

## 福岡工業大学 学術機関リポジトリ

A Consideration for Advancing and Enriching the Education of Programming: Practice through the Lectures and Exercises Performed at Fukuoka junior College of Technology, and Further Investigation for Better Education.

メタデータ	言語: jpn 出版者: 公開日: 2021-01-25 キーワード (Ja): キーワード (En): 作成者: 高橋, 昌也 メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/11478/00001610">http://hdl.handle.net/11478/00001610</a>

## プログラミング教育の推進・充実をめざして

— 福岡工業短期大学の講義・演習を通しての実践と今後の課題 —

高 橋 昌 也 (福岡工業大学短期大学部 OA情報システム学科)

**A Consideration for Advancing and Enriching the Education of Programming: Practice through the Lectures and Exercises Performed at Fukuoka junior College of Technology, and Further Investigation for Better Education.**

Masaya TAKAHASHI

(Department of Office Automation and Information Systems, Jr. College. Fukuoka Institute of Technology)

**Abstract**

Every Educational institution confronts social needs, while it has its own ideology in education. Fukuoka junior College of Technology (F J C T, for short) is no exception. Taking the balance between the two into consideration, we have always made a great effort to keep the quality of our education comprehensibly and reshape the curriculum into a better one, if necessary, to let our students acquire the firm knowledge about information systems during the course of two years.

How should we put our ideals into practice? The most important factor to produce students of high skills depends on each professor who constantly devotes himself to enriching the contents of subjects he teaches. For this reason, the contents of subjects have to be in the constant process of being analyzed, modified and revised.

This paper attempts to analyze the contents of lectures and exercises for programming I performed at F J C T from 1994 to 2001, and investigate future themes for better education.

*Key words: lectures and exercises, contents, verification*

## 1. はじめに

あらゆる教育機関には、それぞれに対する社会のニーズがある。福岡工業短期大学（現、福岡工業大学短期大学部）では、それを『今日、短期大学の卒業生に対して社会が真に求めるものは、事務系及び技術系のあらゆる職種において、情報システムの技術を理解

し駆使用する能力です。』と捉えており、それに対して、『情報技術を中心とする理工系の短期大学として、情報化社会の変化に対応できるように、情報の基礎とソフトウェアを中心にして、技術系以外の分野でも即戦力となりうる専門教育の充実を図り、品性ある豊かな人間性を備えた人材を育て、社会の要請に応じていきたいと考えております。』という教育理念を掲げている。そして、それらのバランスを考えながら、絶えず教育内容を見直し、カリキュラムという形で提示し、2年間の短大教育の中で可能な情報システムの考え方

---

平成14年5月30日受付

を修得させるよう努めている。

しかし、実際に方針どおりに教育を推進し、学生という人材を育てるための大きな要素は、各教員が受け持った授業科目の内容の充実を図ることである。そのためには、授業内容を絶えず検証し、修正・改訂する必要がある。

そこで本稿では、平成6年度から平成13年度まで筆者が福岡工業短期大学（以下、本学と記述する）で行ったプログラミング関連科目の講義・演習の内容を検証し、今後の課題を探っていくことにする。

なお、本学は前期・後期の2学期制であり、各学期の講義は13週（プラス、定期試験、再追試験）となっている。

## 2. プログラミング教育の意義

近年、各種のアプリケーションの普及により、自分でプログラムを作らなくても、ほとんどの業務処理ができるようになった。しかし、計算処理をしながら図形を描くなど、アプリケーションソフトではできない分野もある。<sup>6)</sup>また、一方ではコンピュータの発展に伴って、社会や企業でのコンピュータの利用範囲が広がり、利用形態も多様化してきており、アプリケーションソフトで対応できない業務等を自前のプログラムで処理するケースも出てきている。また、従来より存在する基幹システム等の保守・改良も重要な仕事である。

このような現代社会の状況において、プログラミングは依然として情報処理活動の重要な位置を占めることができる。従って、プログラミングのできる人材を育てるためのプログラミング教育を行うことは、『情報技術を中心とする理工系の短期大学』である本学の社会的役割の重要な要素である。

## 3. プログラミング教育の概要

本学のプログラミング教育は、アルゴリズムやフローチャートなどのプログラミングのために必要不可欠な基礎的な知識を講義中心に解説し、演習問題をとおしてそれらを実際に体験して修得させるシステムをとっている。

具体的には、まず、「プログラミング基礎」という科目で基礎的な知識を解説し、「プログラミング基礎演習」という科目でその習得を図っている。演習のための言語は、電子情報システム学科では「C言語」で

あり、OA情報システム学科では「Visual Basic」である。そして、その後にさまざまな種類の言語とその言語の特性を生かすための処理に関する説明と演習に発展していくのである。また、関連知識として電子情報システム学科では「データ構造」「データ構造演習」という科目も用意されている。

筆者が担当したプログラミング言語は「COBOL」, 「Pascal」, 「Visual Basic」の3種類である。また、「プログラミング基礎」「データ構造」も担当している。

「プログラミング基礎」については以下の4章で、「データ構造」については5章で、「Visual Basic」については6章で、「COBOL」については以下の7章及び8章で、「Pascal」については9章でそれぞれ述べる。

## 4. プログラミング基礎

厳密には「プログラミング基礎/Visual Basic」という科目名で、平成12年度から担当している科目である。OA情報システム学科ビジネス情報処理コースの1年生後期学生を対象にした科目である。

