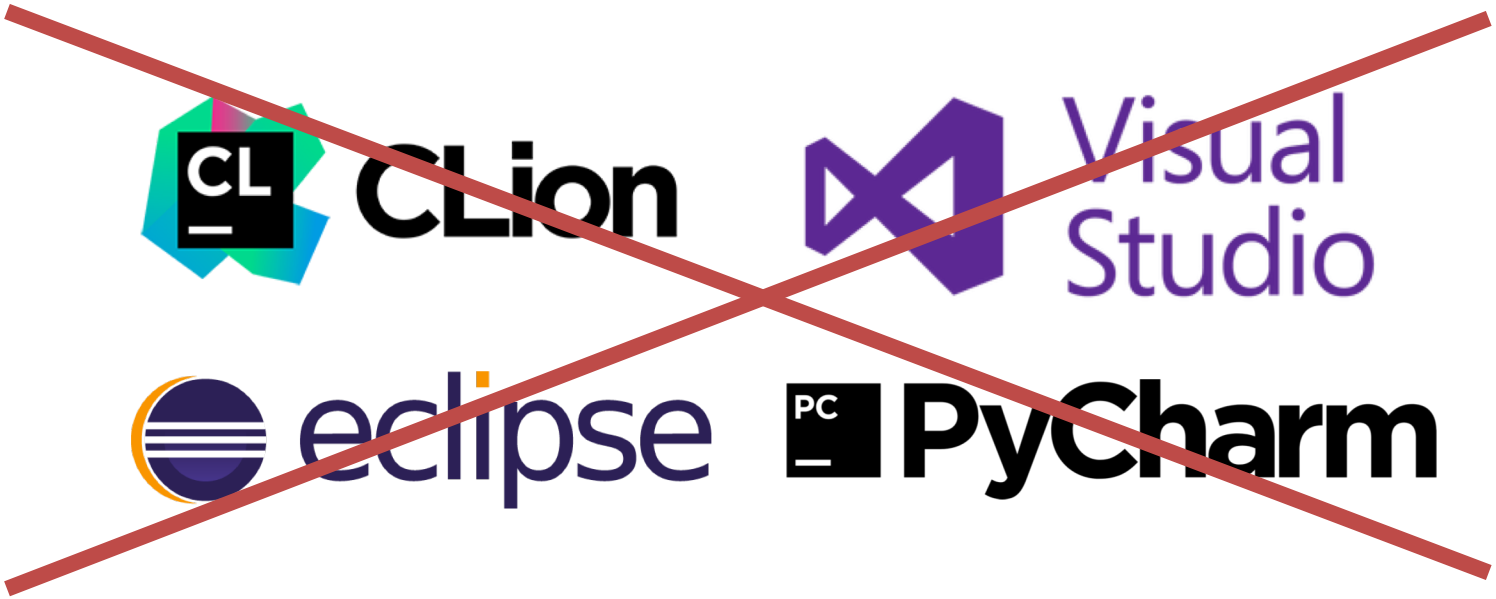
- Example (for cpp development)
  - Visual Studio

    - Developed by Microsoft
    - Most powerful IDE
    - Standard development tool for *Windows* developers
  - CLion

    - Developed by Jetbrain
    - Supports almost every major OSs (linux / mac OS / Windows / …)
    - High productivity & Convenience

IDE Hazard

- However, for novice programmers…
- IDE equals to *Disaster!!!* *(just my opinion…)*
- Why???



IDE Hazard

- Slow update
- Too many functionalities
  - Therefore, Too heavy
  - vulnerable to errors
  - Also, your laptop has limited memory & CPU resources…
  - save them!!
- (IMPORTANT) **HARD TO understand**
  - The fundamentals of programming
  - how your computer works

