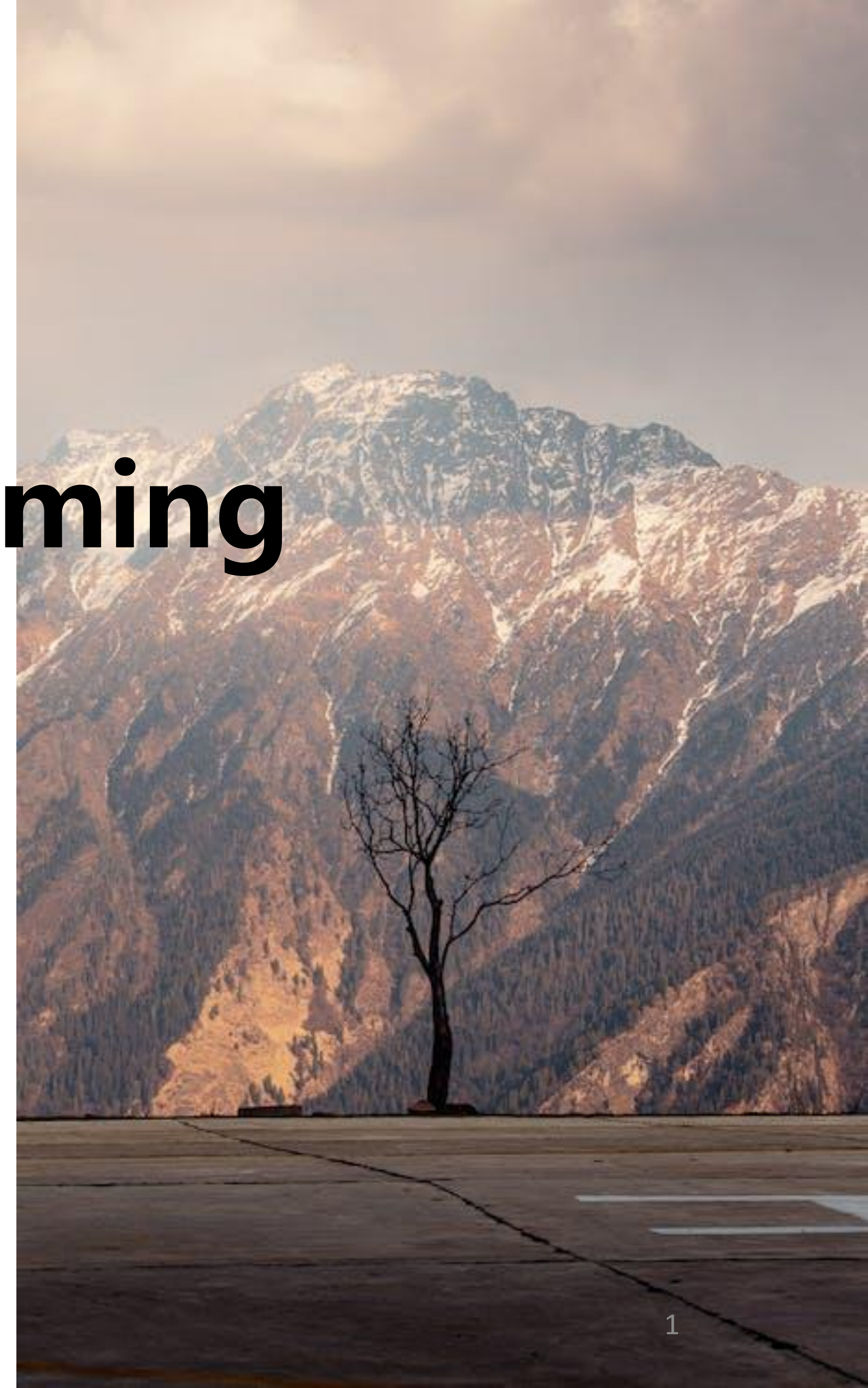


Object Oriented Programming

Lec01: Course Information

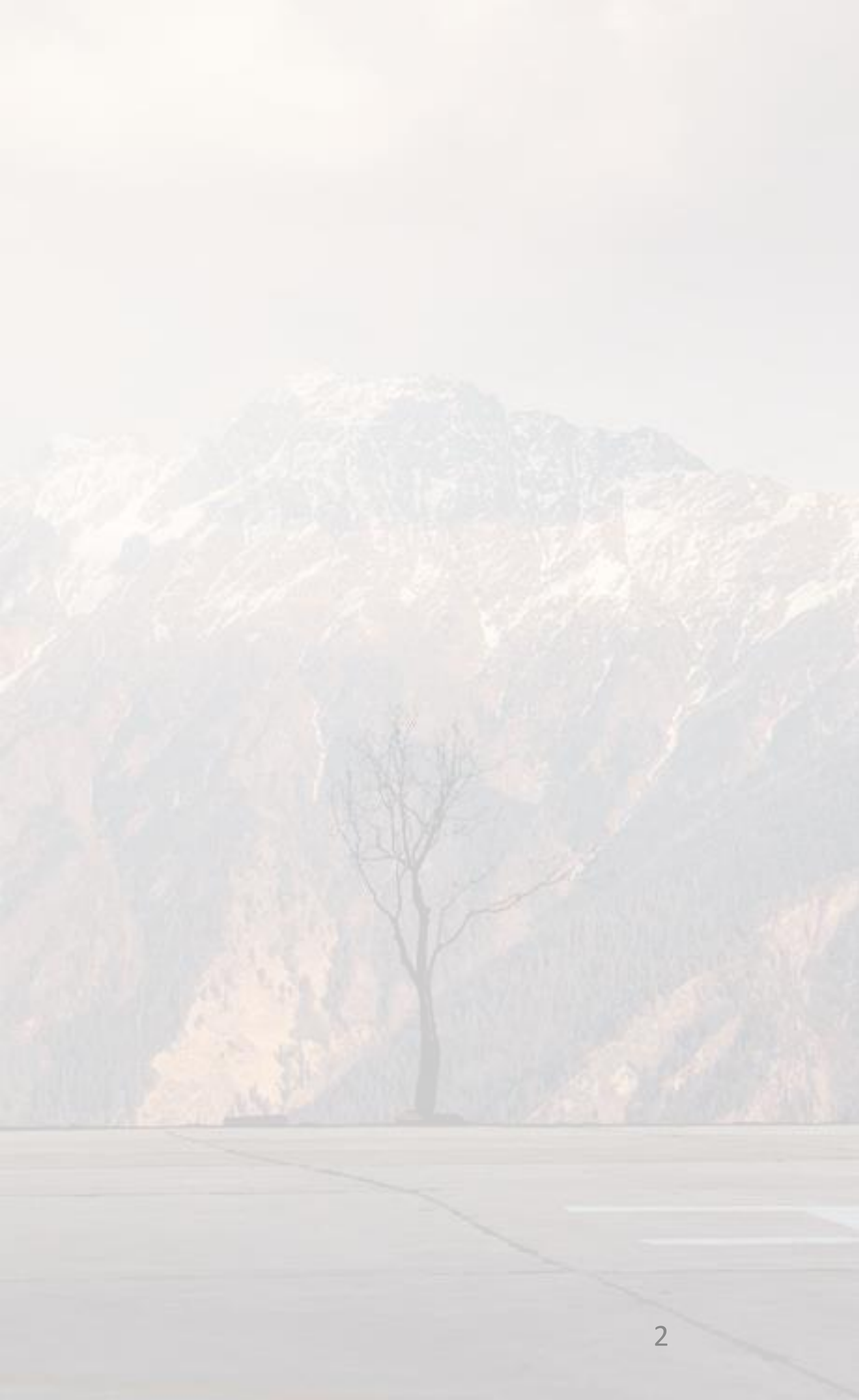
Sun Chin-Yu（孫勤昱）

cysun@ntut.edu.tw

2023/09/11

Course Information

- Course name: Object Oriented Programming
- Course time and place:
  - Lecture (3 Hours)
    - Monday 15:10 – 16:00
    - Friday 13:10 – 15:00
    - The Sixth Teaching Building 327
- Midterm & Final involve both hand-written and computer-based exams
  - Check the Announcement beforehand
- Course Website:
  - http://oop.is1ab.com/gitlab/OOP2023f/announcement
  - 北科i學園



