

科目	プログラミングⅡ (Programming II)		
担当教員	高田 峻介 講師		
対象学年等	電子工学科・3年・通年・必修・2単位【講義・演習】(学修単位I)		
学習・教育目標	A3(100%)		
授業の概要と方針	アルゴリズム、データ構造などのプログラミングを実践的な見地から学習する。前期では、C言語の復習とファイル入出力、およびオブジェクト指向型言語 (Java 言語等) について触れる。後期ではウェブサービスなどと連携した実用的で発展性のある技術について、各種プログラミング言語を用いて広く紹介する。個人開発やチーム開発を通してプログラミングで“何か”を実現する喜びを感じて頂きたい。		
	到達目標	達成度	到達目標別の評価方法と基準
1	【A3】C言語によるプログラミングの基礎を理解できる。		C言語プログラミングの基礎を理解できているかを前期中間試験、レポートで評価する。
2	【A3】プログラミングによりファイル入出力操作をすることができる。		ファイル入出力操作を理解できているかを前期中間試験、レポートで評価する。
3	【A3】オブジェクト指向型プログラムを理解することができる。		オブジェクト指向型プログラムの特徴であるインスタンス化、コンストラクタ、クラスの継承が理解できているかを前期定期試験、レポートで評価する。
4	【A3】提示された問題を解決できるようなプログラミングをおこなうことができる。		プログラミングによる問題解決能力を総合演習、プレゼンテーションとプレゼン関係資料、そのレポートで評価する。
5			
6			
7			
8			
9			
10			
総合評価	成績は、試験35% レポート30% プレゼンテーション10% 総合演習15% プレゼン関係資料10% として評価する。試験点は前期中間試験と前期定期試験の平均を試験成績として評価する。100点満点で60点以上を合格とする。尚、場合により臨時試験を実施する。臨時試験では、70点以上を合格とし、当該試験の点数を60点とする。		
テキスト	配布プリント		
参考書	「プログラミング言語C ANSI規格準拠」: B.W. カーニハン(共立出版) 「Java言語プログラムレッスン 上下」: 結城浩(SBクリエイティブ) 「入門 Python 3」: Bill Lubanovic著, 斎藤康毅監修, 長尾高弘訳(オライリー・ジャパン) 「プログラマの数学 第2版」: 結城浩(SBクリエイティブ)		
関連科目	情報基礎, プログラミングI, ソフトウェア工学, 数値解析, 電子工学実験実習		
履修上の注意事項	本科目では、プログラミングIをさらに実践的なものへと発展させることを目指す。そのため、プログラミングIの内容を十分理解しておく必要がある。加えて、電子工学科におけるプログラミングは電子工学の知識と密接に関わっているため、実験実習などの実習系科目についてもその楽しさを理解しておく必要がある。		

授業計画(プログラミングII)

	テーマ	内容(目標・準備など)
1	C言語の復習1	2年次に学習したC言語の復習を行う。ここでは、標準入出力や四則演算、2進数の計算に必要な論理演算やシフト演算、科学計算に欠かせない算術演算関数について復習する。
2	C言語の復習2	関数と変数についての講義を行う。
3	ポインタ1	C言語では、ポインタを利用してメモリ上のアドレスにアクセスすることができる。ここでは、メモリに割り当てられたポインタ変数がどのような振る舞いを行うかを学び、その使用方法について理解を深める。
4	ポインタ2	配列とポインタ、構造体のポインタ、ポインタのポインタを学習し、その使用方法について理解を深める。
5	ファイル入出力1	プログラミングにおいて、データを読み込んだり、結果を書き出したりする処理は重要である。このような処理ファイルは入出力と呼ばれるが、その概要を示す。また、ファイルにはテキストファイルとバイナリファイルが存在するが、ここではテキストファイルを中心にその入出力方法を学ぶ。
6	ファイル入出力2	科学計算を行う上で、バイナリファイルを扱うことは重要である。ここでは、バイナリファイルの入出力に関して学習を行う。
7	演習	ファイル入出力に関する演習を行う。
8	中間試験	1~7週に関する内容の中間試験
9	試験問題の解答と解説。オブジェクト指向型プログラミングの説明	中間試験問題の解答を示し、その解説をする。その後、オブジェクト指向型プログラミングの説明を行い、手続き型プログラミングとの違いを学習する。また、Java言語ベースであるProcessing言語等を用いてオブジェクト指向型プログラミングを理解する。
10	オブジェクト指向言語1 クラスとインスタンス	クラスとオブジェクト、インスタンスの生成方法を講義し、その使い方を理解する。
11	オブジェクト指向言語2 コンストラクタ	クラスからオブジェクトをインスタンス化する際、その初期化処理を行うコンストラクタという概念が存在する、その使用方法を学び、理解を深める。
12	オブジェクト指向言語3 継承	あるクラスに、新しいメソッドやフィールドを追加して新しいクラスを作ることを継承という。これらの概念を理解し、演習を通して理解を深める。
13	オブジェクト指向言語4 ファイル操作と入出力	オブジェクト指向言語におけるファイル入出力操作を理解する。
14	オブジェクト指向言語演習	オブジェクト指向言語に関する演習を行う。
15	総合演習	これまでの内容に関する総合演習を行う。
16	プログラミング言語解説	前期で学んだプログラミング言語以外に、多くの思想に基づいて設計・利用されているプログラミング言語について学ぶ。
17	Python入門1	各分野で幅広く用いられているスクリプト言語であるPython言語の基本を学ぶ。特徴的なインデントの書き方や変数や配列の扱い、ifやforなどの制御構造について解説する。
18	Python入門2	Pythonは様々なライブラリが充実しており、その使いこなしが重要である。ここではそのライブラリの紹介と使い方について講義する。
19	発展プログラミング言語と個人開発 1	与えられた目標に基づき、各自が希望する課題を設定し、課題に合ったプログラミング言語を選定し、その実装方法を学ぶ。本講義で紹介していない言語でも差し支えない。
20	発展プログラミング言語と個人開発 2	前週に引き続き、課題に取り組む。
21	発展プログラミング言語と個人開発 3	前週に引き続き、課題に取り組む。
22	個人開発プレゼンテーション	19週目から21週目の内容に関する発表を行う(1人2分程度)。
23	アイデア設計 1	前週までに学習した内容および与えられた目標に基づき、3~5人のチームで目標を達成するためのアイデアの設計や実装方法の選定を行う。
24	アイデア設計 2	前週に引き続きアイデアの設計に取り組み、その内容をスライドにまとめる。
25	アイデアプレゼンテーション	23週目から24週目の内容に関するアイデア発表と相互講評を行う(1チーム4分程度)。
26	チーム開発 1	前週のアイデア発表およびその講評に基づき、課題の再設計を行い、チームでのアイデアの実装に取り組む。
27	チーム開発 2	前週に引き続きチーム課題に取り組む。
28	チーム開発 3	前週に引き続きチーム課題に取り組む。
29	チーム開発プレゼンテーション	チーム課題についてのプレゼンテーションを行う(1チーム4分程度)。
30	総括	この講義の総括をおこなうとともに、講義中に紹介しきれなかった事例や今後の発展などについて述べる。
備考	前期中間試験および前期定期試験を実施する。 後期は試験を実施せず、かわりにプレゼンテーションおよびレポートで評価する。	