

木更津工業高等専門学校		開講年度	平成30年度 (2018年度)	授業科目	プログラミング技法Ⅱ
科目基礎情報					
科目番号	0052		科目区分	専門 / 必修	
授業形態	授業		単位の種別と単位数	履修単位: 1	
開設学科	電子制御工学科		対象学年	2	
開設期	後期		週時間数	2	
教科書/教材	内容に合わせて作成した講義資料を配布する；（補助教科書）林晴比古著、『明快入門C』、ソフトバンククリエイティブ、2013年、1900円（＋税）、ISBN-13: 978-4797373264				
担当教員	関口 明生				
到達目標					
本学科を卒業し将来ソフトウェアを作成または利用する業務に携わるための基礎として、プログラミングの基礎的知識と技能を身に付けること。目安として： ☐ 処理の順序を、フローチャートで表現することができる。 ☐ ソースコードにインデントやコメントを用いてプログラムを作成することができる。 ☐ 基礎的なC言語のプログラムの動作を説明することができる。 ☐ 構造化プログラミングにおける順次構造・分岐構造・反復構造の構造化文を、C言語において正しく記述できる。					
ルーブリック					
	理想的な到達レベルの目安		標準的な到達レベルの目安		未到達レベルの目安
他者と協調してプログラミングする基礎能力	コメントの内容や変数名なども分かりやすいプログラムを作成することができる。		作成することができる。ソースコードにインデントやコメントを用いてプログラムを作成することができる。		ソースコードにインデントやコメントを用いてプログラムを作成することができない。
動作を理解して改良・デバッグする能力	基礎的なC言語のプログラムの動作を説明することができるだけではなく、改良するための工夫ができる。		基礎的なC言語のプログラムの動作を説明することができる。		基礎的なC言語のプログラムの動作を理解することができない。
処理構造を系統立てて扱う能力	構造化プログラミングにおける順次構造・分岐構造・反復構造の構造化文を、C言語において正しく適切に工夫して記述できる。		構造化プログラミングにおける順次構造・分岐構造・反復構造の構造化文を、C言語において正しく記述できる。		構造化プログラミングにおける順次構造・分岐構造・反復構造の構造化文の違いがわからない。
処理内容をモジュール化して扱う能力	可読性や再利用性を考慮したうえで、関数を記述して処理内容をより適切に区分することができる。		関数を記述して処理内容を区分することができる。		関数を記述して処理内容を区分することができない。
UNIX環境を操作する基礎能力	基礎的なUNIXコマンドを記憶し、複数組み合わせることで応用することができる。		基礎的なUNIXコマンドを扱うことができる。		基礎的なUNIXコマンドを扱うことができない。
多くの情報を効率的に処理するための基礎能力	データをファイルへ出力・ファイルから入力するプログラムを、アルゴリズムを組み合わせるなど工夫して作成できる。		データをファイルへ出力・ファイルから入力するプログラムを作成できる。		データをファイルへ出力・ファイルから入力するプログラムを作成できない。
アルゴリズムを利用して創作する能力	基礎的なアルゴリズムやデータ構造の動作を説明することができ、自主的に調べてソフトウェアを創作することができる。		基礎的なアルゴリズムの動作を説明することができる。		基礎的なアルゴリズムの動作を説明することができない。
学科の到達目標項目との関係					
教育方法等					
概要	コンピュータを用いてソフトウェアを作るためのプログラミングの基礎的知識と技能を、C言語を用いて学び演習する。より具体的にはメール・エディタ・UNIX操作などの基礎的な操作方法から、順次構造・分岐構造・反復構造・関数・ファイル操作・構造体・各種基礎的アルゴリズムなどのC言語基礎全般を扱う。				
授業の進め方・方法	スライドを用いた講義をなるべく手短かに受けた後で、演習課題によって学習内容を身に付ける。課題は、授業内容を十分に理解すればするほど、取り組みやすい。課題の内容を常に十分理解して、遅延や未提出なく出すことができる。試験勉強に多くの時間を割かなくてもよいと考えられる。原則毎回、授業用プリントを配布する。このプリントを教科書代わりとして授業を進める。このため、各自でA4のファイルを用意して整理しておくこと。もしも整理ができないと、本科目を履修する難易度を自ら高くしてしまうことになる。補助教材「明快入門C」は、辞書あるいはハンドブックとして使用する。このため、人によって授業で用いることは多くないかもしれないが、自習や本科目履修以後にも用いることができる。質問があれば、授業時間内や授業時間外に積極的に担当教員まで申し出ること。				
注意点	プログラミングの知識と技能は、実験実習・情報処理・制御工学・卒業研究ほか多くの専門科目の礎の一つである。すなわち、低学年の他科目や実験実習Ⅰで実施したタッチタイピングと同様に、本科目が身につかないと卒業に支障をきたす。したがって、分からない箇所があれば、積極的に質問あるいは補助教科書などで調べる。卒業後の君が余計な苦勞をしないために、タッチタイピングにより授業に臨むこと。本科目を習得して勉学を重ねると、たとえば簡単なゲームを作ることすら可能となる。				
授業計画					
		週	授業内容	週ごとの到達目標	
後期	3rdQ	1週	【UNIX操作】 「UNIX操作の基礎」 ○基本的なUNIXコマンド（logout, ls, mkdir, rmdir, cd, pwd, echo, cat, cp, mv, rm, head, tail, history） ○パイプとリダイレクト	☐ 初歩的なUNIXコマンドを用いてファイル操作を行うことができる。	
		2週	【関数】 「関数(1)」 ○関数とプロトタイプ宣言の書き方 ○戻り値のある関数 ○引数のある関数（引数の値渡し） ○戻り値と引数のある関数 ☆課題（関数、プロトタイプ宣言、for文、引数、戻り値、整数÷整数、リダイレクト）	☐ 処理のまとまりを関数としてmain()の外に記述することができる。 ☐ 【特に重要】引数と戻り値を用いてデータを受け渡す関数を記述することができる。	

