

## Media Software Engineering

Course Name	Course section (credit/hours)		Elective course(4/5)		course code	M017
	course item				course component	
	Target students Division/major/grade				opening semester	2021 1ST SEMESTER
	Class time and classroom		Tue 10:30~12:00 (IUC419)Thu 09:00~10:30 (IUC419)Fri 1(IUC419) Fri 2(IUC419)		English Grade	A(100%English)
Reference to this course	Credit compositon		Theory(0) + Design(0) + Practice(0)			
	Prerequisite courses		Computer Programming, Object–Oriented Programming			
	Related basic courses					
	Recommanded concurrent courses					
Instructor	Name (title/division)		Teemu H. Laine(Associate Professor, Digital Media)			
	Office Room Number	Sanhak Hall 618	Extension Number		e-mail	tlaine@ajou.ac.kr
	Office hour	Tue 1pm – 3pm		Homepage address		
Teaching Assistant	Name (title/division)					
	Office Room Number		Office phone Number		e-mail	

### 1. Course Introduction

This course teaches students how to build robust, flexible and reusable digital media software. Drawing from the best practices and principles of software engineering, students learn how their software architectures can be taken to the next level. To achieve this, we cover essential topics of software engineering, such as software engineering process, agile software development, object-oriented software design (Unified Modeling Language), software design patterns and application programming interfaces. During the second half of the course, students form teams to iteratively develop a media software using the learned processes and techniques. This course is essential for students who have some programming experience but have not yet learned the importance of good software architecture design and dynamic software development process.

### 2. Course Objectives & course outcome

The learning objectives of the course are as follows:

1. Learn about different software engineering process models.
2. Learn and apply an agile software development method (Scrum)
3. Learn to apply well-known software design patterns to make the architecture of a media software highly flexible, robust and reusable.
4. Learn the basic principles about object-oriented software design and how to apply Unified Modeling Language to design robust and extensible software architectures.
5. Improve software development and team work skills through individual assignments and an agile media software development process.

### 3. Class types and activities

The course is based on theory lectures and practical lab sessions during which the students practice software design and implementation by solving various problems. The theory lectures cover the essential topics on media software development, such as software engineering process models, software architecture design, the UML language, software design patterns, network programming, and Application Programming Interfaces.

Students will receive weekly design and programming lab assignments that help them understand the theoretical content and acquire the skills needed to apply the theories in practice. The lab sessions are interactive, whereby professor and/or TAs provide practical guidance to students. Coding and design demonstrations are also provided. Through practical case studies and lab assignments, students also learn how to leverage rich repositories of Application Programming Interfaces found on the Internet to speed up their media software development.

Later in the course, students will work in teams to iteratively design, implement and evaluate a media software with focus on software architecture design. This team project is based on the knowledge and skills that the students acquire during the course. Scrum, an agile software development method, will be used for the team project. During the team project, weekly lecture and lab slots are dedicated to students to work together, to present their weekly progress (Scrum sprints), and to get help from professor and TAs.

### 4. Teaching Method

- |   |   |
|---|---|
| <input checked="" type="checkbox"/> lecture                                     | <input checked="" type="checkbox"/> discussion and debate         |
| <input checked="" type="checkbox"/> team project(presentation and case studies) | <input checked="" type="checkbox"/> experiments(role-playing,etc) |
| <input type="checkbox"/> designing and production                               | <input type="checkbox"/> on-site learning(on-site training)       |
| <input type="checkbox"/> others   |   |

### 5. Support Systems in Use

- |  |   |   |
|--|---|---|
| <input checked="" type="checkbox"/> AjouBb               | <input type="checkbox"/> automatic recording system     | <input type="checkbox"/> web-based assignment |
| <input checked="" type="checkbox"/> cyber lecture        | <input checked="" type="checkbox"/> online content      |   |
| <input type="checkbox"/> class behavior analyzing system | <input checked="" type="checkbox"/> others ( Chatroom ) |   |

### 6. Teaching Tools

- |   |   |  |
|---|---|--|
| <input checked="" type="checkbox"/> PBL(Problem Based Learning) | <input type="checkbox"/> CBL(Case Based Learning) | <input checked="" type="checkbox"/> TBL(Team Based Learning) |
| <input type="checkbox"/> UR(Undergraduate Research)             | <input type="checkbox"/> FL(Flipped Learning)     | <input type="checkbox"/> DSAL(Data Sciencd Active Learning)  |
| <input type="checkbox"/> others                                 |   |  |

## 7. Evaluation method of course outcome

Evaluation Item	The Number of Times	Evaluation Proportion	Remarks
Attendance		5	Attendance
midterm exam		30	Mid-term exam
final exam		15	Final exam
quiz			
presentation			
discussion			
homework		25	Individual assignments
etc		25	Team project
study hours			

## 8. Textbook and Reference material

Main/Sub	Title	Writer	Publisher	Publication year
	To be announced (all needed materials will be in the lecture notes and website references)			

## 9. Class system and Class shedule

<p>The following topics will be covered (tentative):</p> <ul style="list-style-type: none"> <li>● Software engineering process</li> <li>● Object-oriented design</li> <li>● Unified Modeling Language</li> <li>● Software Design Patterns (e.g. Singleton, Observer, Decorator, Factory, Command, Adapter, Facade, State, Iterator, Composite, etc)</li> <li>● Basics of network programming and client-server architecture (using Java, C#)</li> <li>● Application Programming Interfaces (REST)</li> </ul> <p>NOTE: the lecture plan is tentative and may change during the course.</p>							
---	--	--	--	--	--	--	--

### < Schedule >

\* language : K-korean, E-English

Weeks	Title of lecture	language	time distribution(minutes)			Teaching Method	evaluation method
			theory	design	experiment practice		
1	Course introduction	E				Lectures, demonstrations, discussion, practice	

## &lt; Schedule &gt;

\* language : K-korean, E-English

Weeks	Title of lecture	language	time distribution(minutes)			Teaching Method	evaluation method
			theory	design	experiment practice		
2	Software engineering profess models	E				Lectures, demonstrations, discussion, practice	
3	Object-oriented design 1	E				Lectures, demonstrations, discussion, practice	
4	Object-oriented design 2	E				Lectures, demonstrations, discussion, practice	
5	Software design patterns 1	E				Lectures, demonstrations, discussion, practice	
6	Software design patterns 2	E				Lectures, demonstrations, discussion, practice	
7	Software design patterns 3	E				Lectures, demonstrations, discussion, practice	
8	Mid-term exam	E				Exam	
9	Basics of network programming	E				Lectures, demonstrations, discussion, practice	
10	Using RESTful APIs	E				Lectures, demonstrations, discussion, practice	
11	Creating RESTful APIs	E				Lectures, demonstrations, discussion, practice, team project	
12	Team project	E				Team project	
13	Team project	E				Team project	
14	Team project	E				Team project	
15	Team project presentations	E				Team project	
16	Final exam	E				Exam	

10. Contribution index of the course for attaining ABEEK program outcomes

course outcome	contribution scale
No Data	

11. Analysis of improved matters for the previous semester

13. Reference items