Introduction

C++ Software Training in CORA (COmputational **R**obotics and **A**utonomy)

Organized by Minchan Jung

---

1. Goal

- Introduce GitHub and how it can be used for collaboration and version control.
- Encourage students to collaborate with each other and to help each other learn.
- Provide students with the opportunity to practice open-source collaboration.
- Foster a sense of community and teamwork among the students.

2. Team Building

- Involving 2-3 members, based on prior team-working experience.
- The team must progress together by teaching and learning from each other.
- It is meaningless to excel alone in this section.
- Focus on expanding and deepening your knowledge.

3. General Rules

1. Formatting & Naming convention.
   A. Standard C++ coding styles recommended
   B. Write class names in UpperCamelCase format.
      i. Files containing class will always be named according to the class name.
      ii. For instance: ClassName.hpp/ClassName.h, ClassName.cpp, or ClassName.tpp.
   C. Definition and Declaration should be separated into different files (.cpp file and .hpp or .h file). (except template class)
   D. Your code should be understandable. Every name must have its own meaning.
   E. The code without comments is a bad code!
2. Git commit message convection
   A. The 'git commit message' should be understandable to others.
3. Allowed & Forbidden
   A. Include safeguards like #pragma once
   B. Your classes must be designed in the Orthodox Canonical Form, except when explicitly stated otherwise.

4. Additional Area

GUI (Qt/QML) & Communication (CAN, D-Bus, SOME/IP)