

# プログラミング学習における 知識獲得の場に関する考察

加藤 武信, 尾辻 斗志

## 目 次

はじめに

1. 講義体系の設計
2. コンピュータ実習のための準備段階
3. コンピュータ実習による処理手順検証過程
4. 知識獲得の場と理解度に関する実態分析
5. 知識獲得の場としての CAI の検討

おわりに

## はじめに

情報化と国際化という時代の要請は大学カリキュラムにも反映しつつあり、比較的新しいデータベースや表計算の分野でのコンピュータの活用が試みられている今日であるが、プログラミング実習教育の導入については、各大学とも比較的長い歴史と経験をもっている。

これから考察の対象にする情報講座においても約 10 年ほどの COBOL 実習の経験があるが、情報化の拡充に伴う問題点も多数存在するので、この時点でこれまでの実習状況を整理し実態を分析することによりコンピュータ実習の特性を把握し、問題点の明確化とそれへの対策について検討し、今後に生かしたいと思う。本稿は、実習教育に指導員制を導入した経緯や理念についてまとめた別稿<sup>(注1)</sup>の続編であり、そこで述べた概要の一部を、さらに詳細に展開したものである。本稿において、実習現場に導入した指導員制が受講者の知識獲得にどのように寄与するのか、さらに、コンピュータ実習実績が、知識の明確化にどれほど貢献するかを中心にし、それに関連する問題も含めて実態を踏まえた考察を行う。

## 1. 講義体系の設計

### (1) システムの全般的理解

本稿の対象であるコンピュータ概論は、コンピュータの機能や適用分野などについて幅広く解説し、その背後にある情報とシステムに関する基礎概念を理解することを目的とする。そしてコ

コンピュータの本質や機能を明確に理解するための手段としてコンピュータ実習を導入しているが、実習を行うことによって、受講生が情報設備に触れ、考え、議論をし、確認する知識獲得の場を設定することが重要であると考え。この講座は、正規の授業時間（週1回、通年）においては座学形式の講義を行い、コンピュータ実習は、課外の空き時間を利用してオープンショップ方式（実習生が機械室に自由に入り、セルフサービスで実習を行う方式）で実施している。

ここでは第1表に示すように、システム設計を含めた広義の概念でプログラミングを理解することにする。実践界においては、システム構築にかかわる適用分野は多岐にわたり、その対象となる業務は大規模かつ複雑であるので、プログラミングの上位にシステム設計の段階をおき精巧な手順により実践システムを設計する。それに対して、大学の基礎講座で行う実習課題は極めて小規模であるため、システム設計を特別の段階として認識する必要はなく、それを含んだ広義の概念としてプログラミングを定義することにより、システムの設計、製造ならびにその運用の全

第1表 システム構築の過程

(1:必要な知識)

区 分	プログラミングの過程	知 識 レ ベ ル				
		1	2	3	4	5
シ ス テ ム 構 築	システム設計					
	問題の解析 出力設計 入力設計 プロセス設計 コード設計 仕様書の作成		1	1	1	1
シ ス テ ム 製 造	システム構築					
	処理手順設定過程 処理技法の決定 処理手順の詳細化		1	1	1	1
	文法表現過程 言語の文法に即した記述	1	1	1	1	1
	処理手順検証過程 コンピュータ操作 操作上のトラブル対策 エラーメッセージの意味理解 文法ミスの発見と訂正 論理ミスの発見と訂正 異常終了対策 出力結果の判断	1	1	1	1	1
	操作手引書の作成				1	1
システム運用	本番処理 プログラムの保守					1 1

(注) 広義のプログラミング＝システム構築、狭義のプログラミング＝システム製造

過程を把握し、システム概念を総合的に理解することができる。ただし、ここで考察の対象にしているのは1年生配当の基礎講座であるので、システム設計とシステム運用については、その範囲や手順を示すのみで、専門的な内容に深く立ち入ることはしない。むしろ力点をおくのはシステム製造、すなわち、コンピュータ・プログラムを作成する過程である。

## (2) コンピュータ認識の側面

プログラミング実習においては、コンピュータをいくつかの側面から認識することができる。

まず、コンピュータをその本質であるデータ処理システムとして認識する。この側面は、入力データを加工し、それより高い価値の出力データを産出するシステムとしての理解である。その出力データは行動系の制御や意思決定などの上位システムに組み込まれ、産業界において業務処理や管理の自動化に応用されている。本講座の実習課題を解くことは、このデータ処理の仕組みを構築することを意味する。

