

香川高等専門学校		開講年度	令和03年度 (2021年度)		授業科目	ソフトウェア特別実習 I	
科目基礎情報							
科目番号	211454			科目区分	専門 / 選択		
授業形態	講義・実験・実習			単位の種別と単位数	履修単位: 4		
開設学科	建設環境工学科（2019年度以降入学者）			対象学年	3		
開設期	通年			週時間数	4		
教科書/教材	指導教員より資料を配布する						
担当教員	未 定						
到達目標							
1. 課題に沿ったソフトウェアを開発することができる。 2. ソフトウェアの開発において創造性のある作品を提案し制作することができる。 3. 課題に対する取り組み状況について報告書を作成して担当教員に報告できる。							
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	
ソフトウェア開発力	課題（コンテスト）に対して完成度の高いソフトウェアを開発することができる。			課題（コンテスト）に沿ったソフトウェアを開発することができる。		課題（コンテスト）に沿ったソフトウェアを開発することができない。	
創造性	ソフトウェアの開発において創造性のある作品を提案し、制作することができる。			ソフトウェアの開発において創造性のある作品を提案することができる。		ソフトウェアの開発において創造性のある作品を提案することができない。	
表現力	課題に対する取り組み状況について報告書を作成して担当教員に報告できる。作品をわかりやすく説明できる。			課題に対する取り組み状況について報告書を作成して担当教員に報告できる。		課題に対する取り組み状況について報告書を作成して担当教員に報告することができない。	
学科の到達目標項目との関係							
教育方法等							
概要	将来ソフトウェア技術者としての活躍を強く希望し、十分な基礎学力とプログラミングの素養がある学生に対して、ソフトウェア技術に関するプロジェクト型の実習科目を設け、高度なプログラミング技術の修得機会を提供する。ソフトウェアの開発力を競う各種コンテストへの参加を通じて実践的なプログラミング力の育成を図る。						
授業の進め方・方法	実習で取り組む課題（参加コンテスト等）は担当教員と学生で相談のうえ決定する。実習は放課後、夏休み期間中など、授業が行われていない時間に取り組む。実習課題の取り組み状況は、実施報告書により担当教員に報告する。						
注意点	本科目はソフトウェア特化コースの学生のみ受講できる。 ソフトウェア特化コースは、学科長の推薦を必要とする。 推薦基準は以下の通りとし、（１）～（３）を全て満たすこと。 （１）本人の希望 将来、高度ソフトウェア技術者として活躍することを強く希望しており、ソフトウェア特化コースへの配属を希望していること。 （２）十分な基礎学力、素養 卒業後に高度ソフトウェア技術者として活躍することが期待できる十分な基礎学力、素養を有すること。以下の実績を目安とする。 (a)情報オリンピック予選に参加して優秀な成績を修め、本選に招待選手として出場した。（１または２年次） (b)パソコン甲子園の「プログラミング部門」または「モバイル部門」に参加し、予選を突破し全国大会に出場した。（１～３年次） ただし、チーム内でプログラミング作成において中心的な役割を果たした場合に限る。 (c)プログラミング技術を競う全国規模のコンテストで上記(a)、(b)と同等以上の成果をあげた。ただし、チームで参加している場合は、チーム内でプログラミング作成において中心的な役割を果たした場合に限る。 (d)プログラミングに関する専門科目において、極めて特別優秀な成績を修めた。 （３）生活態度、その他 出席状況が良好で、放課後に実施する本プロジェクトの実習に参加が可能であり、主として休日に実施される各種コンテストに出場が可能であること。 受講手続き 受講を希望するものは、４月２週目までに推薦基準を満たしていることを証明する書類等を持参して、学科長に申し出ること。 なお、学科からの推薦により履修する場合は、この限りではない。						
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. ガイダンス 2. 実習のテーマ選定		実習で取り組む課題（テーマ）を選定する。		
		2週	2. 実習のテーマの選定		実習で取り組む課題（テーマ）を選定する。		
		3週	3. ソフトウェアの設計・開発		課題に沿ったソフトウェアの設計を行い、開発を行うことができる。		
		4週	3. ソフトウェアの設計・開発		課題に沿ったソフトウェアの設計を行い、開発を行うことができる。		
		5週	3. ソフトウェアの設計・開発		課題に沿ったソフトウェアの設計を行い、開発を行うことができる。		
		6週	3. ソフトウェアの設計・開発		課題に沿ったソフトウェアの設計を行い、開発を行うことができる。		
		7週	3. ソフトウェアの設計・開発		課題に沿ったソフトウェアの設計を行い、開発を行うことができる。		
		8週	3. ソフトウェアの設計・開発		課題に沿ったソフトウェアの設計を行い、開発を行うことができる。		

