

苫小牧工業高等専門学校		開講年度	令和06年度 (2024年度)		授業科目	プログラミングⅡ	
科目基礎情報							
科目番号	0008			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	創造工学科 (情報科学・工学系共通科目)			対象学年	3		
開設期	通年			週時間数	前期:2 後期:2		
教科書/教材	教科書:「Cプログラミング」株式会社インフォテックサーブ/参考書:中山 清喬 著「スッキリわかるC言語入門」インプレス, 柴田望洋著「明解C言語 入門編」ソフトバンククリエイティブ, 林晴比古著「新C言語入門ビギナー編」ソフトバンククリエイティブ, William H. Press, et al. 丹慶勝市, 他「ニューメリカルレシビ・イン・シー」, 他多数.						
担当教員	原田 恵雨						
到達目標							
1) 与えられたプログラムの読解, 要求仕様を満たすプログラムの作成ができる. 2) ポインタについて, 実装を通して理解できる. 3) 構造体・共用体・列挙型について, 実装を通して理解できる. 4) ファイル処理について, 実装を通して理解できる. 5) C以外の一つ以上のプログラミング言語について, 基本的な実装ができる.							
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	与えられたプログラムの読解, 要求仕様を満たすプログラムの作成ができる.		与えられたプログラムの読解, 要求仕様を満たすプログラムの基礎的な作成ができる.		与えられたプログラムの読解, 要求仕様を満たすプログラムの作成ができない.		
評価項目2	ポインタについて, 実装を通して理解できる.		ポインタについて, 実装を通して大部分が理解できる.		ポインタについて, 実装を通して理解できない.		
評価項目3	構造体・共用体・列挙型について, 実装を通して理解できる.		構造体・共用体・列挙型について, 実装を通して大部分が理解できる.		構造体・共用体・列挙型について, 実装を通して理解できない.		
評価項目4	ファイル処理について, 実装を通して理解できる.		ファイル処理について, 実装を通して大部分が理解できる.		ファイル処理について, 実装を通して理解できない.		
評価項目6	C以外の一つ以上のプログラミング言語について, 基本的な実装ができる.		C以外の一つ以上のプログラミング言語について, 基本的な理解ができる.		C以外の一つ以上のプログラミング言語について, 基本的な理解ができない.		
学科の到達目標項目との関係							
Ⅰ 人間性 Ⅰ 人間性 Ⅱ 実践性 Ⅱ 実践性 Ⅲ 国際性 Ⅲ 国際性 CP2 各系の工学的専門基礎知識, および実験・実習および演習・実技を通してその知識を社会実装に応用・実践できる力 5 CP2 各系の工学的専門基礎知識, および実験・実習および演習・実技を通してその知識を社会実装に応用・実践できる力 CP4 他者を理解・尊重し, 協働できるコミュニケーション能力と人間力 7 CP4 他者を理解・尊重し, 協働できるコミュニケーション能力と人間力 学習目標 Ⅱ 実践性 学校目標 D (工学基礎) 数学, 自然科学, 情報技術および工学の基礎知識と応用力を身につける 学科目標 D (工学基礎) 数学, 自然科学, 情報技術および工学の基礎知識と応用力を身につける 本科の点検項目 D-i 数学に関する基礎的な問題を解くことができる 本科の点検項目 D-iv 数学, 自然科学, 情報技術および工学の基礎知識を専門分野の工学的問題解決に応用できる 学校目標 E (継続的学習) 技術者としての自覚を持ち, 自主的, 継続的に学習できる能力を身につける 本科の点検項目 E-ii 工学知識, 技術の習得を通して, 継続的に学習することができる							
教育方法等							
概要	この授業では, 2年で学んできたプログラミングの基礎知識を要する. 第2学年で開講されている「プログラミングⅠ」は, 情報処理技術者としてソフトウェア開発を行うために必要なプログラミング技法を修得することが目的であり, 第3学年の「プログラミングⅡ」もそれを踏襲する. しかし, プログラミングⅡでは, より高度なプログラミング概念とその技法を修得する. 授業内容としては, ポインタ, 構造体, ファイル処理, 数値計算, C言語以外の言語, と多岐にわたる.						
授業の進め方・方法	授業は各単元において, 教科書と配布資料に基づき口頭説明した上, 各自プログラムを実装する. 適宜参考コードを示し, 各自が書いたプログラムと比較し, 理解を深める. その際, Unix環境を使用するため, Unixコマンドを使えるようにしておくこと. 関連文書はオンラインで閲覧可とする. 不定期に提出が必要な課題を課すが, 授業時間内に完成しない場合は自習時間を使って完成させること. また, 授業態度を測る指標のひとつとして, ノート (予習・授業・復習) を確認する. 最終評価が合格点に達しない場合, 再試験を行う可能性がある.						
注意点	プログラミング技術向上のためには, 日頃の努力が必要である. したがって, 不定期に出題される課題については, 提出の要・不要を問わず, 必ずすべて完成させるよう努力すること. また, オンライン教材を活用することも望ましい. 原則, 実習室で授業を行うものとするが, 場合によっては遠隔授業に対応する. なお, 自身のPC (オフライン) で開発環境を整えることを強く推奨する. 基本的にプログラムは一人で組むが, 相当考えても問題が解決しない場合は周囲の学生や教員と相談することが望ましい. 当たり前のことだが, そこで教わったことは理解してかつ他の問題にも適用できるようにする努力が必要である. 提出が必要な課題において, 不正コピーが見つかった場合は当該課題点を0にするなどのペナルティを与える.						
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, C開発環境の構築, ポインタ/ポインタを使った配列操作		ポインタについて, 実装を通して理解できる.		
		2週	ポインタ/ポインタを使った配列操作		ポインタについて, 実装を通して理解できる.		
		3週	ポインタ/ポインタを使った配列操作		ポインタについて, 実装を通して理解できる.		
		4週	ポインタと関数		ポインタについて, 実装を通して理解できる.		