コンピュータ・テクノロジーは、現代社会を支える巨大技術の1つであり、銀行オンライン・システムや工場オートメーションにおいて既に高度に構築されている。その中核には、ネットワークにより結合された種々の規模のコンピュータ群を基盤として各種データベースによって統合された技術システムがある。このような技術システムは、業務処理や管理方式を変革し組織における協働体系を変えていく。

文系においては、技術システムよりもむしろ後者の人間システムに関心があるが、その人間システムへの影響を正しく認識するためにも変革の中心的役割を担うコンピュータおよび技術システムの適用分野に与える多くの変革要因を理解するための基礎学習として、コンピュータの本質を理解することは必要である。実習教育の目的として、コンピュータの現時点の技術レベルを明確に理解することが含まれるので、現在、産業界に最も普及している COBOL 言語によるプログラミング実習を大型コンピュータを用いて実施している。

つぎに、コンピュータそれ自体をシステムとして認識することができる。大型コンピュータを動かす OS (オペレーティング・システム) は複雑であるため、一連の関連知識を理解していなければ独力でそれを使用することが困難である。そのことを裏返してみると、コンピュータの複数の側面を理解してはじめてそれを自在に駆使することができる。プログラミング実習の別の狙いは、プログラムの作成過程全体を意識し、大型コンピュータの利用環境の中でプログラムの作成活動を通じてコンピュータを作動させることによって、それ自体を複雑な統合体として把握することである。初学者は初めから独力でコンピュータを1つのシステムとして認識できないけれども、そのことを種々の場において何度も力説することによって、到達すべき目標を実習者に示し、努力させることができる。

さらに、コンピュータをレポート添削マシンとして認識することができる。論理展開能力の訓練においては、各自が組み立てた論理を検証する過程において論理のエラーを発見し、それを訂正するための手段としてコンピュータを使用する。その場合のコンピュータの役割は、実習者の与えるプログラムの論理を忠実かつ迅速に実行し、その実行証跡を出力することであるが、そのことは実習生のレポートをコンピュータが添削することを意味する。すなわち実習生は、コンピュータの出力が課題の正解条件をクリアしているか否かの判定を容易に行うことができ、また正解を得るまで何度も論理の修正と結果の確認を繰り返すことができる。

このようにコンピュータをレポート添削マシンとして位置づける。実習生がレポート（プログラム）をそれに入力しさえすれば、それは苦情の一言もなく何度でも忠実に添削してくれる教師に変身する。

### （3）受講生の行動特性と実習現場における知識獲得の場の設定

講義と実習における受講生の行動特性を比較してみよう。講義は、教師の方針に基づいて該当分野の知識を体系的に受講生に伝達する。教師は、一方においては受講生の種々の能力レベルその他の状況を配慮しながら多くの受講生の知的満足の実現を目指し、他方では必要と判断する目標段階の知識内容まで強引に到達させようとする。ここで考察の対象となっている講義の中心部分は、第1表に示した内容をトップダウン方式で解説する。すなわち講義は、抽象的な概要把握から具体的な明細理解へ向けて、種々の能力レベルの受講生を考慮しながら教師からの一方通行による知識伝達の場である。したがって、受講生は気楽に受講し、その内容を漠然と理解しながら、各自の現在の能力レベルに応じて知識の吸収をはかる。それは、実習に比べて受講生を受容する範囲が広く、実習で自信を無くした受講生の気持をリフレッシュする場でもある。

実習の場合の受講生は、自分の解答の不備部分を自分で訂正するという、いわば自身との格闘である。すなわち、受講生が不勉強であればあるほどその解答にはミスが混入し、その発見と訂正には多くの時間と労力を注がねばならない。さらに、第1表で明確なように、初学者は実習を独力で行うだけの能力レベルまで至っていない。100%の厳密性が要求されるコンピュータ実習においては、自己の設定した論理の不備を指摘される都度コンピュータが自分の思い通りにならないことを知り、自己の能力の無さを痛感し、それに対抗できずに急速に自信と興味を無くして実習現場から逃避しようとする場合が多い。そのような挫折感に耐えながら実習を続行することによって、課題を正解に導くための種々の技法を少しずつ身につける。そして、自身の思い込みがいかにも多く、それに気付くのがいかにも遅く、それを訂正する作業がいかにも大変であるかを痛感する度合いが高い受講生ほど、プログラミングをマスターする道を直進していることになる。

