

教科	工業	科目名	プログラミング技術	単位数	2
学科	機械システム	学年	2 学年	履修区分	必修・機械制御コース
使用教科書	7実教「工業746 プログラミング技術」				
副教材など	プログラミングのためのアプリケーション				

1. 科目の目的

工業の見方・考え方を働かせ、実践的・体験的な学習活動を行うことを通じて、コンピューターのプログラミングに必要な資質・能力を育成する。

2. 授業の内容と進め方

コンピューターのプログラミングをアルゴリズムとプログラム技法の観点から捉え、工業生産や社会生活に関連付けて考察し、実践的・体験的な学習活動を行うことなどを通して、コンピューターのプログラミングができるようになる。そのためにまず、コンピューターによる問題処理の手順を理解し、次に実際のプログラムを作成するための技法を身につける。さらに、ファイル処理、ネットワーク処理、機器制御処理、グラフィック処理などの実際的な応用プログラムによりプログラムの開発方法を体験的に学習する。

3. 学習する上での留意点

演習や実習が主となるので、授業中に具体的な理解できるようになることが大切である。論理的な思考のもと、アルゴリズムを表現できるようになることが必要である。

4. 課題等について

生徒の理解度などを勘案し、教科書の例題や問題を活用する。図表作成にあたっては適切なデータを与える。プログラムと実行結果のハードコピーを提出させる。

5. 成績評価規準(評価の観点及び趣旨)

評価の観点	評価規準
知識・技術	<ul style="list-style-type: none"> ・コンピューターを使用して問題を解決するための処理手順を理解している。 ・文書化、システムの開発手順、プログラムの構造化、モジュール化などの実践的な知識を持ち、効率的な開発の技法を理解している。 ・コンパイラなどの開発用ソフトウェアを適切に操作し、プログラムを作成できる。
思考・判断・表現	<ul style="list-style-type: none"> ・基本的なアルゴリズムと処理手順を実際にプログラミングすることを通して理解している。 ・処理の対象となる問題を正確に分析し、適切な処理手順を考え、プログラムを作成する実践的な能力を身につけている。
主体的に学習に取り組む態度	<ul style="list-style-type: none"> ・コンピューターによる問題処理の手段としてのプログラミングに興味・関心を持っている。 ・基本的なプログラミング言語の知識を学習し活用する意欲を持ち、実際のプログラム開発に主体的に取り組む態度を身につけている。 ・デバッグ、トレースなどの操作を通じて、プログラムが正しく動作しているかの確認を行える技能を有し、期待通りの動作を行うプログラムの作成に主体的に取り組むことができる。

