

1. Course Code

2207

2. Course Title

Fundamentals of Software Engineering

3. Teacher

ITO, Mamoru

4. Term

Fall 2

5. Course Requirements (Courses / Knowledge prerequisite for this course)

None

6. Course Overview and Objectives

Students learn basic knowledge of the development processes and techniques (analysis, design, and verification) necessary for developing practical software systems and products, as well as software project management techniques. Through this course, students will understand the characteristics of software and the social environment surrounding software and will be able to develop their decision-making and behavioral skills to deal with various problems they will face in software projects.

7. Course Outline

- 1 Principles of computer operation
- 2 Fundamentals of data structures and algorithms
- 3 Software engineering overview and ethics
- 4 Basics of the software development processes
- 5 Fundamentals of the project management processes
- 6 Basics of requirements analysis
- 7 Software design basics
- 8 Software testing basics
- 9 Software quality fundamentals
- 10 Object oriented concepts
- 11 Basics of modeling using Unified Modeling Language (UML)
- 12 Basics of modeling using Unified Modeling Language (UML)
- 13 Exercises in modeling with UML
- 14 Exercises in modeling with UML
- 15 Presentation of UML modeling exercise
- 16 Term-end exam (multiple-choice and open-book style)

8. Textbooks (Required Books for this course)

None

9. Reference Books (optional books for further study)

R. S. Pressman, B. R. Maxim. Software Engineering: A Practitioner's Approach. McGraw Hill Higher Education.

IEEE Computer Society. Guide to the Software Engineering Body of Knowledge. IEEE Computer Society Press.

10. Course Goals (Attainment Targets)

- (1) Have a basic understanding of software development life cycle and process models
- (2) Utilize basic techniques in software analysis and design
- (3) Acquire practical decision-making skills required for software project management
- (4) Deepen an understanding of social environments surrounding software development
- (5) Analyze the ethical issues in software development
- (6)
- (7)
- (8)

11. Correspondence relationship between Educational goals and Course goals

Educational goals of the school		Course Goals	
High level ICT skills	Basic academic skills	(1), (2)	
	Specialized knowledge and literacy	(1), (2)	
Human skill (Tankyu skill)	Ability to continually improve own strengths	(2)	
	Ability to discover and resolve the problem in society	Problem setting	(3), (4)
		Hypothesis planning	(3), (4)
		Hypothesis testing	
	Fundamental Competencies for Working Persons	Practice	
		Ability to step forward	(3)
Ability to think through		(3), (4)	
	Ability to work in a team	(1)	
Professional ethics		(3), (4), (5)	

12. Evaluation

Goals	Evaluation method & point allocation					
	examination	Quiz	Reports	Presentation	Deliverables	Other
(1)	○	○		○	○	
(2)	○	○		○	○	
(3)	○	○	○	○		
(4)	○		○	○		
(5)	○		○	○		
(6)						
Allocation	30	30	15	10	15	

13. Evaluation Criteria

Examination	Multiple-choice exam is used to assess students' understanding and application of the course goals. This is an open-book exam that allows students to bring in their own materials and does not require knowledge of the subject matter.
Quiz	Multiple-choice quizzes are used to evaluate the students' comprehension and application of the content in each class. As this is an open-book test, no knowledge is required.

Reports	Evaluate in terms of the appropriateness of the description, the structure of the report, the validity of the argument, and the originality of the proposal.
Presentation	Evaluate the content of the presentation in terms of its purposefulness, the structure of the presentation, the relevance of the argument, and the originality of the proposal.
Deliverables	Evaluate the diagrams and documents created in terms of relevance, clarity, and objectivity.
Other	

14. Active Learning

Hourly percentage of active learning within the whole class time	40%
--	-----

1	Active learning such as problem solving assignment using the knowledge and skills acquired in class.	All the time
2	Active learning such as group works and discussions.	All the time
3	Outcome presentations and feedbacks.	Sometimes
4	Students actively make decisions on how the class should be conducted.	Not at all

15. Notes

This course provides the course materials on Moodle.

16. Course plan

(Notice) This plan is tentative and might be changed at the time of delivery

Lesson 1: Basic principles

Lecture/Discussion 90 min

The Software runs on a computer. The basic knowledge on computers expands an understanding of software development. We will learn how a computer works in this lesson.

