

1. Course Code

2207

2. Course Title

F50e: Introduction to Computer Software

3. Teacher

ITO, Mamoru

4. Term

Fall 2

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

None

6. Course Overview and Objectives

You can learn basic knowledge of the processes and techniques necessary for developing and using software systems. This course will also discuss the latest technologies that promote digital transformation in our society. Through this course, you 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 you will face in software projects.

7. Course Outline

- 1 Basic computer principles
- 2 Fundamentals of data structures and algorithms
- 3 Digital Transformation - Cloud and web technologies
- 4 Digital Transformation - Machine learning and blockchain technologies
- 5 Software engineering and ethics
- 6 Basics of software development processes
- 7 Basics of requirements analysis
- 8 Basics of software design
- 9 Basic concepts of testing and quality
- 10 Object-Oriented paradigm
- 11 Introduction to UML - Behavior diagrams
- 12 Introduction to UML - Structure diagrams
- 13 UML modeling exercise
- 14 UML modeling exercise
- 15 Presentation of modeling exercise
- 16 Term-end examination (multiple-choice and open-book format)

8. Textbooks (Required Books for this course)

None

9. Reference Books (optional books for further study)

Bourque, Pierre, et al. Guide to the Software Engineering Body of Knowledge. 2014.

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	
		Practice	
	Fundamental Competencies for Working Persons	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)						
(7)						
(8)						
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 computer principles	Lecture/Discussion 90 min
-------------------------------------	---------------------------

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

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

Lesson 2: Fundamentals of 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.

- Number system
- Address space
- Major data structures – array, list, stack, queue, and tree
- Algorithms and computational complexity

Lesson 3: Digital Transformation - Cloud and web technologies	Lecture/Discussion 90 min
---	------------------------------

A variety of digital technologies are used to drive digital transformation (DX). This lesson provides an overview of DX, followed by the basics and latest trends in cloud computing and web technologies.

- What is Digital Transformation (DX)?
- Strategy first
- Cloud computing
- Web technologies

Lesson 4: Digital Transformation - Machine learning and blockchain technologies	Lecture/Discussion 90 min
---	------------------------------

This lesson focuses on machine learning and data science among the various digital technologies used to drive digital transformation (DX). We will discuss the expanding possibilities and challenges of these technologies.

- Basics of Machine learning and data science
- AI and society
- Blockchain technologies
- web3

Lesson 5: 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 6: Basics of 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.” A 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 7: Basics of requirements analysis**Lecture/Discussion 90 min**

The role of software engineers is to realize the requirements of customers and users by using the 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 8: Basics of software design**Lecture/Discussion 90 min**

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

- Design viewpoints
- Data Flow Diagram (DFD)
- State Transition Diagram (STD)
- Entity Relationship Diagram (ERD)

Lesson 9: Basic concepts of testing and quality**Lecture/Discussion 90 min**

Software testing and quality management are becoming increasingly important because defects in software have a significant impact on society. We will learn the basic concept of software testing and quality management in this lesson.

- What is software testing?
- Major testing techniques
- Quality and grade
- Software quality model
- Quantitative quality management

Lesson 10: Object-Oriented paradigm**Lecture/Discussion 90 min**

The object-oriented methodology is becoming popular in association with increases 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 model
 - Object-oriented principles
 - Object-oriented analysis
-

Lesson 11-12: Introduction to UML**Lecture/Exercise 180 min**

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

- What is UML?
- Behavior diagrams
- Structure diagrams

Lesson 13-14: UML modeling exercise**Exercise: 180 min**

UML is used to model businesses and processes and analyze, design, and implement software systems. In this lesson, we will focus on various businesses, and through group work, we will model each business using UML.

Lesson 15: Presentation**Presentation 90 min**

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

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.