Course Instructor and TAs

Prof. Sun, Chin-Yu

NTUT CSIE

ISLab Leader

cysun@ntut.edu.tw

Huang, Zheng

110 NTUT CSIE

t110590003@ntut.org.tw

Huang, (Uriah) Han-Xuan

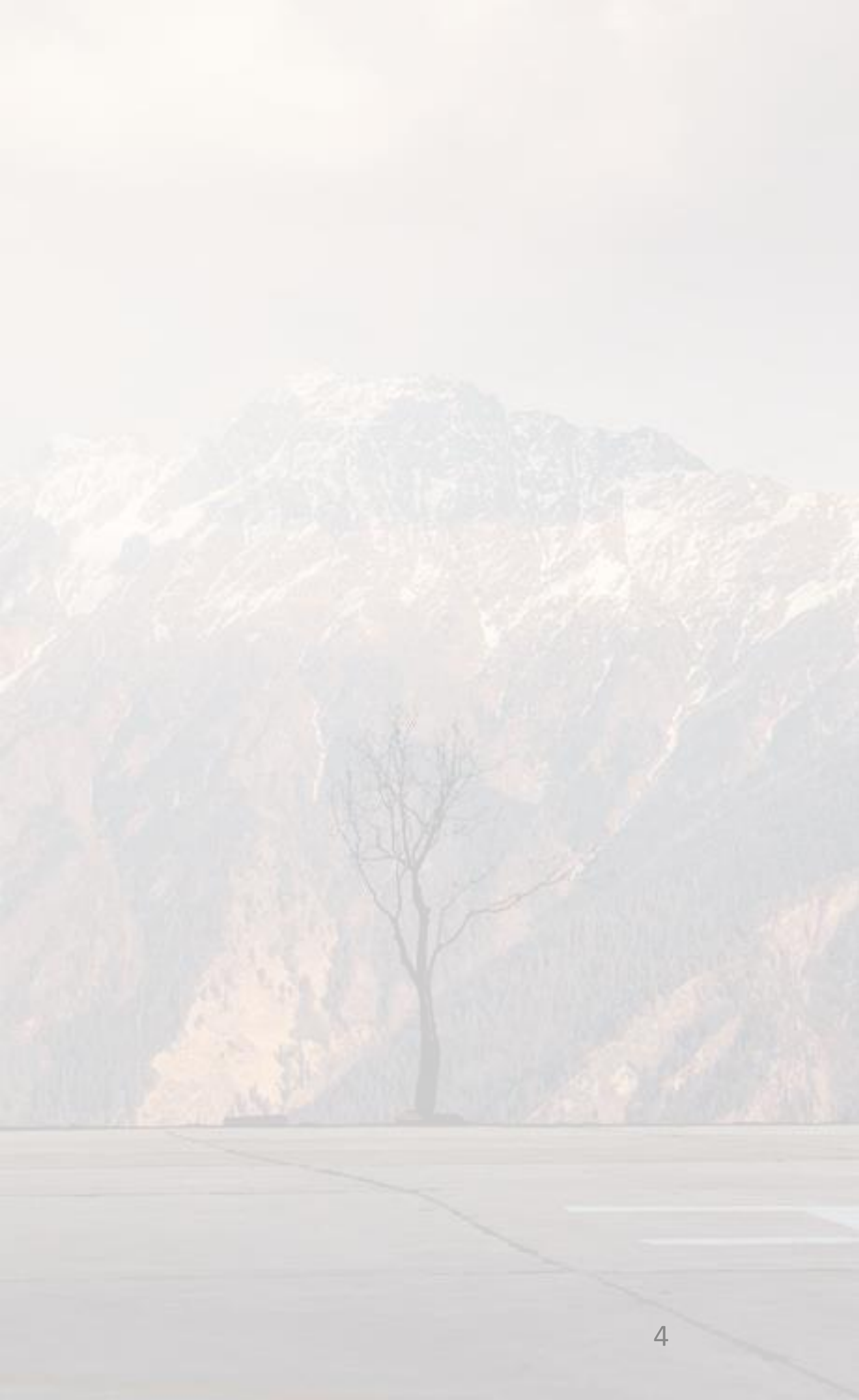
109 NTUT CSIE

CSF Engineer Intern

t109590031@ntut.org.tw

Prerequisites

- Attend & Interact
- Familiar with C or have taken 計算機程式設計（一） and 計算機程式設計（二）
- You may need a laptop for online exams and to complete the homework assignments
- And, of course, you should be willing to learn C++