講義科目の概要は『この科目ではプログラミングとはどのようなものであるかについて学習します。そして、プログラミングのための基本操作やプログラムの用途、言語の種類など、基礎的事項について学びます。また、基本的な制御プログラムを通し、処理手順（アルゴリズム）や図式化の表現法と必要性、構造化プログラミングとはどのようなものかについて学習します。その後、いくつかの簡単なプログラム手法によってプログラミングに親しみが持てるような授業内容を計画しています。<sup>7)</sup>』である。

各週の授業計画は以下のとおりである。<sup>7)</sup>

- (1) プログラミングとは
- (2) 操作法とプログラム処理Ⅰ
- (3) 操作法とプログラム処理Ⅱ
- (4) 操作法とプログラム処理Ⅲ
- (5) 処理手順とフローチャート及び構造化プログラミング
- (6) 条件の判断処理Ⅰ
- (7) 条件の判断処理Ⅱ
- (8) くり返し処理Ⅰ
- (9) くり返し処理Ⅱ
- (10) くり返し処理Ⅲ

- (1) サブルーチン・プロシージャⅠ
- (2) サブルーチン・プロシージャⅡ
- (3) まとめ

筆者は、本科目の第一の目的を『コンピュータが、いかに演算をしたり、記憶したりする能力を備えていると、コンピュータの利用者が「売上統計を出してくれ」とコンピュータに伝えても、それがどのような処理手順から成り立っているか勘を働かして理解し、処理してくれる能力はないのである。そのようなことができるのは、やはり人間だけである。コンピュータの優れた能力を活用するためには、あらかじめその売上統計の処理をどのように進めればよいかの手順を、コンピュータ自身の立場に立って、しかもコンピュータが理解できる形で覚え込ませなければならない。このような処理手順をアルゴリズムと呼び、それをコンピュータが理解できるように一定のルールのもとに表現したものをプログラムとよんでいる。<sup>8)</sup>』を体得させることであると捉え、導入部分（最初の2週程度）でアルゴリズムの基本的特性について述べ、身近なアルゴリズムの例をいくつか示した。

アルゴリズムの特性とは、以下の(1)~(3)のとおりである<sup>2)</sup>。

- (1) 操作が有限である：手順の有限回の繰返しによって結果が求められる。
  - (2) 決定的な手順である：同じデータに同じ手順を利用すれば、必ず同じ結果が得られる。
  - (3) 一般的である：手順の制約条件に適合するデータからは、必ず結果は得られる。
- 例題としては、以下の(1)~(7)を示した。

- (1) 料理のレシピ。これは一見アルゴリズムのようであるが、実際は同じ手順で作っても、作り手の力量によっては異なる味の料理を作ることになるので、『決定的な手順』というアルゴリズムの特性の1つに合致しないので、厳密にはアルゴリズムとはいえない。しかし、求める料理を作るための指針になってはいることが重要なのである。
- (2) パソコンを使って新規文書を作成してフロッピーディスクに保存する一連の操作。重要な点は、手作業ではあるが、誰がやっても、同じ手順で行えば同じように保存されることである。
- (3) 常用対数表から  $\log_{10}x$  を求める方法：  
ここで用いた常用対数表は、100~1009の整数値について、下1桁の数字を横軸に、それ以外の桁の数字を縦軸に配置し、その交わるところにその

値に対応する常用対数の小数点以下の値が記述されている。例えば、245の常用対数の小数点以下の値は、上から15行目の左から6列目に書かれている。この常用対数表を使って、任意の整数の常用対数を求める方法を、この表の範囲内の値、範囲外の値に分けて、それぞれ例題を用いて解説した。これは、人間が常用対数を手計算で求める手順であるが、この手順に従えば、同じデータを与えれば同じ結果を算出できることが重要なのである。

- (4) 統計の度数分布表の作り方：文献[1]の第1章1節の「すぐわかる度数分布表の作り方の公式」に出てくる手順1~4を、例題を用いて解説した。これは、人間が電卓等を使って手計算で度数分布表を作成する手順であるが、この手順に従えば、同じデータを与えれば同じ結果を算出できることが重要なのである。
- (5) 合計計算、2つのデータの入れ替え：  
文献[2]の第1章、1.2節に出てくるアルゴリズムを、例題を用いて解説した。特に、「合計計算」の場合は、最初に合計値を0にする、いわゆる初期化の概念に、「データの入れ替え」の場合は、一方のデータを一時的に保存しておく、いわゆる作業変数（作業領域）の概念に力点を置いた。
- (6) 最大値・最小値の検索、ソーティング：  
「最大値・最小値の検索」については、文献[2]の第1章1.2節に出てくるアルゴリズムを少しアレンジし、データの数を5個くらいにした例題を用いて説明した。ここでは、繰返し回数等に使うカウンタの概念に力点を置いた。「ソーティング」については、バブルソートの考え方を基本的に、「2つのデータの入れ替え」を繰返し行っていることに力点を置いた。
- (7) ユークリッドの互減法・互除法：2つの整数の最大公約数を、文献[3]の第4章1節に出てくる「ユークリッドの互減法」という方法で求める算法を、例題を用いて解説した。また、同じ作業を行う、更に効率が良い「ユークリッドの互除法」を、同じ例題を用いて解説した。

以上により、コンピュータを上手に活用するためには、アルゴリズムが必要であり、アルゴリズムとはどういうものかということ、を、解説した。

次の1週程度で、アルゴリズムを図式化する必要性

について説明した。というのは、実際にコンピュータで問題を処理する時には、相当に複雑な処理ステップからなる場合が多く、だからこそコンピュータに処理を委ねるのである。このようなアルゴリズムを簡条書き等で表現してはかなり煩雑になり、アルゴリズムを理解することは困難になるからである。本科目では、図式化としては最もよく知られているフローチャートについて、文献 [2] の第 1 章 1. 3 節や文献 [3] の第 1 章 3 節をベースにして解説し、例題として上記の「統計の度数分布表の作り方」をフローチャートに表現したものを示した。

