

# OOP Term Project

25.11.18

## Logistics

Due date: **12/9**

(MUST) Submission and Presentation (by Team leader only)

- **C++ Code (single file) → LMS (no email submission)**

- **Report (in any format) → print it out and bring it on the date of presentation. (no email or LMS submission)**

- **Presentation (in class)**

- **Peer evaluation summary (add a short summary in your submission comment field)**

(Optional) Submission (by everyone)

- **Peer evaluation summary (Each member may submit their own evaluation of teammates)**

## Overview

In this project, students will work in groups to practice both designing and implementing object-oriented systems in C++. Every team will **design and implement a stock trading program. In addition to the based features, every team must propose one unique functionality that distinguishes their designs from others.**

Before we start, we form the project groups as follows:

Team A	B	C	D	E	F
김화완	이호준	홍사인	유성주	황동현	김민준
이혜민	임찬형	정지윤	김희수	박찬서	자말
김재노	김진광	박찬엽	김병하	심은제	조민형
권성미	정승아	신수인	한예준	김정현	임정민
김민석	신바다	신지웅	김나경	김준엽	신승민
오승준	김다인	강수빈	임희성	강동현	주예나

As a first step, every team should select **one member as the team leader**. The team leader

- MUST give the final presentation,
- MUST submit all deliverables,
- MUST evaluate teammates' participations,
- Will receive +2 bonus points for his/her work

The project is divided into two main phases:

### **Phase 1. Design Phase (Specification Writing)**

**Expected Output: One specification document including a class diagram.**

Based on the given topic, each group designs a set of C++ classes and writes a clear specification, including:

- Class names and responsibilities
- Data members (with types and brief descriptions)
- Member functions (signatures and short comments)
- Relationships between classes (inheritance, composition, etc.)
- Any additional notes you think are necessary

\* Think about this: your specification will be given to another team, and they will implement the actual program based on your specification. If your specification is not detailed or clear enough, the other team will justifiably complain, and your grade will be reduced.

#### **Requirements:**

- You must include **one class diagram** that shows the overall architecture.
- You may add brief comments in your specification to help others understand your design.
- You do not need to follow exact UML notation, but try your best to follow the style and conventions we learned in class.
- There is no page limit, but you are strongly encouraged to keep specification short and concise.

### **Phase 2. Implementation Phase (Cross-Group Implementation)**

**Expected Output: C++ Code (Single file)**

The specifications will be exchanged between groups in a cycle as follows:

A → B → C → D → E → F → A (Team A's specification goes to Team B, B's → C, ..., F's to A.)

1) Each group will:

- Implement the classes designed by another group.
- Write a test program (main) that demonstrates the functionality of those classes. This test program will be used later for your in-class demo.

2) During implementation, each group must:

- Follow the given specification as closely as possible; however, you may make necessary corrections or additions if you find clear errors or missing parts.
- Identify any unclear, missing, or unrealistic parts in the design.
- Propose improvements or comments about the original design.

### **Phase 3. Final Deliverables**

At the end, each group will:

#### **Submit**

- 1) One code file: Your implementation based on another group's design (single C++ source file via LMS)
- 2) One report (printed) including both
  - YOUR TEAM's original specification
  - Presentation slides (these should include a summary of the other team's design that your team implemented.)
- 3) Peer evaluation summary
  - (Optional) Each team member may report any teammate who they believe did not participate properly in the project. If a member receives multiple negative reports, he or she may receive a penalty.
  - (MUST) The team leader MUST write a short evaluation of each member's participation in the comment field of the LMS submission window.

#### **Present**

In each team's presentation, the team leader will

- 1) Demo their program,
- 2) Explain, at least:
  - What was easy or difficult to implement from the specification
  - Which design decisions were good or problematic
  - Any other interesting experiences, challenges, or lessons learned

#### **Score guidelines (total 25pts)**

Item	Guidelines	Score
Phase 1. Design	<ul style="list-style-type: none"><li>- How detailed and clear is the specification?</li><li>- Are the relationships between classes well defined and meaningful?</li><li>- Did other teams report any complaints or serious challenges when using your specification?</li></ul>	5pts
Phase 2. Review and Implementation	<ul style="list-style-type: none"><li>- Does the implementation follow the original specification as closely as possible?</li><li>- Does the main program work properly and clearly demonstrate the intent of the class architecture?</li><li>- Do the critiques and review of the original design make sense?</li></ul>	5pts
Report	<ul style="list-style-type: none"><li>- Does the report include all required items?</li><li>- Original specification</li><li>- Review and presentation report (reflection on design, implementation, and cross-group experience)</li></ul>	6pts
Presentation	<ul style="list-style-type: none"><li>- Is the explanation clear and well structured?</li><li>- Is the team demo performed effectively?</li><li>- Does the team describe lessons learned?</li></ul>	7pts
Peer Review	<ul style="list-style-type: none"><li>- Team members can evaluate each other's participation in the project.</li><li>- A member with clearly negative participation may receive 0 point.</li></ul>	2pts
Bonus	<ul style="list-style-type: none"><li>- The team leader will receive +2 points.</li><li>- The whole team members will receive +1 point if the leader gives the presentation in English.</li></ul>	(3pts)
	Total	25pts (+3pts)