

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Ⅱ				科目コード	T1112A3		
配当期	後期	授業実施形態	通常				単位数	6 単位		
担当教員名	福間 更生	履修グループ	1A(JT/KS/SI)				授業方法	演習		
実務経験の内容	<p>市販のパッケージソフトウェアのテスターからIT業界に入り、パッケージソフトウェア、土木系・教育系・鉄道系・小売業向けなどの企業向けシステム、公共システムの開発プロジェクトに携わり、今までの経験としてはテスター1年、プログラマー6年、システムエンジニア10年、プロジェクトマネージャと管理職を5年経験し、開発言語はC#、Visual C++、Visual Basic、Java、Objective C、Swiftなど多彩な言語での開発経験、開発工程は要件定義から全工程の経験がある。その他、学校の教員としてC#などの講義や実習も8年経験しており、IT業界での経験も踏まえた指導を行う。</p>									
学習一般目標	<p>本校では、IPA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。</p> <p>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ー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Ⅱ」では、タスク「単体テストの実施」の中の「テスト技法」のスキルのうち初歩的なスキル、タスク「プログラミング」の中で業務システム構築の際に必要なとなる「データベース操作」のスキルなどの習得を目指す。</p>									
授業の概要および学習上の助言	<p>モジュール論理設計の設計書に記述された処理手順を理解し、それに基づいたコーディングが行えるように演習を実施する。また設計書の作成もできるようにしていく。作成したプログラムについては、テスト技法にのっとってホワイトボックステストやブラックボックステストを実施し、単体テストと結合テストが行えるようにする。授業後半では、データアクセス技術を用いたデータベースプログラミングを学び、CRUD操作のコーディングを行う。最終課題の実務システム演習では、初級PGの「タスク」であるモジュール論理設計、プログラミング、単体テストの仕様作成、実施の総合演習を行う。本科目ではウォーターフォールモデルを基本とする開発工程を理解し、1年前期の「プログラミングⅠ」で学んだプログラミングに必要な基礎知識やスキルを活用して、各開発工程の成果物に基づく実習課題に取り組んでほしい。</p>									
教科書および参考書										
履修に必要な予備知識や技能	「プログラミングⅠ」で学んだプログラミングに必要な基礎知識									
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# SQL Server									
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								
	1/2/4	モジュール論理設計の設計書を理解・作成でき、設計書に基づいたコーディングができる								
	1/2/4	データベースへのアクセス技術を理解し、CRUD操作のコーディングができる								
	1/2/4	テスト仕様書に基づいて、ホワイトボックステストとブラックボックステストが実施できる								
	1/2/4	ツールを用いたバージョン管理ができる								
	3/5	専門知識や技能を修得するために自ら継続的に学習する成したプログラムの不具合を発見し、デバッグなどを利用して修正することができる								
達成度評価	学部D	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
		1.知識・理解		10	30					40
		2.思考・判断		5	15					20
		3.態度							5	5

価	P	4.技能・表現	5	15					20
		5.関心・意欲						15	15
	総合評価割合		20	60				20	100
評価の要点									
評価方法		評価の実施方法と注意点							
試験									
小テスト		学期末に期末理解度確認テストを実施してその成績で評価する。							
レポート		授業中に出題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	設計書に基づいたプログラミング(復習) ・設計書の理解(擬似言語を用いた処理手順) ・基本文法のコーディングとデバッグの復習	講義・演習	
第2回	テスト技法(単体テスト)を理解する ・テスト仕様書に基づいたテストコードの作成 ・ホワイトボックステストの実施	講義・演習	
第3回	テスト技法(単体テスト、結合テスト)を理解する ・ブラックボックステストの実施	講義・演習	
第4回	テスト技法の演習 ・ブラックボックステストの演習 プログラミング演習	講義・演習	
第5回	プロパティを理解する クラスの継承を理解する プログラミング演習	講義・演習	
第6回	Listクラスの使い方を理解し、foreach文による繰り返し処理を作成する ツールを用いた基本的なバージョン管理を理解する(Git) ・Gitの基本操作の理解 ・Gitを用いたバージョン管理の実施	講義・演習	
第7回	データアクセス技術を理解する(データプロバイダ) ・データプロバイダを用いたコーディングの理解 ・CRUD操作(Create(登録)、Read(参照)、Update(更新)、Delete(削除)機能)のコーディング練習	講義・演習	
第8回	データアクセス技術を理解する(データプロバイダ) ・パラメータを用いたCRUD操作のコーディング演習 プログラミング演習	講義・演習	

第9回	<p>データアクセス技術を理解する(Entity Framework)</p> <ul style="list-style-type: none"> Entity Frameworkを用いたコーディングの理解 CRUD操作のコーディング練習 	講義・演習	
第10回	<p>データアクセス技術を理解する(Entity Framework)</p> <ul style="list-style-type: none"> Entity Frameworkを用いたCRUD操作のコーディング演習 プログラミング演習 	講義・演習	
第11回	<p>実務システム演習</p> <ul style="list-style-type: none"> Entity Frameworkの実務的なコーディングを練習する 	講義・演習	
第12回	<p>実務システム演習</p> <ul style="list-style-type: none"> Entity Frameworkの実務的なコーディングを練習する 	講義・演習	
第13回	期末理解度確認テスト	小テスト	
第14回	<p>課題解決型授業1</p> <p>プログラム演習</p>	<p>遠隔授業</p> <p>実施時期:6期</p>	
第15回	<p>課題解決型授業2</p> <p>プログラム演習</p>	<p>遠隔授業</p> <p>実施時期:8期</p>	