一関工業高等専門学校	開講年度	平成31年度 (2019年度)	授業科目	ソフトウェア工学
科目基礎情報				
科目番号	0058	科目区分	専門 / 選択	
授業形態	講義	単位の種別と単位数	履修単位: 2	
開設学科	電気情報工学科	対象学年	4	
開設期	通年	週時間数	2	
教科書/教材	教科書: 松本啓之亮, "ソフトウェア工学", 森北出版 / 教材: オリジナルテキスト (当該科目moodle上に掲載)			
担当教員	秋田 敏宏			

到達目標

- ① ソフトウェアの開発における問題点から, ソフトウェア工学の重要性を理解できる。
- ② ソフトウェア開発プロセスを理解し, 要求分析からテストに至る流れとその内容を理解できる。
- ③ オブジェクト指向の概要を理解し, UMLによるモデリングができる。
- ④ 複数人による共同開発を通じて, ソフトウェア開発における協調性とモデリングの重要性を理解できる。

【教育目標】 C, D
【学習・教育到達目標】 C-2, D-1

【キーワード】 開発プロセス, 要求分析, オブジェクト指向, UML, ソフトウェア設計, 実装, ソフトウェアテスト, 品質特性

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
ソフトウェアの開発における問題点から, ソフトウェア工学の重要性を理解できる。	ソフトウェア工学という学問が誕生した背景とソフトウェア工学の重要性を詳細に説明できる。	ソフトウェア工学という学問が誕生した背景とソフトウェア工学の重要性をおおよそ説明できる。	ソフトウェア工学という学問が誕生した背景とソフトウェア工学の重要性を説明できない。
ソフトウェア開発プロセスを理解し, 要求分析からテストに至る流れとその内容を理解できる。	V字モデルについて, 各プロセスを詳細に説明できる。	V字モデルについて, 各プロセスをおおよそ説明できる。	V字モデルについて, 各プロセスを説明できない。
オブジェクト指向の概要を理解し, UMLによるモデリングができる。	オブジェクト指向の考え方を理解しており, ソフトウェアの機能・構造・振る舞いをUMLにより表現できることを理解し, 複数の図で一貫性のとれたモデリングができる。	オブジェクト指向の考え方をだいたい理解しており, ソフトウェアの機能・構造・振る舞いをUMLにより表現できることを理解し, 基本的なモデリングができる。	オブジェクト指向の考え方を理解できていない。または, ソフトウェアの機能・構造・振る舞いをUMLにより表現できることを理解できていない。
複数人による共同開発を通じて, ソフトウェア開発における協調性とモデリングの重要性を理解できる。	ソフトウェア開発における共同作業を十分に行うことができ, モデリングを適用した開発ができる。	ソフトウェア開発における共同作業を行うことができる。	ソフトウェア開発における共同作業を行うことができない。

学科の到達目標項目との関係

教育方法等

概要	ソフトウェア開発論と開発プロセスという観点から, 必要となる知識を学習することで, 年々肥大化・複雑化するソフトウェアへの要求に適用する要素を身につけることが目的です。また, 近年のソフトウェア開発では, 複数人による共同作業も行われていることから, 演習を通じて実践的なソフトウェア開発手法を理解することが目的です。
授業の進め方・方法	<ul style="list-style-type: none"> ・ 授業資料はmoodle上の本科目のサイトよりダウンロードして事前にその内容を読んでおくこと。その授業資料を中心に授業を進めます。なお, 授業内容に関連した課題を与える。 ・ 授業開始前までに各自PCを起動しておくこと。 ・ プログラミングでは, C言語を使用します。組込みソフトウェア開発演習では, LEGO Mindstormsを使用します。
注意点	<p>課題や演習では, C言語によるプログラミングがあります。C言語のプログラミングについて十分に復習しておくこと。</p> <p>【事前学習】 授業内容を確認し, 教科書を一読しておくこと。その際, 教科書の太字で書かれている重要語句は和名・英語名・意味を含めて確認しておくこと。C言語によるプログラミングについても事前に構文などを確認しておくこと。</p> <p>【評価方法・評価基準】 試験 (60%), 課題 (40%) で評価する。詳細については, 第1回目の授業で告知する。ソフトウェア工学に関する知識の理解の程度を評価する。また, 課題はソフトウェア設計に関する課題とソフトウェア開発演習による課題により評価する。総合成績60点以上を単位修得とする。</p>

