

熊本高等専門学校		開講年度	平成31年度 (2019年度)		授業科目	プログラミングII	
科目基礎情報							
科目番号	HI1305			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: 4		
開設学科	人間情報システム工学科			対象学年	3		
開設期	通年			週時間数	4		
教科書/教材	①新訂・新C言語入門 シニア編, 林晴比古, ソフトバンククリエイティブ社, ②詳細! Python 3 入門ノート, 大重美幸, ソーテック社						
担当教員	島川 学,大隈 千春						
到達目標							
プログラミング I で学習したC言語をさらに習熟することができ、さらにPython言語の基礎を習得することができる。 C言語については構造化プログラミング技法に心がけ、可読しやすく効率的なプログラムを書くことができる。 ライブラリ関数の挙動や制御構造の概念を理解し、与えられた問題を解決する方法を設計してプログラムで実現できる。 基本的なアルゴリズムとして探索や整列に関するアルゴリズムについても学び、効率も考慮したプログラムの作成に配慮できる。 Python言語についてはC言語で学習した内容を踏まえて、新しいプログラミング言語を習得し、データ処理等に必要となる基本手法を活用できる。							
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	
ファイル処理	テキストファイル、バイナリファイルを利用した、実用的なプログラムを作成できる。			テキストファイル、バイナリファイルに対して適切な入出力プログラムを作成できる。		テキストファイル、バイナリファイルを用いたプログラムを理解できない。	
構造体	構造体の宣言や変数の種々の宣言方法、それぞれの初期化や入出力、メンバの参照について理解してプログラムに応用できる。			構造体の宣言や構造体型の変数の基本的な宣言方法、構造体の初期化と入出力とメンバの参照について理解できる。		構造体の宣言や構造体型の変数の基本的な宣言方法、構造体の初期化と入出力とメンバの参照について理解できない。	
自己参照構造体	自己参照構造体の意味を理解できる。静的チェーンや動的チェーンを生成操作した応用プログラムを作成できる。			自己参照構造体の構造を理解でき、プログラムを作成できる。		自己参照構造体の構造を理解できない。	
Python言語の基礎	Python言語での数値や変数の取り扱いや制御構造などの基礎を理解し、基本的なプログラムを作成できる。さらに、Python言語の特徴を活かした効率的なプログラム記述ができる。			Python言語での数値や変数の取り扱いや制御構造などの基礎を理解し、基本的なプログラムを作成できる。		Python言語の基礎が理解できず、プログラムを作成できない。	
Python言語の活用	Python言語でテキストファイルの入出力、グラフ描画、数値データ処理ができる。さらに、Python言語の特徴を活かした効率的なプログラム記述ができる。			Python言語でテキストファイルの入出力、グラフ描画、数値データ処理ができる。		Python言語でテキストファイルの入出力、グラフ描画、数値データ処理ができない。	
学科の到達目標項目との関係							
教育方法等							
概要	前期は2年次「プログラミング I」で学習したC言語についてさらに習熟する。ファイル操作や構造体などのデータ構造の仕組みを理解し、構造的プログラミング技法の習得にも心がけ、よりよいプログラムが書けるようになる。 後期はPython言語について学習する。新しいプログラミング言語を学習することで、言語処理系の違いを理解するとともに、プログラミング言語に依存しない論理的な処理手順やアルゴリズム、データ処理等に必要となる基本手法を学ぶ。						
授業の進め方・方法	本科目は、2年生「プログラミング I」に続く科目である。前期は、プログラミング言語をC言語とし、2年次に学習したこととの定着を図りながら、ファイルの取り扱い、自己参照構造体などを学ぶ。プログラミングは主にUNIX環境で行う。後期は、プログラミング言語をPython言語とし、基本データや変数、制御構造などの基本から学習する。プログラミングは主にWindows環境で行う。 本科目では講義だけでなく、演習を通じて実際のプログラムの挙動を確認しながら内容理解を深める。						
注意点	本科目は4単位であり規定授業時数は120時間である。 成績評価は、筆記試験、プログラミング実技、および演習課題レポートによって行う。 授業内で行う演習については、期限内に報告書を提出する。						
授業計画							
		週	授業内容			週ごとの到達目標	
前期	1stQ	1週	ファイル処理 (テキストファイル)			テキストファイルに対する入出力を含むプログラムを書ける	
		2週	コマンドライン引数			コマンドライン引数を利用したプログラムを作成できる。	
		3週	プログラミング演習(1)			テキストファイルを処理するプログラムを作成できる。	
		4週	プログラミング演習(1)			前週に同じ	
		5週	ファイル処理 (バイナリファイル)			バイナリファイルに対する入出力を含むプログラムを書ける	
		6週	プログラミング演習(2)			バイナリファイルを処理するプログラムを作成できる。	
		7週	プログラミング演習(2)			前週に同じ	
		8週	中間試験				
	2ndQ	9週	中間試験の答案返却・解説				
		10週	構造体			構造体の構造や性質を理解できる。	
		11週	プログラミング演習(3)			構造体を用いたプログラムを作成できる。	
		12週	自己参照構造体			自己参照構造体を利用した、線形リストを構成することができる。	

後期		13週	プログラミング演習(4)	自己参照構造体を用いて、スタック構造を持つプログラムを作成できる。
		14週	プログラミング演習(4)	前週に同じ
		15週	定期試験	
		16週	定期試験の答案返却・解説	
	3rdQ	1週	Pythonプログラムの基本	基本データ、変数、基本分布を理解できる。
		2週	標準ライブラリ	標準ライブラリを利用できる。
		3週	制御構造	制御構造を理解し、プログラムを記述できる。
		4週	リスト、タプル、集合	リスト、タプル、集合の概念を理解し、基本的なプログラムを記述できる。
		5週	ユーザー定義関数(1)	ユーザー定義関数を記述し、活用できる。
		6週	ユーザー定義関数(2)	前週に同じ
		7週	中間試験	
		8週	中間試験の答案返却・解説	
	4thQ	9週	テキストファイル入出力(1)	テキストファイル入出力ができる。
		10週	テキストファイル入出力(2)	前週に同じ
		11週	グラフ描画(1)	グラフ描画ができる。
		12週	グラフ描画(2)	前週に同じ
		13週	NumPyの活用(1)	NumPyを活用できる。
		14週	NumPyの活用(2)	前週に同じ
		15週	定期試験	
		16週	定期試験の答案返却・解説	

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
基礎的能力	工学基礎	情報リテラシー	情報リテラシー	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知っている。	3	
				与えられた基本的な問題を解くための適切なアルゴリズムを構築することができる。	3	
				任意のプログラミング言語を用いて、構築したアルゴリズムを実装できる。	3	
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	
				主要な言語処理プロセッサの種類と特徴を説明できる。	3	
				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。	3	
				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。	3	
				主要な計算モデルを説明できる。	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	
			ソフトウェア	アルゴリズムの概念を説明できる。	3	
				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。	3	
				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。	3	
				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。	3	
				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	
			システムプログラム	形式言語の概念について説明できる。	4	

				オートマトンの概念について説明できる。	4	
				コンパイラの役割と仕組みについて説明できる。	4	
				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。	4	
				正規表現と有限オートマトンの関係を説明できる。	4	
評価割合						
			筆記試験	実技試験	演習	合計
総合評価割合			30	30	40	100
基礎的能力			20	20	30	70
専門的能力			10	10	10	30