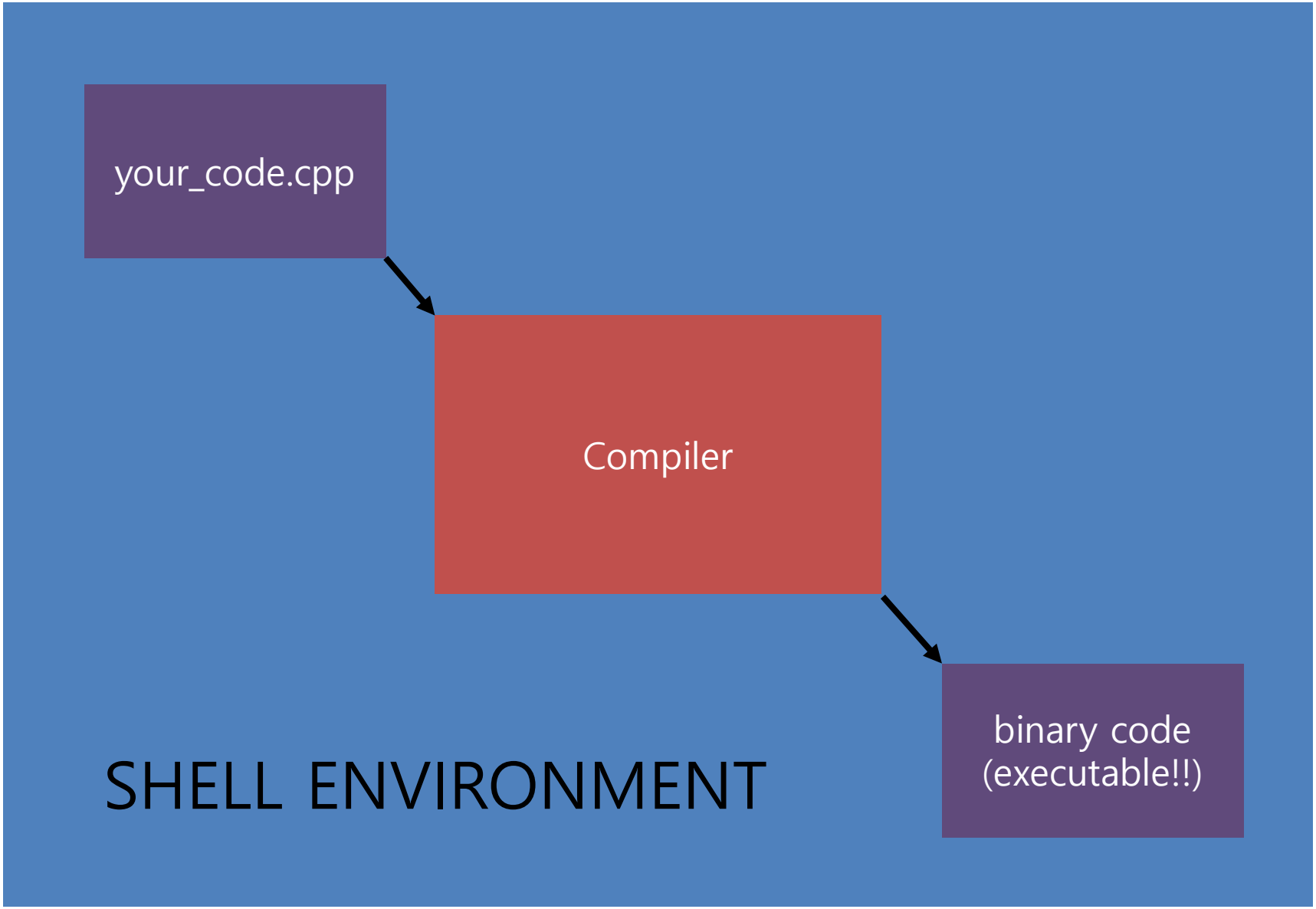
Your OWN Dev environment

- Therefore, you need to setup your own dev environment!!!
- This will help you
  - to understand the entire process of programming
  - to earn background knowledge to study computer system
  - **to customize your DEV ENVIRONMENT (most attractive feature)**
- And will help your computer
  - to save its resources
  - to be sensitive to latest techs (by setting & updating the environment on your own)

Basic components to develop something

- Editor
  - your have to “write” or “edit” the code!!
  - possible to just use *Notepad*...but less convenient
  - ∃ lots of fancy code editors!!
- Compiler
  - okay, you write the code, but computer cannot understand it
  - you need to “compile” the code which your computer can execute
- (CLI) Shell environment
  - conventional environment to write, compile, and execute code
  - TUI: command line interface - powerful tool with only a few commands

Why we need them?

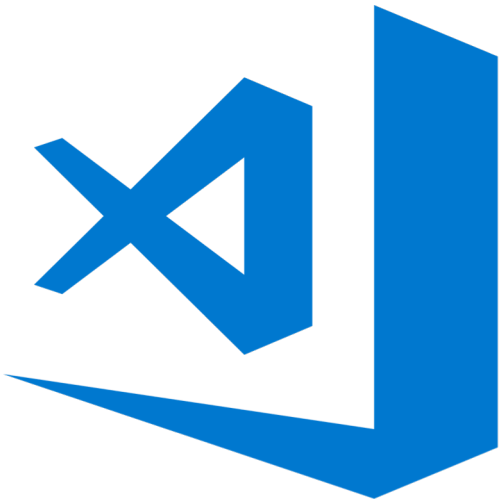


In this course

- In this course, I recommend to use
  - editor
    - Visual Studio Code
  - Compiler
    - gcc compiler
  - Shell
    - Window : command prompt & mingw environment
    - Mac OS : iterm
    - Linux : Maybe your already an expert of customizing dev environment..

Visual Studio Code

- Developed by *Microsoft*, released in 2016
- Currently, "Most famous" code editor
- Platform-free
  - supports Linux, mac OS, Windows
- Powerful extension environments
  - customize your own editor!!
  - supports almost all major programming languages
  - you can even develop your own extension
- * *Visual Studio Code is not Visual Studio!!*