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要	ソフトウェア工学の概要を理解できる。
		2週	ソフトウェア工学概要	プロセスについて理解できる。
		3週	ソフトウェア要求分析	データフロー図を理解できる。
		4週	ソフトウェア要求分析	状態遷移図を理解できる。
		5週	ソフトウェア要求分析	ER図を理解できる。
		6週	ソフトウェア要求分析・課題演習	演習課題を通じて要求分析について理解できる。
		7週	ソフトウェア要求分析・課題演習	演習課題を通じて, 要求分析について理解できる。
		8週	中間試験	
	2ndQ	9週	オブジェクト指向開発	オブジェクト指向の概要とモデリングについて理解できる。
		10週	オブジェクト指向開発	機能を表す図を理解でき, ツールにより作図できる。
		11週	オブジェクト指向開発	構造を表す図を理解でき, ツールにより作図できる。
		12週	オブジェクト指向開発	振る舞いを表す図を理解でき, ツールにより作図できる。
		13週	オブジェクト指向開発	複数の図の関係を理解できる。
		14週	オブジェクト指向開発・課題演習	演習課題を通じて, モデリングについて理解できる。

後期		15週	期末試験	
		16週	前期のまとめ	これまでの内容を再確認し、ソフトウェア工学の理解を深めることができる。
	3rdQ	1週	ソフトウェア設計	外部設計について理解できる。
		2週	ソフトウェア設計	内部設計について理解できる。
		3週	ソフトウェアテストと品質	ソフトウェア品質について理解できる。
		4週	ソフトウェアテストと品質	テストケースについて理解できる。
		5週	組込みソフトウェア開発演習：導入	開発環境の使い方を理解できる。
		6週	組込みソフトウェア開発演習：要素技術	アクチュエータに関するプログラムの基本を理解できる。
		7週	組込みソフトウェア開発演習：要素技術	センサに関するプログラムの基本を理解できる。
		8週	中間試験	
	4thQ	9週	組込みソフトウェア開発演習	課題に対する要求分析ができる。
		10週	組込みソフトウェア開発演習	課題に対する外部設計ができる。
		11週	組込みソフトウェア開発演習	課題に対する内部設計ができる。
		12週	組込みソフトウェア開発演習	演習課題に対する実装・テストができる。
		13週	組込みソフトウェア開発演習	演習課題に対する実装・テストができる。
		14週	組込みソフトウェア開発演習	演習課題に対する実装・テストができる。
15週		成果報告会とまとめ	製作物のプレゼンテーションができ、ソフトウェア開発のプロセスを確認できる。	
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	プログラミング	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		
			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		
		コンピュータシステム	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。	4		
			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。	4		
			プロジェクト管理の必要性について説明できる。	4		
	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。		4			
	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。		4			
	分野別の工学実験・実習能力	情報系分野【実験・実習能力】	情報系【実験・実習】	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、標準的な開発ツールや開発環境を利用して記述できる。	4	
				ソフトウェア生成に利用される標準的なツールや環境を使い、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	
				ソフトウェア開発の現場において標準的とされるツールを使い、生成したロードモジュールの動作を確認できる。	4	
				フローチャートなどを用いて、作成するプログラムの設計図を作成することができる。	4	
				問題を解決するために、与えられたアルゴリズムを用いてソースプログラムを記述し、得られた実行結果を確認できる。	4	
				要求仕様に従って標準的な手法によりプログラムを設計し、適切な実行結果を得ることができる。	4	

評価割合

	試験	課題	合計
総合評価割合	60	40	100
基礎的能力	0	0	0
専門的能力	60	40	100
分野横断的能力	0	0	0