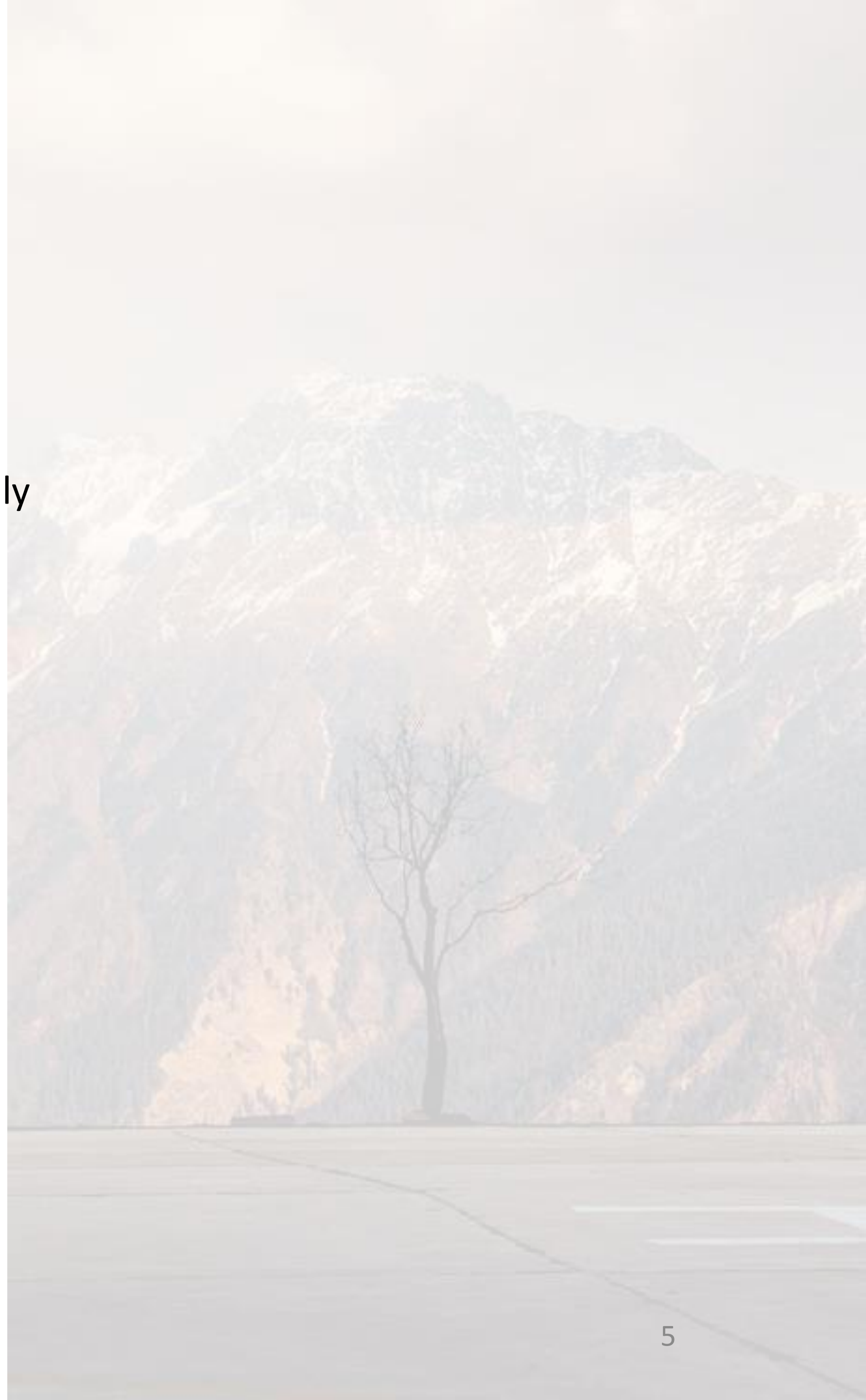


Course Grading Policy

- **Assignment: 30%**
    - Read the problem content and complete the tasks
    - Submit the homework to GitLab and trigger the judge system (Jenkins) automatically

- **Midterm Exam: 35%**
    - Hand-written exam: Answer the problem about the concept of OOP
    - Online exam: Complete the task and submit the code to Jenkins

- **Final Exam: 35%**
    - Hand-written exam: Answer the problem about the concept of OOP
    - Online exam: Complete the task and submit the code to Jenkins



Examination

- The examination:
  - Hand-written exam: 60 minutes
  - Online exam: 3 hours
    - **11/10 (Fri.) 18:00 – 21:00**
    - **1/5 (Fri.) 18:00 – 21:00**
- Survey of the examination participation:
  - Please fill it to report the participation
  - If you do, you will get +1 point on the final grade :D
  - If you do not, you will lose -3 point on the final grade >:(

Schedule

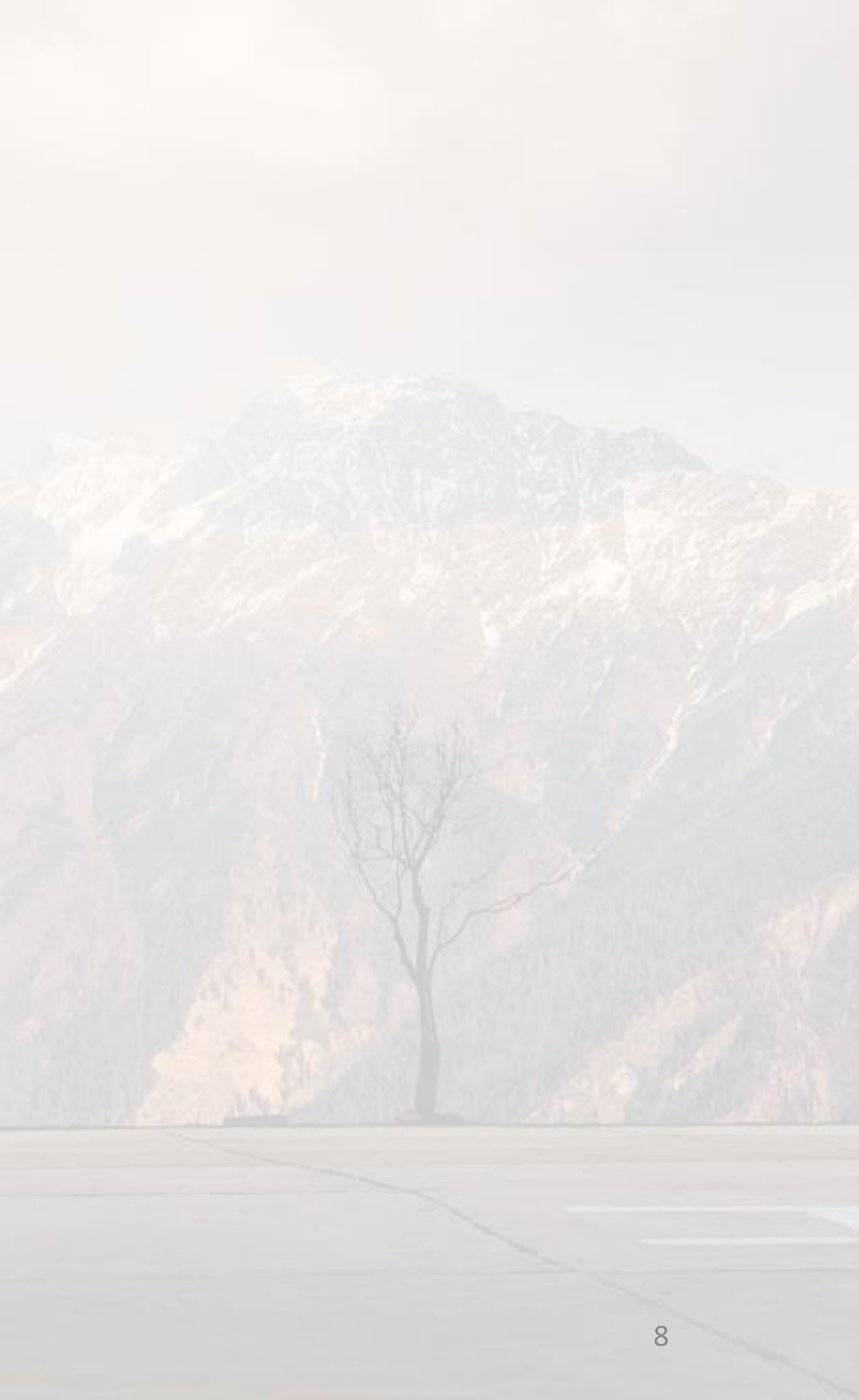
W	Date	Lecture	Homework
1	09/11, 09/15	Lec01: Course Information, Environment Introduction, and OOP Conception	
2	09/18, 09/22	Lec02: Class Introduction, and Essential STL Introduction	
3	09/25, **09/29**	Lec02: Class Introduction, and Essential STL Introduction / No class due to Moon Festival	
4	10/02, 10/06	Lec03: Encapsulation	
5	**10/09**, 10/13	No class due to the bridge holiday of Nation day / Lec03: Encapsulation	
6	10/16, 10/20	Lec04: Inheritance	
7	10/23, 10/27	Lec04: Inheritance	
8	10/30, 11/03	Lec05: Polymorphism	
9	**11/06**, **11/10**	**Physical Hand-Written Midterm** / **Online Computer-based Midterm**	
10	11/13, 11/17	Lec05: Polymorphism	
11	11/20, 11/24	Lec06: Composition & Interface	
12	11/27, 12/01	Lec07: Factory	
13	12/04, 12/08	Lec08: Dependency Injection	
14	12/11, 12/15	Lec09: Efficiency	
15	12/18, 12/22	Lec10: RAII	
16	12/25, 12/29	**Physical Hand-Written Final**, Flexible time	
17	**01/01**, 01/05	No class / **Online Computer-based Final**	
18	01/08, 01/12	No class & Let’s you guys to get prepared for other finals	

