

# COURSE SYLLABUS

2020 Fall

PROFESSOR: Eunseok Lee (이은석)

---

OFFICE:	Building of Corporate Collaboration Center , Room 85466/85457
OFFICE HOURS:	4:30-5:30 MON & WED; 1:00-3:00 THR; 4:30-5:30 FRI
OFFICE PHONE:	031-290-7135 (010-4751-8833)
E-MAIL ADDRESS:	leees@skku.edu
WEB PAGE:	<a href="http://selab.skku.ac.kr">http://selab.skku.ac.kr</a>
COURSE PAGE:	<a href="http://www.icampus.ac.kr">http://www.icampus.ac.kr</a> (강의운영/과목자료실)
CLASS HOURS:	

---

## A. DESCRIPTION

This course involves an **overview of modern software engineering techniques**. The techniques correspond with software's life cycle, range from requirement engineering to testing, which cover imperative knowledge for software engineers. The comprehensive knowledge is classified into three groups of issues. Each group covers the following:

- Basic concepts and notions on SE
- Fundamental knowledge on main activities and work products during the process of SE
- Global standards and trends on SE like Agile method, UML, CMMI and SPICE

## B. ORGANIZATION

This course will be conducted in a form of **flipped class** with lecturer's online lecture (called **pre-class**), off-line class(called **in-class**) answering each Q&A sheet for every off-line class, and a couple of presentations for team project by the attendees. For the lecture, one text book and numbers of related books and technical papers will be employed. There are quizzes and comprehensive exams such as midterm and final.

**\*Flipped Class:** It is a switch version of lecture and application.

- A flipped class can be divided into pre-class and in-class.

The pre-class is an online class and students should study the lecture video provided by the lecturer in advance before the off-line class.

The in-class is an off-line class like this, and you guys should fill out the Q & A sheet provided by the lecturer individually or through discussion with team members, using the knowledge learned through the pre-class.

Besides the questions that the lecturer provides, each student has to create two or more questions for each chapter.

Consequently, students have to make answer sheet for the questions that both the lecturer provides and students create.

# If you do not study pre-class in advance, you will not be able to solve the problem during the in-class and you will cause trouble for others.

### C. COURSE OBJECTIVES

1. To introduce students, who have a few backgrounds and experience, to the necessary minimum knowledge on software engineering.(Engineers with Modern SE techniques)
2. To emphasize students ethical and professional responsibilities of software engineers.(Ethical Engineers)
3. To introduce students to currently emerging issue on SE techniques such as UML, agile method, TDD(test-driven development), SPICE and CMMI.(Updated Engineers)
4. To make students experience small scale project with a team to apply SE techniques to be learned. (Interactive & Collaborative Engineers).
5. To raise student to be a competitive engineer with a capability to select the best practice or make the best decision on engineering processes.(Smart Engineers)

### D. COURSE TOPICS

The course will cover the following topics:

1. Introduction to Software Engineering
2. System Engineering
3. Software Processes
4. Agile Software Development
5. Requirement Engineering
6. UML(Unified Modeling Language)
7. System Modeling
8. Architectural Design
9. OMT(Object Modeling Technique)
10. Project Management
11. Project Estimation
12. Design and Implementation
13. Software Testing
14. Software Process Assessment and Improvement(optional)

### E. TEXT AND Related Papers

1. Software Engineering (10th), Ian Sommerville 2016.
2. Software Engineering - a practitioner's approach (8th). R. Pressman. McGrawHill 2015. (*Translated version by Eunseok Lee 2016*)
3. Other SE related papers and reviews

### F. GRADING PLAN

Coursework will be weighted as follows:

1. Team project	35%				
2. Q/A sheet		20%			
3. Midterm exam		20%			
4. Final exam		20%			
5. Attendance				5%	
	35%	+	60%	+	5% = 100%

#### TEAM PROJECT:

A team project will be continued during the semester. The theme of the team project this semester is "Internet Commerce". I ask you to carry out the project using trend technologies, such as AR, VR, video streaming, big data analysis, and machine learning techniques.

I recommend that you first find a site that you can refer to, and then modify and change it with your ideas. That is, you can borrow existing techniques for your projects. All activities and outcomes should be open and controlled through the Github, and finally published as an open source SW. Make your Github ID first and just begin your activity. It is strongly recommended that you use **more than three open source SWs** for your project.

Each team is required to make some documents, such as *proposal*, *requirements specification*, *design specification*, *code*, and so on, according to a guideline. The TP will be basically graded with *originality* (20%), *complexity and scale* (20%), *document completeness* (20%), and *system completeness* (40%). However, the details will be discussed at a separate time in a separate document.

Each team would be organized within five members under ordinary conditions.

#### MIDTERM and FINAL EXAM:

The midterm and final exam will be comprehensive to check overall knowledge on the whole of contents of the lecture.

#### ATTENDANCE:

Attendance will be graded as follows:

No absences	5
One absence	4
Two absences	3
Three absences	2
Four or more absences	F

Absences for which a medical or court excuse is provided (professional letterhead required) will be recorded but not figured in the attendance grade. Likewise, *one* absence for which *advance* notice is given by email or in person will not be figured in the attendance grade. *Any significant tardy or early departure from class will be figured as an absence.*

***Also, anyone who has more than four class-long, unexcused absences will receive an "F" grade for the COURSE.*** This is one of College's rules.

## TENTATIVE SCHEDULE

Weeks	TOPIC/ACTIVITY
Week 1	Orientation / Introduction to software engineering
Week 2	System engineering
Week 3	Software processes
Week 4	<b>Proposal Presentation of Team Project</b>
Week 5	Agile software development
Week 6	Agile software development
Week 7	Requirements engineering
Week 8	UML(Unified Modeling Language)
Week 9	System modeling / OMT(Object Modeling Technique)
Week 10	Architecture design
Week 11	Design and Implementation

Week 12	Project management / Estimation
Week 13	Software testing
Week 14	<b>Final Presentation of Team Project</b>
Week 15	<b>Final Exam</b>