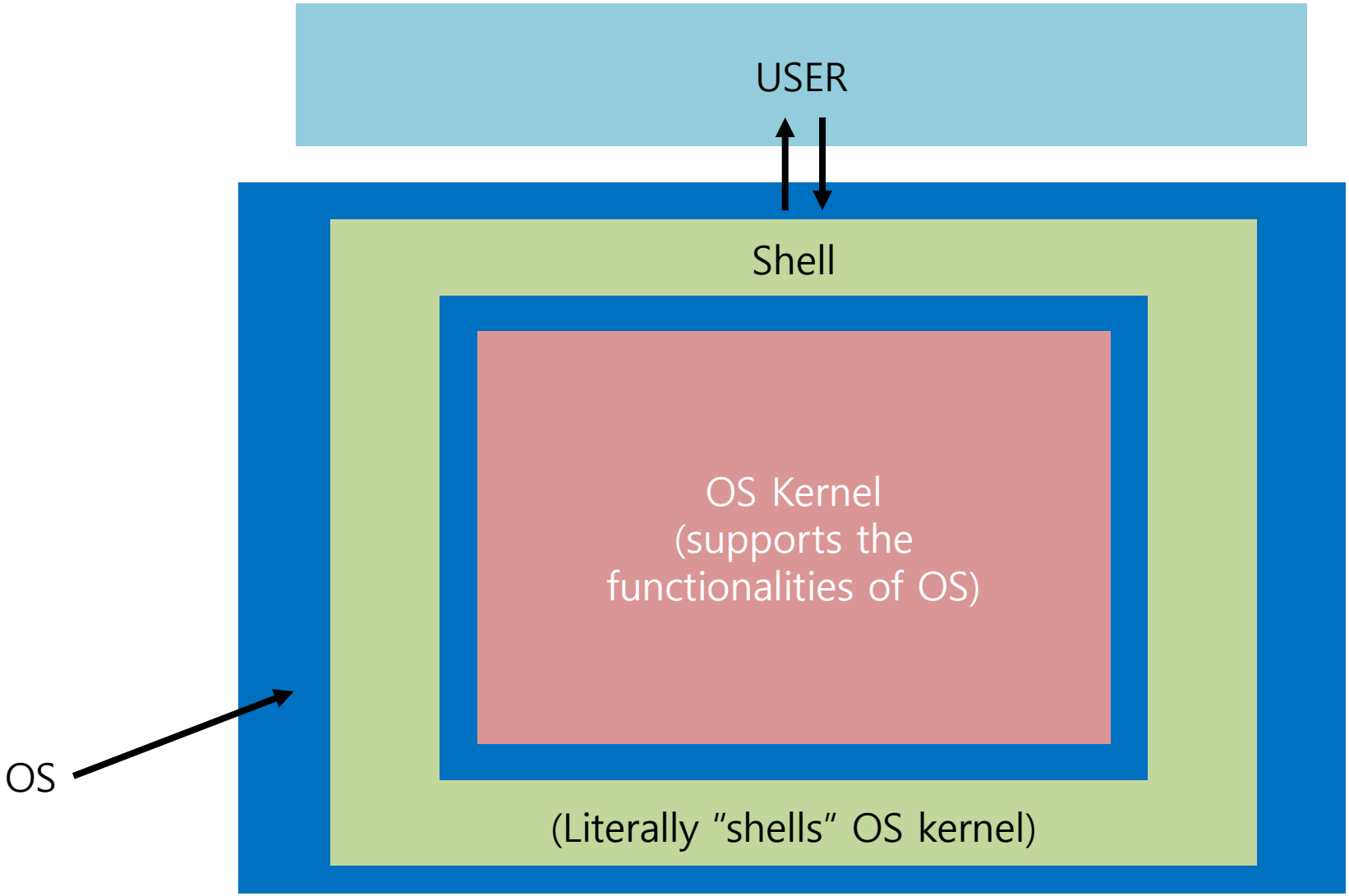
GCC Compiler

- GNU Compiler Collection
- initially developed by *Richard Stallman*
- open source, free, multi-platform
- supports
  - c, cpp, objective-c, objective-cpp
  - Fortran, Java, Go, Pascal
  - and else…

Shell / OS

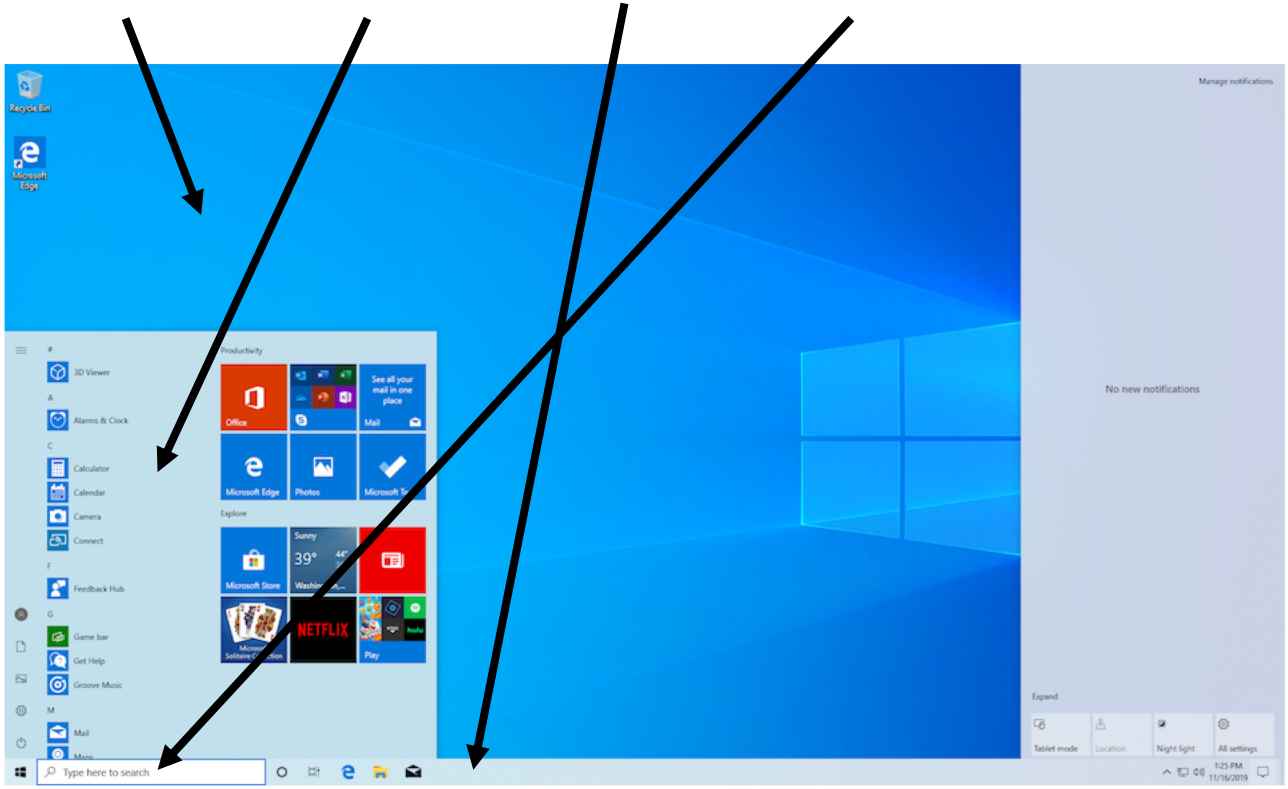
- Shell
  - What is shell?
  - User interface for access to an **OS**'s services *(Wikipedia)*
  - Actually, part of OS
  - So, to understand shell, you must know the basic of OS!!
- OS (Operating System)
  - Manages computer hardware resources (CPU, memory, HDD…)
  - Manages processes / programs
  - Provides file system (including directory, file, link(shortcut))

Shell / OS


USER
Shell
OS Kernel
(supports the
functionalities of OS)
(Literally "shells" OS kernel)
OS

