

宇部工業高等専門学校		開講年度	令和03年度 (2021年度)		授業科目	プログラミングⅡ	
科目基礎情報							
科目番号	32023			科目区分	専門 / 必修		
授業形態	講義			単位の種別と単位数	履修単位: 3		
開設学科	制御情報工学科			対象学年	2		
開設期	通年			週時間数	3		
教科書/教材	「新・明解C言語入門編」柴田望洋 著 (ソフトバンク・パブリッシング)						
担当教員	田辺 誠,勝田 祐司						
到達目標							
講義と演習を通してC言語の基本的なプログラミング技術を習得することを目的とする。 本講義の到達目標は以下の通りである。 (1) 基本的なプログラムの作成・実行ができる。 (2) C言語の各構文について説明できる。 (3) コンパイル時・実行時に出力されるメッセージの内容を説明できる。 (4) 作成したプログラムの機能や振る舞いを他人に分かりやすく説明できる。							
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	最低限の到達レベルの目安 (可)	未到達レベルの目安			
評価項目1	自身で応用課題を設定し、課題を解決するプログラムを作成・実行できる。	教科書に掲載されている発展課題のプログラムを作成・実行できる。	基本的なプログラムの作成・実行ができる。	基本的なプログラムの作成・実行ができない。			
評価項目2	C言語の各構文について理解し、応用的なプログラムを作成できる。	C言語の各構文について理解し、基礎的なプログラムを作成できる。	C言語の各構文について理解できる。	C言語の各構文について理解できない。			
評価項目3	コンパイル時・実行時に出力されるメッセージの内容を説明でき、種々のエラーに対処できる。	コンパイル時・実行時に出力されるメッセージの内容を説明でき、初歩的なエラーに対処できる。	コンパイル時・実行時に出力されるメッセージの内容を説明できる。	コンパイル時・実行時に出力されるメッセージの内容を説明できない。			
評価項目4	複雑なプログラムの機能や振る舞いについてわかりやすく説明できる。	プログラムの機能や振る舞いについてわかりやすく説明できる。	プログラムの機能や振る舞いについて説明できる。	プログラムの機能や振る舞いについて説明できない。			
学科の到達目標項目との関係							
教育方法等							
概要	第1、3、4学期開講 C言語の基本技術として、配列、関数、文字列、ポインタ、構造体を学習する。 また、リダイレクトやコンパイラのオプションなどについても学習する。						
授業の進め方・方法	本講義は、1回当たり4単位時間連続して講義と演習を実施する。情報処理演習室で、初めに解説を行った後、プログラミング演習を行う。コンピュータ、ネットワークの利用についてはマナーを守ること。ホームルームからの移動は速やかに行き、遅れないようにすること。						
注意点	第4学期のみ試験を行います。また、第1学期末に補講が入りますので、注意すること。 教科書を持参し、細部について参照ながら理解を深めること。また、教科書の授業内容の範囲を熟読し、復習しておくこと。 演習が時間内に終了しない場合もあり得るが、放課後などを利用して実施し、期限までに必ず提出できるようにすること。 課題レポートの提出期限は厳守すること。原則、再試験は実施しない。 プログラミングが得意な学生は、与えられた課題をこなすだけで満足せず、追加課題を教員に求めること。						
授業の属性・履修上の区分							
☐ アクティブラーニング		☐ ICT 利用		☐ 遠隔授業対応		☐ 実務経験のある教員による授業	
授業計画							
		週	授業内容	週ごとの到達目標			
前期	1stQ	1週	ガイダンス 入出力の復習	シラバスから学習の意義、授業の進め方、評価方法を理解できる。 出力と変数、入力、条件分岐について、種々のデータに関するレポート作成ができる。			
		2週	関数：教科書第6-1節を学習する。	関数の呼び出しを含むプログラムを読めるようになり、関数の定義を自分で書けるようになる。			
		3週	関数の設計：教科書第6-2節を学習する。	教科書6-2節を理解し、値を返さない関数や仮引数を受け取らない関数を作成することができる。			
		4週	関数の応用：教科書第6章の例題をもとに関数利用時の注意事項や探索プログラムへの応用例を学習する。	配列を引数とする関数に関する注意点を理解するとともに、線形探索のアルゴリズムを理解し、プログラムを作成することができる。			
		5週	基本型：教科書第7章を学習する。	C言語に現れる様々な基本型の違いについて理解し、状況によって使い分けができる。整数の2進表現について理解し、目的に応じたビット演算を行うことが出来るようになる。			
		6週	いろいろなプログラム：教科書第8章を学習する。	関数形式マクロや列挙体などを用いたプログラムを作成できるようになる。また、ソートや再帰などのアルゴリズムを理解し、実行結果を予想できるようになる。			
		7週	文字列の基本・文字列の配列：教科書第9-1節および第9-2節を学習する。	文字列や文字列リテラルの基本を理解し、プログラムを作成できるようになる。			
		8週	文字列の操作：教科書第9-3節を学習する	文字列の長さを求める関数や文字列内の数字文字の出現回数を求めるプログラムについて理解し、実行結果を予想できるようになる。			
	2ndQ	9週					

