

茨城工業高等専門学校		開講年度	平成30年度 (2018年度)		授業科目	プログラミング応用	
科目基礎情報							
科目番号	0012			科目区分	専門 / 必修		
授業形態	講義			単位の種別と単位数	履修単位: 2		
開設学科	電子情報工学科(2016年度以前入学生)			対象学年	3		
開設期	通年			週時間数	2		
教科書/教材	資料配布						
担当教員	小飼 敬						
到達目標							
1. 様々な分野のための応用プログラムをで作成できる。 2. オペレーティングシステムの役割と構成を理解する。 3. 計算モデルと言語処理系の概要・仕組みを理解する。 4. 整列や平方根計算などの簡単なアルゴリズム・数値計算例について理解する。 5. 代表的なソフトウェア開発方法論について理解する。 6. コンピュータネットワークの基礎を理解し簡単なプログラムを作成する。							
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	ソースコードの意味と位置付けを理解し説明できると共に、ソフトウェア開発において管理できる。			ソースコードの意味と位置付けを理解し説明できる。		ソースコードの意味と位置付けを理解できず、説明もできない。	
評価項目 2	オペレーティングシステムの役割と構成を理解し説明できると共に、関連プログラムを開発できる。			オペレーティングシステムの役割と構成を理解し説明できる。		オペレーティングシステムの役割と構成を理解できず、説明もできない。	
評価項目 3	ソフトウェアと知的財産権の関係を理解し説明できると共に、開発ソフトウェアに対し知的財産権を明確にするための処置を行うことができる。			ソフトウェアと知的財産権の関係を理解し説明できる。		ソフトウェアと知的財産権の関係を理解できず、説明もできない。	
評価項目 4	計算モデルと言語処理系の概要・仕組みを理解し説明できると共に、簡単な言語処理系を設計・実装できる。			計算モデルと言語処理系の概要・仕組みを理解し説明できる。		計算モデルと言語処理系の概要・仕組みを理解できず、説明もできない。	
評価項目 5	整列や平方根計算などの簡単なアルゴリズム・数値計算例について理解し、具体的なアルゴリズム・プログラムとして実現できる。			整列や平方根計算などの簡単なアルゴリズム・数値計算例について理解している。		整列や平方根計算などの簡単なアルゴリズム・数値計算例について理解できない。	
評価項目 6	代表的なソフトウェア開発方法論について理解すると共に、具体的な手法・記法・評価を実践できる。			代表的なソフトウェア開発方法論について理解している。		代表的なソフトウェア開発方法論について理解していない。	
評価項目 7	コンピュータネットワークの基礎を理解し、簡単なプログラムを作成できると共に、単純なネットワークシステムを設計できる。			コンピュータネットワークの基礎を理解し、簡単なプログラムを作成できる。		コンピュータネットワークの基礎を理解できず、関連するプログラムも作成できない。	
学科の到達目標項目との関係							
学習・教育到達度目標 (A)(イ) 学習・教育到達度目標 (B)(ロ)							
教育方法等							
概要	プログラミングを様々な問題に応用する方法を学ぶ。応用プログラムの開発手法・評価の他、プログラム処理系、オペレーティングシステム、アルゴリズム、ソフトウェア開発方法論、ネットワークプログラミング、知的財産権との関係などのそれぞれ基礎について、講義と演習を通して理解を深める。						
授業の進め方・方法	講義およびプログラミング演習で進める。教科書はなく、講義内容や演習問題は各時間ごとに示す。						
注意点							
授業計画							
		週	授業内容		週ごとの到達目標		
前期	1stQ	1週	プログラミングとソフトウェア実装		実践的なソフトウェアの構成を理解し、ソフトウェア実装時に構成を判断・設計できる。		
		2週	ソースコードの管理 (1)		適切な修正方法、パッチ、リポジトリの考え方、ソースコードの配布方法を理解し、ソフトウェア実装時に適用できる。		
		3週	ソースコードの管理 (2)		プログラムを実行しながら不具合を発見する方法を理解し、ソフトウェア実装時に適用できる。		
		4週	ソフトウェアと知的財産権		コンピュータソフトウェアにおける知的財産権の考え方を理解し、ソースコードの著作権の適用方法を実践できる。		
		5週	オペレーティングシステムとプログラミング (1)		オペレーティングシステムの歴史的背景、カーネル、システムコール、プロセス、プロセス間通信について理解し、関連するプログラムを作成できる。		
		6週	オペレーティングシステムとプログラミング (2)		オペレーティングシステムにおける記憶管理、ファイルシステム、ファイル入出力を理解し、関連するプログラムを作成できる。		
		7週	(中間試験)				
		8週	計算モデル (1)		オートマトンの考え方、状態遷移機械、チューリングマシンを理解し、特定のモデルについて図式化できる。		
	2ndQ	9週	計算モデル (2)		文法と言語、文脈自由文法、仮想機械の位置付けや役割を理解し、説明できる。		