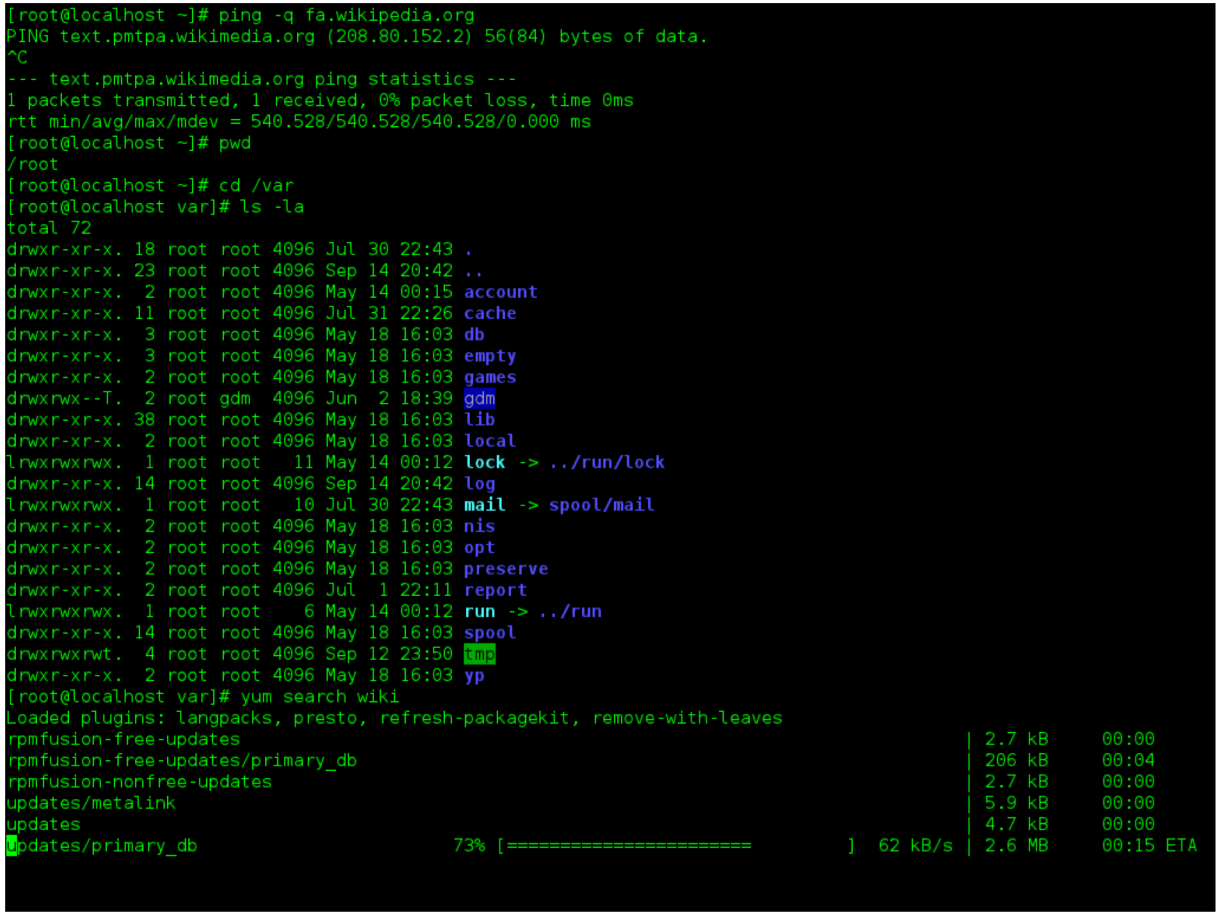
Shell

- You're already using shell!!
  - Windows shell
    - Desktop, Start menu, Taskbar, Explorer…



Shell

- In this course, we'll use CLI shell!!
- Maybe familiar to CS students



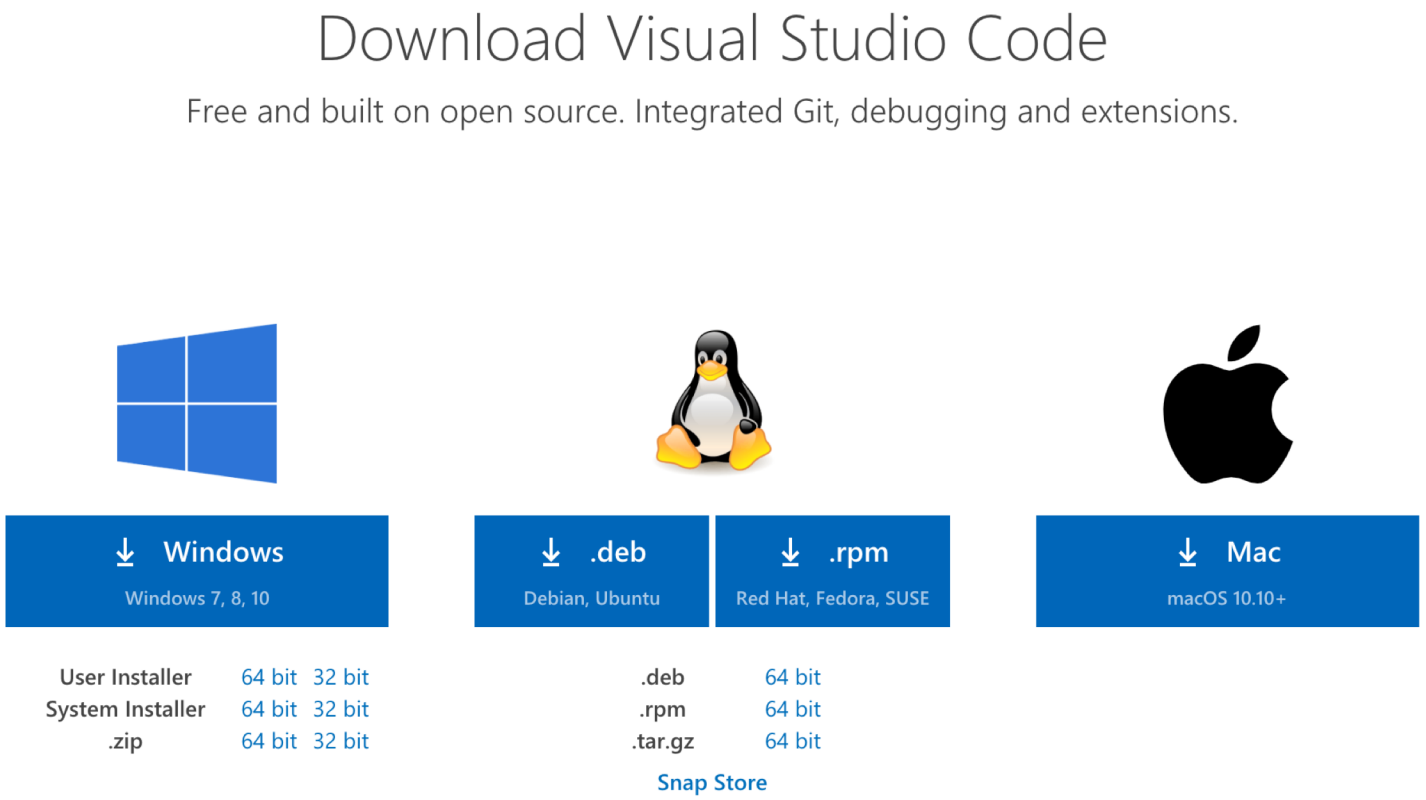
CLI Shell

- Convenient, light, and fast
  - If you become familiar…
- Can manage entire OS / Dev environment
  - with only few commands
  - for beginners: `ls`, `cd`, and `cat` commands are sufficient!!
  - we'll see them soon

Let’s start the setting!!

