

熊本高等専門学校		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	HI1513	科目区分	専門 / 選択		
授業形態	授業	単位の種別と単位数	学修単位: 2		
開設学科	人間情報システム工学科	対象学年	5		
開設期	通年	週時間数	1		
教科書/教材	なし				
担当教員	神崎 雄一郎				
到達目標					
オブジェクト指向にもとづいてソフトウェアの設計・実装・テストを行う方法の基礎を理解し、それを生かして、与えられたテーマに沿ったソフトウェアを開発できる。					
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		
ソフトウェア開発プロセス	ソフトウェア開発プロセスについて理解し、最近の動向を含めて説明できる。	よく知られたソフトウェア開発プロセスについて理解し、説明できる。	ソフトウェア開発プロセスについて説明できない。		
UMLを用いた分析・設計	ユースケース図、クラス図などの記述ルールについて理解し、自ら考えた要求仕様や設計をダイアグラムで表現できる。	ユースケース図、クラス図などの記述ルールについて理解し、与えられた課題をダイアグラムで記述できる。	ユースケース図、クラス図などを記述できない。		
オブジェクト指向プログラミング	「標準的な到達レベルの目安」に加え、創造力を生かした応用課題を解くことができる。	クラス、オブジェクト、継承、カプセル化、抽象メソッド、多態性といったオブジェクト指向プログラミングの基本項目やスレッドについて理解し、これらを生かしたプログラムをJavaで実装することができる。	オブジェクト指向プログラミングに関する基礎的な課題を解くことができない。		
ソフトウェアテスト	「標準的な到達レベルの目安」に加え、創造力を生かした応用課題を解くことができる。	ソフトウェアテストの技法について理解し、Java言語を対象にした単体テストを実施できる。	Java言語を対象にした単体テストを実施できない。		
学科の到達目標項目との関係					
教育方法等					
概要	ソフトウェア開発に関する幅広い技術の基礎を身に付けることを目的とした講義である。オブジェクト指向を基盤としたソフトウェアの分析、設計、実装、テストの技法などの学習を通して、ソフトウェアを効率的に開発するための方法を体得することを旨とする。				
授業の進め方・方法	本科目は、コンピュータを用いた演習を中心に進められる。評価基準は、一年を4つに分けた学期別に次のとおりとする。 ・ 前期Iおよび後期I：演習課題60点、実装テスト30点、および、Webクイズ・練習問題10点 ・ 前期IIおよび後期II：演習課題90点、Webクイズ・練習問題10点 総合評価(各学期の平均)が60点以上の場合、目標達成とみなす。演習課題の点数は、ソフトウェア開発演習において提出された成果物の内容によって決められる。 実装テストは、身に付いた実装能力を評価するためのテストである。また、Webクイズは、理解度を確認するためのWeb上でのクイズ(小テスト)である。				
注意点	4年次の選択科目「オブジェクト指向プログラミング」を履修済みであることが望ましい。 本科目は2単位の科目であり、規定授業時数は60時間である。レポート作成などで年間に30時間の自主的な学習が求められる。				
授業の属性・履修上の区分					
<input type="checkbox"/> アクティブラーニング		<input checked="" type="checkbox"/> ICT 利用		<input type="checkbox"/> 遠隔授業対応	
<input type="checkbox"/> 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	ソフトウェア工学概説	ソフトウェア工学の役割や、関連技術のアウトラインについて説明できる。	
		2週	オブジェクト指向プログラミングの基礎 (1)	クラス、オブジェクト、継承、カプセル化といったオブジェクト指向プログラミングの基本項目について理解し、関連する実装課題を解くことができる。	
		3週	オブジェクト指向プログラミングの基礎 (2)	同上	
		4週	オブジェクト指向プログラミングの基礎 (3)	同上	
		5週	抽象クラス・インターフェイス (1)	抽象メソッド、抽象クラス、(Javaの)インターフェイスの意義について理解し、関連する実装課題を解くことができる。	
		6週	抽象クラス・インターフェイス (2)	同上	
		7週	抽象クラス・インターフェイス (3)	同上	
		8週	マルチスレッド (1)	マルチスレッドについて理解し、マルチスレッドを用いた簡単なアプリケーションを実装できる。	
	2ndQ	9週	マルチスレッド (2)	同上	
		10週	定期実装テスト		
		11週	ソフトウェアテスト(1)	ソフトウェアテストの技法について理解し、Java言語を対象にした単体テストが実施できる。	
		12週	ソフトウェアテスト(2)	同上	
		13週	ソフトウェアテスト(3)	同上	
		14週	ソフトウェアテスト(4)	同上	

		15週	レビュー	演習課題や定期実装テストについてのレビュー（クラス内での議論と解説）を通して、関連する話題についての理解を深めることができる。
		16週		
後期	3rdQ	1週	ソフトウェア開発プロセス(1)	ソフトウェア開発の各工程の目的や代表的なソフトウェア開発プロセスモデルについて説明できる。
		2週	ソフトウェア開発プロセス(2)	同上
		3週	分析・設計(1)	オブジェクト指向にもとづくソフトウェアの分析・設計の技法について理解し、UMLによって簡単なソフトウェアの分析・設計を行うことができる。
		4週	分析・設計(2)	同上
		5週	分析・設計(3)	同上
		6週	分析・設計(4)	同上
		7週	ソフトウェア開発総合演習(1)	与えられたテーマに沿ったソフトウェアの開発（設計と実装）を行うことができる。
		8週	ソフトウェア開発総合演習(2)	同上
	4thQ	9週	定期実装テスト	同上
		10週	ソフトウェア開発総合演習(3)	同上
		11週	ソフトウェア開発総合演習(4)	同上
		12週	ソフトウェア開発総合演習(5)	同上
		13週	ソフトウェア開発総合演習(6)	同上
		14週	ソフトウェア開発総合演習(7)	同上
		15週	レビュー	演習課題や定期実装テストについてのレビュー（クラス内での議論と解説）を通して、関連する話題についての理解を深めることができる。
		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
評価割合					
		演習課題	定期実装テスト	クイズ・練習問題	合計
総合評価割合		75	15	10	100
専門的能力		75	15	10	100