

科目名	プログラミング実習 1							年度	2025	
英語科目名	Programming 1							学期	前期	
学科・学年	I T スペシャリスト科	1 年次	必／選	必	時間数	60	単位数	2	種別※	実習
担当教員	白幡知之			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】 本授業は、「アルゴリズムとデータ構造」の科目と連携し補完しあいながら進めていく。簡便なプログラム言語であるPythonを通してプログラミングの基礎的な能力を身につけてもらうことを目的として行う。										
【科目の概要】 アルゴリズムの説明と演習を通して、プログラミングの概念や基本的な考え方を学ぶ。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)などについて基礎から学習する。										
【到達目標】 プログラムを作成するにあたって、言語の文法の習得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に当てはめて作成できるようになることが目標である。更に、アルゴリズムを十分理解したうえでプログラミング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)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を目標とする。（論理的な思考能力・実習に基づく専門能力・問題解決力）										
【授業の注意点】 テキスト、問題集、ノート、筆記用具、ノートパソコンを必ず持参すること。毎回の授業は、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。授業時間内に終わらなかった演習問題があった場合には、各自で次の授業までに終わらせておくこと。										
評価基準＝ルーブリック										
ルーブリック 評価	レベル3 優れている			レベル2 ふつう			レベル 1 要努力			
到達目標 A	代入する値に合わせた変数名を付け、値を代入することができ、その変数の値をコンソールへ出力することができる			意味の分からない変数名に値を代入することができ、その変数の値をコンソールへ出力することができる			変数と値の出力方法について理解できていない			
到達目標 B	複数の条件があった場合でもif文、ネストを使用して処理することができる			1つの条件であればif文を利用して処理することができる			判断構造について理解できていない			
到達目標 C	while文を利用し、繰り返し処理ができ、またbreak文やcontinue文による処理の制御をすることができる。			while文を利用し、繰り返し処理ができる。			while文による繰り返し処理ができない			
到達目標 D	コレクションのそれぞれのデータ構造と特徴を理解し、データの登録や削除、for文による繰り返し処理をすることができる			List型のみデータの登録や削除、for文による繰り返し処理をすることができる			コレクションが理解できない			
到達目標 E	引数・戻り値のありなしを理解した上で自分で関数を定義し、利用することができる			関数は定義できるが、引数・戻り値が理解できてはいるが、利用することはできない			関数が理解できない			
【教科書】 スッキリわかるPython入門（インプレス）										
【参考資料】										

【成績の評価方法・評価基準】									
試験・課題、小テスト、平常点									
※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 1					年度	2025	
英語表記		Programming 1					学期	前期	
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	Pythonの導入	開発環境構築を行える	1	オリエンテーション	授業の内容と目的について理解している			1	
			2	Pythonの概要	プログラミング言語Pythonのインストール方法を理解している				
			3	開発環境の構築	VSCodeを利用した開発環境構築方法を理解している				
2	プログラミングの基礎	開発環境を利用したプログラミングと実行ができる	1	ファイルの作成と拡張子	プログラムの作成に必要なファイルを拡張子を指定して作成できる			3	
			2	コンソールへの出力	Pythonプログラムの基本形を理解し、コンソール出力するプログラムが書ける				
			3	開発環境での実行	作成したプログラムを実行の仕方を理解する				
3	定数と変数	定数と変数、データ型を理解し、これらを活用したプログラムが作成できる	1	定数	整数、浮動小数点数、文字列の違いを理解する			3	
			2	変数と代入演算子	変数の作成方法と変数へデータの代入方法を理解する				
			3	データ型	定数や変数に登録された値にはデータ型があることを理解する				
4	式と演算子	各種演算子を活用し、式を組み入れたプログラムが作成できる	1	四則演算	四則演算を利用したプログラムの作成ができる			3	
			2	剰余算(%）、累乗(**)	剰余算や累乗演算を利用したプログラムが作成できる				
			3	演算子の優先順位	演算子の優先順位を理解する				
5	判断構造	判断構造を活用したプログラムが作成できる	1	比較演算子	比較演算子と論理値型を理解する			3	
			2	if～else文	if文の書き方を理解し、1つの条件を利用したプログラムが作成できる				
			3	if～elif～else文	複数の条件を利用したプログラムが作成できる				
6	多分岐構造	入れ子（ネスト）の考え方とインデントを理解し、多分岐構造を活用したプログラムが作成できる	1	インデント	インデントを理解する			3	
			2	if文のネスト	if文の中にif文を入れる入れ子（ネスト）の考え方を理解し、プログラムが作成できる				
7	反復構造	反復構造を理解し、処理を繰り返すプログラムを作成できる	1	while文	while文を利用したプログラムが作成できる			3	
			2	無限ループ	無限ループに陥った際に強制的にプログラム停止ができる				
			3	break文、countinue文	breaad文、continue文を利用したプログラムが作成できる				
8	デバッガの利用	ブレークポイントを利用したデバッグができる	1	ブレークポイント	ブレークポイントの設置の仕方とデバッグ実行の仕方を理解する			3	
			2	デバッグ	ブレークポイントで止めたプログラムの変数の値を確認する方法を理解する				
			3	ステップ実行、継続	ステップ実行の仕方、プログラムの継続の仕方を理解する				
9	コレクション(1)	Listを利用したプログラムが作成できる	1	Listの定義	Listを利用した複数のデータを管理するデータ構造を理解し、プログラムが作成できる			3	
			2	要素の参照、追加と削除	作成した複数のデータから要素を1つ取り出したり、追加、削除できるプログラムを作成できる				
			3	for文による繰返し処理	for文を利用した、List型の繰返し処理のプログラムが作成できる				
10	コレクション(2)	辞書型、セット型を利用したプログラムが作成できる	1	辞書型	辞書型のデータ構造を理解し、プログラムが作成できる			3	
			2	セット型	セット型のデータ構造を理解し、プログラムが作成できる				
			3	for文による繰返し処理	List同様に、for文による繰返し処理のプログラムが作成できる				
11	コレクション(3)	リストの要素にリストや辞書など、複数のコレクションを組み合わせたプログラムが作成できる	1	リストとリスト	リストの要素にリストを入れる、多次元のデータ構造を理解し、プログラムが作成できる			3	
			2	辞書とリスト	辞書の値にリストを入れるデータ構造を理解し、プログラムが作成できる				
			3	繰返し処理	複数のコレクションで構成されたデータ構造を繰返し処理するプログラムが作成できる				
12	組み込み関数とライブラリ	よく利用される組み込み関数と乱数などのライブラリを利用したプログラムが作成できる	1	組み込み関数	int, float, len, rangeなどの組み込み関数を利用したプログラムが作成できる			3	
			2	ライブラリ	乱数や日付ライブラリを利用したプログラムが作成できる				
13	ユーザ定義関数	関数の概念を理解し、関数の作成・呼び出すプログラムが作成できる	1	関数とは	関数の概念を理解している			3	
			2	defキーワード	ユーザ定義関数の書き方を理解し、関数プログラムを作成できる				
			3	ユーザ定義関数の呼び出し	自分で作成した関数の呼び出し方を理化学し、プログラムが作成できる				
14	関数	引数や戻り値、関数のメソッドを理解し、関数を活用したプログラムが作成できる	1	引数	引数がある関数のプログラムが作成できる			3	
			2	戻り値	戻り値がある関数のプログラムが作成できる				

		きる					
15	総合演習	これまで学んだ内容を総合的に活用したプログラムが作成できる	1	総合演習	変数、関数、コレクション、判断構造、繰返し処理を組み合わせたプログラムが作成できる		3

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