後期		10週	言語処理系（１）	コンパイラの構成，プリプロセッサ，リンカ，ローダ，オペレーティングシステムとの関係について理解し，説明できると共に，関連するアプリケーションプログラムを活用できる．
		11週	言語処理系（２）	字句解析，構文解析について理解すると共に，簡単な処理系を実装できる．
		12週	言語処理系（３）	インタプリタの種類，スクリプト言語，ガーベージコレクションについて理解し説明できる．
		13週	プログラム実行の制御（１）	バッチ処理とコマンドインタプリタについて理解し，活用できる．
		14週	プログラム実行の制御（２）	コマンドインタプリタを用いた応用例を実践できる．
		15週	（期末試験）	
		16週	総復習	
	3rdQ	1週	再帰処理（１）	再帰の仕組みを理解し，プログラムとして記述・実現できる．
		2週	再帰処理（２）	再帰処理とリスト・スタックについて理解し，プログラムとして記述・実現できる．
		3週	アルゴリズム（１）	整列の概要と単純な並べ替え手法について理解し，再帰を用いたソートアルゴリズムをプログラムとして実現できる．
		4週	アルゴリズム（２）	検索の役割と単純な検索手法について理解し，整列との組合せを含むプログラムとして実現できる．
		5週	アルゴリズム（３）	コンピュータによる数値計算の概要と役割を理解し，具体例として平方根を求める手法をプログラムとして実現できる．
		6週	アルゴリズム（４）	時間計算量と領域計算量について理解し説明できると共に，簡単な例に対して計算した結果の意味を説明できる．
		7週	（中間試験）	
		8週	ソフトウェア開発方法論（１）	ソフトウェア開発の様々な捉え方について理解し，説明できる．
	4thQ	9週	ソフトウェア開発方法論（２）	グループ開発やシステム開発に対する考え方や手法を理解し，説明できる．
		10週	ソフトウェア開発方法論（３）	複数の具体的なソフトウェア開発方法論について理解すると共に，各方法論における手法・記法・評価を簡単な例に適用できる．
		11週	ソフトウェア開発方法論（４）	Unified Modeling Language (UML) の主要記法を理解し表現できると共に，オブジェクト指向開発の簡単な例に適用できる．
		12週	コンピュータネットワーク（１）	コンピュータネットワークの基本的な仕組みとプロトコル階層モデルについて理解し，説明できる．
		13週	コンピュータネットワーク（２）	クライアント／サーバモデルについて理解すると共に，活用例について説明できる．
		14週	コンピュータネットワーク（３）	クライアント／サーバモデルに基づく簡単な構成のプログラムを実装・評価できる．
		15週	（期末試験）	
		16週	総復習	

評価割合							
	試験	小テスト	相互評価	態度	ポートフォリオ	その他	合計
総合評価割合	80	20	0	0	0	0	100
基礎的能力	0	0	0	0	0	0	0
専門的能力	70	20	0	0	0	0	90
分野横断的能力	10	0	0	0	0	0	10