Goal

- We expected you learning how to implement the code with OOP paradigm
  - Make your code better, not longer, bigger, or mess up
  - Object-based paradigm helps your code more intuitional

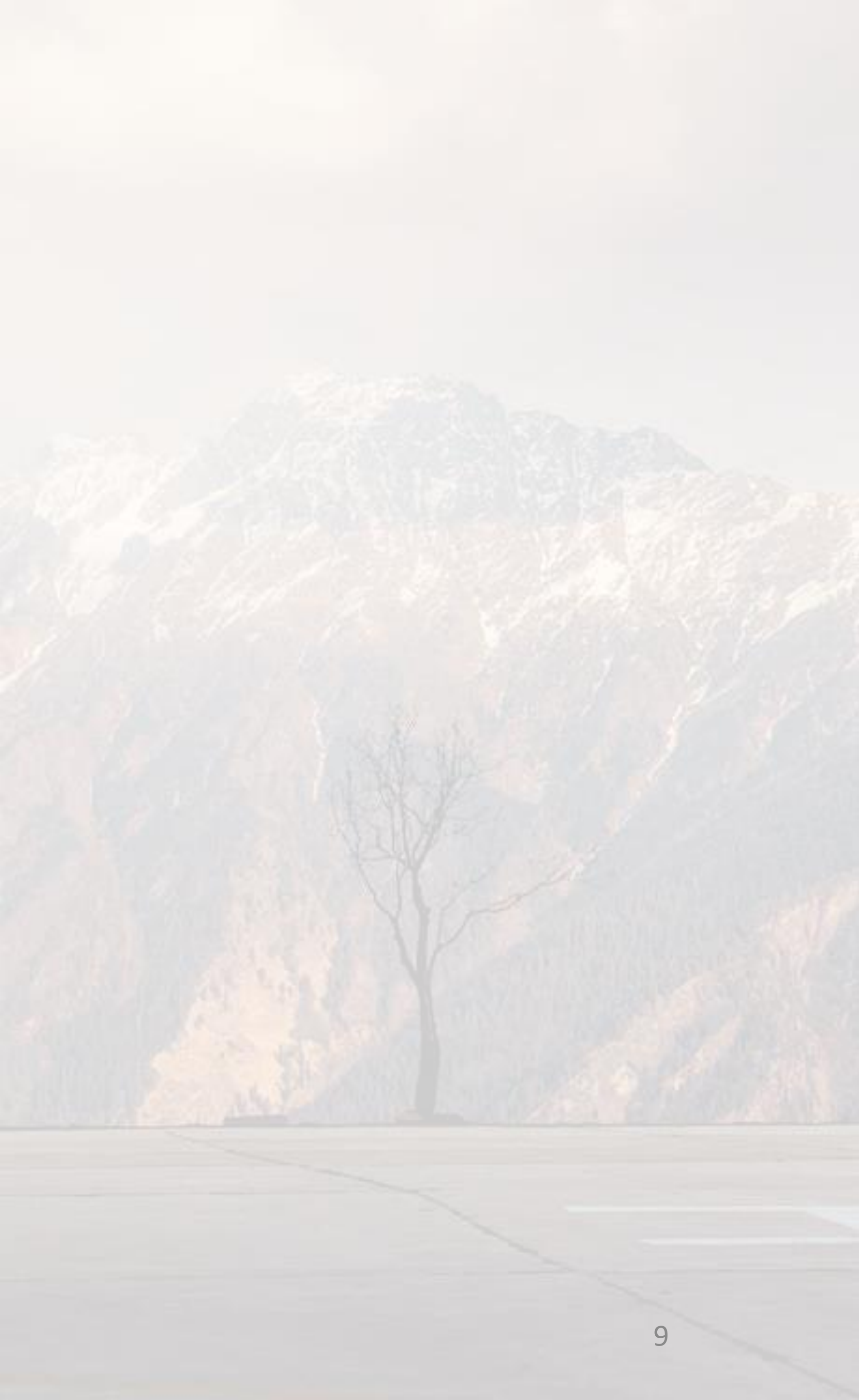
- We expected you familiar with how to write code better
  - Not faster
  - Write the unit tests and integration tests to confirm your code work smoothly
  - Check the code work smoothly every commit with CI tool
  - Check the test coverage to know the miss
  - Check the memory-check tool to check the memory leak

- We expected you familiar with how to use Git



The editor or IDE

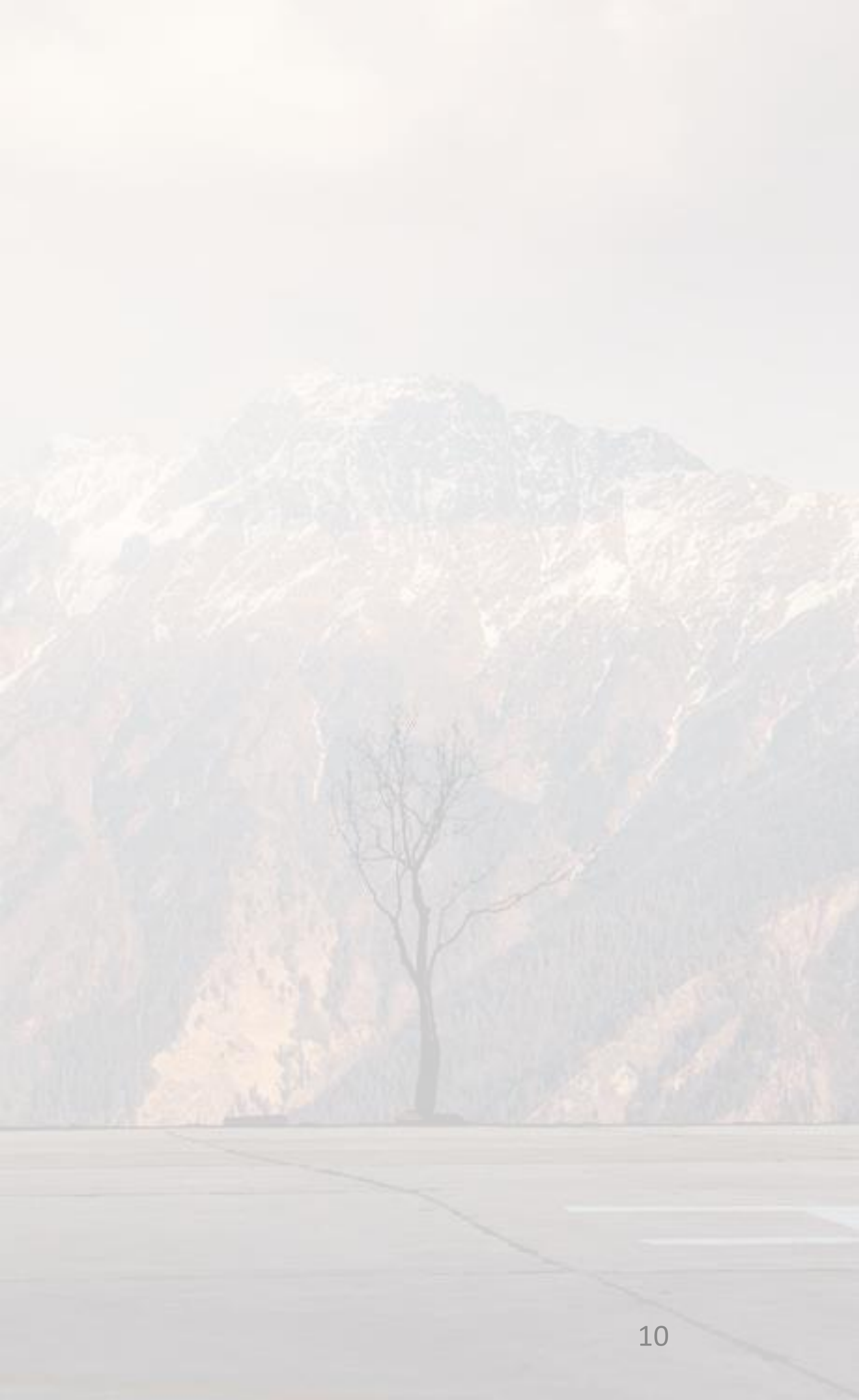
- Suggest to use VSCode, CLion, or Visual Studio
- With the syntax highlight and various tool help you coding better



