

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 for this course) and Important Information

None

6. Course Overview and Objectives

You will learn the fundamentals of the processes and techniques required to develop and deploy software systems. This course also discusses the latest technologies that are driving the digital transformation of our society. Through this course, you will understand the characteristics of software and the social environment surrounding software, and you will be able to develop your decision-making and behavioral skills to deal with various problems you will face in software projects.

7. Course Outline

- 1 Basic computer principles
- 2 Data structures and algorithms
- 3 Digital Transformation - Cloud computing and web design
- 4 Digital Transformation - Machine learning and blockchain
- 5 Software engineering and ethics
- 6 Software development processes
- 7 Software requirements
- 8 Software design
- 9 Software testing and quality
- 10 Object-Oriented concepts
- 11 Introduction to UML - Behavior diagrams
- 12 Introduction to UML - Structure diagrams
- 13 UML modeling exercise
- 14 UML modeling exercise
- 15 Modeling exercise presentation
- 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	
	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)						
(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
-------------------------------------	------------------------------

Software runs on a computer. A basic knowledge of computers will improve your understanding of software development. In this lesson, we will learn how a computer works.

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

Lesson 2: Data structures and algorithms	Lecture/Discussion 90 min
--	------------------------------

Data structures and algorithms should be considered when designing a computer program. This lesson introduces students to 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 computing and web design	Lecture/Discussion 90 min
---	------------------------------

A variety of digital technologies are being 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	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 growing opportunities 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
---	------------------------------

With the popularization of computers, software is becoming increasingly important. We will discuss the reality surrounding software development after understanding the characteristics of software and learning the necessity of software engineering.

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

Lesson 6: Software development processes	Lecture/Discussion 90 min
--	------------------------------

A "process" can be defined as "a set of related or interacting activities that transform inputs into outputs. A good process is needed to produce good outputs. We will get an overview of software life cycle process models and the importance of process improvement.

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

Lesson 7: Software requirements	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: Software design**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 inconsistent. We should collect their requirements thoroughly and analyze them systematically. We will review the concepts of requirements and the flow of requirements analysis.

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

Lesson 9: Software testing and quality**Lecture/Discussion
90 min**

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

- 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 gaining popularity in connection with the increasing size and complexity of software. This methodology is used not only for programming, but also for requirements analysis and software design. This lesson focuses on object-oriented analysis and design using 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 widely 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 important 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 to analyze, design, and implement software systems. In this lesson, we will focus on different 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 will be administered to assess each student's level of understanding. Your answers to these questions will all be scored by computer.