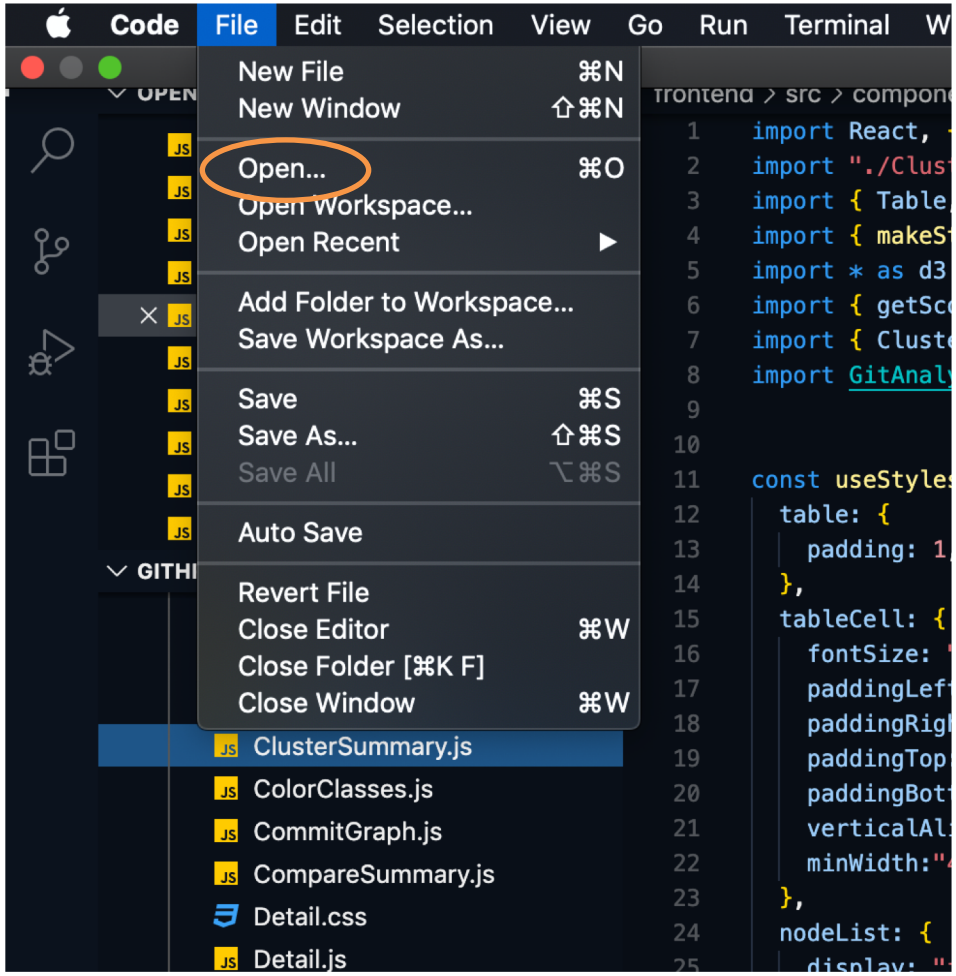
Visual Studio code

- Reference official download page (Very easy)
- https://code.visualstudio.com/download



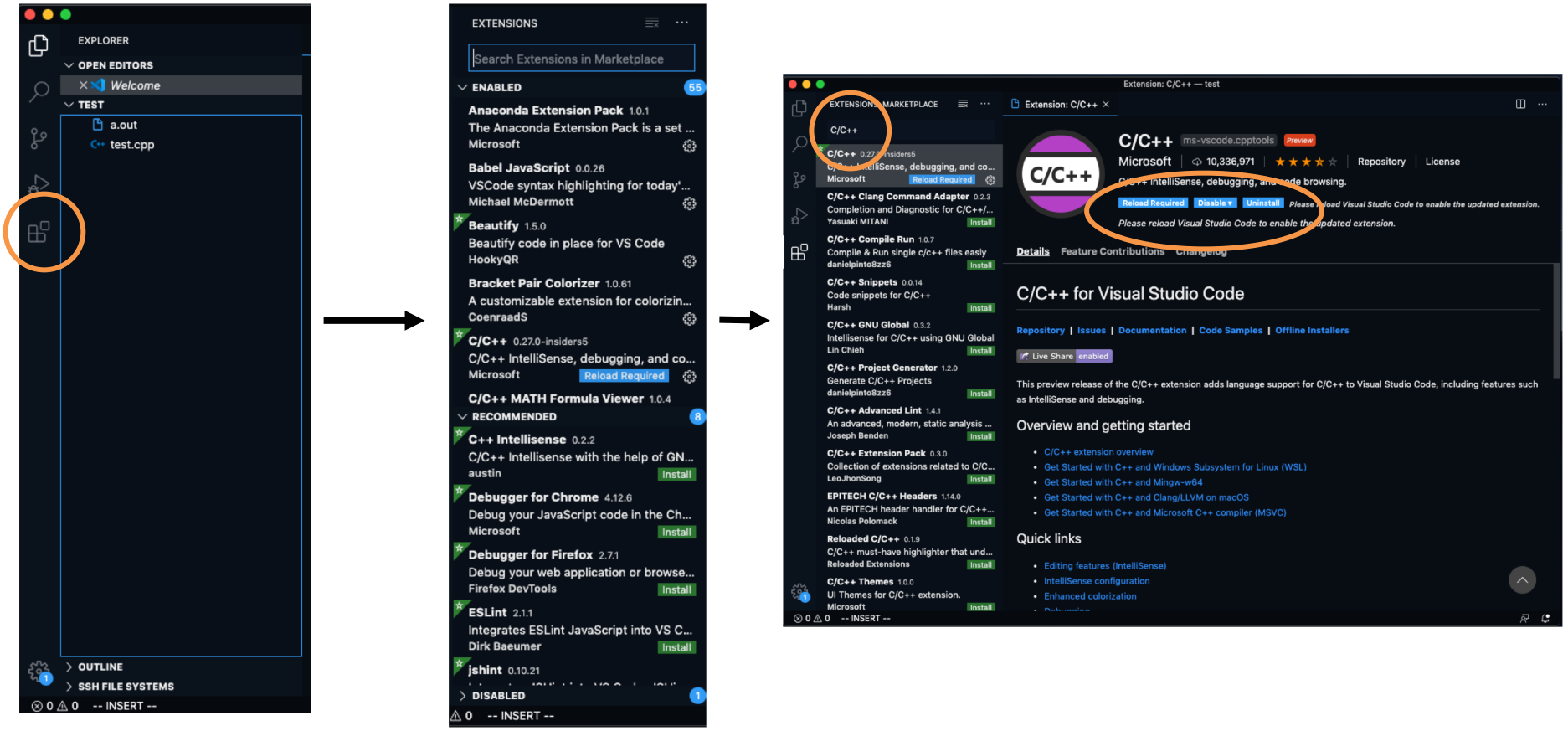
Visual Studio code

- Select working directory by accessing File >> Open menu
- Now you're free to use the editor!!



