

奈良工業高等専門学校		開講年度	令和02年度 (2020年度)	授業科目	プログラミング基礎
科目基礎情報					
科目番号	0031		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: 1	
開設学科	情報工学科		対象学年	2	
開設期	前期		週時間数	2	
教科書/教材	参考書に基づく。適宜補助教材を配布する。				
担当教員	松村 寿枝,市川 嘉裕				
到達目標					
<p>順次、分岐、繰り返しの制御構造を使いこなすことを例に、後に続くプログラミング学習をスムーズに進めるための基礎を築くことを目標とする。</p> <p>具体的には、下記の項目ができるようになること。</p> <ul style="list-style-type: none"> <li>・デバッグの手順を理解してバグをスムーズに取り除くことができる</li> <li>・仕様や要求(文章等)から手順を理解してフローチャート等を用いて整理し、プログラムを作成することができる</li> <li>・インデント、コメント、変数名、関数名等に気を使って読みやすいソースコードを記述することができる</li> <li>・授業時間外で予習ができる</li> </ul>					
ループリック					
	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) デバッグ	デバッグの手順を理解してバグをスムーズに取り除くことができる		デバッグの手順を理解してバグを発見することができる		デバッグの手順を理解できない
(評価項目2) フローチャート	仕様や要求(文章等)から手順を理解してフローチャート等を用いて整理し、プログラムを作成することができる		仕様や要求(文章等)から手順を理解してフローチャート等を用いて整理することができる		プログラムの手順を整理できない
(評価項目3) ソースコード	インデント、コメント、変数名、関数名等に気を使って読みやすいソースコードを記述することができる		インデント、コメント、変数名、関数名等に自分用のルールをもつことができる		インデント、コメント、変数名、関数名等に自分用のルールがない
(評価項目4) 自己学習	授業時間外で予習ができる		疑問点を解消するためにそれを言語化することができる		疑問点を言葉にできない
学科の到達目標項目との関係					
準学士課程(本科1～5年)学習教育目標(2)					
教育方法等					
概要	以後プログラミングを行ううえで重要な要素となるデバッグ、フローチャートの利用、ソースコードの適切な記述、自己学習の習慣については、一度教わっただけで身につくことはない。そのため、各要素について課題や解説等の中で繰り返し登場させることで、学生の中での意識化・習慣化を図る。				
授業の進め方・方法	可能な限り短時間の講義に加えて、授業中の課題と、授業時間外の予習課題を軸に進める。授業中の課題は理解の確認と促進のために、予習課題は自己学習の推進のために重要である。自己学習のためにはわからないことを明確にする力も重要であることから、疑問点を言語化する訓練を随所で行う。理解の向上とモチベーションの維持のために、授業は3週を1単元としてメリハリをもたせる。具体的には各週で、 (1) ビジュアルプログラミング、 (2) Javaプログラミング、 (3) 応用課題 を行う。 またグループ単位の課題を設けることで、知識の定着を図る。				
注意点	<p>授業に欠席した場合は必ず補習によって取り戻すこと。</p> <p>【参考書】 Javaプログラミングに関する推薦本 (1) スッキリわかるJava入門 第2版, 中山清喬/国本大悟 著, インプレス (2) 新・明解Java入門, 柴田望洋 著, SBクリエイティブ ※後期のプログラミングIでは(1)が教科書になります</p> <p>Scratchプログラミングに関する推薦本 ・Scratchで学ぶ プログラミングとアルゴリズムの基本, 中植正剛/太田和志/鴨谷真知子 著, 日経BP社</p> <p>【自己学習】 事前学習 ……あらかじめ次の講義内容に該当する部分を教科書等で調べて、理解できるところ、理解できないところを明らかにしておくこと。 事後展開学習 ……演習、課題は必ず解いて、締め切りまでに提出をすること。理解が不十分な場合は参考書の練習問題に取り組んでおくこと。</p>				
学修単位の履修上の注意					
授業計画					
		週	授業内容	週ごとの到達目標	
前期	1stQ	1週	ガイダンス	授業の目標を理解し、以後の授業のための準備を行う	
		2週	ビジュアルプログラミングで順次処理	Scratchの使い方を覚え、動作を組み合わせて目的の動作を実現する	
		3週	Javaプログラミングで順次処理	Javaによるプログラミングの方法と変数と入出力の使い方を覚え、動作を組み合わせて目的の動作を実現する	
		4週	応用課題	デバッグの方法を覚え、プログラムの問題を発見する	
		5週	ビジュアルプログラミングで分岐処理	フローチャートの使い方を覚え、Scratchで分岐処理を含む動作を実現する	
		6週	Javaプログラミングで分岐処理	Javaで分岐処理を含む動作を実現する	
		7週	応用課題	ここまでの知識を利用するグループ課題に取り組む	
		8週	中間試験	中間試験	
	2ndQ	9週	ビジュアルプログラミングで繰り返し処理	Scratchで繰り返し処理を含む動作を実現する	

		10週	Javaプログラミングで繰り返し処理	Javaでソースコードのきれいな記述法を覚え、繰り返し処理を含む動作を実現する
		11週	応用課題	ここまでの知識を利用するグループ課題に取り組む
		12週	総合演習	ここまでの知識を利用する課題に取り組む
		13週	総合演習	ここまでの知識を利用する課題に取り組む
		14週	総合演習	ここまでの知識を利用する課題に取り組む
		15週	期末試験	期末試験
		16週	答案返却	答案返却

### 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	プログラミング	代入や演算子の概念を理解し、式を記述できる。	3	前2,前3,前4,前8,前15
				プロシージャ(または、関数、サブルーチンなど)の概念を理解し、これらを含むプログラムを記述できる。	2	前12,前13,前14,前15
				変数の概念を説明できる。	2	前2,前3,前4,前8
				データ型の概念を説明できる。	2	
				制御構造の概念を理解し、条件分岐を記述できる。	2	前5,前6,前7,前8,前15
				制御構造の概念を理解し、反復処理を記述できる。	2	前9,前10,前11,前15
				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記述できる。	3	前7,前11,前14,前15
				要求仕様に従って、いずれかの手法により動作するプログラムを設計することができる。	2	前7,前11,前14,前15
				要求仕様に従って、いずれかの手法により動作するプログラムを実装することができる。	2	前7,前11,前14,前15

### 評価割合

	試験	発表	相互評価	態度	ポートフォリオ	その他	合計
総合評価割合	40	0	10	10	40	0	100
基礎的能力	0	0	10	10	0	0	20
専門的能力	40	0	0	0	40	0	80
分野横断的能力	0	0	0	0	0	0	0