

| | | | | | | | | | | |
|-------|-------------------------|-----|---------|---|-----|---------|-----|-----------|------|----|
| 科目名 | プログラミング実習 2 | | | | | | | 年度 | 2026 | |
| 英語科目名 | Programming Practicum 2 | | | | | | | 学期 | 後期 | |
| 学科・学年 | ITスペシャリスト科 | 1年次 | 必/選 | 必 | 時間数 | 90 | 単位数 | 3 | 種別※ | 実習 |
| 担当教員 | 菊池(深) | | 教員の実務経験 | | 有 | 実務経験の職種 | | システムエンジニア | | |

【科目の目的】

プログラミング言語を使用してプログラミング（コーディング）をすることができるようになる。また、プログラミング言語の文法を習得するだけでなく、アルゴリズムを理解し、自身が考える処理の流れ図に当てはめて作成できるようになることを目的とする。

【科目の概要】

本授業では、アルゴリズムのプログラムへの実装方法と実行環境の使用方法を習得する。プログラムを作成する上で基本となる型や変数、演算、制御構造(条件文、繰り返し文)、配列などについて基礎から基本文法を学習する。そのうえで、基本構造を組み合わせることで複雑な処理や判定が行えるように理解を深めていく。

【到達目標】

プログラムを作成するにあたって、プログラミング言語の文法を習得するだけでは自由にプログラムを作成することができない。アルゴリズムを理解し、自身が考える処理の流れ図に当てはめて作成できるようになることが1つめの目標である。更に、プログラミングを作成する上で基本となるデータ型や変数、演算、制御構造(条件文、繰り返し文)などを利用したプログラムの読解と作成ができるようになることで、論理的な思考能力・実習に基づく専門能力・問題解決力を身に付けることが最終的な目標である。

【授業の注意点】

教科書、ノート、筆記用具、ノートパソコンを必ず持参してください。本授業は前の回までの授業の内容が身につけていないことを前提に行うため、必ず復習を行うこと。毎回の授業でわからないことがあれば、積極的に質問して疑問点を解消するように心掛けること。万が一、止むを得ない理由で授業を欠席する場合は、欠席した回の授業動画を視聴し、教科書や補助資料も活用して自己学習を行って、翌週の授業に備えてほしい。

評価基準＝ルーブリック

| ルーブリック評価 | レベル5 優れている | レベル4 よい | レベル3 ふつう | レベル2 あと少し | レベル1 要努力 |
|----------|-------------------------------------------------------|-------------------------------------------------|---------------------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| 到達目標 A | Pythonの特徴と用途を理解し、その内容を他者へ教えることができる。 | Pythonの特徴と用途を理解し、その内容を他者から質問された範囲で説明することができる。 | Pythonの特徴と用途を理解している。 | Pythonの特徴と用途を理解しようと努力をしている。 | Pythonの特徴と用途を理解していない。 |
| 到達目標 B | 文字列を画面表示したり、キーボード入力したデータを使用することができる。 | 教科書を参照しながら、文字列を画面表示したり、キーボード入力したデータを使用することができる。 | 不明点を質問し、教科書を参照しながら、文字列を画面表示したり、キーボード入力したデータを使用することができる。 | 指導を受けながら、文字列を画面表示したり、キーボード入力したデータを使用することができる。 | 指導を受けても、文字列を画面表示したり、キーボード入力したデータを使用することができない。 |
| 到達目標 C | 代表的な3つの制御構造を実装することができる。 | 教科書を参照しながら、代表的な3つの制御構造を実装することができる。 | 不明点を質問し、教科書を参照しながら、代表的な3つの制御構造を実装することができる。 | 指導を受けながら、代表的な3つの制御構造を実装することができる。 | 指導を受けても、代表的な3つの制御構造を実装することができる。 |
| 到達目標 D | 関数やクラスを作成し、部品化することでそのメリットを活かすことができる。 | 教科書を参照しながら、関数やクラスを作成し、部品化することでそのメリットを理解できる。 | 不明点を質問し、教科書を参照しながら、関数やクラスを作成することができる。 | 指導を受けながら、関数やクラスを作成することができる。 | 指導を受けても、関数やクラスを作成することができない。 |
| 到達目標 E | 外部モジュールを活用して、ルーチンワークを自動化するなどの問題解決を行うプログラムを作成することができる。 | 教科書を参照しながら、外部モジュールを活用したプログラムを作成することができる。 | 不明点を質問し、教科書を参照しながら、外部モジュールを活用したプログラムを作成することができる。 | 指導を受けながら、外部モジュールを活用したプログラムを作成することができる。 | 指導を受けても、外部モジュールを活用したプログラムを作成することができない。 |