さらに、分かりやすいプログラムを作る必要性が高くなっているという視点から、構造化プログラミングと段階的詳細化についても解説した。構造化プログラミングとは、簡単にいうと、入口と出口は 1 つずつになるように、「順次構造」、「選択構造」、「繰返し構造」の 3 つの基本構造の組み合わせとしてプログラムを作るという考え方である。3 つの基本構造それぞれと、その変形である「多重選択構造」、「空処理」、「Do-While 型」、「Do-until 型」などについても説明した。また、段階的詳細化とは、まず 1 つの処理問題を、いくつかの独立したモジュール (ルーチンともいう) に細分化し、次に各モジュールのアルゴリズムを具体的に展開する方法である。このレベルにおいてもアルゴリズムが複雑ならば、さらに各モジュールをサブモジュール (サブルーチン) に分割して、各サブモジュールのアルゴリズムを具体的に展開するのである。

このようにして、3 週程度にわたってプログラミングに関する基本的事項を解説した後、文献 [9] をテキストとして、上記で述べた授業計画(2)~(12)に従って、以下の内容で講義した。

- (1) 順次構造による簡単な計算
  - ① 簡単な計算
  - ② 変数の使い方
  - ③ 関数計算
- (2) 選択構造
  - ① If 文による選択処理
  - ② Case による選択処理
- (3) 繰返し構造
  - ① For 文による繰返し処理
  - ② Do-while 文による繰返し処理
  - ③ Do-until 文による繰返し処理
- (4) 配列の利用
  - ① 一次元配列
  - ② 二次元配列
  - ③ 配列を使った問題例

(5) サブルーチンについて

- ① サブルーチンとは何か
- ② サブルーチンを使った問題例

なお、テキストの各例題のアルゴリズムは、フローチャートを書いて説明した。

平成12年度生には上記の内容で講義したが、少し消化不良であったので、平成13年度生にはアルゴリズムの特性と 7 つの例題の分量を減らし、文献 [9] をメインにして講義を行った。

## 5. データ構造

平成 8 年度・9 年度に担当した科目である。電子情報システム学科情報処理コースの 2 年生前期学生を対象にした科目である。

授業の概要は『効率のよいプログラムを作るためには、アルゴリズムだけでなく、その処理対象のデータをコンピュータ内でどう表現しどう扱うかも重要な問題となる。そこでこの講義では、まずプログラミング言語 (例えば Pascal) が直接提供する基本データ型について概要を講述する。さらに、それを土台にして自由に定義できる構造型のデータであるリスト、キュー、スタック、木構造、等の基本的な性質、定義と操作の方法、応用、等について講述する。』<sup>7)</sup>である。

各週の授業計画は以下のとおりである。<sup>7)</sup>

- (1) データ構造とは何か—計算量との関係など
- (2) 基本データ型：主に単純型データの整理
- (3) 基本データ型：主に構造型のデータの整理
- (4) ポインタ変数、動的変数、文字列など
- (5) スタック：概念、基本演算、応用など
- (6) キュー：概念、基本演算、応用など
- (7) 線形リスト：概念、基本演算、応用など
- (8) その他のリスト：概念、基本演算、応用など
- (9) ハッシング：概念、基本演算、応用など
- (10) 木構造 I：概念、基本演算など
- (11) 木構造 II：2 分木探索、平衡木など
- (12) グラフ：概念、基本演算、応用など

なおこの時期は、前期・後期の講義は 12 週 (プラス、定期試験、最追試験) となっている。

講義内容は以下のとおりである。

まず、最初の 2 週程度を使って、「データ構造の位置づけ」について説明した。内容は次のとおりである。

コンピュータで問題を処理するためには、処理の対象をコンピュータで扱えるようにモデル化 (=データ

構造化)し、処理の具体的な手続きを見だし(=アルゴリズムの構成)、それらをコンピュータが受け付けられるような記述に変換する(=プログラミング)という手順をとる。従って、問題が扱う事象をどのようにデータ構造化するかが、処理効率やアルゴリズムの作り方に影響を与えるのである。

また、データ構造のよい例・悪い例ということで、文献[5]の「家計図データ」とそれを改良したものをを用いて、両者の同じ質問に対する解答を見つける手間の違いについて説明した。さらに、計算量・領域複雑度という概念についても説明した。例題として「検索」をあげ、逐次検索と2分検索の2つの方法を示し、それぞれの比較の最大の回数を求めて、計算量の違いを示した。

その後、3週目以降は、テキストとして、文献[8]を参考にして作成したプリントを適宜配布した。

上記で述べた授業計画(2)~(4)に従って、3~6週目までは、以下の内容で講義した。

#### (1) 基本データ型 [単純型データ]

- ① 標準基本形…整数型、実数型、論理型、文字型、それぞれの型の例題
- ② 区間型…データの宣言方法と利点、例題
- ③ 配列…1次元・2次元・3次元の各配列と各要素の参照方法、例題
- ④ 列挙型…データの宣言方法と利点、例題

#### (2) 基本データ型 [構造型データ]

- ① Packed 配列…配列との相違点、データの宣言方法と利点、相互のデータの移動方法、例題
- ② レコード型…基になるデータのイメージ、データの宣言方法、レコードの各フィールドの参照方法、例題
- ③ with 文…with 文の意味、データの宣言方法、例題
- ④ 可変レコード型…レコード型との相違点、データの宣言方法、例題