Build

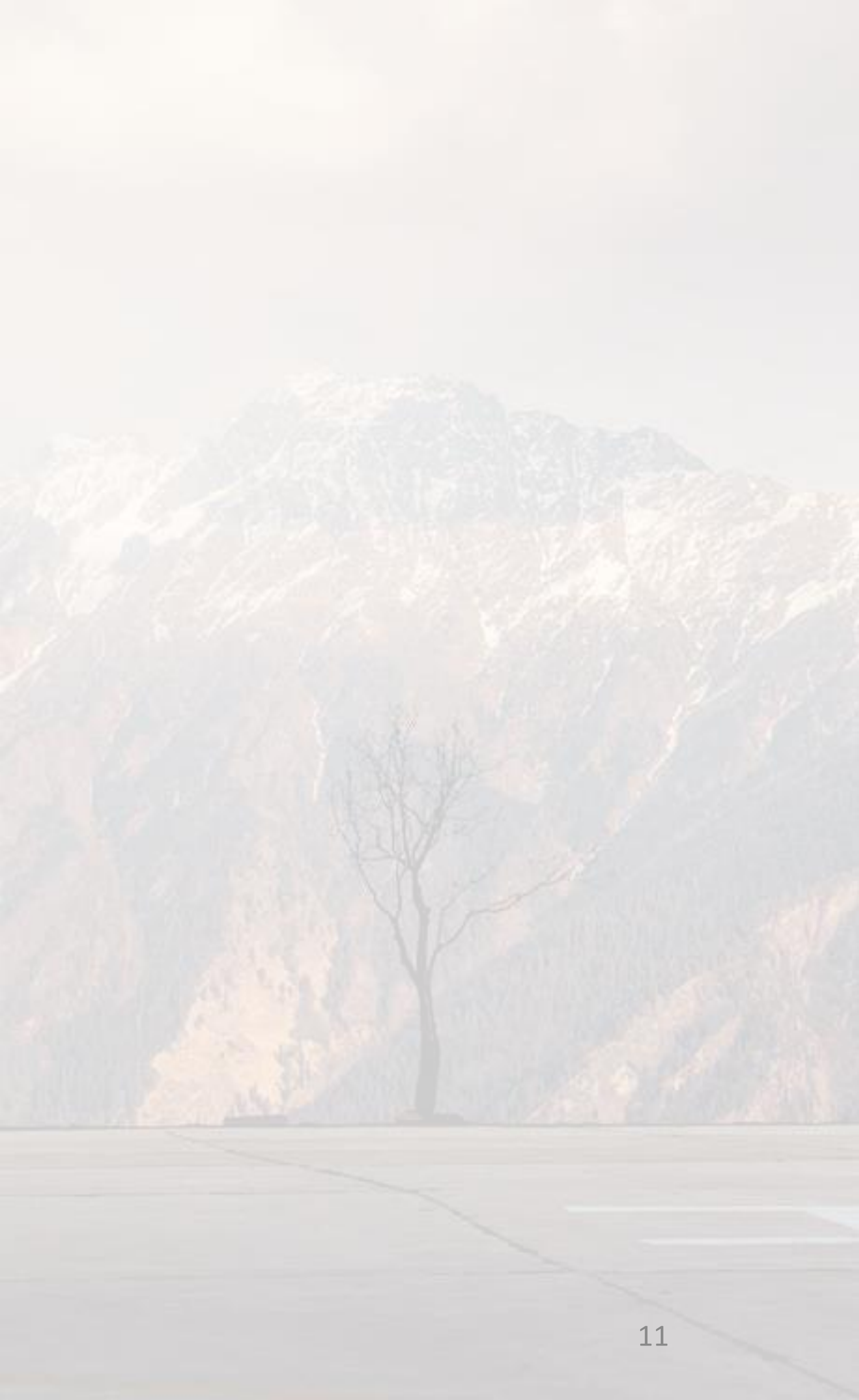
- We use CMake to build the project

- The CMakeLists.txt will complete when the homework publish
- You don’t need to modified the CMakeList.txt file



How to implement your code

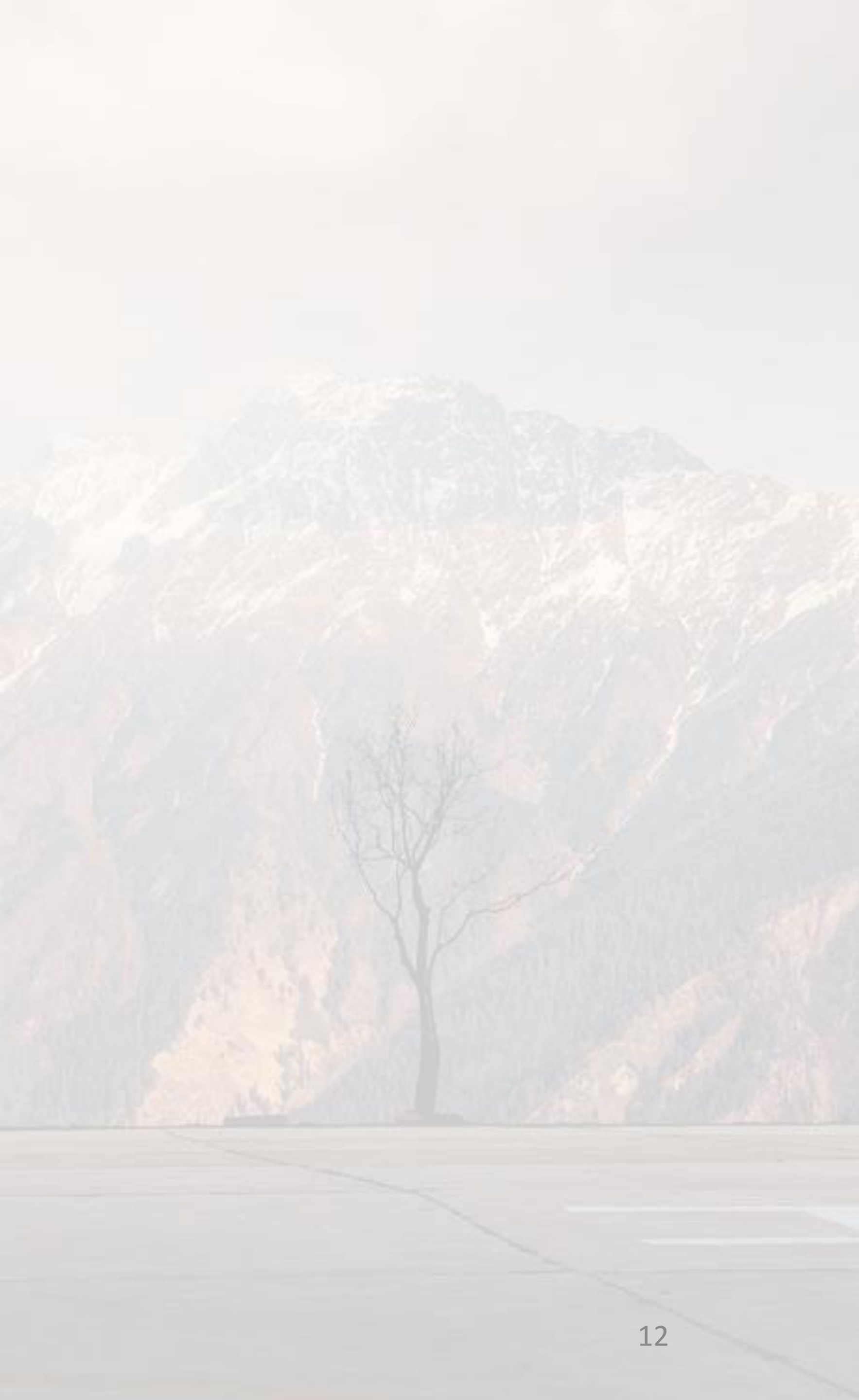
- Write the code
- Write the tests with Google Test
- Build and execute the project with CMake
- Check the memory leak with Valgrind in Jenkins
- Check the tests coverage with Gcovr in Jenkins