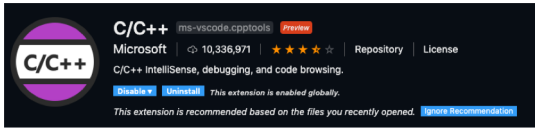
Visual Studio Code

- To earn the full power of VS Code…
- You must install proper extensions!!

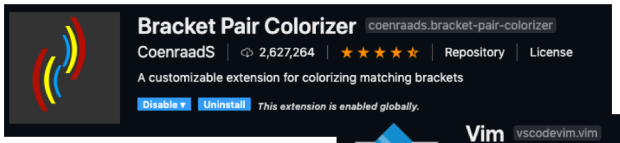


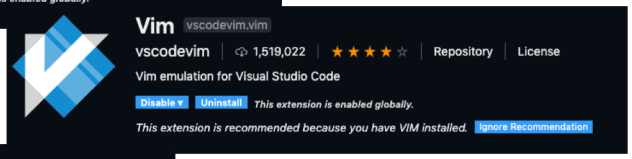
Visual Studio Code

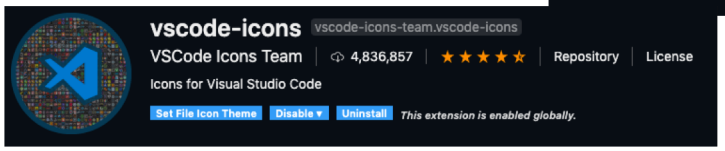
- Installing extensions
  - Search it / and install it!!
  - Very easy….I guess
- Essential extensions for cpp dev:
  - C/C++



- Recommendation:
  - Bracket Pair Colorizer
  - vim (if you're familiar to vim editor)
  - vscode-icons

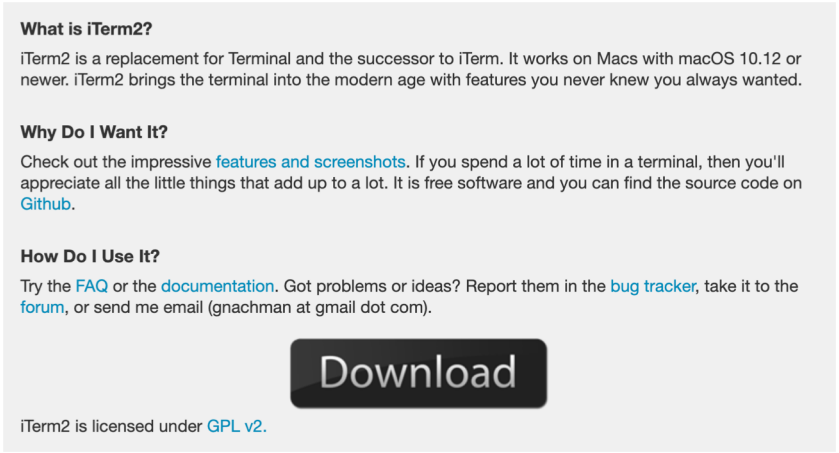






Shell

- Linux
  - Use default cli shell (Terminal)
- mac OS
  - mac provides default cli shell…but ∃ a better alternative
  - install iterm2
  - https://www.iterm2.com/



What is iTerm2?

iTerm2 is a replacement for Terminal and the successor to iTerm. It works on Macs with macOS 10.12 or newer. iTerm2 brings the terminal into the modern age with features you never knew you always wanted.

Why Do I Want It?

Check out the impressive features and screenshots. If you spend a lot of time in a terminal, then you'll appreciate all the little things that add up to a lot. It is free software and you can find the source code on Github.

How Do I Use It?

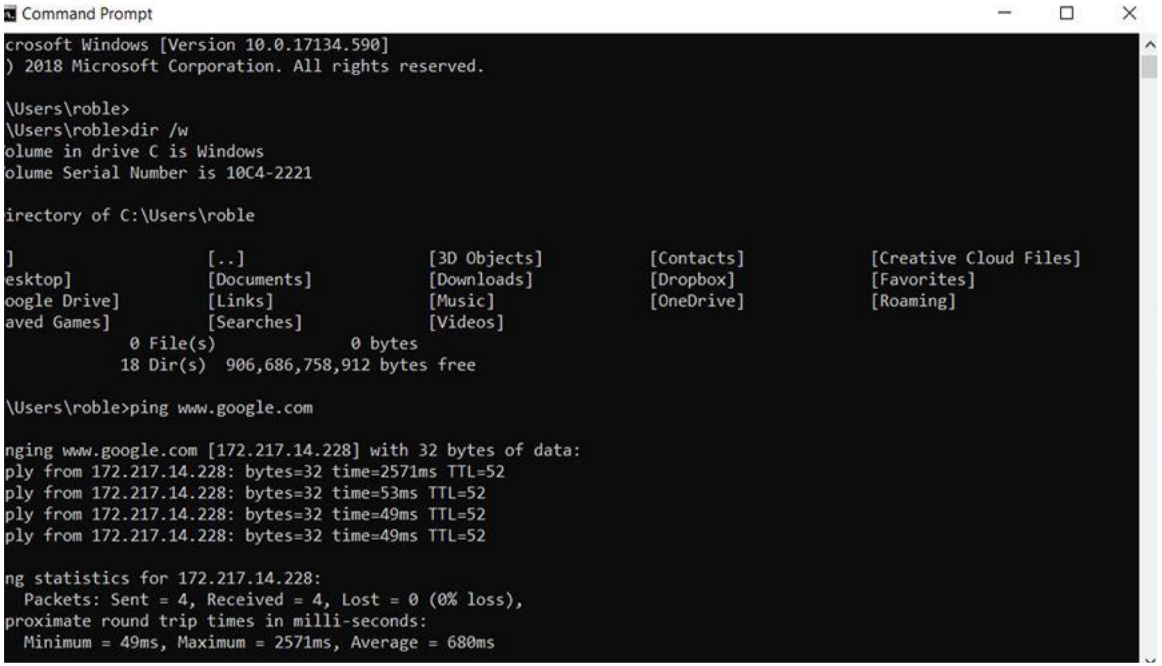
Try the FAQ or the documentation. Got problems or ideas? Report them in the bug tracker, take it to the forum, or send me email (gnachman at gmail dot com).

