

## 1. 科目コード

1207

## 2. 科目名

F50: コンピュータ・ソフトウェア入門

## 3. 担当教員

伊藤 守 (Mamoru Ito)

## 4. 開講期

春1期

## 5. 履修要件(前提科目)・重要情報

特になし

## 6. 科目の目的・概要

コンピュータ科学の基礎から, AIやクラウド等の最新技術, ソフトウェア開発のライフサイクルに至る広範な知識を習得し, 専門家としての倫理観を養います. さらに, UMLを活用した分析・設計演習とプレゼンテーションを通じて, 実務における複雑な課題をモデル化し, 解決に導くための高度な専門能力を育成します.

## 7. 授業概要

- 1 コンピュータの動作原理
- 2 データ型とデータ構造
- 3 クラウドコンピューティング/Web技術の基礎
- 4 AI・機械学習の基礎
- 5 ソフトウェア工学の概要と倫理
- 6 ソフトウェア開発プロセス
- 7 ソフトウェア要求
- 8 ソフトウェア設計
- 9 ソフトウェアテスト/ソフトウェア品質
- 10 オブジェクト指向技術
- 11 UMLの基礎－振る舞い図
- 12 UMLの基礎－構造図
- 13 AI駆動型ソフトウェア開発演習1
- 14 AI駆動型ソフトウェア開発演習2
- 15 プレゼンテーション
- 16 期末試験(多肢選択式・オープンブック方式)

## 8. 教科書

なし

## 9. 参考書

H. Washizaki, eds., Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), Version 4.0, IEEE Computer Society, 2024; [www.swebok.org](http://www.swebok.org). (ダウンロード可)

## 10. 科目の学習目標

- (1) コンピュータシステムの基礎を理解し, システム全体の仕組みを説明できる
- (2) ソフトウェアライフサイクルの基本を理解し, 目的に応じた設計手法を選択できる
- (3) 開発プロセスの基本を理解し, 適切なマネジメント手法を活用できる
- (4) UMLの基本記法を習得し, ビジネスやシステムの構造や動作をモデル化できる
- (5) 最新の技術トレンドを把握し, 社会に与えるインパクトを技術的側面から分析できる
- (6) ソフトウェアの開発・運用において直面する倫理的課題に対して適切な判断を下すことができる
- (7)
- (8)

## 11. 本学の教育目標と科目の学習目標との対応

教育目標		学習目標	
高度ICT スキルの修得	基礎的素養	(1) (2)	
	専門知識および業務応用力	(3) (4)	
人間力 (=探究力) の修得	自ら強みを磨き続ける力	(5)	
	自ら社会における 課題を発見し、 解決する力	課題設定	(3) (4) (5)
		仮説立案	(3) (4) (5)
		仮説検証	
		実行	
	社会人基礎力	前に踏出す力	(3) (4)
		考え抜く力	(3) (4)
チームで働く力		(3) (6)	
職業倫理の修得		(5) (6)	

## 12. 評価方法と配点

学習目標	達成度評価方法と配点					
	期末試験	小テスト	レポート	発表	成果物	振り返り
(1)	○	○		○	○	○
(2)	○	○		○	○	○
(3)	○	○	○	○		○
(4)	○		○	○		○
(5)	○		○	○		○
(6)						
(7)						
(8)						
配点	30	25	10	10	20	5

## 13. 評価基準

期末試験	多肢選択式試験により、学習目標に掲げた基礎的事項に関する理解度と応用力を評価する。教材持ち込み可能なオープンブック方式の試験のため、知識は問わない。
小テスト	多肢選択式テストにより、各授業で実施した内容に関する理解度と応用力を評価する。オープンブック方式のテストのため、知識は問わない。
レポート	記載内容の合目的性、レポートの構成、議論の妥当性、提案の独創性の観点から評価する。
発表	発表内容の合目的性、発表の構成、議論の妥当性、提案の独創性の観点から評価する。
成果物	作成した図表やドキュメントの妥当性及び分かりやすさ・客観性の観点から評価する。
その他	振り返り:各授業後に学生が提出する振り返りコメントを、理解度、考察・洞察力、応用力の観点から評価する。