After you implement your code

- Push the necessary file to GitLab
  - You should be familiar with how to use Git
  - You should not push the binary file to GitLab

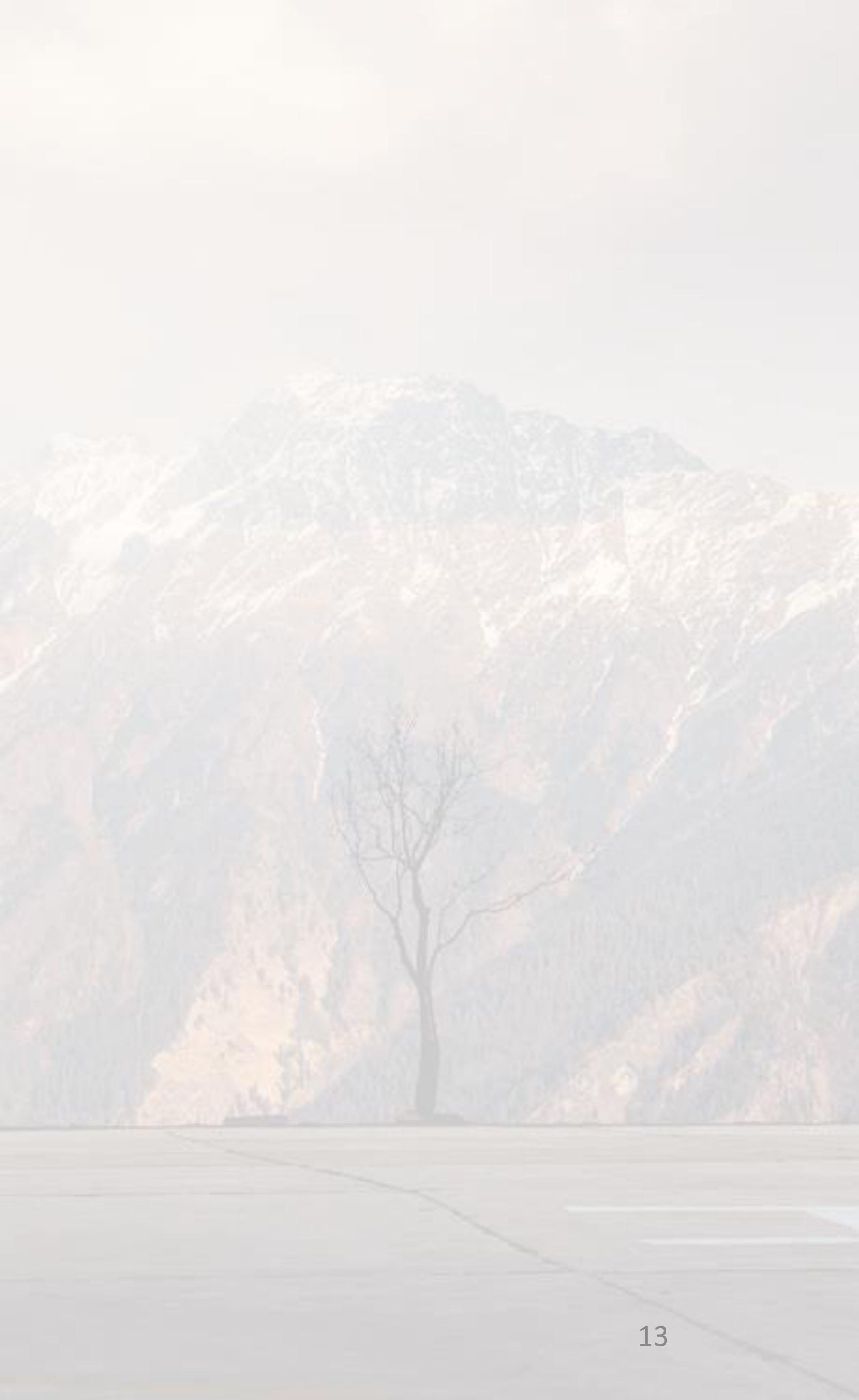
- The GitLab account will be active in 9/15



How we check your code (1/2)

- When the code pushed to GitLab, it should trigger the Jenkins Job automatically