このような初学者の逃避を可能な限り防止するためには、実習現場において実習生が困惑する

ような状況に対して積極的に対応するためのサポート体制がなければならない。すなわち、実習生の設定する論理に柔軟に対応するだけの能力を持ち、その遭遇する多くの技術的障害を除去し、さらにそれを激励するサポート体制が必要である。実習室にはいつもサポート能力をもつ何人かの利用者がいて、初学者がいつでも気軽に質問できる環境が確立していれば、実習室に抱くプレッシャーを少しでも緩和することができる。コンピュータに関する技術と知識は、頻繁に活用することによって得られる経験で裏打ちされてはじめて実習現場で威力を発揮するものである。地道な努力により集積した知識を他人の技術障害の解決に役立てることは、さらに経験を豊富にすることでもある。そのような意味からも、中級程度の能力のある人は、積極的に初学者を指導することによってその知識を確実なものにでき、指導をする人もそれを受ける人も共に恩恵を享受する。

このような互恵的な知識交換の場の設定は、融通の利かないコンピュータに対抗するために人間の協力体制を確立することであり、それによって実習室の活性化を促進することができる。幸いにも我々は、実習現場でそのような役割を果たすサポート要員を育成して指導員組織として編成しその運用を試みてきたが、現在では必要不可欠な存在にまで成長している。その育成および活動過程については既に別稿で紹介したとおりである<sup>(注2)</sup>。

#### (4) 知識獲得の場と指導員組織

初学者を対象とするプログラミング実習においては、初学者の各時点の能力をよく見極め、彼らの若干の努力で達成可能な課題を設定すること、解題に必要な施設・設備が十分に使用できることは言うに及ばず、それに加えて、技術的障害に対処できる懇切丁寧なサポート体制が確立していることが重要である。実習現場においてそのようなサポート活動を行う学生を指導員組織として位置づけ、初学者の重要な知識獲得の場として設定してきた。学生が学生を指導することには賛否両論あると思うが、生起する技術問題をマンツーマンで十分にサポートする体制が充実していればいるほど、初学者の学習効果は大きくなり、それは同時に、指導員レベルの能力をもった学生を育成できることをも意味する。

知識獲得の場として、公式の場である講義のほかに、実習現場における指導員を加えることにする。さらに、学生相互（すなわち友人）の知識交換も奨励している。進度が遅れ気味の場合、その周囲には理解している仲間が既に存在しているので、簡単な事項は友人相互で解決できる。最終的な解答のコピーではなく、課題について相互に議論し、吸収し合う環境を意図的に創出することが実習現場のレベルアップに貢献すると考える。

したがって本講座における知識獲得の場として、講義、友人、個人勉強のほかに、実習現場における指導員を含めた4つを認識する。

## 2. コンピュータ実習のための準備段階

### (1) 課題体系

本講座の実習課題の体系は、第2表に示すように初級例題、初級応用課題、中級課題、上級課題の4つのグループから構成されている。初級例題は、元来1つの課題を7段階で表わしたものであり、解答過程をすべて示した最初の例題から開始して、段階的に条件を追加した上位例題に進むように構成されている。この過程は、上位グループの基礎力を養成するために講義では丁寧に解説し、また実習現場においても多数の受講生に基本的な事柄を反復的に説明するために、負担の大きい部分である。

初級応用課題は、初級例題と同一レベルの類似問題である。受講生は初級例題における解答の基本部分を応用する方式で解題をし、初級例題で学習した知識が理解できたか否かを確認することをその目的とする。さらに、中級、上級と課題が進むにつれて、論理展開が複雑になるよう設定している。この課題体系は、COBOL 訓練用ソフトに組み込まれていて、その追加登録や変更を自由に行うことができる。

第2表 実習課題体系

区分	問題	実習内容
初級例題	1	給与計算処理（データ入力と明細プリント）
	2	給与計算処理（データ・カウント）
	3	給与計算処理（見出し処理）
	4	給与計算処理（支給額平均）
	5	給与計算処理（改頁処理）
	6	給与計算処理（頁付け）
	7	給与計算処理（サブルーティン処理）
初級応用課題	8	数表作成
	9	売上計算
	10	成績処理
	11	給与計算
	12	給与計算（頁付け）
13	数列発生	
中級課題	14	テーブル格納
	15	成績処理（新旧コード判定）
上級課題	16	クラス別成績処理
	17	2次元テーブル操作
	18	内部ソート
	19	外部ソート

