

2026 年度シラバス

科目分類/Subject Categories			
学部等/Faculty	/工芸科学部 : /School of Science and Technology	今年度開講/Availability	/有 : /Available
学域等/Field	/設計工学域 : /Academic Field of Engineering Design	年次/Year	/2年次 : /2nd Year
課程等/Program	/情報工学課程・課程専門科目 : /Specialized Subjects for Undergraduate Program of Information Science	学期/Semester	/前学期 : /First term
分類/Category	/:/	曜日時限/Day & Period	/金 3-4 : /Fri.3-4

科目情報/Course Information				
時間割番号 /Timetable Number	12215201			
科目番号 /Course Number	12260080			
単位数/Credits	2			
授業形態 /Course Type	講義 : Lecture			
クラス/Class				
授業科目名 /Course Title	ソフトウェア工学 : Software Engineering			
担当教員名 / Instructor(s)	/水野 修/崔 恩静 : MIZUNO Osamu/Choi Eunjong/			
その他/Other	インターンシップ実施科目 Internship	国際科学技術コース提供科目 IGP	PBL 実施科目 Project Based Learning	DX 活用科目 ICT Usage in Learning
				○
	実務経験のある教員による科目 Practical Teacher	○	ソフトウェア開発実務経験のある教員・外部講師による講義を行う。	
科目ナンバリング /Numbering Code				

授業の目的・概要 Objectives and Outline of the Course	
日	<p>情報化社会の発展と共に社会の隅々にまでコンピュータが行き渡り、それを制御するソフトウェアの開発が必須となっている。ソフトウェア工学の分野は平均的な開発者がいかに品質の高いソフトウェアを作成できるか、を目的として 40 年に渡って発展を続けた。しかし、今日でもソフトウェアの開発は本質的に難しい状況は変わっていない。本講義ではソフトウェア開発の設計、コーディング、テストといった各段階について詳説し、これからのソフトウェア開発に必要な知識を身につける。また、適宜最新の研究動向についても解説する。</p> <p>担当教員の一部は</p>
英	<p>Nowadays, computers spread into society and thus the role of software to control the computers becomes important. Software Engineering aims to establish the technologies for developing the software easier and have been developed for 40 years. However, the software crisis has not been solved for 40 years, either. In this lecture, the essence of software engineering will be discussed and we will explore the state of art technologies as well as traditional software engineering disciplines. A part of professors have practical experience in software development, and will have lectures in a particular area in the software engineering based on their experience.</p>

学習の到達目標 Learning Objectives	
日	<p>ソフトウェア工学の歴史とその果たすべき役割についての理解を深める。</p> <p>ソフトウェア開発プロセスについて理解を深める。</p> <p>要求分析工程について理解を深める。</p> <p>設計工程についての理解を深める。</p> <p>設計工程において利用される技法について理解する。</p>

	<p>コーディング工程についての理解を深める。 テスト工程の必要性について理解を深める。 テストケース設計技法について理解する。 テストの妥当性評価や有効性の評価について理解する。 オブジェクト指向設計技法について理解を深める。 オブジェクト指向設計を実際に行い、構造化設計との違いを確認する。 ソフトウェアの再利用についての理解を深める。 ソフトウェア開発プロジェクトの管理について理解する。 ソフトウェア開発の規模・工数推定手法について学習する。 本講義において学習・実践してきたソフトウェア工学の技法について総括を行う。</p>
英	<p>Understand the history and the role of software engineering. Understand the software development process. Understand the requirements analysis phase. Understand the software design phase. Understand the techniques in the software design. Find the difficulties in software coding Understand the necessity of software testing. Find the difficulties in software testing phase. Able to apply the testing methodologies in practice. Understand the notion of object oriented design. Practice the object oriented design. Understand the reuse of software. Understand the software process management. Understand the software project management and effort estimation. Conclusion.</p>

学習目標の達成度の評価基準 / Fulfillment of Course Goals (JABEE 関連科目のみ)