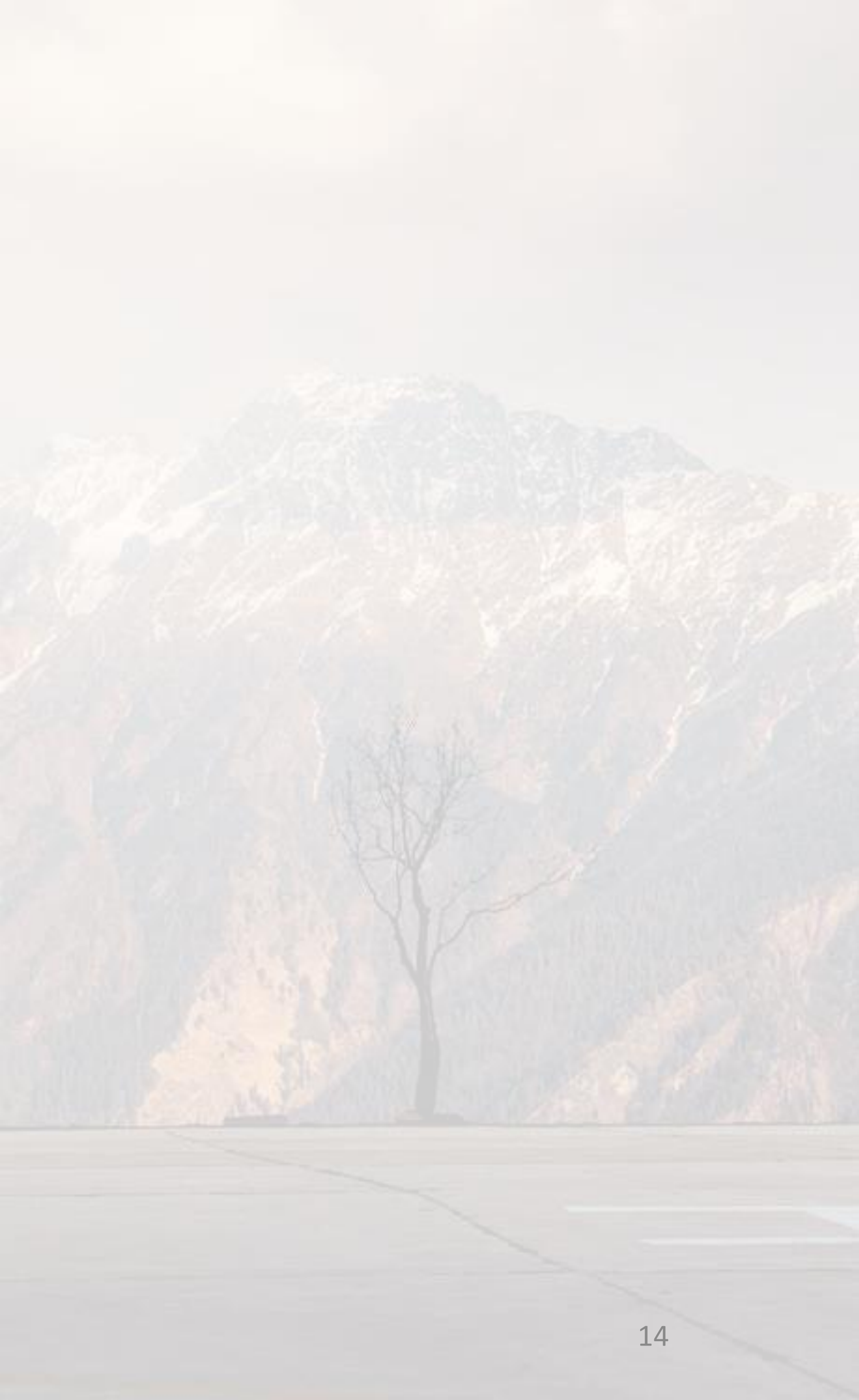
- The Jenkins will execute HW Job
  - HW Job will execute your work without replacement
  - Just test the Job is work smoothly in the test environment
  - Check the test coverage is higher than the margin
  - Check the work is not memory leak



How we check your code (2/2)

- After Jenkins finished to execute HW Job, it will execute TA Job
  - Replace test file and cmake file (i.e. CMakeLists.txt, files.cmake…)
  - Re-execute the work with replacement file
  - Check the work smoothly and doesn’t have memory leak

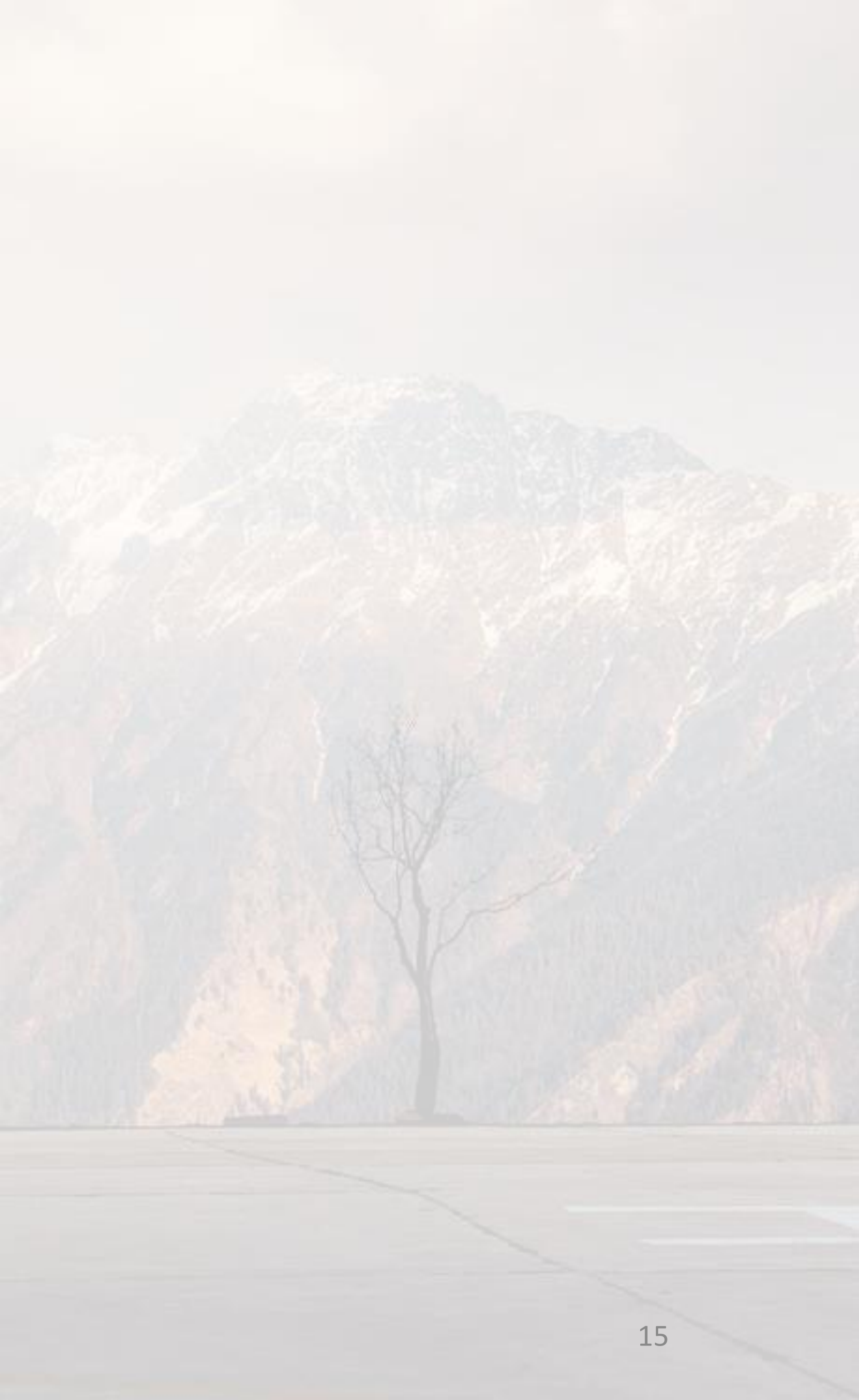
- The Jenkins account will be active in 9/15