#### (3) 動的データ構造 - I

- ① 動的変数とは…動的変数の意味とその利点、例題
- ② ポインタ変数…ポインタ変数の意味、データの宣言方法、指し示すデータの参照方法、ポインタ変数の生成と解放の方法、レコード型のポインタ変数、それぞれの例題

なお、いずれの場合のデータ構造も、宣言方法と手続き部での使用方法を豊富な例題を用いて、詳細に説

明した。ここまでは、プログラムの宣言部のところで宣言することによって実現できる、いわば、物理的なデータ構造である。

次は、そのようなデータ構造を利用して行う基本的でかつ重要な操作や、プログラムを実行させることによって実現できる論理的なデータ構造について、上記で述べた授業計画(5)(6)に従って、7~9週目で、以下の内容で講義した。

#### (1) ストリング-文字列の操作

- ① ストリング(文字列)の操作…長さを求めるアルゴリズム、特定の位置の文字を取り出すアルゴリズム、結合のアルゴリズム、特定の位置から特定の長さの部分文字列を取り出すアルゴリズム、例題
- ② パターン・マッチング…パターン・マッチングとは何か、単純なアルゴリズムとその例題、クヌーク・モリス・プラット(KMP)のアルゴリズムとその原理及びその例題

#### (2) スタック

- ① スタック…スタックとは何か、スタックへの挿入、スタックからの削除、それぞれの例題
- ② スタックの応用としての回文…回文とは何か、回文かどうかの判定のアルゴリズム、例題
- ③ スタックの物理的構造…配列による構造

#### (3) キュー

- ① キュー…キューとは何か、キューへの挿入、キューからの削除、それぞれの例題、
- ② キューの物理的構造…配列による構造

最後に、「動的データ構造-II」として、上記で述べた授業計画(7)~(11)に従って、10~12週目で、以下の内容で講義した。

#### (1) 線形リスト

- ① 定義と概念…形状(データ部とポインタ部)、アクセス方法、例題、利点と欠点
- ② データの挿入と削除…例題
- ③ Pascal による実現…新しい線形リストの作成、データの挿入・削除
- ④ 線形リストの型…単方向リスト、環状リスト、双方向リスト、例題

#### (2) 木構造

- ① 定義と概念…木の定義、順序木、木に関する諸定義、リスト構造による表現、例題
- ② 2分木…2分木の定義、リスト構造による表現、例題

以上、2年間の講義を通して、ほぼ当初の授業計画どおりに進行することができたが、最初の「データ構造とは何か」を丁寧に、時間を懸けすぎたため、授業計画の後半の(9)と(12)は積み残しになってしまった。

また、データ構造とアルゴリズムに関する出版物は各出版社からたくさん出ているが、そのほとんどが本学の学生のテキストとするにはレベルが高すぎる内容であったが、文献[8]は、「データ構造演習」という関連科目で当時使用していたプログラミング言語である Pascal (現在は C 言語が使用されている) と関連づけた記述が多いため、本学の学生のテキストとして選定したのであるが、やはり使いづらいテキストであった。今後は自作のテキストを用意する必要があるように思われる。

## 6. Visual Basic

Visual Basic を扱った科目のうち、筆者が担当したものの科目名としては、「プログラミング基礎演習/Visual Basic」がある。平成12年度から担当している科目である。OA 情報システム学科ビジネス情報処理コースの1年生後期学生を対象にした科目である。

「プログラミング基礎」と2コマ続きで開講している。演習の概要は、『「プログラミング基礎」での学習事項を、パソコンにより順次演習していきます。演習を通してそれらの理解を深め、各処理内容にさまざまな工夫を加えることができるようになっています。初心者が違和感なくプログラミングに親しめるような演習形態にしていきたいと考えています。』<sup>7)</sup>である。

各週の授業計画は、第4章の各週の授業計画(1)~(13)に対応したものである。

毎週の90分を前半45分と後半45分に分け、前半の45分は文献[9]をテキストとして、以下の内容の例題を、モニター画面を通して、教師用パソコンで実演しながら説明した。

- (1) Visual Basic の概要
  - ① Visual Basic の概要
  - ② Visual Basic の起動・終了・画面
  - ③ ウィンドウの操作
- (2) 文字を表示するプログラム
  - ① 文字の表示
  - ② プログラムの実行・終了・保存・呼出し
  - ③ 簡単なコードの記述
- (3) 順次構造による簡単な計算

- ① 簡単な計算
- ② 変数の使い方
- ③ 関数計算
- (4) 選択構造
  - ① If 文による選択処理
  - ② Select Case 文による選択処理
  - ③ オプションボタンとチェックボックスの使い方
- (5) 繰返し構造
  - ① For...Next 制御構造による繰返し処理
  - ② Do...Loop 構造による繰返し処理
- (6) 配列の利用
  - ① 一次元配列
  - ② 二次元配列
  - ③ 配列を使ったプログラム例
- (7) ジェネラルプロシージャ
  - ① Sub プロシージャ
  - ② Function プロシージャ

ただし、Visual Basic というプログラミング言語は Pascal や COBOL のような従来の言語に比べて、画面の設計や操作がはるかに簡単であり、例題にはそれらの要素もふんだんに取り入れているので、併せて説明する必要があった。

また、後半の45分は演習時間に充てた。演習問題は、「順次構造」「選択構造」「繰返し構造」に関する内容の必修問題(完成度はどうあれ、絶対に提出を義務づけた問題)を各1題と、テキストの1~7章の各章の章末問題をオプション(提出は任意、ただし提出すると1題につき何点ずつかのボーナスを加点する)とした。