## 14. アクティブラーニング(A:行っている B:やや行っている C:行っていない)

授業時間全体に占めるアクティブラーニングの時間的な割合		40%
1	授業で得られた知識や技能を活用し、出題された問題を解いたり、課題に取り組むなど能動的学習を行う	A
2	グループワークで課題に取り組み、学生同士が自由に発言することで何らかの課題に取り組むなど能動的学習を行う	A
3	能動的学習の成果を発表し、そのフィードバックを得て自ら主体的に振り返り、学習効果を高める	A
4	学生自身が主体となって、授業における学習の方向性を定める	C

## 15. 備考

- ・授業では教科書は利用しませんが、受講者には授業資料を公開します。
- ・昼の授業は教室での対面授業とオンライン授業を組み合わせで行いますが、可能な限り教室に出席することをお勧めします。
- ・夜授業はオンライン授業のみとなります。
- ・オンライン授業では、出席者を確認するため、ビデオをオンにしてください。
- ・授業の一部を収録し、授業期間中、受講者にはオンデマンド配信する予定です。

## 16. 授業計画

(注)授業計画は、あくまでも予定であり、実施時に、適時、追加・変更・修正等が生じる場合があります。

### 第1回 コンピュータの動作原理 (講義 45分、演習45分)

ソフトウェアはコンピュータの上で動作します。コンピュータがなぜ動くのかを正しく把握しておく、ソフトウェアの理解が深まります。本授業ではコンピュータの動作原理を学習します。

1. 本講義の進め方(学習目標、成績評価方法、評価基準など)
2. コンピュータの構成(CPU、メモリ、I/O、クロック)
3. フォン・ノイマン・アーキテクチャ
4. メモリの階層構造
5. コンピュータの性能

### 第2回 データ型とデータ構造 (講義 45分、演習45分)

データの扱い方を規定する形式をデータ型、データを効率的に保存および整理するための表現形式をデータ構造と呼びます。本授業ではデータ型とデータ構造の基礎を学びます。

1. 記数法
2. データ型の概要
3. アドレス空間
4. 代表的なデータ構造(配列、リスト、スタックとキュー、木構造)

---

### 第3回 クラウドコンピューティング／Web技術の基礎

(講義 45分、演習45分)

---

現代のビジネスインフラである「クラウドコンピューティング」と「Web技術」の全体像を習得します。技術の理解だけでなく、ビジネスの意思決定において、いつ、どのようにこれらの技術を採用すべきか、その判断能力を養います。

1. クラウドの3形態 (IaaS/PaaS/SaaS) の理解, ビジネスにおける活用
2. ビジネスにおけるクラウドの活用
3. Webシステムの基本的な仕組み
4. 最新のWebトレンド, Webページ的设计

---

### 第4回 AI・機械学習の基礎

(講義 45分、演習45分)

---

現代ビジネスの基盤となっているAI (人工知能) と機械学習の基本構造を理解します。数式を最小限に抑え、直感的なツールやグループワークを通じて「AIで何ができるのか」、 「どのように学習が進むのか」を体系的に習得することを目標とします。

1. AI・機械学習, 統計的手法と深層学習, 主要な3つの学習手法
2. ニューラルネットワークの仕組みと, 深層学習 (ディープラーニング) の革新性
3. LLM (大規模言語モデル) の基礎概念
4. モデル構築のプロセス

---

### 第5回 ソフトウェア工学の概要と倫理

(講義 45分、演習45分)

---

コンピュータの爆発的な普及に伴い、社会におけるソフトウェアの重要性はますます高まっていますが、ソフトウェア開発は容易ではありません。本授業では、ソフトウェアの特徴を理解した上で、ソフトウェア開発を取り巻く現状と課題を議論し、ソフトウェア工学の必要性を学びます。

1. ソフトウェアの特徴
2. ソフトウェアの重要性
3. ソフトウェア開発を取り巻く環境
4. ソフトウェア工学の役割と倫理

---

### 第6回 ソフトウェア開発プロセス

(講義 45分、演習45分)

---

プロセスはプロダクト (製品, 成果物) を生み出す工程であり、プロダクトはプロセスの結果として生み出させるものです。従って、良いソフトウェアを開発するためには良いソフトウェアプロセスが必要です。本授業ではソフトウェアのライフサイクルにおけるプロセスモデルを概観し、プロセス改善の意義と狙いを学習します。