## (2) 処理手順設定過程

処理手順設定過程は、システム設計の確定事項を満足させるデータ加工手順を創出するための処理技法を選択し、それに基づいてその手順設定の論理展開を一定の図と記号を用いて表現しながら段階的に詳細化することを目的とする。この過程において、データ処理の主要な局面ごとに複数の技法が存在し、中級程度の実習課題の場合でも3種類から5種類はあるのが通常である。図と記号を用いて具体化する方法には、コンピュータの開発と共に考案され普及した流れ図方式と、1970年代に産業界に普及し、今日の主流であるSP (Structured Programming; 構造化プログラミング) 方式とがある。流れ図方式は、データ加工の流れ(すなわち時間的経過)を素直に図化するので比較的単純な課題を分析するのに適しているが、課題が複雑になるにつれて論理展開の負担が乗算的に増大し理解しにくいプログラムになっていく。それに対してSP方式は、流れ図の欠点を回避し理解しやすいプログラムを作成するための方法で、処理機能を大機能から小機能へと細分化する過程をSP図によって段階的に示す。流れ図は初学者に理解しやすく、またSP図は複雑な課題を解くために有利である。プログラミングの方法として両者は共に関連性があるので、最初に流れ図方式で実習し、基本事項がある程度理解された時点でSP方式に転換する方法を採用している。

## (3) 文法表現過程

文法表現過程(coding)では、詳細流れ図または詳細SP図をプログラミング言語の文法に即した表現に変換する作業を行うが、ここで初めてプログラムの形式となる。本講座ではステップ数の多いことが特徴であるCOBOL言語を採用しているので(簡単なものでも100ステップ程度)、そのままでは多人数の実習には不向きである。

現在稼働している教育ソフトは、実習生が設定する部分を手続部のみに限定し、しかも予めパターン化したカードを組み合わせてそれを完成するようにしている。その方法によれば、通常の場合と比較してプログラム作成の負担を初級例題において全体の5%に、それ以外の課題については10%程度に軽減したので、4単位の講座にCOBOLによる実習を導入することができた。

## 3. コンピュータ実習による処理手順検証過程

第1表に示したように狭義のプログラミングすなわちシステム製造過程は、すでに検討した処理手順設定過程と文法表現過程に加えて、次に考察する処理手順検証過程より構成される。

処理手順検証過程(debugging)は、前の段階で作成したCOBOLプログラムをコンピュータによって機械語に翻訳し、さらにそれを実行して翻訳段階の文法ミスと実行段階の論理ミスを発見し、その原因を突き止めて訂正を行う一連の作業である。この過程は、人間が設定した論理を

コンピュータに実行させて所定の出力結果が得られるか否かによって論理の正当性を判定するのであり、いわゆるそれは、人間とコンピュータによる論理の添削過程である。

コンピュータに与えるデータ加工手順は、曖昧さをまったく許さない100%の正確性を要求するのに対して、能力や使用できる資源に制約のある人間が設定する論理には、ミスが混入しているのが通常である。単純な不注意、知識不足や分析の不十分さ、さらにコミュニケーションの不備などによって発生するミスは、プログラムの中に深く混入しているが、その発見と訂正作業の負担は、課題が複雑になればなるほど大きくなる。

さてプログラマエラーの発見とその訂正作業は、マシーン上と机上の双方で行われる。前者は、プログラムをマシーンで実行させて処理の証跡をとり、場合によってはマシーンと細かいやり取りをしながらエラーを個別的に追跡するのであるが、その方法に偏重すると時間の消費量に比較して生産性は低くなる。後者は、マシーンによる実行の前後で流れ図、SP図あるいは処理の証跡を含む種々の資料を机上において相互関連的に確認したり、分析したりする作業である。机上における総合的な検討は、錯綜した状態を整理し、複数のミスを一挙に除去するために有効である。このようなミスの発見と訂正の作業は、マシーン上と机上との組み合わせと時間配分を適切に行うことが必要であるが、経験が皆無である初学者が独力でそれを完了することは困難なために実習現場における指導者のサポートが不可欠である。

処理手順検証過程には、第1表に示したようにコンピュータ操作から出力結果の判断までの6つの作業があるが、実習現場においてはこれらの側面が同時に発生するので、実習者が独力で目的を遂行するにはその全側面に対処できる能力をもっていなければならない。初学者は、そのような能力が皆無の状態から学習を開始して部分的かつ段階的にそれらの能力を習得していくのである。したがって初学者が実習を円滑に行うためには、未習得の側面を何らかの方法によってサポートする体制を確立する必要がある。

本講座においては、プログラミングにおける受講生の知識獲得の筋道を第1表のように設定した。この表において5段階の知識レベルが示されているが、レベル1から3までは本講座の対象範囲であり、レベル4以上は初期学習を終了した学生の目指す目標である。指導員の知識レベルは、4以上であることが望ましい。表中に数字の1が存在する項目は、そのレベルの知識獲得が進行していることを示す。レベル1に見るように、実習に関して初学者が学習する最初の目標は、「言語の文法に即した表現」と「コンピュータ操作」である。そして、講義および実習の進行につれて課題を系統的に解くことによって種々の知識や技術を獲得しながら、レベルを向上させていく。このようにコンピュータ実習による学習過程は、ボトムアップによって行われる。