2年間、このような演習形態をとったが、プログラミングに興味がある学生が少ないせいか、オプション問題まで積極的に取り組む学生が非常に少なく(テキストには解答が付いているにもかかわらず!)、必修問題のみしか提出しなかった学生がほとんどであったのは残念である。

## 7. COBOL-I

ここでは、COBOL を扱った科目で筆者が担当した科目のうちの、COBOL の基礎的事項を扱う科目群である、「プログラミング及び演習 I」、「プログラミング及び演習 III」、「プログラミング及び演習 IV-1」、「プログラミング/COBOL」及び「プログラミング演習/COBOL」という科目について述べる。平成6年度から13年度まで、毎年度担当している科目であ

る。

「プログラミング及び演習Ⅳ-1」は、電子情報システム学科情報処理コースの1年生後期学生を対象にした科目であり、「プログラミング及び演習Ⅰ」は、OA情報システム学科の1年生後期学生を対象にした科目である。また、「プログラミング及び演習Ⅲ」は、電子情報システム学科情報処理コースの2年生前期学生を対象にした科目であり、「プログラミング/COBOL」, 「プログラミング演習/COBOL」は、それぞれOA情報システム学科の2年生前期学生を対象にした科目である。

それらの講義科目の概要は若干の違いはあるが、概ね『事務処理用プログラミング言語として長い歴史を持ち、実務的にも最も広く使用されているCOBOLについて、その特徴、歴史及び文法(COBOL85に準拠)を体系的に学習する。プログラム設計とプログラムの構造化についても学ぶ。また、事務処理に必要な基本処理パターンである「媒体変換」「印刷行数の制御」「データチェック」「グループトータル」などのアルゴリズムについて学習する。』である。(詳細は文献[7]を参照すること。)

各週の講義の授業計画は以下のとおりである。<sup>7)</sup>

- (1) アルゴリズムと一般理解
- (2) 構造化設計とアルゴリズムの基本パターン
- (3) プログラミングとその手段、及びフローチャート
- (4) COBOL 文法 (DIVISION と SECTION)
- (5) COBOL 文法 (定義, 命令)
- (6) 媒体変換と簡単な集計, 帳票, 印刷行数の制御 (問題分析, 領域図, アルゴリズム)
- (7) データチェックとグループトータル (処理方法とそのアルゴリズムの解説)

また、演習科目の概要も若干の違いはあるが、概ね『上記の講義科目で学ぶ様々な処理方法と、アルゴリズムやCOBOL文法の理解を深めるために、「媒体変換」「印刷行数の制御」「データチェック」「グループトータル」を課題としたプログラム仕様書を提示し、ネットワーク環境下でプログラミング演習を行う。』である。(詳細は文献[7]を参照すること。)

各週の演習の授業計画は以下のとおりである。<sup>7)</sup>

- (1) サンプルプログラムによる端末操作
- (2) 演習問題(1): 与えられたプログラムの入力, 文法デバッグ, 実行, アルゴリズムのデバッグ
- (3) 演習問題(2): 媒体変換と簡単な集計, 帳票, 印

#### 刷行数の制御

##### (4)演習問題(3): グループトータル

テキストとして文献[2]や[3]を指定したが、世間一般で使用されているCOBOL85と、本学の演習室で使用されているCOBOLシステムに、ファイルの対応関係に関する部分(Select句)など、若干の相違点があるため、それに合わせてテキストの内容を再構成したプリントを配布した。また、この授業では、COBOLが得意とする事務処理のプログラムのうちの最も基礎的なものについて説明した。

講義内容は以下のとおりである。

まず、第1週を中心に、「プログラミング基礎」で学んだアルゴリズムに関する復習を簡単に行い、高級言語(COBOLもこれに含まれる)・機械語とコンパイラやインタプリタの関連についての復習を行った(COBOLはコンパイラ型の言語である)。

次に、第2~5週程度で、「最も単純な一覧表の作成」というテーマで、文献[3]の第5章1節の「例題1」のアルゴリズムとプログラムを以下の要領で解説した。

##### (1) プログラミングの流れ

- ① 「例題1」の説明…使用するファイル、データの形式、入力例、処理内容と処理条件など。
- ② 問題分析と領域図の作成…プログラムで使用するデータの形式、その項目の名前や型と桁数の決定など。
- ③ フローチャートの作成…②で決定した名前と領域図で示したデータフローを基に、処理のアルゴリズムを図式化する。
- ④ COBOLのプログラムの構造…各DIVISION, SECTIONでの記述方法。

##### (2) コーディングについての注意事項

- ① 文字集合とコーディングの文字
- ② 予約語と利用者語
- ③ COBOLプログラムに関する注意点

##### (3) コーディングの実際

- ① 本学演習室用の「例題1」のプログラム例
- ② (1)-①の入力例に対応する出力例
- ③ ①のプログラム例の詳細説明
- ④ 関連事項…うちPERFORM文と、そとPERFORM文。

さらに、第6~8週程度で、「機能を付加した一覧表の作成」というテーマで、文献[3]の第5章2節の「例題2」のアルゴリズムとプログラムを以下の要



領で解説した。

(1) プログラミングの流れ

- ① 「例題2」の説明…使用するファイル、データの形式、入力例、処理内容と処理条件など。
- ② 問題分析と領域図の作成…プログラムで使用するデータの形式、その項目の名前や型と桁数の決定など。
- ③ フローチャートの作成…②で決定した名前と領域図で示したデータフローを基に、処理のアルゴリズムを図式化する。