		3週	「関数(2)」 ○引数のある関数(引数のアドレス渡し) ○定数形のマクロと関数形のマクロ ☆課題(関数、値渡し、アドレス渡し、マクロ、文字列、配列)	□【特に重要】引数の値渡しとアドレス渡しの違いを理解してプログラムを作成することができる。 □マクロを用いたプログラムを作成することができる。
		4週	「関数(3)」 ○ローカル変数とグローバル変数 ○自動変数と静的変数 ○変数のスコープ(有効範囲) ○ソースファイルの分割 ☆課題(関数、プロトタイプ宣言、グローバル変数、静的変数、スコープ、★最大値を逐次計算するアルゴリズム)	□グローバル変数や静的変数の違いについて理解してプログラムを作成できる。 □ソースファイルを分割して作成しコンパイルするための方法を理解する。
		5週	「関数(4)」 ○コマンドライン引数 ○再帰関数 ○関数を作成する利点 ☆課題(コマンドライン引数、atoi()、atof()、再帰関数、最大公約数、★ユークリッドの互除法)	□コマンドライン引数を受け取るプログラムを作成することができる。 □通常の関数と再帰関数の違いを説明することができる。
		6週	【ファイル操作】 「ファイル出力」 ○ファイルの書き込み ○CSVファイルのグラフ描画 ☆課題(ファイル出力、数学関数、コマンドライン引数、RC回路の電圧の時間変化、for文)	□実行結果をファイルに書き込むプログラムを作成することができる。
		7週	「ファイル入力」 ○ファイルの読み込み ☆課題(ファイル入力、EOF、合計と平均、for文)	□データの記録されたファイルを読み込み、処理を行うプログラムを作成することができる。
		8週	「後期中間試験」(予定) ○プログラミング関連の英単語 ○プログラミングに関する基礎用語 ○基本的なUNIXコマンド ○値とアドレスの区別 ○プログラム作成	後期中間試験までの内容を説明あるいは実践できる。 特に: □関数を記述して処理内容を区分することができる。 □基礎的なUNIXコマンドを扱うことができる。
	4thQ	9週	○後期中間試験の解答と解説	□試験で間違った内容から、何に集中して復習すれば効率的に知識・技術を身に付けることができるか、前向きに取り組むことができる。
		10週	【ソートアルゴリズム】 「スワップ・ソート・中央値」 ○メモリの値の入れ替え(swap) ○並び替えのアルゴリズム(★選択ソート、★バブルソート、★挿入ソート) ○平均値と中央値 ☆課題(コマンドライン引数、アドレス渡し、ソートアルゴリズム、中央値)	□基礎的なソートアルゴリズムを用いたプログラムを作成することができる。 □平均値と中央値の違いを説明することができる。
		11週	【構造体と動的確保】 「構造体」 ○構造体の宣言・定義・初期化・使用 ○構造体の配列とポインタ ○構造体の値渡し・アドレス渡し ☆課題(構造体、値渡し・アドレス渡し、乱数、無限ループ、アロー演算子)	□構造体を用いたプログラムを作成することができる。 □構造体を配列・ポインタ・関数を使って扱うことができる。
		12週	動的確保「メモリの動的確保・セグメント」 ○メモリの動的確保と解放 ○メモリセグメント ☆課題(構造体、動的確保、コマンドライン引数、平均アルゴリズム)	□メモリの動的確保と解放を用いたプログラムを作成することができる。 □スタックオーバーフローやメモリリークなどの、メモリセグメント内の構造に関連した現象を説明することができる。
		13週	【データ構造の基礎】 「LIFO構造の作成と操作」 Otypedef宣言 OLIFO構造 ☆課題(LIFO構造メモリ、構造体、動的確保、コマンドライン引数、関数形マクロ、★逆ポーランド記法によるスタック計算機)	□構造体やメモリの動的確保を用いてLIFOメモリのデータ構造を実現する方法を知り、作成済みのソースファイルを利用して比較的複雑なソフトウェアを完成させることができる。 □LIFOメモリを用いてスタック計算機のアルゴリズムがどのように動作しているか説明することができる。
		14週	「FIFO構造の作成と操作」 OFIFO構造 ☆課題(FIFO構造メモリ、構造体、動的確保、★平均を逐次計算するアルゴリズム)	□構造体やメモリの動的確保を用いてFIFOメモリのデータ構造を実現する方法を知り、作成済みのソースファイルを利用して比較的複雑なソフトウェアを完成させることができる。
		15週	「後期定期試験」(予定) ○プログラミング関連の英単語 ○条件式の書き方 ○プログラミングに関する基礎用語 ○プログラミングに関連する基本操作 ○値とアドレスの区別 ○プログラム作成	後期定期試験までの内容を説明あるいは実践できる。 特に: □データをファイルへ出力・ファイルから入力するプログラムを作成できる。 □基礎的なアルゴリズムの動作を説明することができる。
		16週	○後期定期試験の解答と解説	□今までの学習内容でできたことと、十分にできなかったことを認識し、来年度以降に生かす意志がある。

評価割合							
	試験	発表	相互評価	課題と態度	ポートフォリオ	その他	合計
総合評価割合	50	0	0	50	0	0	100
基礎的能力	0	0	0	0	0	0	0
専門的能力	50	0	0	50	0	0	100
分野横断的能力	0	0	0	0	0	0	0