第1レベルにおいて、処理手順設定過程と処理手順検証過程に数字1が表示されていない作業項目もコンピュータ実習には加わってくるのであるが、それに対する適切なサポートが得られな



いと、実習生は遭遇するトラブルに対処することができず多大なプレッシャーを受けて実習現場から逃避してしまう。その場合のコンピュータは学生に対する拷問機となり、頭脳のトレーニング・マシンとしての積極的な役割を果たさなくなる。その対策として、学生が安心して実習室に足を向け、努力に相応する成果が得られるような利用体制の確立が望まれる。このような主旨のもとに、COBOL 訓練用教育ソフトと指導員組織を導入し、長期にわたって運用してきたが、その果たした役割は大きいと判断する。

大型コンピュータによる実習の形態は、端末を利用した TSS (会話) 方式が今日の主流であるが、プログラミング学習の初期においては、その方法には難点がある。TSS の場合、プログラムの作成と OS への指令などはすべて端末のキーボードを通して行う必要があるが、それはキーボード操作の訓練、OS コマンドおよび画面操作というコンピュータ操作の負担増加を意味する。すなわち TSS においては、第 1 表における言語の文法表現過程と処理手順検証過程の大半の作業がコンピュータ操作をベースに進められるが、プログラミング学習とコンピュータ操作は本来異なる分野である。これまでの TSS による実習指導の経験では、実習生はプログラミング学習以前にコンピュータ操作の壁に遭遇し、本来の目的を達成しないうちに逃避するケースが多い。我々が使用している教育ソフトはカードをベースとするバッチ方式であるが、それはコンピュータ操作の負担を極力少なくすることができ、また実行証跡に基づいた机上での思考が中心になるように配慮してある。これまでの実績から判断すると、COBOL をこの方式で習得し、プログラミングの基礎力が養成された後で TSS を経験する段階的な学習は、効果的である。

#### 4. 知識獲得の場と理解度に関する実態分析

実習現場に指導員制を導入して初学者の遭遇する技術的障害に積極的に対応し、それを除去することの重要性と有効性に関して、さらに知識の理解度に作用するコンピュータ実習の役割と効果を把握するために、受講生に対する次のようなアンケート調査に基づいて、それらの実態を明らかにしよう。

まず1988年度に実施したアンケート調査において知識獲得の場と実習実績の関係分析から、次のことを把握してみたい。

- ① 受講生が活用する 4 つの知識獲得の場の割合構成から受講者の学習パターンの類型化を行い、各類型の特性を明確にしたい。
- ② 指導員組織に対する評価を、指導員から受けたサポート内容に対する受講生の理解や印象の側面から明らかにする。

さらに、1989年度に実施した調査においては、実習回数と講義内容の理解度に関する次の点の把握を試みる。

- ㊦ 実習回数とテストの成績との関係を明らかにする。
- ㊧ 講義で解説している重要な知識事項の理解が実習回数とどのような関係にあるのかを明らかにする。

### (1) 1988年度調査にみる指導員制の役割と評価

今日まで長年にわたって指導員を配置した実習体制をとってきたが、それが最高に充実したのが1988年度であった。その時の指導員(3, 4年生)は、過去2年間継続的に先輩指導員に協力して組織活動を経験しており、構成員がその年度当初にうまく役割を分担して負荷を平準化し、毎週開催する勉強会によって能力向上をはかり、実習室でのサポートを各自の勉強や経験として生かし、知識と価値を共有した立派な指導員組織が完成していた。また指導員には常駐できる控室があって実習室の管理を行ったので、ほぼ朝9時から夕方19時過ぎまで交替で実習室に詰め、指導員の自主性と創意工夫を尊重した実習室運営を行った。このような環境のなかで、受講生にとって利用しやすくかつ有効な種々の配慮がなされた。受講生の多くは課題のチェックを指導員に求め、指導員は受講生の考え方を尊重した理解しやすい説明を心掛けたので、受講生と指導員との間に良好な信頼関係が確立し、実習室は活気にあふれた。

そのような状況において、最良の指導員組織が知識獲得の場として果たした役割と受講生の指導員への評価がどの程度であるかを明確にすることは、今後の実習体制を拡充するために大いに参考になると判断したので、最終講義日にアンケート調査を行ってその実態把握を試みた。以下、その調査結果に基づいて考察を行う。

