

鈴鹿工業高等専門学校		開講年度	令和04年度 (2022年度)		授業科目	プログラミングⅡ	
科目基礎情報							
科目番号	0040		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	電子情報工学科		対象学年		2		
開設期	前期		週時間数		2		
教科書/教材	教科書：「やさしいC++第4版」高橋麻奈著（ソフトバンク）参考書：「C++実践プログラミング(第2版), Steave Ouline(著), 望月康司(監訳), O'REILLY, 「Effective C++(第2版)」, Scott Meyers (著), 吉川 邦夫(訳), アスキー, 「プログラミング言語C++第3版」, Bjarne Stroustrup (著), 長尾 高弘(訳), アスキー						
担当教員	青山 俊弘						
到達目標							
プログラミングの基本となる諸概念について理解し, C++によって, 関数, クラスを使った簡単なプログラムを作成することができる。							
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	各種制御文を用いて実践的なプログラムを書ける		各種制御文を用いて基本的なプログラムを書ける		各種制御文を用いて基本的なプログラムを書けない		
評価項目2	関数を用いて実践的なプログラムを書ける		関数を用いて基本的なプログラムを書ける		関数を用いて基本的なプログラムを書けない		
評価項目3	ポインタ, 参照を用いて実践的なプログラムを書ける		ポインタ, 参照を用いて基本的なプログラムを書ける		ポインタ, 参照を用いて基本的なプログラムを書けない		
評価項目4	基本的なクラスを用いて実践的なプログラムを書ける		基本的なクラスを用いて基本的なプログラムを書ける		基本的なクラスを用いて基本的なプログラムを書けない		
評価項目5	継承を用いて実践的なプログラムを書ける		継承を用いて基本的なプログラムを書ける		継承を用いて基本的なプログラムを書けない		
学科の到達目標項目との関係							
教育方法等							
概要	プログラミングⅡでは, プログラミングIに引き続き, C++言語の文法を会得し, C++言語を用いてプログラミングできる知識と技術を習得する。この授業ではC++言語のみではなく, プログラミング一般の方法やオブジェクト指向に関する知識についても学習する。また, 演習を通じてC++言語仕様以外に, 簡単な基本的なデータ構造やアルゴリズムについても学習する。						
授業の進め方・方法	全ての内容は, 学習・教育到達目標の（基礎）に関連する。 授業は講義、演習、実習をバランス良く行う。演習と実習は習熟度別に選択となる。 「授業計画」における各週の「到達目標」はこの授業で習得する「知識・能力」に相当するものとする						
注意点	<到達目標の評価方法と基準> 授業計画の「到達目標」1-13の確認を, これらの範囲を網羅した中間試験, 定期試験と, レポート、小テスト等で行う。評価結果が百点法で60点以上の場合に目標の達成とする。 <学業成績の評価方法および評価基準> 試験結果を70%, レポート、小テスト等を30%で評価する。 それぞれの期間ごと100点満点で評価し, これらの平均値を小数点以下切り捨てたものを最終評価とする。 試験のクラス中央値が65点未満の場合, 30点以上の取得した者に再試験を行う場合がある。 <あらかじめ要求される基礎知識の範囲> プログラミングI, 電子情報工学序論の知識が必要。また, 課題は数学や物理を参考に出題するので, これらの基本的な知識が必要。 <レポート等> 授業の理解を深めるためと, プログラム作成技術を向上させるため, 基本的に毎週, プログラミング作成のレポート課題を課す。遠隔授業の場合はWebブラウザでプログラムを入力できる環境が必要である。						
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	1年の復習1		1. 変数および, if, while, for, switchの各文法を使い, プログラムを作成できる。引数, 戻り値の概念を理解し, 値渡し, アドレス渡し, 参照渡しの違いについて理解し, 関数を使ってプログラムを作成できる。関数テンプレート, テンプレートクラス, 関数のオーバーロードについて理解する。		
		2週	アドレス, ポインタ		2. 変数がアドレスで指定されるメモリ領域を保持し, このアドレスとポインタとの関係を理解する		
		3週	配列とポインタ		3. 配列とポインタの関係を理解し, プログラムを作成できる		
		4週	スコープ, 記憶寿命		4. 型と変数, 記憶寿命とスコープの概念を理解し, プログラムを作成できる		
		5週	enum, typedef, 構造体		5. 列挙型, 構造体のデータ構造について理解し, プログラムを作成できる		
		6週	動的なメモリ確保		6. 動的メモリを確保する必要性を理解し, 動的メモリを使いプログラムを作成できる		
		7週	復習		これまでに学習した内容を説明し, プログラム作成に応用できる		
		8週	クラスとオブジェクト		7. クラスとオブジェクトについて説明できる		
	2ndQ	9週	クラスの定義, コンストラクタ		8. 簡単なクラスを作成し, プログラムを作成できる		
		10週	参照		9. 参照の概念を理解し, 説明でき, プログラムで利用できる		
		11週	継承		10. 継承の概念を理解し, 派生クラスを作成してプログラムを作成することができる		

		12週	仮想関数	11. 仮想関数の概念を理解し、派生クラスを作成してプログラムを作成することができる
		13週	抽象クラス	12. 純粹仮想関数、抽象クラス、仮想クラス、多重継承の概念を理解し、プログラムを作成できる
		14週	演算子関数	13. 演算子関数の概念を理解し、プログラムを作成できる
		15週	復習	これまでに学習した内容を説明できる
		16週		

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	
				プロシージャ(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	
				与えられた問題に対して、それを解決するためのソースプログラムを記述できる。	3	
				ソフトウェア生成に必要なツールを使い、ソースプログラムをロードモジュールに変換して実行できる。	3	
				与えられたソースプログラムを解析し、プログラムの動作を予測することができる。	3	
				要求仕様に従って、標準的な手法により実行効率を考慮したプログラムを設計できる。	3	
				要求仕様に従って、いずれかの手法により動作するプログラムを設計することができる。	3	
				要求仕様に従って、いずれかの手法により動作するプログラムを実装することができる。	3	
				要求仕様に従って、標準的な手法により実行効率を考慮したプログラムを実装できる。	3	

評価割合

	試験	課題	相互評価	態度	発表	その他	合計
総合評価割合	70	30	0	0	0	0	100
配点	70	30	0	0	0	0	100