- Introduction
- Computer organization (CPU, memory, I / O, clock)
- Von Neumann architecture
- Memory hierarchy
- Program performance

Lesson 2: Data structures and algorithms

Lecture/Discussion 90 min

The data structures and algorithms should be considered for us to design a computer program. This lesson will provide students with the introduction of data structures and algorithms.

- Address space and virtual memory
- Major data structures – array, list, stack, queue, and tree
- Major algorithms - sorting algorithms and search algorithms
- Computational complexity

Lesson 3: Software engineering and ethics

Lecture/Discussion 90 min

The software grows increasingly important along with the popularization of computers. We will discuss the reality surrounding software development after understanding of the features of the software and learn the necessity of software engineering.

- Features of software
- Importance of software
- Environment surrounding software development
- Role of software engineering and ethics

Lesson 4: Software development processes

Lecture/Discussion 90 min

A “Process” can be defined as a “set of interrelated or interacting activities, which transforms inputs into outputs”. Good process is required to produce good outputs. We will learn the overview of software life cycle process models and the meaning of process improvement.

- Definition of software process
- Life cycle models
- Present situation and issues on software process
- Meaning of software improvement

Lesson 5: Project management processes

Lecture/Discussion 90 min

Generally, software is developed by a project team. The project team should be managed adequately. This lesson will clarify a project, project management, project lifecycle, and project organizations.

- Definition of project and project management
 - Project life cycle
 - Relationship with organizations and stakeholders
 - Trend in project management standards
-

Lesson 6: Requirements analysis

Lecture/Discussion 90 min

The role of software engineer is to realize the requirements of customers and users by use of software. But their requirements are sometimes ambiguous and lack consistency. We should acquire their requirements exhaustively and analyze them systematically. We will marshal the concepts of requirements and flow of requirement analysis.

- Difference between needs wants and demands
- Functional requirements and non-functional requirements
- Requirements analysis techniques
- Requirements modeling

Lesson 7: Software design

Lecture/Discussion 90 min

The optimum design technique should be selected based on the target and objectives of software development. This lesson introduces major software design techniques such as structured design and object-oriented design.

- Architectural design
- Structured design
- Object-oriented design

Lesson 8: Software testing

Lecture/Discussion 90 min

Software testing is becoming important because defects in software have the significant impact on the society. We will learn the positioning of software testing, kinds of software testing, and testing techniques in this lesson.

- Necessity and limitation of software testing
- Flow of software development and testing phases
- White box test and black box test
- The major testing techniques

Lesson 9: Software quality

Lecture/Discussion 90 min

One of the objectives of software engineering is to develop high-quality software. Management on software quality is more important than that of hardware quality because software is invisible. We will understand the whole picture of software quality and necessary activities to achieve the required quality in this lesson.

- Difference between quality and grade
 - Software quality model
 - Quantitative quality management
 - Software design review
-

Lesson 10: Object-oriented methodology**Lecture/Discussion 90 min**

Object-oriented methodology is becoming popular in association with increasing in size and complication of software. This methodology is used not only for programming but also for requirements analysis and software design. This lesson will focus on object-oriented analysis and design by the use of UML modeling.

- A brief history of object-oriented methodology
- Object-oriented analysis
- Modeling and UML diagrams
- Object-oriented design

Lesson 11-12: Unified modeling language (UML)**Lecture/Exercise 180 min**

UML is becoming commonly-used with object-oriented technology. UML stands for Unified Modeling Language, which is a useful tool for analysis and design of complex software systems. We will learn how to describe major diagrams.

- Overview
- Use Case Diagram
- Activity Diagram
- Class Diagram, Object Diagram
- Sequence Diagram

Lesson 13-14: Exercises in UML modeling**Exercise: 180 min**

Software analysis and design includes various activities from requirements analysis to implementation, which holds extremely important position in software developments. In the following three lessons, exercises in the analysis and design of software systems are conducted through group work. After the exercises, each group of the students makes a presentation on the results of group work.

- Exercises in structural and behavioral modeling
- Exercises in analysis and design of software
- Exercises in drawing UML diagrams
- Presentations

Lesson 15: Presentation**Presentation 90 min**

Each group conducts a presentation of the results of UML modeling.

- Presentation

Term-end Examination**Examination: 90 min**

A multiple-choice exam is conducted to evaluate the level of understanding of each student. Your answers to these questions will all be processed by computer.

- Multiple-choice exam
-