Download

iTerm2 is licensed under GPL v2.

Shell

- Windows
  - use “command prompt” (already installed)
  - press (⊞ `Window key`) + `R` and type “`cmd`”
  - reference https://www.lifewire.com/how-to-open-command-prompt-2618089



GCC compiler environment

- Linux
  - https://linuxize.com/post/how-to-install-gcc-compiler-on-ubuntu-18-04/
- mac OS
  - open iterm2 terminal (cli shell)
  - install ”brew”
    - paste this command in the terminal
    - `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)”`
  - install “gcc”
    - run the command: `brew install gcc`
    - check whether the installation finished successfully or not by
    - running the command: `gcc -v`

GCC compiler environment

- Windows
  - we'll use MinGW!!
    - Minimalist GNU for Windows
    - provides GCC compiler for Windows
    - you should first install MinGW, and then integrate it to Windows
  - http://www.mingw.org/wiki/Getting_Started
  - Follow the instruction!!
  - **Korean ver: https://mingtrace.tistory.com/446**

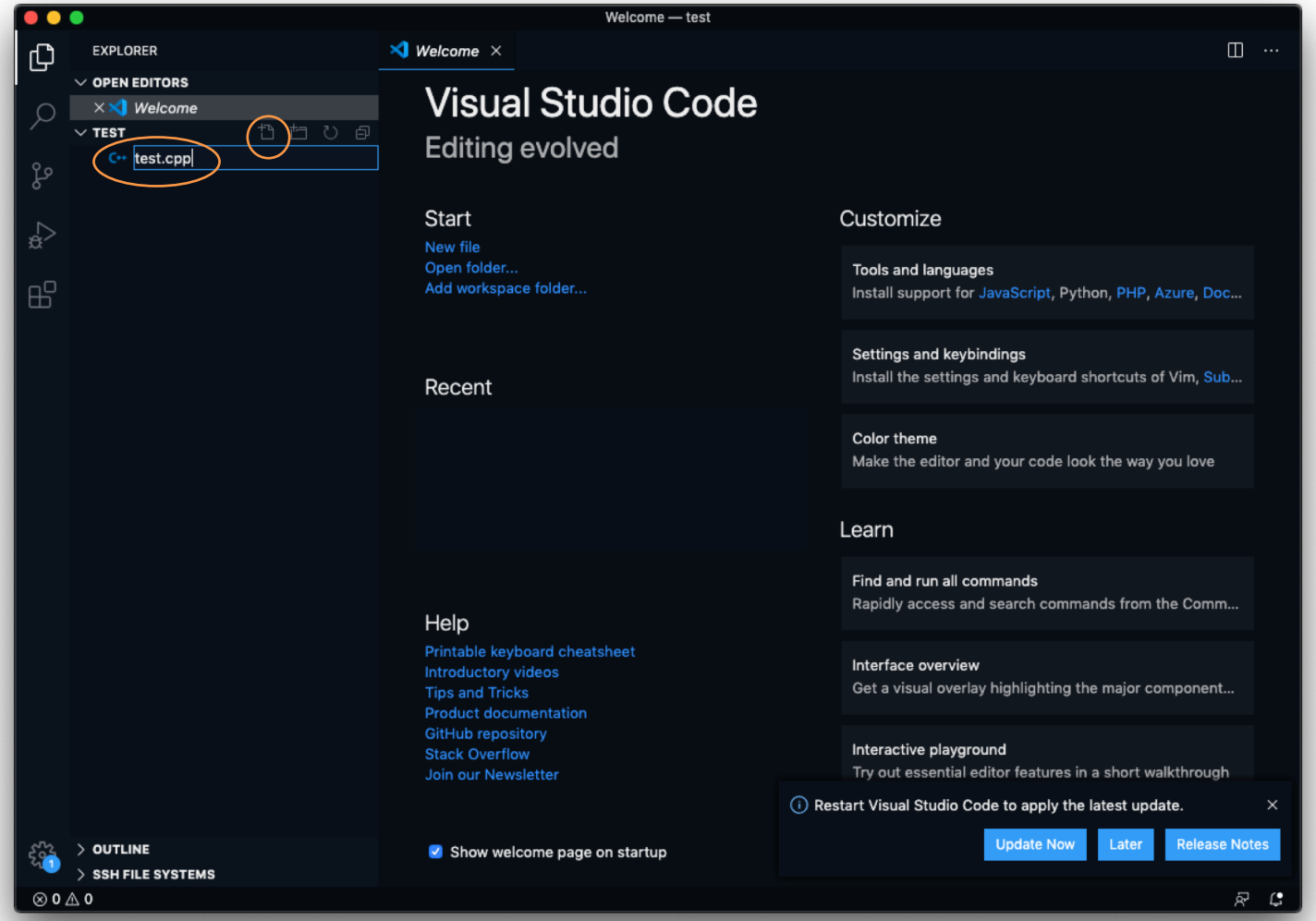
Okay, let's test the environment!