		5週	問題演習	ポインタについて、実装を通して理解できる。
		6週	構造体	構造体・共用体・列挙型について、実装を通して理解できる。
		7週	構造体	構造体・共用体・列挙型について、実装を通して理解できる。
		8週	構造体・列挙型	構造体・共用体・列挙型について、実装を通して理解できる。
	2ndQ	9週	問題演習	与えられたプログラムの読解，要求仕様を満たすプログラムの作成ができる。
		10週	ファイル処理	ファイル処理について，実装を通して理解できる。
		11週	ファイル処理	ファイル処理について，実装を通して理解できる。
		12週	ファイル処理	ファイル処理について，実装を通して理解できる。
		13週	問題演習	与えられたプログラムの読解，要求仕様を満たすプログラムの作成ができる。
		14週	Java開発環境の構築	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		15週	Java開発環境の構築	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		16週	定期試験	
後期	3rdQ	1週	オブジェクト指向プログラミング，Java概要	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		2週	CとJavaとの文法の比較	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		3週	クラス，オブジェクト概要	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		4週	メソッド，コンストラクタ，オーバーロード，多態性	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		5週	メソッド，コンストラクタ，オーバーロード，多態性	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		6週	継承，インターフェース，抽象クラス	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		7週	継承，インターフェース，抽象クラス	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		8週	標準ライブラリ	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
	4thQ	9週	標準ライブラリ	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		10週	標準ライブラリ	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		11週	例外処理	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		12週	例外処理	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		13週	ファイル処理	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		14週	ジェネリクス	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		15週	ジェネリクス	C以外の一つ以上のプログラミング言語について，基本的な実装ができる。
		16週	定期試験	

評価割合				
	定期試験	ノート	課題	合計
総合評価割合	60	20	20	100
基礎的能力	30	5	10	45
専門的能力	30	10	10	50
分野横断的能力	0	5	0	5