

高知工業高等専門学校		開講年度	令和04年度 (2022年度)		授業科目	ソフトウェア工学	
科目基礎情報							
科目番号	I4034			科目区分	専門 / 必修		
授業形態	講義			単位の種別と単位数	学修単位: 2		
開設学科	SD 情報セキュリティコース			対象学年	4		
開設期	後期			週時間数	2		
教科書/教材	教科書は使わず、随時、授業プリントを配布する。						
担当教員	岡村 修司						
到達目標							
1. 標準的かつ実践的なプログラミング書法を理解し、その書法によってプログラミングをすることができる。 2. オブジェクト指向技術について理解し、オブジェクト指向にもとづいて、データ構造やアルゴリズムを設計することができる。 3. UMLを用いたソフトウェアシステムのモデリングについて理解し、ソフトウェアをモデリングできる。							
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		標準的かつ実践的なプログラミング書法を理解し、その書法によってプログラミングをすることができる。		標準的かつ実践的なプログラミング書法を理解し、説明できる。		標準的かつ実践的なプログラミング書法を理解できない。	
評価項目2		オブジェクト指向技術について理解し、オブジェクト指向にもとづいて、データ構造やアルゴリズムを設計することができる。		オブジェクト指向技術について理解し、説明できる。		オブジェクト指向技術について理解できない。	
評価項目3		UMLについて理解し、ソフトウェアをモデリングできる。		UMLについて理解し、説明できる。		UMLについて理解できない。	
学科の到達目標項目との関係							
学習・教育到達度目標 (C)							
教育方法等							
概要		プログラミング基礎、プログラミングⅠを通じてC言語による基本的なプログラミングについて学んだ。本授業では、これらの授業で得た知見を発展させ、特定の言語に依存しないソフトウェアシステム開発の方法論について学ぶ。まず、標準的かつ実践的なプログラミング書法を理解し、その書法に従ってプログラムを記述することを目指す。さらに、オブジェクト指向に基づいたデータ構造やアルゴリズムの設計を行うことを目指す。最後に、ソフトウェアシステムのモデリングを行うためにUMLについて学ぶ。					
授業の進め方・方法		基本的に演習形式で行う。スライドを用いて、学習内容の説明を行う。学習内容をまとめたプリントを配布するので、これを参照にプログラミングを行う。プログラミングを通じて、授業内容を理解する。授業はLinux環境で行う。プログラミング中に他の学生と相談してもよいが、各自主体的に取り組むことが重要である。レポートを作成することで、学習内容を確認し、技術文書の作成法を学ぶ。					
注意点		試験の成績70 %、平素の学習状況等（課題）を30 %とし、総合的に評価する。成績評価は中間と期末の各期間の評価の平均とする。事前学習として、配布するプリントを読み、授業内容を理解しておくこと。授業を受けても理解できない内容は、質問するなどして、理解するための努力を行うこと。作業中は他の学生と相談してもよいが、各自主体的に取り組むことが重要である。事後学習として、授業で学んだ内容をレポートにまとめ、指定された期日までに提出すること。 プログラミング基礎、プログラミングⅠ、プログラミングⅡおよびアルゴリズムとデータ構造の内容をよく理解しておくこと。					
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		
後期	3rdQ	1週	ガイダンス ソフトウェア工学とは		授業の進め方、評価法などについて理解する。ソフトウェア工学の概要について理解する。		
		2週	プログラミング書法（1）		標準的かつ実践的なプログラミング書法を理解し、これにしたがったコーディングが行える。		
		3週	プログラミング書法（2）		標準的かつ実践的なプログラミング書法を理解し、これにしたがったコーディングが行える。		
		4週	プログラミング書法（3）		標準的かつ実践的なプログラミング書法を理解し、これにしたがったコーディングが行える。		
		5週	プログラミング書法（4）		標準的かつ実践的なプログラミング書法を理解し、これにしたがったコーディングが行える。		
		6週	プログラミング書法（5）		標準的かつ実践的なプログラミング書法を理解し、これにしたがったコーディングが行える。		
		7週	オブジェクト指向技術の概要（1）		オブジェクト指向の基本的な概念（オブジェクト、クラス、情報隠蔽、データ抽象化、動的束縛、継承など）について理解し、基本的なプログラムを記述することができる。		
		8週	オブジェクト指向技術の概要（2）		オブジェクト指向の基本的な概念（オブジェクト、クラス、情報隠蔽、データ抽象化、動的束縛、継承など）について理解し、基本的なプログラムを記述することができる。		
	4thQ	9週	オブジェクト指向技術の概要（3）		オブジェクト指向の基本的な概念（オブジェクト、クラス、情報隠蔽、データ抽象化、動的束縛、継承など）について理解し、基本的なプログラムを記述することができる。		
		10週	オブジェクト指向技術の概要（4）		オブジェクト指向の基本的な概念（オブジェクト、クラス、情報隠蔽、データ抽象化、動的束縛、継承など）について理解し、基本的なプログラムを記述することができる。		