(2) コーディングの実際

- ① 本学演習室用の「例題2」のプログラム例
- ② (1)-①の入力例に対応する出力例
- ③ ①のプログラム例の詳細説明

第9～10週あたりで、例題を離れて、「DATA DIVISION の詳説」、「PROCEDURE DIVISION の補足」というテーマで、文献[4]も参考にしながら、以下の要領で解説した。

(1) ファイル・ブロック・レコードの関係

- ① ファイルとレコードの概念
- ② 磁気テープや磁気ディスク上のファイル

(2) レコードとデータ項目との関係

- ① レコードとデータ項目(データ名)の説明とその例題。
- ② 基本項目・集団項目とレベル番号の説明とその例題。
- ③ PICTURE 句・VALUE 句の説明とその例題。

(3) 数字項目・英数字項目の説明と集団項目の取り扱い、その例題。

(4) 算術文と転送の規則…算術文とその例題、転送(MOVE 文)のさまざまな規則とその例題。

(5) 表意定義と編集機能の説明と例題。

(6) 印刷行数とページ替え…行送り・改ページ。

最後に、第11～13週目程度で、「一次コントロールをもつ一覧表の作成」というテーマで、文献[3]の第6章1節の「例題3」のアルゴリズムとプログラムを以下の要領で解説した。ただし、文献[3]ではソートの処理も含まれているが、それは省き、データはコントロールの対象になる項目の順に並んでいるものとした。解説の要領は「例題2」と同様である。なお、ソートに関しては、以下の第8章で扱う科目群の中で解説することにした。

演習内容は以下のとおりである。

まず第1週は、サンプルプログラムを通して、演習

室のマシン環境、プログラムファイル・データファイルの作成方法、コンパイル・リンク・実行の方法等を解説し、実際に、サンプルプログラムの入力から実行までを体験させた。

次に、第2～4週程度を「演習問題1」に充てた。この問題は上記の演習問題(1)に対応するものであり、筆者の側でプログラムとデータを与えており、学生はそれらをプログラムファイルとデータファイルに分けて入力し、コンパイル・リンク・実行をするだけである。ただ、マシン環境に慣れないのと英文によるシンタックスエラーや実行エラーの表示に戸惑い、なかなか簡単に完成させることができなかった学生が多かったのである。

また、第5～7週程度を「演習問題2」に充てた。この問題は、上記の演習問題(2)の前半部分に対応するものであり、上記の「最も単純な一覧表の作成」というテーマの「例題1」や「演習問題1」と同様のアルゴリズムに従えば簡単にプログラムを作成することができる問題である。それに気が付いた学生は、「演習問題1」のプログラムをベースに手早く完成させていった。

さらに、第8～10週程度を「演習問題3」に充てた。この問題は、上記の演習問題(2)の後半部分に対応するものであり、上記の「機能を付加した一覧表の作成」というテーマの「例題2」と同様のアルゴリズムに従えば簡単にプログラムを作成することができる問題である。この問題は、「例題2」が「例題1」をベースに機能を付加したのと同様に、「演習問題2」をベースに機能を付加すればよい問題である。この頃になると、一部の学生は、「演習問題2」のときと同様に、「演習問題3」のプログラムをベースに手早く完成させていた。ただ、この段階でも、コンパイルが終わらないまま提出期限を迎えてしまう学生も結構いた。

最後に、第11～13週を「演習問題4」として、グループトータルの問題に充てた。この問題は、上記の演習問題(3)に対応するものであり、上記の「一次コントロールをもつ一覧表の作成」というテーマの「例題3」と同様のアルゴリズムに従えば簡単にプログラムを作成することができる問題である。この問題になると、アルゴリズムを理解できてプログラムを完成させることができた学生は少なくなってしまう。

以上、8年間の講義を通して、最初の数年は、ほぼ当初の授業計画どおりに進行させてきたが、「例題3」のところから急に学生の理解が悪くなっていたので、

ここ数年は「例題3」の部分のカットし、「例題2」までを丁寧に解説するようにしている。

また、演習問題も授業計画では3題としていたが、当初計画の演習問題(2)を2題に分割した方が、講義の進め方と対応がとれて、学生が理解しやすかったため、演習問題を4題としていた。しかし、「例題3」の部分のカットに伴い、グループトータルの演習問題もカットし、ここ数年は当初計画の演習問題(2)まで計3題の演習問題となっている。

また、演習時の問題点として、シンタックスエラーや実行エラーの英文表示がわからない学生が多いため、同じエラーで別々の学生に何度も対応しなければならなかったり、1クラスの人数が多すぎ（常時70人前後、100人を超えることもあった）て1コマの中で全員に対応できないことが多かった。

改良点としては、講義の時に配布するプリントが多いため、平成13年度から web からプリントを引き出せるようにした。

## 8. COBOL—II

ここでは、COBOL を扱った科目で筆者が担当した科目のうちの、上記第7章で述べた科目群の延長線上にある内容を扱う科目群として位置づけられている、「プログラミング及び演習Ⅳ-2」、「プログラミング応用演習/COBOL」という科目について述べる。それぞれ平成6年度・7年度と、10年度・11年度に担当した科目である。

「プログラミング及び演習Ⅳ-2」は、電子情報システム学科情報処理コースの2年生前期学生を、「プログラミング応用演習/COBOL」は、OA システム学科の2年生後期をそれぞれ対象にした科目である。

