

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	応用プログラミング	
科目基礎情報							
科目番号	0009			科目区分	専門 / 必修		
授業形態	演習			単位の種別と単位数	履修単位: 2		
開設学科	未来創造工学科 (情報・ソフトウェア系)			対象学年	3		
開設期	通年			週時間数	2		
教科書/教材	教科書 明解C言語入門編柴田 望洋ソフトバンク2310円/参考書 C実践プログラミング 第3版Steve Ouallineオライリー・ジャパン4725円						
担当教員	小池 敦						
到達目標							
1. 代表的なプログラミング言語であるC言語を実用的なレベルで使えるようになる。 2. 基本的なアルゴリズムを理解することで、応用的なプログラムを自ら作成できるようにする。 3. アルゴリズムの一般性を理解することで、C言語以外のプログラミング言語に応用することができる。 【キーワード】プログラミング、C言語、アルゴリズム、データ構造							
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	
関数とライブラリ	与えられた問題に対し、適切な入出力を設定し自作関数を作成できる。また、標準ライブラリや外部ライブラリの利用法を自分で調査し、それを利用した応用的なプログラムを作成出来る。			与えられた問題に対し、適切な入出力を設定し自作関数を作成できる。また、標準ライブラリや外部ライブラリを利用してプログラムを作成出来る。		与えられた問題に対し、適切な入出力を設定し自作関数を作成できる。また、標準ライブラリや外部ライブラリを利用してプログラムを作成することが出来ない。	
ポインタの応用	ポインタを使ったメモリへのアクセス、メモリの動的利用などポインタを用いた応用的なプログラムを自ら作成することができる。			ポインタを使ったメモリへのアクセス、メモリの動的利用などポインタを用いたプログラムを作成することができる。		ポインタを使ったメモリへのアクセス、メモリの動的利用などポインタを用いたプログラムを作成することが出来ない。	
構造体・基本的なデータ構造	構造体やそれを応用したリストなどの基本的なデータ構造を、与えられた問題に対し適切に設計でき、与えられた問題を解決する応用的なプログラムを作成出来る。			構造体やそれを応用したリストなどの基本的なデータ構造を適切に設計できる。		構造体やそれを応用したリストなどの基本的なデータ構造を適切に設計出来ない。	
C言語の仕組み	コンパイラ・プリプロセッサなどC言語周辺の知識を利用したプログラムを自ら設計し作成出来る。			コンパイラ・プリプロセッサなどC言語周辺の知識が理解でき、その知識を利用したプログラムを作成できる。		コンパイラ・プリプロセッサなどC言語周辺の知識が理解できない。	
構造化プログラミング	自作関数や再起処理を利用し、与えられた問題に対する適切なプログラムを見ずらから設計し作成することができる。			自作関数や再起処理を利用したプログラムを作成出来る。		自作関数や再起処理について理解できない。	
プログラム開発過程、プログラム作成演習	与えられた問題に関する適切な問題解決法を導き、プログラム開発過程に基づいたグループ開発により、問題へのソリューションとなるプログラムを作成できる。また、制作物に関する適切なプレゼンテーションができる。			与えられた問題に関する適切な問題解決法を導き、グループ開発により、問題へのソリューションとなるプログラムを作成できる。また制作物に関する適切なプレゼンテーションができる。		与えられた問題に関する適切な問題解決法を導くことが出来ない。	
学科の到達目標項目との関係							
教育方法等							
概要	プログラミング・情報処理の内容を更に進め、C言語を用いた制御情報工学分野のためのプログラミング及び、情報処理のより実践的な技術を身につける。						
授業の進め方・方法	プログラミングの技術力向上のためには多くの演習問題を解く必要がある。したがって、本講義では多くの演習を行いながら進め、最後にまとめとしてグループによるプログラム制作演習をしてもらう。また、ほぼ毎週課題を課すので提出日を守って提出すること。						
注意点	【注意点】本授業では、多くの課題が課されるが、それらは全て以下の評価基準に含まれていることに注意すること。 【事前学習】教科書をよく読み2年生までの学習内容を理解してから望むこと。 【評価法・評価基準】試験結果(60%)、課題(40%)で評価する。詳細は第1回目の授業で告知する。 期末試験と作品制作によりC言語によるプログラミングの基礎の理解の程度を評価する。ただし、演習時間を多く取っているため授業態度の著しく悪いものや課題の未提出が25%を超えたものについては評価対象としないものとする。 総合成績60点以上を単位修得とする。						
授業計画							
		週	授業内容		週ごとの到達目標		
前期	1stQ	1週	授業に関するガイダンス		開発環境や授業の進め方について理解できる。		
		2週	C言語基本事項の復習 (1)		プログラミング、及び情報処理の内容復習のための演習プログラム作成ができる。		
		3週	C言語基本事項の復習 (2)		プログラミング、及び情報処理の内容復習のための演習プログラム作成ができる。		
		4週	関数とライブラリ (1)		C言語標準ライブラリ等の様々な関数ライブラリの使い方を理解できる。		
		5週	関数とライブラリ (2)		C言語標準ライブラリ等の様々な関数ライブラリの使い方を理解できる。		
		6週	応用プログラミング演習 (1)		様々なライブラリ関数を使った実践的なプログラムを作成できる。		
		7週	応用プログラミング演習 (2)		様々なライブラリ関数を使った実践的なプログラムを作成できる。		
		8週	中間試験				
	2ndQ	9週	ポインタの応用		ポインタを使った、メモリ操作やアクセスの仕組みについて理解できる。		