後期	3rdQ	10週			
		11週			
		12週			
		13週			
		14週			
		15週			
		16週			
		1週	関数に関するプログラミング演習：課題を課す	与えられた仕様にに基づき、関数の実装ができるようになる。	
		2週	文字列に関するプログラミング演習：課題を課す	文字列を扱うプログラムに適語補充を行い、プログラムを完成できるようになる。	
		3週	ポインタ：教科書第10-1節および第10-2節（途中まで）を学習する。	メモリ空間上のアドレスを指すポインタの概念を理解し、アドレス演算子やポインタ型などを用いたプログラムの挙動を説明できるようになる。	
		4週	ポインタと関数・配列：教科書第10-2節（途中から）および第10-3節を学習する。	ポインタを引数として受け取る関数に関するプログラムの作成ができるようになる。	
		5週	文字列とポインタ：教科書第11-1節および第11-2節を学習する。	配列とポインタの共通点と相違点を理解し、これらを用いたプログラムの作成ができるようになる。	
		6週	文字列を扱うライブラリ関数：教科書第11-3節を学習する。	文字列を扱う代表的なライブラリ関数に関する仕様を理解し、これらを読み出すプログラムの作成ができるようになる。	
		7週	構造体：教科書第12-1節（途中まで）を学習する。	構造体の必要性について理解し、目的に応じた構造体を定義できるようになる。構造体や構造体のポインタを引数に持つ関数の作成ができるようになる。	
		8週	構造体：教科書第12-1節（途中から）および第12-2節を学習する	typedefや->演算子など、構造体を用いたプログラムの可読性を向上する方法を学び、これらを用いたプログラムが読めるようになる。また、構造体のメンバーに別の構造体を含むプログラムについて理解し、実行結果が予想できるようになる。	
	4thQ	9週	ファイル入出力：教科書第13章および配布資料に基づいて演習を行う。	標準入出力の概念を理解し、指示に従ってリダイレクトやパイプを用いたコマンドの実行ができるようになる。また、ファイル入出力関数を用いたファイルへの入出力のプログラムを指示に基づいて作成できるようになる。	
		10週	コンパイルオプションとエラーメッセージ：配布資料に基づいて演習を行う。 オリジナルプログラム：構造体を用いた自由課題に取り組む。	これまでに用いたコンパイルオプションについて整理し、指示が無くても自分で考えてオプション指定が出来るようになる。また、コンパイル時や実行時のエラー状況をもとに、エラーの原因をつきとめ、修正する方法を習得する。自由課題では、題材を自ら選び、その題材に適した構造体を定義できるようになる。	
		11週	ファイル分割：配布資料に基づいて演習を行う。 オリジナルプログラム：構造体を用いた自由課題に取り組む。	指示に基づいて、単一ファイルのソースコードを順を追ってファイル分割できるようになる。また、ヘッダファイルとインクルードの役割について理解し、これらを用いたプログラムの構成を理解できるようになる。また、makeコマンドを用いたコンパイルを指示に従ってできるようになる。自由課題では、定義した構造体のデータを表示するプログラムをmain関数内に作成できるようになる。	
		12週	課題演習：オンラインテストに基づく演習を行う。 オリジナルプログラム：構造体を用いた自由課題に取り組む。	これまでの学習範囲を対象とするオンラインテストに取り組み、自分の理解度について再確認できる。解説が必要な単元について自分で判断し、教員に示すことができる。自由課題では、つまづいた個所に関するフィードバックをもとに、エラーへの対処を進めることができる。	
		13週	課題演習解説：オンラインテストのフィードバックを行う。 オリジナルプログラム：構造体を用いた自由課題に取り組む。	オンラインテストの解説を聞き、オンラインテストレベルの問題が解けるようになる。自由課題では、機能ごとの関数化やデータの充実、ファイル入出力等、これまでに学んだ内容を理解度に応じて採り入れることができる。	
		14週	総合説明、試験準備	これまでの疑問点を質問することで疑問点の解消ができる。	
		15週	（定期試験）		
		16週	試験返却 授業改善アンケートの実施	試験問題の解説を通じて間違った箇所を理解し、正しい解答を説明できる。また、授業改善アンケートを行う。	

モデルコアカリキュラムの学習内容と到達目標						
分類		分野	学習内容	学習内容の到達目標	到達レベル	授業週
専門的能力	分野別の専門工学	情報系分野	プログラミング	代入や演算子の概念を理解し、式を記述できる。	3	
				ブローシャ(または、関数、サブルーチンなど)の概念を理解し、これらを含むプログラムを記述できる。	3	
				変数の概念を説明できる。	3	
				データ型の概念を説明できる。	3	
				制御構造の概念を理解し、条件分岐を記述できる。	3	
				制御構造の概念を理解し、反復処理を記述できる。	3	
				主要な言語処理プロセッサの種類と特徴を説明できる。	4	
				ソフトウェア開発に利用する標準的なツールの種類と機能を説明できる。	4	
				プログラミング言語は計算モデルによって分類されることを説明できる。	4	

				主要な計算モデルを説明できる。			4	
評価割合								
	試験	発表	相互評価	態度	ポートフォリオ	レポート	合計	
総合評価割合	50	0	0	0	0	50	100	
知識の基本的な理解	25	0	0	0	0	25	50	
思考・推論・創造への適用力	25	0	0	0	0	25	50	
汎用的技能	0	0	0	0	0	0	0	
態度・志向性(人間力)	0	0	0	0	0	0	0	
総合的な学習経験と創造的思考力	0	0	0	0	0	0	0	