【教科書】

やさしいPython

【参考資料】

【成績の評価方法・評価基準】

課題 90% 授業毎の学習内容の理解度向上と知識の定着を目的とした課題によって評価する。
平常点 10% 積極的な授業参加度や課題の提出状況によって評価する。

※種別は講義、実習、演習のいずれかを記入。

| 科目名 | | プログラミング実習 2 | | | 年度 | 2026 |
|------|------------|-----------------------------|-----------------|-------------------------------|------|------|
| 英語表記 | | Programming Practicum 2 | | | 学期 | 後期 |
| 回数 | 授業テーマ | 各授業の目的 | 授業内容 | 到達目標＝修得するスキル | 評価方法 | 自己評価 |
| 1 | 実習のための環境構築 | 実習のための環境構築ができる | 1 Pythonプログラム | プログラミング言語Pythonの特徴を理解できる | 1 | |
| | | | 2 環境構築 | 実習のための環境構築ができる | | |
| | | | 3 最初のプログラム | 最初の1本目の簡単なプログラムを作成できる | | |
| 2 | 変数とデータ型 | 変数を使用して基本的なプログラミングができる | 1 開発の流れ | プログラム開発の流れを理解できる | 1 | |
| | | | 2 2種類のエラー | 2種類のエラーの違いを理解できる | | |
| | | | 3 変数 | 変数に値を格納することができる | | |
| 3 | 変数とデータ型 | 変数を使用して複雑な計算を行うプログラミングができる | 1 演算子 | 演算子を使って変数の値に演算することができる | 1 | |
| | | | 2 複合的な演算子 | 複合演算子を使って演算することができる | | |
| | | | 3 キーボード入力 | キーボード入力を行い変数に値を格納できる | | |
| 4 | 変数とデータ型 | データ型の特徴を理解してプログラミングができる | 1 データ型の種類 | 代表的なデータ型の種類と特徴を理解できる | 1 | |
| | | | 2 format関数 | format関数を利用して変数の値を画面表示できる | | |
| 5 | コレクション | リストを使用してプログラミングができる | 1 コレクションの特徴 | コレクションの特徴を理解できる | 1 | |
| | | | 2 リストの特徴 | リストの特徴を理解できる | | |
| | | | 3 リストの利用 | リストを利用してデータを格納できる | | |
| 6 | コレクション | ディクショナリを使用してプログラミングができる | 1 ディクショナリの特徴 | ディクショナリの特徴を理解できる | 1 | |
| | | | 2 ディクショナリの利用 | ディクショナリを利用してデータを格納できる | | |
| 7 | 条件分岐 | 条件分岐を使用してプログラミングができる | 1 代表的な制御構造 | 代表的な3つの制御構造を理解できる | 1 | |
| | | | 2 条件分岐の基本構造 | if文を使用して基本的な条件分岐ができる | | |
| | | | 3 条件分岐の演算子 | 条件分岐に使用する演算子の種類を理解できる | | |
| 8 | 条件分岐 | 複雑な条件分岐の処理をプログラミングできる | 1 条件分岐のバリエーション | 条件分岐のバリエーションを理解できる | 1 | |
| | | | 2 様々な分岐処理 | バリエーションに応じた様々な分岐処理ができる | | |
| | | | 3 ネスト | ネストの概念が理解できる | | |
| 9 | 繰り返し | 繰り返しを使用してプログラミングができる | 1 while文 | while文を利用して繰り返し処理ができる | 1 | |
| | | | 2 for文 | for文を利用して繰り返し処理ができる | | |
| | | | 3 無限ループ | 無限ループになった場合の対処ができる | | |
| 10 | 関数 | 関数の概念を理解できる | 1 関数によるメリット | 関数による部品化するメリットが理解できる | 1 | |
| | | | 2 関数の基本 | 関数を使用して基本的な処理ができる | | |
| | | | 3 引数 | 引数の概念を理解できる | | |
| 11 | 関数 | 関数における変数の特徴を理解できる | 1 ローカル変数 | 関数における変数の特徴を理解できる | 1 | |
| | | | 2 仮引数と実引数 | 仮引数と実引数の概念を理解できる | | |
| | | | 3 戻り値 | 戻り値の概念を理解できる | | |
| 12 | オブジェクト | オブジェクトの概念を理解できる | 1 オブジェクト | すべての値がオブジェクトであったことを理解できる | 1 | |
| | | | 2 オブジェクトの生成 | オブジェクトの生成方法を理解できる | | |
| | | | 3 オブジェクトの設計図 | オブジェクトの設計図について理解できる | | |
| 13 | オブジェクト | オリジナルの設計図 | 1 オリジナルの設計図 | オリジナルの設計図を作成することができる | 1 | |
| 14 | モジュール | モジュールを利用してプログラミングができる | 1 モジュール | モジュールの概念が理解できる | 1 | |
| | | | 2 モジュールの取込み | モジュールの取込んで利用することができる | | |
| | | | 3 外部ライブラリ | 外部ライブラリの概念が理解できる | | |
| 15 | Pythonの可能性 | 様々な場面で活用できるPythonの可能性を理解できる | 1 ルーチワークの自動化 | ルーチワークの自動化する例を知っている | 1 | |
| | | | 2 ウィンドウアプリケーション | ウィンドウアプリケーションが簡単に作成できる例を知っている | | |
| | | | 3 データ分析・機械学習 | データ分析・機械学習に活用されていることを知っている | | |

評価方法：1. 小テスト、2. パフォーマンス評価、3. その他
自己評価：S：とてもよくできた、A：よくできた、B：できた、C：少しできなかった、D：まったくできなかった
備考 等