1. ソフトウェアプロセスの定義
2. ライフサイクルモデル
3. ソフトウェアプロセスの現状と課題
4. プロセス改善の意義と狙い

---

### 第7回 ソフトウェア要求

(講義 45分、演習45分)

---

ソフトウェア技術者の仕事は顧客や利用者の要求をソフトウェアによって実現することですが、顧客や利用者の要求は曖昧で一貫性に欠けることが多く、このままではソフトウェアが開発できません。関係者から要求を獲得し、分析し、明確にする必要があります。本授業では、要求に関する概念を整理し、要求分析の流れを解説します。

1. ニーズ, 要求, 要件の違い
2. 機能要求, 非機能要求
3. 要求分析の流れ
4. 要求分析モデリングの概要

---

## 第8回 ソフトウェア設計

(講義 45分, 演習45分)

---

良いソフトウェアを作るには良い分析と設計が必要です. どんなに優れたプログラムであってもユーザのニーズが実現できていなければ役に立ちません. また, 再利用性や保守性が悪いとビジネスには利用できません. 本授業では代表的な設計手法を取り上げ, 比較検討を行います.

1. アーキテクチャ設計
2. データフロー・ダイアグラム (DFD)
3. 状態遷移図 (STD)
4. 実体関連図 (ERD)

---

## 第9回 ソフトウェアテスト/ソフトウェア品質

(講義 45分, 演習45分)

---

ソフトウェアの欠陥は社会に大きな影響を与えるため, ソフトウェアテストと品質管理の重要性はますます高まっています. この授業では, ソフトウェアテストと品質管理の基本的な考え方について学びます.

1. ソフトウェアテストとは?
2. 主なテスト技法
3. 品質と等級
4. ソフトウェア品質モデル

---

## 第10回 オブジェクト指向技術

(講義 45分, 演習45分)

---

ソフトウェアの大規模化, 複雑化に伴って, オブジェクト指向技術が普及してきました. オブジェクト指向にはさまざまな特徴がありますが, 本授業ではその再利用性, 保守性に着目して, オブジェクト指向の理解を深めます.

1. モジュール化, 構造化
2. オブジェクト, クラス
3. カプセル化, データ隠蔽
4. インヘリタンス, ポリモーフィズム

---

## 第11回 UMLの基礎－振る舞い図

(講義 45分, 演習45分)

---

UML (Unified Modeling Language) とは, ソフトウェア開発においてシステムの構造や振る舞いを視覚的に表現するための標準的な図 (ダイアグラム) の記法 (モデリング言語) です. この授業では, システムの動的な振る舞いや相互作用を表す振る舞い図について学びます.

1. UML (Unified Modeling Language) の概要
2. ユースケース図 (Use Case Diagram)
3. アクティビティ図 (Activity Diagram)

---

## 第12回 UMLの基礎－構造図

(講義 45分, 演習45分)

---

UML (Unified Modeling Language) とは, ソフトウェア開発においてシステムの構造や振る舞いを視覚的に表現するための標準的な図 (ダイアグラム) の記法 (モデリング言語) です. この授業では, システムの静的な構造を表す構造図を学びます.

1. クラスとオブジェクト
2. クラス図
3. オブジェクト図

---

第13回, 第14回 AI駆動型ソフトウェア開発演習

(演習 180分)

---

グループ演習を通じ, 要求分析・モデリングから実際に動作するプロトタイプの完成まで, 現代的なソフトウェア開発のワークフローを疑似体験します. ソフトウェアの「設計図 (UML)」を論理的に描く能力と, それを「AI (LLM)」に正しく伝える技術を習得します.

1. UMLによる要求分析・モデリング
2. LLMを用いたコード生成と動作確認

---

第15回 プレゼンテーション

(発表 90分)

---

グループごとに開発したアプリのデモとプレゼンテーションを行います. 全員が発表できるように工夫してください.

1. 開発したアプリのデモ
2. グループ演習を通じて得た気づきと考察の発表

---

第16回 期末試験

(試験 90分)

---

学習目標の到達度を評価するため, 期末試験を実施する予定です.

1. 選択式試験 (テキスト持ち込み可)