#### (a) 知識獲得の場に関する実態分析

知識獲得の場の調査に関して、1988年度の受講生に対して次の質問をした。すなわち受講生各自が4つの知識獲得の場(講義, 指導員, 友人, その他)から1年間に得たと思う知識の割合を、合計で100%になるように回答を求めた(回答者218名)。

ここで設定した4種の知識獲得の場は、なるだけ平均している学習状態が理想であると判断するので、フォーマル度の高い講義と指導員のそれぞれから獲得する知識の割合構成を次のように限定することによって、良好(以下タイプAという)と偏重(以下タイプBという)の2つに分け、学習効果は前者が後者を上回るという仮説を設定した。

- ㊦ 講義と指導員から獲得する割合が双方とも20%以上45%以下である場合をタイプAとする。
- ㊧ 上記に該当しない場合をタイプBとする。

まず、タイプA(回答者の28%)とタイプB(回答者の72%)との比較を中心に第3表と他の関連する図表を併用しながら論議を進めよう。

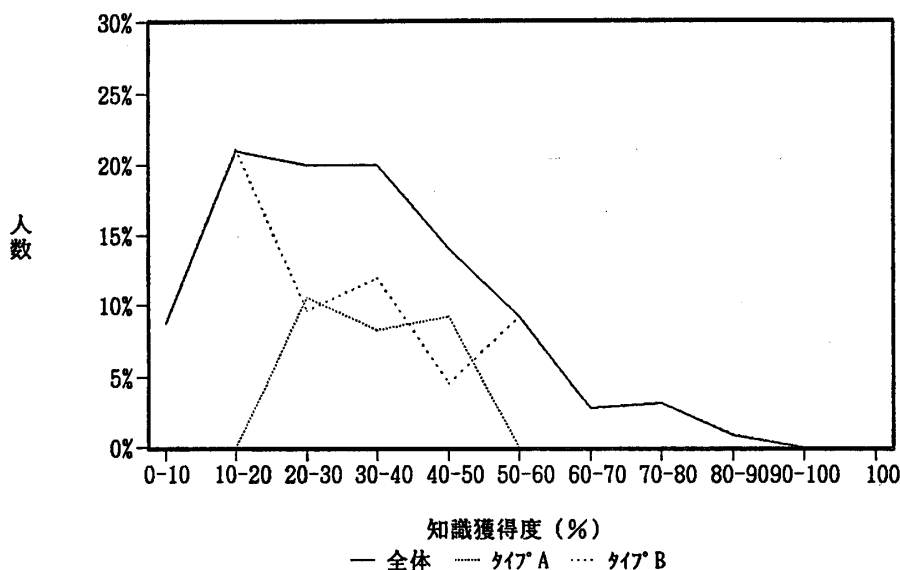
第3表 実習分析概括表

項目	タイプA	タイプB	全体
講義	30.3%	26.9%	27.9%
指導員	27.4%	25.5%	26.0%
友人	31.1%	37.8%	35.9%
個人勉強	11.2%	9.8%	10.2%
課題数(題)	8.9	8.8	8.8
実習回数(回)	40.7	37.7	38.6
回答者数(人)	61	157	218
回答者(%)	28.0	72.0	100.0

① 講義からの知識獲得の分布

受講生が講義に出席して知識を獲得することは、本来的な学習方法である。第1図から明らかのように、講義から獲得する知識量は、10%から40%までの受講生が最も多く、40%を越えるに従って漸減している。タイプAの場合、定義通りに20%から50%幅に分布しているが、タイプBもその幅にほぼ同数の受講者が存在することがわかる。その平均は、全体で27.9%、タイプA 30.3%、タイプB 26.9%であるが、平均値としては妥当な状態であると判断する。しかしながら後述するように、タイプBは20%未満かあるいは45%を越える受講者が多いため、その平均値は一応の目安にしかならない。

講義は、他の知識獲得の場に比べて学生に対して平等に働きかけ、また、いったん履修届を出せば、学生からの主体的な行動無しに知識を獲得できる場である。講義は、単位取得のためのノルマを課してそれを強制する反面、学生を広く受け入れ、学生の勉学意欲を高め、理解しやすい