それらの講義内容は若干の違いはあるが、概ね『引き続き、表と表操作、ファイルの概念及びファイル処理形態について学習するとともに、それらを課題としたプログラミング演習も行う。』である。（詳細は文献[7]を参照すること。）

各週の授業計画は以下のとおりである。<sup>7)</sup>

- (1) 整列・分類
- (2) 表の定義と OCCURS 句
- (3) 表の検索と SEARCH 命令
- (4) ファイルの概念とファイル編成方法
- (5) ファイル処理のアルゴリズム（照合）
- (6) まとめ

各週の演習の授業計画は以下のとおりである。<sup>7)</sup>

- (1) 演習問題(1): 整列・分類
- (2) 演習問題(2): 表操作による度数分布表とヒストグラムの作成
- (3) 演習問題(3): ファイル処理

テキストとして文献[2]や[3]を指定したが、上記第7章と同様に、テキストの内容を再構成したプリントを配布した。

講義内容は以下のとおりである。

まず、第1～3週程度で、「整列・分類」というテーマで、文献[2]の第2章1節の「例題1」のデータ形式を例題に、アルゴリズムとプログラムを以下の要領で解説した。

- (1) プログラミングの流れ
  - ① 整列・分類（ソート）とは何か。
  - ② 問題の説明…使用するファイル、データの形式、入力例、処理内容と処理条件など。
  - ③ フローチャートの作成…②で決定したファイル名を基に、処理のアルゴリズムを図式化する。

### (2) コーディングの実際

- ① 本学演習室用のプログラム例
- ② (1)–①の入力例に対応する出力例
- ③ ①のプログラム例の詳細説明

### (3) ソート文の一般形

次に、第4～9週程度で、「表操作」というテーマで、『学生  $n$  人の数学、英語、物理の点数データから度数分布表を作成する。ただし、 $1 \leq n \leq 100$ 、得点の範囲は0～100点であり、度数分布表の階級の範囲は0点～9点、10点～19点、20点～29点、…80点～89点、90点～100点とする。』という例題を用いて、アルゴリズムとプログラムを以下の要領で解説した。

- (1) 表の考え方と添字の概念
- (2) 表の定義
  - ① OCCURS 句（反復）と表要素の概念
  - ② REDEFINE 句（再定義）と表の初期値設定
- (3) 問題（上記の例題）の説明
  - ① 入力例とその論理構造
  - ② 入力例より出力される度数分布表の例とその論理構造
- (4) データ構造
  - ① 入力データの物理的なデータ形式。
  - ② 度数分布表作成のために操作する表の物理的なデータ形式
  - ③ 度数分布表の物理的なデータ形式

- (5) 度数分布表の作り方
  - ① 操作する表の初期値設定
  - ② 操作する表の値の更新方法
- (6) コーディングの実際
  - ① 本学演習室用のプログラム例
  - ② 実際の入力例
  - ③ ②の入力例に対応する出力例
  - ④ ①のプログラム例の詳細説明

最後に、第10～13週で、「ファイル処理」というテーマで、文献[2]の第6章6.1～6.2節を以下の要領で解説した。

- (1) ファイル処理とは何か
  - ① マスターファイルとトランザクションファイル
  - ② ファイルの編成方法…順次編成, 相対編成, 索引人事編成, VSAM。
  - ③ ファイル処理の形態…照合, 併合, 更新, 抽出, 分配, 生成。
- (2) ファイル処理の基本形
  - ① 例題…文献[2]の第6章6.2節の「例題7」の内容の説明
  - ② 上記①のアルゴリズムの説明

演習内容は以下のとおりである。

まず、第1～3週程度を「演習問題1」に充てた。この問題は上記の演習問題(1)に対応するものであり、上記の「整列・分類」というテーマの「例題1」と同様のアルゴリズムに従えば簡単にプログラムを作成することができる問題である。それに気が付いた学生は、「演習問題1」のプログラムをベースに手早く完成させていった。

また、第4～10週程度を「演習問題2」に充てた。この問題は、上記の演習問題(2)に対応するものであり、上記の「表操作」というテーマの「例題」と同様のアルゴリズムに従えば、度数分布表作成の部分については、簡単にプログラムを作成することができる問題である。それに気が付いた学生は、「例題」のプログラムをベースに手早く完成させていった。しかし、ヒストグラムについては、完成に到る学生は少なかった。与えられた例題に極めて近い内容のものはこなせても、思考を要する問題になると対応できなくなるようである。

最後に、第11～13週を「演習問題3」に充てた。この問題は、上記の演習問題(3)に対応するものであり、上記の「ファイル処理」というテーマの問題であった

が、上記の「例題7」もアルゴリズムしか説明できなかったため、この演習もプログラム作成まではさせられず、フローチャートの作成と、入力例に対する結果のシミュレーションしかできなかった。

なお、上記第7章の「グループトータルとその演習問題」を上記第7章の科目群の授業からカットするようになってからは、本章の科目群の最初に実施するようになった。また、それに伴い、「ファイル処理とその演習問題」は実施できなくなった。

また、平成6～7年のプログラミング及び演習Ⅳ-2については、電子情報システム学科の学生はかなりの数の学生が受講していたが、平成8年度に、この授業内容のCOBOLに関する科目が廃止された。平成10年度・11年度の「プログラミング応用演習/COBOL」については、OA情報システム学科の学生に開講していた科目であったが、受講する学生の数が大変少なく、プログラミングに興味がある学生がほとんどおらず、卒業要件単位を充たすために受講する学生が多かったため、学習効果が上らなかった。従って、カリキュラム改訂と同時にこの科目を廃止とした。

