

Domain Analysis and Software Design

Course Name	Course type (credit/hours)		Elective course(3/3)		Course code	F067
	Target students Division/major/grade		Software and Computer Engineering/Junior		Opening semester	2020 1ST SEMESTER
	Class time and classroom		Tue F(Pal309)Thu E(Pal309)		English Grade	A(100%English)
Reference to this course	Prerequisite courses		컴퓨터프로그램설계, 객체지향프로그래밍			
	Related basic courses		컴퓨터프로그래밍			
	Recommended concurrent courses					
	Related advanced courses		종합설계프로젝트			
Instructor	Name (title/division)		Kristin Chung(Assistant Professor, Software and Computer Engineering)			
	Office Room Number	팔달관1010호	Office phone Number	1644	e-mail	
	Office hours	Wed. 3:00 ~ 4:30PM		Homepage address		
Teaching Assistant	Name (title/division)					
	Office Room Number		Office phone Number		e-mail	

1. Introduction

Modern software development requires very complicated tasks. A structured approach is needed to deal with these complicated development tasks. Software Development Method is an approach to develop software utilizing the software basic model which is the basic theory, and the software process which is the structured development steps of the development activities.

This course will teach the Object Oriented Analysis and Design Method as the software basic model and the RUP (Rational Unified Process) as the development process. And UML(Unified Modeling Language) will be used as a modeling language.

The detailed contents of the course are listed as below:

1. Analysis

- ① Requirements Analysis using Use Case Modeling
- ② Software Analysis using Domain Modeling
- ③ Documentation methods using UML

2. Design

- ① Object Oriented Design Principles
- ② Object Oriented Design Methods
- ③ Documentation methods using UML
- ④ Design Patterns

3. Development Process

- ① Basic concepts and applications of RUP

4. Modeling Language : UML

- ① Use Case Diagram
- ② Class Diagram
- ③ Sequence Diagram
- ④ Communication Diagram
- ⑤ Activity Diagram
- ⑥ State Chart Diagram
- ⑦ Component Diagram
- ⑧ Deployment Diagram

2. Course Objectives

1. Teaching Objectives (교육목표)

- 1) Students should analyze, design and implement the software engineering problem by themselves.
- 2) Students should be able to apply their programming skill to the industry problems.
- 3) Students should produce a reliable software through verification.
- 4) Students should learn how to use the existing and the open-source code and apply to their problem solving.

2. Class achievements (교과목 학습성과)

- 1) Students understand and apply the object-oriented development process (학습성과 3, 5)
- 2) Students perform software system analysis and design methods using UML which is a Modeling language and apply this knowledge to the class projects. (학습성과 3, 4)
- 3) Students develop a team-proposed project considering quality factors like safety, external constraints, and economic factors. (학습성과 8)
- 5) Students effectively communicate the technical contents to others through documentation and review of the software artifacts (학습성과 6)

3. Class types and activities

- Lecture on the basic analysis and design methods, and evaluate presentations on the design projects
- Students need to apply the Object-Oriented analysis and design knowledge on the design team project using UP process and UML

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 type="checkbox"/> experiments(role-playing,etc) |
| <input checked="" 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 type="checkbox"/> cyber lecture | <input type="checkbox"/> online content | |
| <input type="checkbox"/> class behavior analyzing system | <input type="checkbox"/> others | |

6. Teaching Tools

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

7. Knowledge and ability required for taking this course

Should be able to use one of the object oriented programming language (C++, Java, C#, Python).
Should be able to use one of the UML diagram drawing tool like RSA. (will provide the introductory RSA tool training)

8. Method of Evaluation

Evaluation Item	The Number of Times	Evaluation Proportion	Remarks
Attendance		5%	
midterm exam		20%	exam
final exam		20%	exam
quiz		10%	
presentation		5%	presentation evaluation
discussion			
homework		35%	Design project evaluation
etc		5%	TOPCIT 결과
study hours	12 hours		

9. Textbook and supplementary material

Main/Sub	Title (Web-site)	Writer	Publisher	Publication year
Main	ApplyingUMLAndPatterns,3rdEd.	C.Larman	PrenticeHall	2005
Ref.	DesignPatterns:ElementsofReusableObject-OrientedSoftware	Gamma외3인	Addison-Wesley	1995
Ref.	The Unified Modeling Language User Guide, 2ndEd.	Booch, Rumbaugh, Jacobson	Addison-Wesley	2005

10. Class system and Class shedule

Detailed contents of this course are:

1. Object Oriented Analysis
 - Requirements analysis
 - Domain analysis methods
 - Documentation methods using UML
2. Object Oriented Design
 - Principles of Object Oriented Design
 - Object Oriented Design methods
 - Documentation methods using UML
3. Design Patterns
 - Creational Patterns, Structural Patterns, Behavioral Patterns
4. Modeling language : UML

The above four concepts are proceeded in parallel:

Preliminary Requirements analysis → Preliminary Requirements modeling → Preliminary Analysis → Analysis modeling (UML) → Preliminary design patterns → Advanced Requirements analysis → Advanced Requirements modeling → Advanced Analysis → Advanced Analysis modeling → Advanced Design → Advanced Design modeling (UML) → Advanced Design patterns

Team Project development:

Wk2: Team Forming

Wk3: UML tool RSA training

Wk4: Project Proposal / Inception Report & Presentation

Wk9: Elaboration 1 (E1) Report due

Wk10: E1 Presentation

Wk14: Final (E2) Report due

Wk15: Final Project Presentation

< Class Schedule >

* language : K-korean, E-English

Weeks	Topics	language	Instructor	Teaching Method	Evaluation Method	Matter to be prepared
1	Introduction to Object Oriented Analysis & Design		Kristin Chung	강의		
2	Requirements analysis & Use Case modeling		Kristin Chung	강의		
3	Domain modeling		Kristin Chung	강의		
4	Applying Requirements analysis – System Sequence Diagram		Kristin Chung	강의, 발표		
5	Applying Requirements analysis – Operation Contracts		Kristin Chung	강의		
6	Requirements to Design – UML & package		Kristin Chung	강의		
7	Design principles and patterns – OOD & GRASP		Kristin Chung	강의		
8	Midterm exam		Kristin Chung	시험		

< Class Schedule >

* language : K-korean, E-English

Week s	Topics	lang uag e	Instructor	Teaching Method	Evaluation Method	Matter to be prepared
9	Use Case Realization – applying GRASP patterns		Kristin Chung	강의, 발표		
10	Project presentation 1		Kristin Chung	강의, 발표		
11	More GRASP design patterns		Kristin Chung	강의		
12	Introduction to Design patterns – Gof		Kristin Chung	강의		
13	Software Architecture		Kristin Chung	강의		
14	Applying Design patterns – I		Kristin Chung	강의		
15	Final Project presentation		Kristin Chung	발표		
16	Final exam		Kristin Chung	시험, 발표		

11. Other items of notification