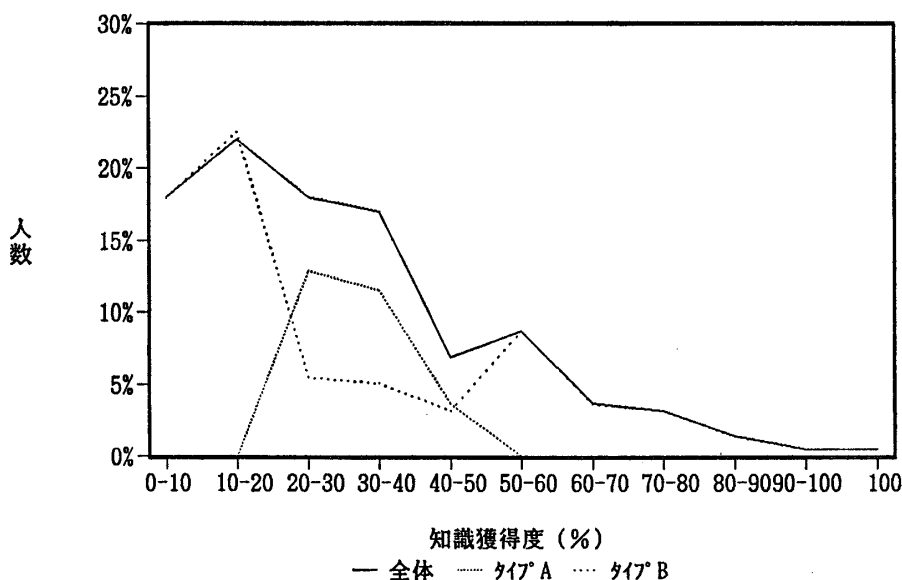


第1図 講義からの知識獲得の分布  
(x軸の目盛は「以上-未満」を示す)

ように課題の体系を設定して通常でその条件をクリアできるように配慮する。講義においては、基本的な知識はすべて提供するのであるが、その吸収が不十分なために講義に出席しただけでは独力で実習を行う段階に至らないのが通常である。それを補完するために実習現場においてサポート機能をもった指導員組織を配置している。さらに友人との積極的な知識交換や個人勉強を奨励することにより、バランスのとれた知識獲得の場を設定することを今日まで試みてきた。

### ② 指導員からの知識獲得の分布

第2図において示されるように、指導員から知識を獲得する割合が10%から30%までである受講生が最も多く、それ以上は漸減して、講義の場合とほぼ類似した傾向を示している。20%から45%の幅では、タイプBがタイプAの半分ほどであるのが講義と異なっている。その平均は、全体26.0%、タイプA27.4%、タイプB25.5%であり、それぞれは講義より若干少ないけれどもほぼそれに近い割合であるのは興味深い。この年度の指導員は、人数、個人能力および組織力において最高に活動した。すなわち、実習現場において彼らは実習生と積極的に対話し、論理展開



第2図 指導員からの知識獲得の分布  
(x軸の目盛は【以上-未満】を示す)

に関する議論において指導員の方で用意した模範的な解答を押し付けるのではなく、実習生の考えを尊重して正解に導く態度が彼らに感銘を与え、多くの実習生の信頼を得ていたように思う。この数字は指導員の積極的な指導意欲によって実現されたものであり、今後とも容易に達成され得ないような活動レベルであると思う。

### ③ 友人からの知識獲得の分布

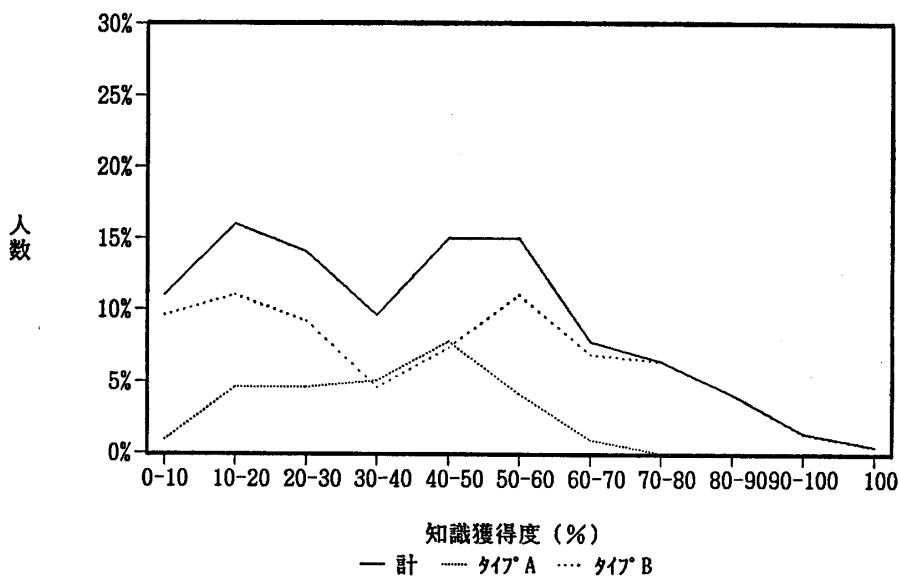
友人から知識を獲得する状況は、これまでとは大いに異なっている。第3図によると、10%から70%までの長い幅にわたって平らな山が2つあり、それ以降漸減している。すなわち、友人に