日	
英	

授業計画項目 Course Plan

No.		項目 Topics	内容 Content
1	日	ソフトウェア工学の歴史とその果たすべき役割	ソフトウェア工学が認知されたきっかけとなるソフトウェア危機から始まり、ソフトウェア工学という分野が何を目指しているのかについて述べる。
	英	The history and the role of software engineering	Lectures on the 40 years history of software engineering and the role of the software engineering.
2	日	ソフトウェア開発プロセス	ソフトウェア開発を「プロセス」と見なす考え方を紹介し、その各工程についての概要を述べる。
	英	Software development process	An introduction of software development process.
3	日	要求分析工程	顧客の要求を仕様化する要求分析工程について、そこで使われる技法の解説などを行う。
	英	Requirements analysis	An introduction of requirements analysis phase.
4	日	設計工程(1):設計工程の概論	ソフトウェアの骨組みを作成するとも言える設計工程について解説する。
	英	Software design (1)	An introduction of software design phase.
5	日	設計工程(2):モジュール分割とその評価	ソフトウェア設計で重要な役割を果たすモジュール分割について、その技法と評価手法について述べる。
	英	Software design (2)	Software design (2)
6	日	コーディング工程	実際にプログラムを作成するコーディング工程について解説する。
	英	Coding	An introduction of coding phase.
7	日	テスト工程(1): テスト工程の役割	ソフトウェア開発において最も重要な工程とも言えるテスト工程について、その意義を解説する。
	英	Software testing (1)	An introduction of software testing phase.

8	日	テスト工程(2): テストケース設計技法	テストにおいて利用するテストケースの生成手法について述べる.
	英	Software testing (2)	An introduction of methods in testing phase.
9	日	テスト工程(3): テストの妥当性評価	テスト工程が妥当に行われたかを判断する基準について述べる.
	英	Software testing (3)	How to evaluate the validity and effectiveness of software testing?
10	日	オブジェクト指向設計(1): 概論	オブジェクト指向設計技法について, その中心となる考え方について解説する.
	英	Object oriented design (1)	An introduction of the object oriented design.
11	日	オブジェクト指向設計(2): UML	一般的なオブジェクト指向設計記述言語, UML, について解説する.
	英	Object oriented design (2)	An application of the object oriented design into practice.
12	日	オブジェクト指向設計(3): 再利用の促進	ソフトウェアの再利用を促進する方法について述べ, その利点, および, 弊害について解説する.
	英	Object oriented design (3)	An introduction of software reuse.
13	日	ソフトウェア開発プロジェクト管理	ソフトウェア開発プロセス全体を管理する手法について概要を述べる.
	英	Software project management	An introduction of software project management.
14	日	ソフトウェア開発規模と工数見積	ソフトウェア開発を管理するために必要となる規模や工数の見積技法について, 解説する.
	英	Effort estimation	An introduction of effort estimation in software development.
15	日	ソフトウェア工学の総括	ソフトウェア工学という分野の現状と, これからの展開について考え, 本講義の総括を行う.
	英	Conclusions	Conclusion of the entire lecture.

履修条件 Prerequisite(s)

日	ソフトウェア開発に関する基本的な技術 (プログラミング経験等)・知識が推奨される.
英	A fundamental skill for software development such as programming is recommended.

授業時間外学習 (予習・復習等)

Required study time, Preparation and review

日	・毎回の授業に対し, 2時間の予習・復習を要する.
英	* Needs 2 hours preparations and reviews.

教科書/参考書 Textbooks/Reference Books

日	Moodle にて講義資料を配付する.
英	Texts will be delivered in Moodle.

成績評価の方法及び基準 Grading Policy

日	講義中の小テスト, 及び, 数回のレポート(または期末試験)を総合して判断する. 小テストの評価を 50%, レポートの評価を 50%とし, その合計点が 60 点以上を合格とする.
英	Small assignments and several reports (or exams) are evaluated as scores of 100 points. Small assignments are 50 points and reports are 50 points.

留意事項等 Point to consider

日	・各講義回でミニレポートを出題し, その回答をもって出席と見なす. ・レポートの作成にあたっては, 適切な引用等を心がけること. また, 剽窃・盗用行為は発見次第厳正に対処する.
英	* In each lecture, we have small assignments and will be evaluated. * Be careful about quotation of other papers and books. * Do not plagiarize programs and reports.