## 9. Pascal

Pascalを扱った科目のうち、筆者が担当したものの科目名としては、「データ構造演習」がある。平成8年度・9年度に担当した科目である。電子情報システム学科情報処理コースの2年生前期学生を対象にした科目である。

「データ構造」と2コマ続きで開講している。演習の概要は、『この演習では、「データ構造」で学んだいろいろなデータ構造をコンピュータ上に実際に作り出して操作させ、その意味を理解させる。その場合に、プログラミング言語が直接提供する基本データ型である単純型、ポインタ型、構造型に関する演習問題をまず考えさせる。次に、それらを組み合わせて、任意のデータ構造を作り出せることを学ばせる。なお、この演習のプログラミング言語としては、入門用プログラミング言語のPascalを使う。』<sup>7)</sup>である。

実際の演習問題としては、以下の4題である。

- (1) 与えられたPascalプログラムを入力して実行させる。
- (2) フィボナッチ数列の最初の50個を求めて出力させる。ただし、配列を使用したプログラムと使用しないプログラムをそれぞれ作る。

(3) レコード型のデータ構造を利用して、打者なら打数と安打から打率を求めて、打率の順にソートさせる、投手なら投球回数と自責点から防御率を求めて、防御率の順にソートさせる。

(4) パターン・マッチングのアルゴリズムを実際に行わせる。ただし、「単純な方法」は必ず提出させ、「KMPの方法」はオプションとする。

上記の演習問題について、(1)は1年生の時の「プログラミング基礎」, 「プログラミング基礎演習 I」等で習得しているはずの Pascal プログラムの操作を、実際にどの程度理解しているかを探るための問題であったが(それによつては以後の演習問題の出題方法を再検討する必要が生じるかもしれない), さすがにこの問題は90分以内にはほぼ全員が完成させていた。従つて、演習問題(2)~(4)は当初の予定どおり、講義で説明したデータ構造を使ってデータを処理するプログラムを作成する問題とした。

演習問題(2)は、配列を使って求める方は概ね完成させていたが(こちらが演習の主題である), 配列を使わない方は、やはりあまり完成までこぎつけていなかった。

演習問題(3)は、データが比較的馴染みのあるものなので、こちらが思ったよりは理解できていたため、かなりの学生が完成にこぎつけていた。

演習問題(4)については、「単純な方法」については、アルゴリズムは説明していたが、実際にプログラムにするのが難しかったようで、配列を利用するところまでは、ほとんどの学生がわかっていたが、For 文の入れ子構造の作り方を理解している学生は少なかった。しかし、仲間で助け合つてどうにか完成にはこぎつけていた。オプション問題にはほとんどの学生が手もつけていなかったことが残念である。

## 10. 結 言

本論文では、筆者が1994年4月に福岡工業短期大学に赴任してから、2001年度までに担当したプログラミング言語関連科目についての講義・演習の内容を振り返り、検証し、今後の課題を探ってみた。ただ、検証した科目数が多く、ほとんどを各科目、または同じような内容の科目群の講義・演習内容に絞った検証と、そこから導き出された若干の課題を記述したにとどまった。

授業の実施内容等は今後の課題となるが、内容を検

証するためには、自分の講義内容と比較対照できる材料等が必要となる。

プログラミング教育方法論は確立されたわけではないが、さまざまな提案は出されている。例えば、文献 [6] では以下のような10項目の指針が提案されている。

- (1) プログラミングの何を教えるべきか
- (2) プログラムと処理系の教え方
- (3) プログラミング言語の選び方
- (4) プログラム内蔵式の教え方
- (5) メモリー概念の教え方
- (6) 変数の教え方
- (7) 実行順序の制御方法の教え方
- (8) プログラムの作り方の教え方
- (9) デバッグ方法の教え方
- (10) プログラム高速化の教え方

今後の課題は、これらの指針と自分の講義内容を比較・検討し、よい提案は積極的に取り入れて、自分の授業内容の改善に役立て、実状に合わないものは、自らの方法論を確立し、世の中に提案していくことであろう。

最後に、最近プログラミングに興味を示さなくなった学生が目だつて多くなってきている。特に、GUIの要素の強い Visual Basic はともかく、そうでない COBOL については、その傾向が顕著である。そこで、来年度からは COBOL に代わつて、GUIの備わつた Java などの script 系の言語を採用する予定である。この試みについての検証も、今後報告していきたい。

## 参 考 文 献

- [1] 石村貞夫: “すぐわかる統計解析”, (第1版第25刷), 東京図書, (2001).
- [2] 小島崇弘. 内野明. 町田欣弥: “例解構造化 COBOL”, (改訂版第4刷), 実教出版, (1996).
- [3] 遠山暁. 海老澤榮一. 村田潔: “ストラクチャードプログラミング COBOL”, (改訂版第6刷), 実教出版, (2000).
- [4] 中山二夫, 太田宗雄: “JIS 準拠 COBOL 文法”, (第2版第17刷), 共立出版, (1993).
- [5] 萩原宏, 西原清一: “現代データ構造とプログラム技法”, (第1版第1刷), オーム社出版局, (1987).
- [6] 平林雅英: “パソコン・インターネットの教え方”, (第1版第2刷), 共立出版, (2001).

- [7] 福岡工業短期大学編：“福岡工業短期大学学生  
便覧”，福岡工業短期大学，(1994-2001).
- [8] 村本正生：“データ構造とPascal プログラミン  
グ”，(第1版第2刷)，近代科学社，(1995).
- [9] 若山芳三郎：“学生のための Visual Basic”，(第  
1版第4刷)，東京電機大学出版局，(2000).