後期	2ndQ	9週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		10週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		11週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		12週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		13週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		14週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		15週	3. ソフトウェアの設計・開発	課題に沿ったソフトウェアの設計を行い、開発を行うことができる。
		16週		
	3rdQ	1週	4. ソフトウェアの開発・試験	ソフトウェアの開発と試験を行うことができる。
		2週	4. プログラムの開発・試験	ソフトウェアの開発と試験を行うことができる。
		3週	4. プログラムの開発・試験	ソフトウェアの開発と試験を行うことができる。
		4週	4. プログラムの開発・試験	ソフトウェアの開発と試験を行うことができる。
		5週	4. プログラムの開発・試験	ソフトウェアの開発と試験を行うことができる。
		6週	5. 作品発表の準備	ソフトウェアの作品発表の準備を行うことができる。
		7週	5. 作品発表の準備	ソフトウェアの作品発表の準備を行うことができる。
		8週	5. 作品発表の準備	ソフトウェアの作品発表の準備を行うことができる。
	4thQ	9週	5. 作品発表の準備	ソフトウェアの作品発表の準備を行うことができる。
		10週	5. 作品発表の準備	ソフトウェアの作品発表の準備を行うことができる。
		11週	6. 作品発表 (コンテストへの参加など)	ソフトウェアの作品発表を行うことができる。
		12週	6. 作品発表 (コンテストへの参加など)	ソフトウェアの作品発表を行うことができる。
		13週	6. 作品発表 (コンテストへの参加など)	ソフトウェアの作品発表を行うことができる。
		14週	6. 作品発表 (コンテストへの参加など)	ソフトウェアの作品発表を行うことができる。
		15週	6. 作品発表 (コンテストへの参加など)	ソフトウェアの作品発表を行うことができる。
		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
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。	1	
				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。	1	
				変数の概念を説明できる。	1	
				データ型の概念を説明できる。	1	
				制御構造の概念を理解し、条件分岐を記述できる。	1	
				制御構造の概念を理解し、反復処理を記述できる。	1	
				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。	1	
				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。	1	
				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。	1	
				主要な言語処理プロセッサの種類と特徴を説明できる。	1	
				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。	1	
				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。	1	
				主要な計算モデルを説明できる。	1	
				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。	1	
				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。	1	
				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。	1	
				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。	1	
			ソフトウェア	アルゴリズムの概念を説明できる。	1	
				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。	1	
				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。	1	
				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。	1	

				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。	1	
				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。	1	
				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。	1	
				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することを説明できる。	1	
				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。	1	
				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。	1	
				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。	1	
				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。	1	
				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。	1	
			システムプログラム	コンピュータシステムにおけるオペレーティングシステムの位置づけを説明できる。	1	
				プロセス管理やスケジューリングなどCPUの仮想化について説明できる。	1	
				排他制御の基本的な考え方について説明できる。	1	
				記憶管理の基本的な考え方について説明できる。	1	
				形式言語の概念について説明できる。	1	
				オートマトンの概念について説明できる。	1	
				コンパイラの役割と仕組みについて説明できる。	1	
				形式言語が制限の多さにしたがって分類されることを説明できる。	1	
				正規表現と有限オートマトンの関係を説明できる。	1	

評価割合

	プログラム	報告書	合計
総合評価割合	80	20	100
ソフトウェア開発力	70	0	70
創造性	10	10	20
表現力	0	10	10