

函館工業高等専門学校		開講年度	令和04年度 (2022年度)		授業科目	情報処理 I	
科目基礎情報							
科目番号	0037			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: 2		
開設学科	社会基盤工学科			対象学年	2		
開設期	通年			週時間数	2		
教科書/教材	筧捷彦監修, 後藤良和・佐久間修一・高田大二著 「初級C言語」 (実教出版) / 適宜プリント配布						
担当教員	山崎 俊夫,越智 聖志						
到達目標							
1. C言語のプログラミング方法を理解して自ら基本的なプログラムが作成できる。 2. アルゴリズムを理解してプログラミングすることができる。 3. 教員の指導を受けて基本的なプログラムから高度な応用プログラミングができる。							
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	
評価項目1	デバッグを通じて独力でプログラムのエラーを修正し、応用的なプログラムを作成できる。			教員の指導によりプログラムのエラーを修正し、応用的なプログラムを作成できる。		プログラムのエラーを修正して、応用的なプログラムを作成することができない。	
評価項目2	アルゴリズムを理解し、これに基づく正しいフローチャートを独自に作成できる。			アルゴリズムに基づいたフローチャートを教員の指導を受けて作成できる。		アルゴリズムを理解できない、もしくはフローチャートを作成できない。	
評価項目3	応用的なデータ処理を行うプログラムを作成できる。			簡単なデータ処理を行うプログラムを作成できる。		簡単なデータ処理を行うプログラムが作成できない。	
学科の到達目標項目との関係							
函館高専教育目標 B 函館高専教育目標 C							
教育方法等							
概要	情報処理 I では、第 1 学年で学んだプログラミング入門に引き続き、C言語を用いてソフトウェア作成に必要な基礎概念及びソフトウェアの作成工程を学習する。本科目の履修に当たっては第 1 学年のプログラミング入門の内容について復習しておく必要がある。						
授業の進め方・方法	第 1 学年のプログラム入門と同様に、以下の点に留意して授業に取り組むこと。 ・プログラム開発にはエラーがつきものである。なぜエラーとなったのかを考えることがプログラミング能力の向上につながる。 ・見やすいプログラムを書くこと。見やすいプログラムは、処理手順が理解しやすく、一般的にエラーも少ない。 ・構造化プログラミングの技法についてもしっかりと身につけ、積極的に演習に取り組むこと。 後半の演習については、「成果品実技」の達成度評価となる。各演習問題の採点基準は以下のとおりとする。 ①その授業時間内に教員のチェックを受け、正しいプログラムが提出できた。… 100点 ②次回の授業日前日17：00までに正しいプログラムを提出できた。… 80点 ③演習担当教員が定めるその期の提出締切日までに正しいプログラムが提出できた。… 60点 ④③の提出締切日までに正しいプログラムが提出できなかった。… 0点 ③の期間については、提出したプログラムに間違いがある場合は指摘する。完成するまで何度でも提出可とする。 中テスト、前期期末試験、学年末試験はH Rにて筆記試験を実施する。前期中テスト、前期期末試験、後期中テスト、学年末試験の評価割合は各10%である。小テスト4回の評価割合は各5%である。						
注意点							
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(0.5h), プログラミング入門 (第 1 学年) の復習(1.5h)	授業の進め方、年間計画を理解する。 昨年度授業の理解を確認する。			
		2週	アルゴリズムとフローチャート(2.0h)	プログラミングの手順 (アルゴリズム) を理解する。 フローチャートの作成方法を理解する。			
		3週	2 次元配列(2.0h)	2 次元配列を使用した簡単なプログラムを作成できる。			
		4週	繰返し処理 (for文) (2.0h) (2 次元配列との組み合わせ)	2 重のfor文の処理過程を理解し、使用できる。			
		5週	文字列(2.0h)	文字列を使用した簡単なプログラムを作成できる。			
		6週	文字配列(2.0h)	文字配列を使用した簡単なプログラムを作成できる。			
		7週	小テスト(0.5h), 応用プログラミング (その 1) (1.5h)	前期前半部分の理解を確認する。			
		8週	前期中テスト				
	2ndQ	9週	答案返却・解答解説(1.0h), 復習問題(1.0h)	間違った問題の正答を理解できる。 復習問題を理解し、正しいプログラムを作成できる。			
		10週	関数の定義と宣言(2.0h)	関数の意味を理解し、関数を自由に定義できる。			
		11週	関数を使ったプログラム(2.0h)	関数を使ったプログラムを作成できる。			
		12週	文字列関数を用いたプログラミング(2.0h)	文字列関数を使うことができる。 文字列関数を用いたプログラムを作成できる。			
		13週	数学関数を用いたプログラミング(2.0h)	数学関数を使うことができる。 数学関数を用いたプログラムを作成できる。			
		14週	小テスト(0.5h), 応用プログラミング (その 2) (1.5h)	これまでの学習内容を理解したうえで、応用的なプログラムを作成できる。			
		15週	前期期末試験				
		16週	答案返却・解答解説(1.0h), 復習問題(1.0h)	間違った問題の正答を理解できる。 復習問題を理解し、正しいプログラムを作成できる。			

後期	3rdQ	1週	配列とポインタ(2.0h)	配列とポインタを理解し、関数における変数の受け渡しができる。
		2週	関数における配列、ポインタの受け渡し(2.0h)	関数における配列、ポインタの受け渡しができる。
		3週	ファイル入出力（その1）(2.0h)	ファイルによる入力ができる。
		4週	ファイル入出力（その2）(2.0h)	ファイルによる出力ができる。
		5週	時間処理関数を用いたプログラミング(2.0h)	時間処理関数を使うことができる。 時間処理関数を用いたプログラムを作成できる。
		6週	前処理系(2.0h)	前処理指令によるプログラムの書き換えが理解できる。
		7週	小テスト(0.5h), 応用プログラミング（その3）(1.5h)	後期前半部分の理解を確認する。
		8週	後期中テスト	
	4thQ	9週	答案返却・解答解説(1.0h), 復習問題(1.0h)	間違った問題の正答を理解できる。 復習問題を理解し、正しいプログラムを作成できる。
		10週	構造体変数の定義(2.0h)	構造体変数を宣言し定義することができる。
		11週	構造体変数の初期化と代入(2.0h)	構造体変数の初期化と数値の代入を行うことができる。
		12週	構造体の配列とポインタ(2.0h)	構造体の配列を理解できる。 構造体へのポインタを理解し、使用できる。
		13週	プログラミング技法(2.0h)	単純なプログラムから構造体を用いた高度なプログラムを作成できる。
		14週	小テスト(0.5h), 応用プログラミング（その4）(1.5h)	これまでの学習内容を理解したうえで、応用的なプログラムを作成できる。
		15週	学年末試験	
		16週	答案返却・解答解説(1.0h), 復習問題(1.0h)	間違った問題の正答を理解できる。 復習問題を理解し、正しいプログラムを作成できる。

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
評価割合					
	試験	小テスト	成果品実技	合計	
総合評価割合	40	20	40	100	
基礎的能力	0	0	0	0	
専門的能力	40	20	40	100	
分野横断的能力	0	0	0	0	