

茨城工業高等専門学校		開講年度	令和04年度 (2022年度)		授業科目	プログラミングⅠ	
科目基礎情報							
科目番号	0014			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	国際創造工学科 情報系			対象学年	2		
開設期	通年			週時間数	2		
教科書/教材	適宜プリントを配布, 参考書: 石田晴久「入門ANSI-C」(実教出版), 中山清喬「スッキリわかるC言語入門」(インプレス)						
担当教員	丸山 智章,安細 勉						
到達目標							
1. コンピュータの原理と仕組みを理解し、プログラミングの基礎技術を身につける。 2. 複雑な手順やデータを扱うプログラムを作成するための知識や技術を身に付ける。 3. ソフトウェア開発環境の仕組みを理解し、プログラミングに応用するための技術を身に付ける。 4. ソフトウェアシステムとしての構成を考え構築するための基礎知識を理解する。							
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	コンピュータの原理と仕組みを理解し、プログラミングの基礎技術を身につけると共に、複数の問題例に適用できる。			コンピュータの原理と仕組みを理解し、プログラミングの基礎技術を身につける。		コンピュータの原理と仕組みが理解できず、プログラミングの基礎も身につけていない。	
評価項目2	複雑な手順やデータを扱うプログラムを作成するための知識や技術を身に付けると共に、多様な問題の解決のために適用できる。			複雑な手順やデータを扱うプログラムを作成するための知識や技術を身に付ける。		手順やデータを扱うプログラムを作成できない。	
評価項目3	ソフトウェア開発環境の仕組みを理解して説明でき、プログラミングに応用するための技術を身に付ける。			ソフトウェア開発環境の仕組みを理解し、プログラミングに応用するための技術を身に付ける。		ソフトウェア開発環境の仕組みを理解できず、プログラミングへの応用技術も身に付いていない。	
評価項目4	ソフトウェアシステムとしての構成を考え構築するための基礎知識を理解し、説明できると共に、簡単なソフトウェア構築問題に適用できる。			ソフトウェアシステムとしての構成を考え構築するための基礎知識を理解する。		ソフトウェアシステムとしての構成を考え構築するための基礎知識が理解できない。	
学科の到達目標項目との関係							
学習・教育到達度目標 (A)							
教育方法等							
概要	C言語を用いてプログラミングの様々な処理方法について学ぶと共に、コンピュータソフトウェアの開発の基礎を学ぶ。更に、ソフトウェアシステムの基本的な仕組みに焦点を当て、講義や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。講義内容や演習問題は各時間ごとに示す。成績評価は、演習課題を40%、定期試験を60%として評価する。学年成績は、前期と後期の成績の平均とし、60点以上を合格とする。						
注意点	成績評価は、すべての演習課題を完了した学生に対して行う。						
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	プログラミングの基礎		プログラムについて理解し、簡単な文字出力プログラムをコーディングし実行できる。		
		2週	値の種類と演算子		値の種類や型について理解し、簡単な四則演算を行うプログラムを作成・実行できる。		
		3週	変数と代入		変数の概念や宣言方法について理解し説明できると共に、変数への代入や値の取り出し、四則演算を行うプログラムを作成できる。		
		4週	条件分岐 (1)		関係演算子について理解し、基本的な条件式を構成することができると共に、簡単な条件分岐プログラムを作成できる。		
		5週	条件分岐 (2)		論理演算子について理解し、条件式を組み合わせたプログラムを作成できる。また、値に応じ複数分岐するプログラムを作成できる。		
		6週	繰り返し (1)		単純な繰り返しを行うプログラムを作成できる。		
		7週	(中間試験)				
		8週	繰り返し (2)		前判定・後判定の繰り返しについて理解し説明できる。また、繰り返しを複数段階組み合わせたプログラムを作成できる。		
	2ndQ	9週	関数 (1)		関数の仕組みを理解し、簡単な関数をプログラムとして作成できる。		
		10週	関数 (2)		再帰関数について理解し、簡単な再帰処理プログラムを作成できる。		
		11週	配列 (1)		配列の基本的な考え方を理解し説明できると共に、配列を用いた代入や値の取り出しを行うプログラムを作成できる。		
		12週	配列 (2)		多次元配列について理解し説明できると共に、簡単なプログラム例を作成できる。		

後期		13週	アルゴリズムとプログラミング（１）	計算手順や流れ図について理解し説明できると共に、プログラム実行を実践できる。
		14週	アルゴリズムとプログラミング（２）	並べ替えなどの基本的なアルゴリズムを用いたプログラムを理解し作成できる。
		15週	（期末試験）	
		16週	総復習	
	3rdQ	1週	プログラム実行環境	プログラムの実行とライブラリの関係について理解し、プログラミングにおいて活用できる。
		2週	プログラムの分割作成（１）	プログラムのモジュール化の意義を理解し説明できると共に、アルゴリズム設計やコーディングにおいて実践できる。
		3週	プログラムの分割作成（２）	プログラムコードの様々なモジュール化手法について理解し、プログラミングにおいて活用できる。
		4週	様々な情報の数値表現	数値や文字などの表現方法、値の種類や型の変換について理解し、プログラミングにおいて活用できる。
		5週	演算子の活用（１）	関係演算子を活用したプログラミング手法について理解する。
		6週	演算子の活用（２）	論理演算子を活用したプログラミング手法について理解する。
		7週	（中間試験）	
		8週	データ表現の基礎（１）	アドレスとポインタの概念について理解し、プログラミングにおいて活用できる。
	4thQ	9週	データ表現の基礎（２）	関数や配列におけるアドレスとポインタの活用について理解し、プログラミング技術として実践できる。
		10週	データ表現の基礎（３）	構造体について理解し、プログラミングにおいて様々な種類の変数を組合せたデータ表現とその活用を行うことができる。
		11週	データ表現の基礎（４）	構造体の応用方法について理解し、プログラミングにおいて様々な種類のデータ表現を実装できる。
		12週	ファイル入出力（１）	ファイル入出力処理の基本的な流れについて理解し、プログラムとして実装できる。
		13週	ファイル入出力（２）	ファイル入出力を使った様々な処理について理解し、プログラムとして実装できる。
		14週	総合的なプログラム	文字列やリスト構造など、配列や構造体などを応用した様々なデータ表現に対する処理プログラムについて理解し、プログラミング技術として実践できる。
		15週	（期末試験）	
		16週	総復習	

評価割合

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