6. 評価の方法

定期考査の成績、課題や提出物及び学習への取り組み状況などによる総合評価

学期	月	学習内容 (単元・考査等)	学習のねらい	備考
一 学 期	4	第1章 アルゴリズムとシステム開発 1節 アルゴリズム 2節 プログラム開発環境	<ul style="list-style-type: none"> 身近な例を用いてアルゴリズムの役割を理解させる。 流れ図に用いる主な図記号を理解させる。 順次型・選択型・繰返し型のアルゴリズムを理解させる。 システム開発の流れと各段階の作業内容を理解させる。 システム開発における文書化の意味とその重要性を理解させるとともに、アローダイアグラムなどの図表について理解させる。 OSの機能を理解させる。 プログラム言語の種類と特徴を理解させ、コンパイラ言語で、実行可能プログラムを作成するまでの手順を理解させる。 	第1章は座学(講義)が中心となるので、実習室でなく教室を利用して授業を行うこともある。
	5	第2章 プログラミング技法I 1節 基本的なプログラム	<ul style="list-style-type: none"> 実習が中心となるので、コンパイラの使い方などの基本的な操作方法を覚えさせる。 C言語のプログラムの書き方について理解させる。 定数の種類、変数の型と記憶領域の関係を理解させ、取り得る値の範囲や変数名の付け方および型宣言について理解させる。 デバッグの概要を理解させ、プログラムの誤りを修正できるようにする。 標準入出力関数の使い方理解させ、簡単な入出力のプログラムを書けるようにする。 算術演算子の優先順序と結合規則を理解させ、適切な演算式を書けるようにする。 複合代入演算子や増分演算子、減分演算子などC言語独特の演算子についてその働きを理解させる。 	第2章以降、実習が中心の授業になる。コンピュータの起動、終了などの基本操作を理解する。コンパイラを利用するので、コンパイラの使用法、エラーの見方、エラーの訂正方法などをよく理解する。整数演算・実数演算・混合演算について例から学ぶ。
	6	2節 プログラムの制御構造	<ul style="list-style-type: none"> 選択における制御文の使い方を理解させる。 関係演算子・等価演算子を利用した条件式の書き方を理解させる。 論理演算子の働きについて理解させ、二つ以上の条件式を組み合わせる方法を理解させる。 演算子の優先順序と結合規則について理解させ、分かりやすい条件式を書けるようにする。 繰返しにおける制御文の使い方を理解させる。 永久ループについて説明し、break文とcontinue文の使い方を理解させる。 	キーボードの入力の早さや、理解の程度などの個人差が出やすい教科であるので、早く進んでいる生徒には、節末問題や章末問題を解かせたり、課題を与える。
	7	3節 配列とポインタ	<ul style="list-style-type: none"> 配列を用いる利点を理解させ、配列の宣言について説明し、配列のサイズと要素の添え字の関係を理解させる。 文字型配列と文字列の関係およびナル文字の意味とその働きについて理解させる。 多次元配列について理解させる。 ポインタによりメモリ上のアドレスを扱えることを理解させ、アドレス演算子・間接参照演算子の働きを理解させる。 配列とポインタの関係を理解させる。 	ポインタがC言語の学習においてひとつの壁になることはよく知られている。理解の遅い生徒には後の学習のためにも、補習を行ってでもある程度理解させる。
二 学 期	9	第3章 プログラミング技法II 1節 関数	<ul style="list-style-type: none"> 関数の概念とC言語における関数の意味を理解させる。 関数の型と引数について説明し、プロトタイプ宣言がなぜ必要か理解させ、関数の作り方を理解させる。 プリプロセッサの種類と働きを理解させる。 数学関数の種類と使い方について理解させる。 変数の有効範囲と記憶域クラスについて説明する。 関数には値を渡す関数と、アドレスを渡す関数があることを理解させる。 	本節の内容は広範囲にわたるため、効率的な理解が必要である。
	10	2節 標準化とテスト技法	<ul style="list-style-type: none"> 標準化の必要性を理解させ、プログラム処理の流れを分析するために、状態遷移図などの図表が使用されることを理解させる。 開発効率について理解させ、構造化プログラミングの必要性和モジュール化について理解させる。 構造化プログラムの基本構造を理解させる。 	
	11	第4章 応用的プログラム 1節 構造体とデータ構造	<ul style="list-style-type: none"> 構造体の概念を理解させ、その宣言や初期化の方法を理解させるとともに、構造体の参照や構造体を利用した演算ができるようにする。 具体的な利用例を示して連結リストの概念を理解させる。 二分探索の原理を理解させ、双方向リストの違いを認識させる。 	構造体もC言語の学習におけるひとつの壁になる。構造体の考え方はオブジェクト指向のプログラムにもつながるため、じゅうぶんに理解させる必要がある。
三 学 期	1	第5章 入出力設計 1節 ネットワークの利用	<ul style="list-style-type: none"> クライアントとサーバの役割など基本的なLANに関する知識を確認させる。 階層的なファイル管理システムについて説明し、ファイルやディレクトリの性質を理解させる。 ネットワーク用のプログラムを作成し、クライアントとサーバ間でのデータ送受信の仕組みについて理解させる。 ftpについて説明し、実習により理解を深める。 	工業情報数理、ハードウェア技術、ソフトウェア技術の教科書を併用して説明を補足する。
	2	2節 制御用ICの活用	<ul style="list-style-type: none"> 制御用ICの構成を理解させ、制御用プログラムの開発手順や入出力ポートの設定について理解させる。 制御用プログラムを作成し、実際に制御を行うことにより、C言語による制御技術を理解させる。 	ハードウェア技術、電子計測制御の授業内容との関連に留意する。
	3			