1. Make test directory in your desktop

2. Execute visual studio code, and open the directory:
   - Path will be similar to `~/Desktop/test`

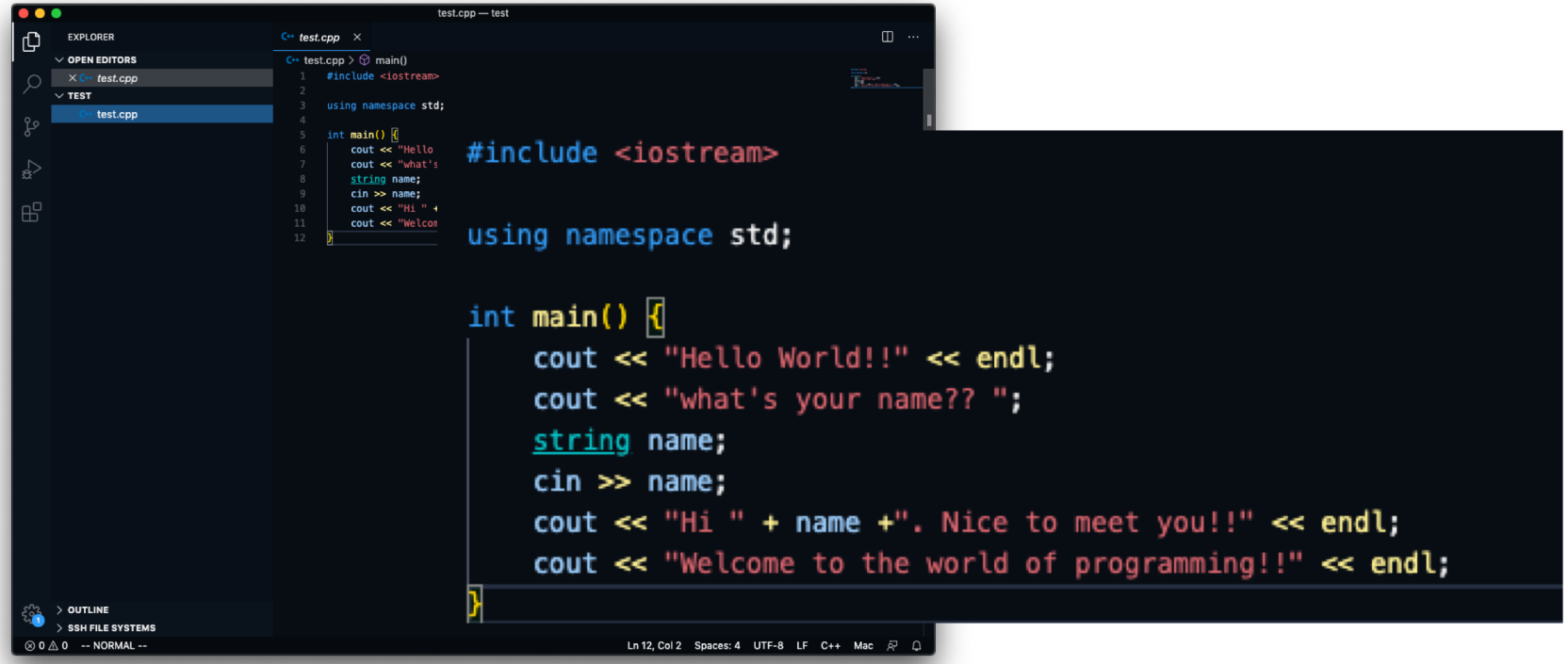
Okay, let's test the environment!

3. select "new file" icon and create new file: "`test.cpp`"



Okay, let's test the environment!

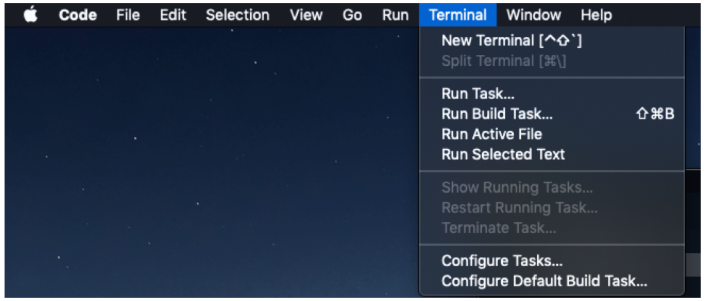
4. Write Test code



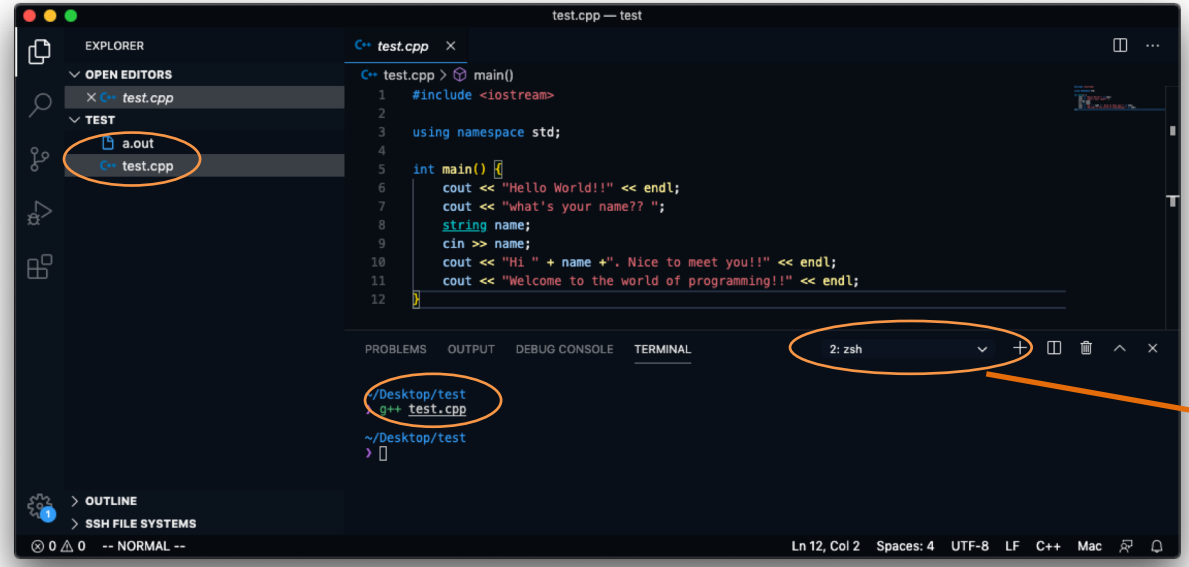
- The code is available at : https://github.com/jeonhyun97/cpp_4_undergraduates/blob/master/missing_session/lecture1/code/test.cpp
- but I highly recommend you to type the code in your own

Okay, let's test the environment!

- Open terminal (cli shell) by using Menu Tab:
- Terminal >> New Terminal



- run the command: `g++ test.cpp`



new executable file will be created!!
max / linux : `a.out`
windows: `a.exe`

For windows, change this tab to "command prompt"

Okay, let's test the environment!

- Now ready to run the code!!
- run the command:
  - Windows: `a.exe`
  - mac OS / Linux: `./a.out`
- The program will ask your name...answer
- And it will say hello to you!!

Thank you!!

contact: jeonhyun97@postech.ac.kr