

大阪情報コンピュータ専門学校 授業シラバス (2024年度)

専門分野区分	プログラミング基礎	科目名	プログラミング I				科目コード	T1111B1		
配当期	前期	授業実施形態	通常				単位数	6 単位		
担当教員名	清水 素彦	履修グループ	1E(MP/SP)				授業方法	演習		
実務経験の内容	プログラマ、システムエンジニアとしてIT企業で5年間、また、社内SEとして外食チェーン企業の情報システム部門で12年間にわたり、販売管理システムや物流システムなどの開発や導入・運用に従事した。これらの実務経験に基づき、企業現場での仕事(タスク)に必要な基礎知識やスキルを指導する。									
学習一般目標	本校では、IPA(独立行政法人 情報処理推進機構)が提唱し、企業で人材戦略のために広く活用されているiCD(iコンピテンシ ディクショナリ)を参照し、「企業現場で遂行される実際のタスク(業務)」をベースとして本校独自の「タスク・専門スキルモデル」を構築し、それをもとにカリキュラムを構成している。 科目「プログラミング I ~ III」では、アプリケーション開発においてプログラマが遂行するタスクである「プログラミング」と「単体テストの実施」「結合テストの実施」が実施できることを目指す。そのために、本科目「プログラミング I」では、タスク「プログラミング」の中で初心者が学習を始めるにあたり必要となる「基本的なプログラミング」のスキルの修得を目指す。プログラム言語はC#を使用する。									
授業の概要および学習上の助言	まず、視覚的なプログラミングの体験をとおしてプログラムの直感的理解を得るとともに、プログラム開発の全体像の概要を理解する。その後、様々なフォームアプリケーションやコンソールアプリケーションの作成・演習をとおしてC#の基本文法を修得するとともに、基本アルゴリズムやデータ構造を使用したコーディング・デバッグの方法を学ぶ。また、授業期間の後半では、プログラミングにおいて重要な概念となるメソッド(関数)の作成や使用について学習する。授業時間内に演習課題に取り組む時間も多く設けるため、積極的に演習課題に取り組み、授業内容の理解に努めること。 プログラムの開発環境を各自のノートPCに構築するので、授業時間内だけでなく、授業時間外でも自宅等で継続した学習を進めることが可能である。プログラミングスキルの向上には実習の量が重要な要素となるため、授業時間外でも自主的・自発的な学習を行うことが望ましい。									
教科書および参考書	「3ステップでしっかり学ぶ C#入門[改訂2版]」技術評論社									
履修に必要な予備知識や技能	特になし									
使用機器	各自の個人所有パソコンを使用する									
使用ソフト	Visual Studio 2022、C#									
学習到達目標	学部DP(番号表記)	学生が到達すべき行動目標								
	3/5	プログラミングに興味関心を持ち、学習に取り組むことができる								
	1/2/4	実習環境の基本的な機能を利用し、標準クラスを利用してWindowsフォームアプリやコンソールアプリを作成することができる								
	1/2/4	簡単なクラスやメソッドの記述ができ、複数のメソッド(関数)を利用してプログラムを作成することができる								
	1/2/4	設計書に基づいて、簡単なプログラムを作成することができる								
	1/2/4	作成したプログラムの不具合を発見し、デバッガなどを利用して修正することができる								
達成度評価	評価方法	試験	小テスト	レポート	成果発表(口頭・実技)	作品	ポートフォリオ	その他	合計	
	学部DP	1.知識・理解	5	5	15					25
		2.思考・判断	5	5	15					25
		3.態度							10	10
		4.技能・表現	10		20					30

	5.関心・意欲							10	10
	総合評価割合	20	10	50				20	100
評価の要点									
評価方法	評価の実施方法と注意点								
試験	プログラムの作成能力を期末試験の成績で評価する。								
小テスト	授業中に実施する理解度確認テストの成績で評価する。								
レポート	授業中に出题する課題、課題解決型授業に出题する課題に対する提出物の完成度を判断する。 ただし、提出物を自ら作成しなかった場合や期限までに提出しなかった場合は、原則として評価しない。								
成果発表(口頭・実技)									
作品									
ポートフォリオ									
その他	授業への出席、取り組みなどを含め総合的に判断する。								

授業明細表

授業回数	学習内容	授業の運営方法	学習課題(予習・復習)
第1回	プログラミング体験(Scratch) プログラム開発の全体像を理解する プロジェクトの作成方法を理解する	講義・演習	外部サイト ・Scratch
第2回	コンソールアプリの簡単な作成方法を理解する 初歩的なWindowsフォームを作成する C#ソースコードの構成、書き方などを理解する	講義・演習	外部サイト ・TechFUL
第3回	演算と変数、およびデータ型を理解する 代入や四則演算など順次処理を作る 設計書の読み方を理解する	講義・演習	
第4回	if文による条件分岐を作る 制御構造として選択処理の構造を理解する	講義・演習	
第5回	do文、while文による繰り返し処理を作る 制御構造として反復処理の構造を理解する	講義・演習	
第6回	for文による繰り返し処理を作る 制御構造として反復処理の構造を理解する 配列を用いてプログラムを作成する	講義・演習	
第7回	多重ループの繰り返し処理を作る 多次元配列を用いてプログラムを作成する 理解度確認テスト(1)	講義・演習	
第8回	ソースコードを分割する(メソッド・関数) 例外処理を理解する	講義・演習	
第9回	オブジェクト指向プログラミングの考え方の基礎を理解する	講義・演習	
第10回	簡単なクラスを作成する	講義・演習	

第11回	switch文による多分岐を作る 制御構造として選択処理の構造を理解する	講義・演習	
第12回	ProgramクラスとMainメソッドを理解する 基本アルゴリズムをコーディングする ・探索(線形探索、二分探索) ・整列(選択ソート、交換ソート、挿入ソート)	講義・演習	
第13回	理解度確認テスト(2)	講義・演習	
第14回	課題解決型授業1 プログラム演習	遠隔授業 実施時期:2期	
第15回	課題解決型授業2 プログラム演習	遠隔授業 実施時期:4期	