After described the goal

- Check you already familiar with C
- Check you don’t be afraid with English

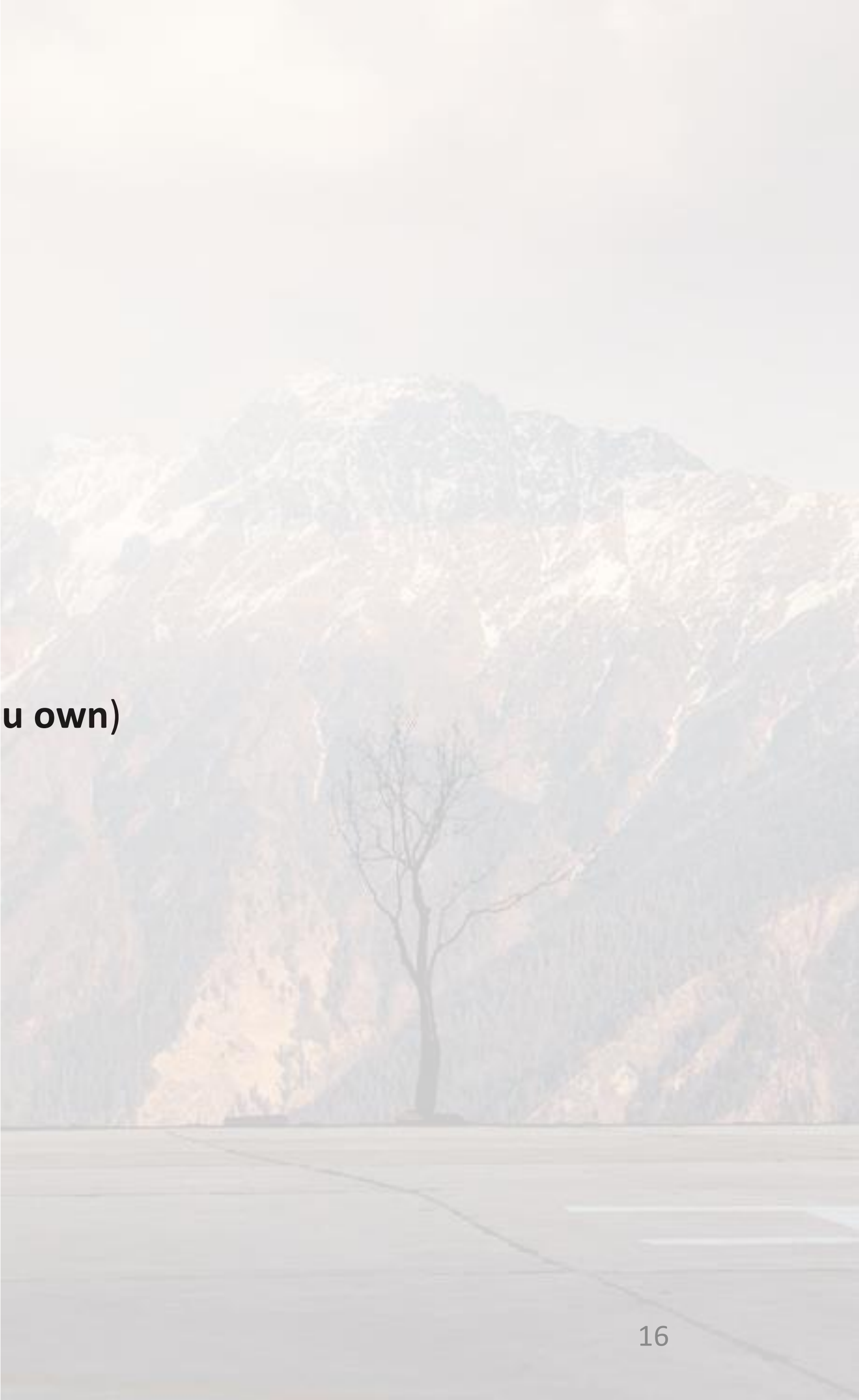
- It will be good to join the course and make your code better and better



TA Time

- Location: Hong-Yue Tech. Research Building 1222
- Time: Monday 18:00 – 20:00
- Start from 9/18

- It’s **not force** to attend TA time, won’t add the score, feel free to seeking help in TA time
  - How to use Git, VSCode, Jenkins, GitLab, etc… (It’s good to familiar these tool **on you own**)
  - How to solve the issue in homework
  - Discuss the concept or confusion of OOP
  - Technique discussion

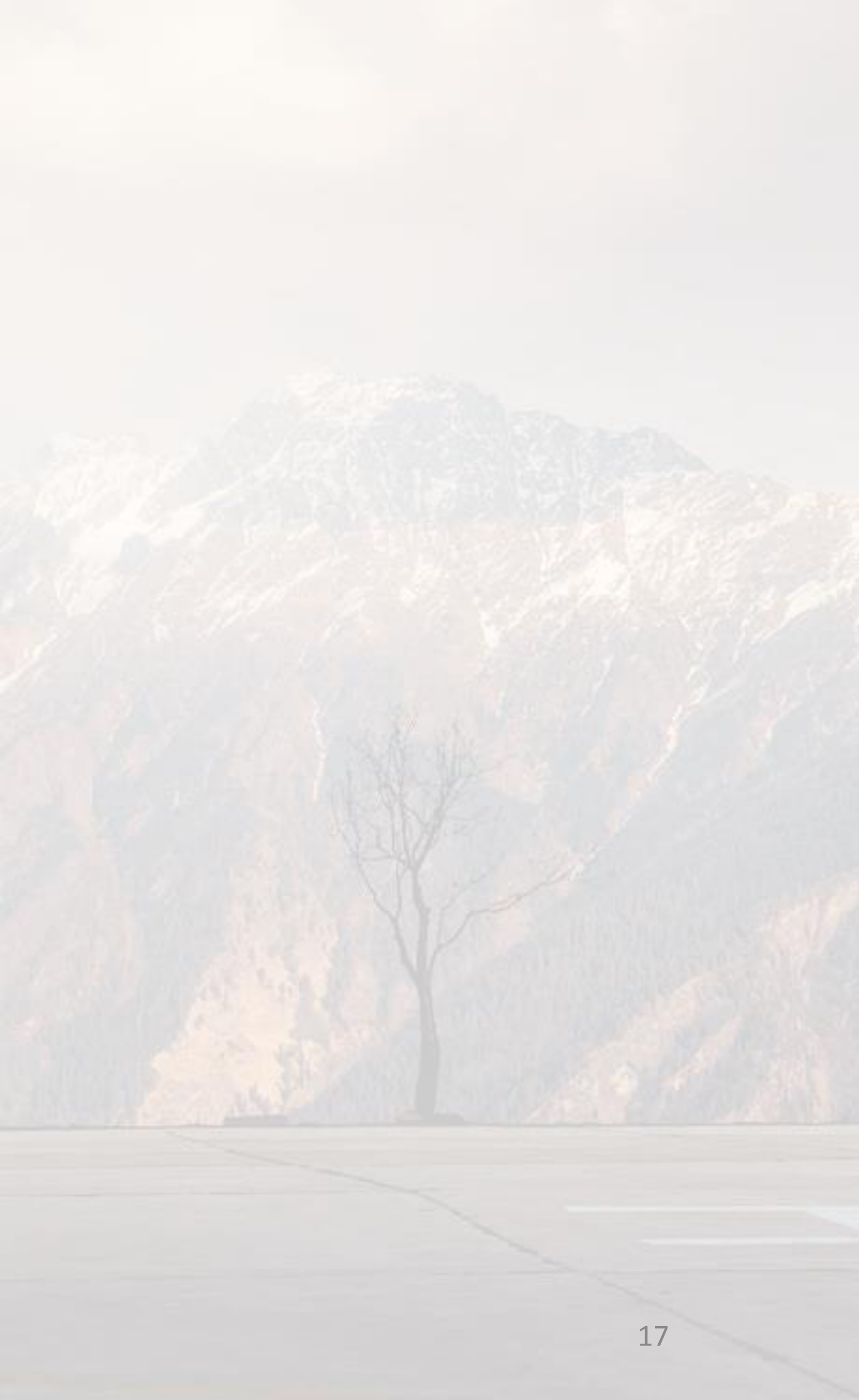


Homework

- Should start from homework 0 to homework 8

- Every homework should contain these information:
  - Due date: The deadline of homework. Late submit is not allow
  - Folder Architecture: The files should be wrap with the specific folder
  - Problem Content: A story
  - Task: Some task you should complete (i.e. solve the problem, write the tests…)
  - Note: Essential tips can help you coding better
  - *Noice Meme*

- *The homework 0 should be published in 9/15, let you guys familiar with the environment*



Thanks!

Course Registration