		11週	オブジェクト指向技術の概要（５）	オブジェクト指向の基本的な概念（オブジェクト、クラス、情報隠蔽、データ抽象化、動的束縛、継承など）について理解し、基本的なプログラムを記述することができる。
		12週	オブジェクト指向技術の概要（６）	オブジェクト指向の基本的な概念（オブジェクト、クラス、情報隠蔽、データ抽象化、動的束縛、継承など）について理解し、基本的なプログラムを記述することができる。
		13週	UML（１）	ソフトウェアシステムの設計、実装、配置を行う場合の、UMLの利用法について理解する。
		14週	UML（２）	ソフトウェアシステムの設計、実装、配置を行う場合の、UMLの利用法について理解する。
		15週	アルゴリズム解析	プログラムが消費する資源を見積もるための手法を理解する。
		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	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。	4	
			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。	4	
			変数の概念を説明できる。	4	
			データ型の概念を説明できる。	4	
			制御構造の概念を理解し、条件分岐を記述できる。	4	
			制御構造の概念を理解し、反復処理を記述できる。	4	
			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。	4	
			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。	4	
			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。	4	
			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	
			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。	4	
			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。	4	
			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。	4	
			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。	4	
		ソフトウェア	アルゴリズムの概念を説明できる。	4	
			与えられたアルゴリズムが問題を解決していく過程を説明できる。	4	
			同一の問題に対し、それを解決できる複数のアルゴリズムが存在しうることを説明できる。	4	
			整列、探索など、基本的なアルゴリズムについて説明できる。	4	
			時間計算量によってアルゴリズムを比較・評価できることを説明できる。	4	
			領域計算量などによってアルゴリズムを比較・評価できることを説明できる。	4	
			コンピュータ内部でデータを表現する方法(データ構造)にはバリエーションがあることを説明できる。	4	
			同一の問題に対し、選択したデータ構造によってアルゴリズムが変化しうることを説明できる。	4	
			リスト構造、スタック、キュー、木構造などの基本的なデータ構造の概念と操作を説明できる。	4	
			リスト構造、スタック、キュー、木構造などの基本的なデータ構造を実装することができる。	4	
			ソフトウェアを中心としたシステム開発のプロセスを説明できる。	4	
			ソースプログラムを解析することにより、計算量等のさまざまな観点から評価できる。	4	
			同じ問題を解決する複数のプログラムを計算量等の観点から比較できる。	4	
		コンピュータシステム	ネットワークコンピューティングや組込みシステムなど、実用に供せられているコンピュータシステムの利用形態について説明できる。	4	
			デュアルシステムやマルチプロセッサシステムなど、コンピュータシステムの信頼性や機能を向上させるための代表的なシステム構成について説明できる。	4	

			集中処理システムについて、それぞれの特徴と代表的な例を説明できる。	4	
			分散処理システムについて、特徴と代表的な例を説明できる。	4	
			システム設計には、要求される機能をハードウェアとソフトウェアでどのように実現するかなどの要求の振り分けやシステム構成の決定が含まれることを説明できる。	3	
			ユーザの要求に従ってシステム設計を行うプロセスを説明することができる。	3	
			プロジェクト管理の必要性について説明できる。	3	
			WBSやPERT図など、プロジェクト管理手法の少なくとも一つについて説明できる。	3	
			ER図やDFD、待ち行列モデルなど、ビジネスフロー分析手法の少なくとも一つについて説明できる。	3	
評価割合					
		試験	課題	合計	
総合評価割合		70	30	100	
基礎的能力		40	10	50	
専門的能力		30	20	50	