後期	3rdQ	10週	応用プログラミング演習（３）	ポインタの仕組みについて理解し、ポインタを使った実践的なプログラムを作成できる。
		11週	構造体、基本的なデータ構造（１）	C言語でよく利用される基本的なデータ構造について理解できる。
		12週	構造体、基本的なデータ構造（２）	C言語でよく利用される基本的なデータ構造について理解できる。
		13週	応用プログラミング演習（４）	基本的なデータ構造を使ったプログラムが作成できる。
		14週	応用プログラミング演習（５）	基本的なデータ構造を使ったプログラムが作成できる。
		15週	期末試験	
		16週	前期のまとめ	期末試験の解答・まとめを自己確認できる。
	4thQ	1週	C言語の仕組み（コンパイラ、プリプロセッサ等）	コンパイラやプリプロセッサなどの仕組みが理解できる。
		2週	C言語の仕組み（コンパイラ、プリプロセッサ等） 2	コンパイラやプリプロセッサなどの仕組みが理解できる。
		3週	応用プログラミング演習（６）	外部ライブラリやプリプロセッサを利用した応用的なプログラムが作成できる。
		4週	構造化プログラミング（関数応用、再帰処理等） 1	関数の高度な使い方や再帰処理について理解できる。
		5週	構造化プログラミング（関数応用、再帰処理等） 2	関数の高度な使い方や再帰処理について理解できる。
		6週	応用プログラミング演習（７）	これまでの学習内容を使った総合的なプログラム作成ができる。
		7週	応用プログラミング演習（８）	これまでの学習内容を使った総合的なプログラム作成ができる。
		8週	中間試験	
		9週	プログラム開発過程、プログラム作成演習（グループに分かれこれまでの学習内容を用いてプログラムの設計、実装、評価の演習を行う） 1	プログラム開発過程について理解し、これまで学習してきたことを用いた応用的プログラムの作成ができる。
		10週	プログラム開発過程、プログラム作成演習（グループに分かれこれまでの学習内容を用いてプログラムの設計、実装、評価の演習を行う） 2	プログラム開発過程について理解し、これまで学習してきたことを用いた応用的プログラムの作成ができる。
		11週	プログラム開発過程、プログラム作成演習（グループに分かれこれまでの学習内容を用いてプログラムの設計、実装、評価の演習を行う） 3	プログラム開発過程について理解し、これまで学習してきたことを用いた応用的プログラムの作成ができる。
		12週	プログラム開発過程、プログラム作成演習（グループに分かれこれまでの学習内容を用いてプログラムの設計、実装、評価の演習を行う） 4	プログラム開発過程について理解し、これまで学習してきたことを用いた応用的プログラムの作成ができる。
		13週	プログラム開発過程、プログラム作成演習（グループに分かれこれまでの学習内容を用いてプログラムの設計、実装、評価の演習を行う） 5	プログラム開発過程について理解し、これまで学習してきたことを用いた応用的プログラムの作成ができる。
		14週	制作物のプレゼンテーション	課題制作物についてのプレゼンテーションが出来る。
		15週	期末試験	
		16週	後期のまとめ	期末試験の解答・まとめを自己確認できる。

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	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
				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。	3
				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。	3
				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。	3
				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しうることを説明できる。	3
				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。	3
				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。	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。	3
	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】	基本的な論理演算を行うことができる。	3
				基本的な論理演算を組合わせて、論理関数を論理式として表現できる。	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、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	
				標準的な開発ツールを用いてプログラミングするための開発環境構築ができる。	4	
				要求仕様にあったソフトウェア(アプリケーション)を構築するために必要なツールや開発環境を構築することができる。	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	発表	資料作成	合計	
総合評価割合	60	20	15	5	100	
基礎的能力	60	20	0	0	80	
創造的能力	0	0	15	5	20	