ほとんど頼らない受講生から全面的に教えてもらう受講生まで幅広く存在している。タイプAの分布幅は70%までに対してタイプBのそれは全域に及んでいるのが特徴である。タイプAとタイプBの分布状態では、分布幅の狭い前者の方が良い状態である。その平均は、全体35.9%、タイプA31.3%、タイプB37.8%であるが、この数字は、講義や指導員よりも高く、さらにタイプAよりもタイプBの方がかなり上回っている。

受講生との関係が友人であって教える能力をもった学生グループは、この年度に確かに存在した。すなわち、我々が顧問をしているコンピュータ関係サークルに入部した20名程度の1年生に対して、指導員による週3回のマンツーマン指導と春夏の休暇を利用した集中研修による特訓が行われたのであるが、それによって能力の向上した1年生が周囲の学生に知識を与えたと思われる。友人から教わる他の状況は、課題を終了した友人の解答をコピーすることであるが、その状況は常に存在する。その防止策として指導員による出力表を見ながらの簡単な口頭質問式のチェックを導入したので、ある程度効き目があったと思う。

受講生が質問をしたりサポートを受けたいと思う場合、教師や指導員よりも気軽にその目的を達成できる友人を利用することをこのデータは示している。少なくとも、初学者が友人に対するように気軽に質問できる環境を欲しがっていることは、容易に推測される。

プログラミング実習において、友人から教わることの是非については賛否両論あると思うが、解答のコピーのみではなく、それに至る過程も含めた知識交換は大に行うべきであり、友人間で解決できる簡単な事項の質問や確認などは、積極的に行うことを奨励している。

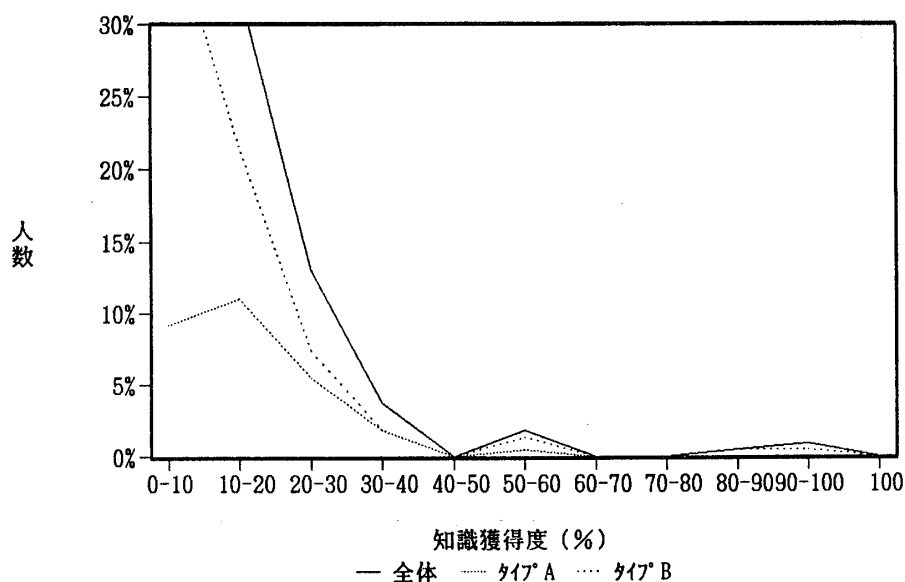


第3図 友人からの知識獲得の分布  
(x軸の目盛は「以上-未満」を示す)

#### ④ 個人勉強による知識獲得の分布

知識獲得の場の最後はその他であるが、それは参考書などを用いた予習・復習が中心であるので、それを個人勉強に言い換えることにする。第4図にあるように、個人勉強による知識獲得は、他とは著しく相違する。すなわち、その数字が10%までである受講生が全体の半数近く（48%）もいて、それから40%まで急カーブで下降している。タイプAの場合、その最高は10%以上20%未満の11.0%であり比較的バランスの良い分布をしていて、タイプBと好対照をなしている。

その平均は、全体10.2%、タイプA 11.2%、タイプB 9.8%であり、その差異はわずかであるが、両タイプの分布の様相は異なる。このような結果は予想しなかったのであるが、周囲の学生にも確認して判明したことは、この数字は知識獲得量の割合であって、消費した勉強時間ではな

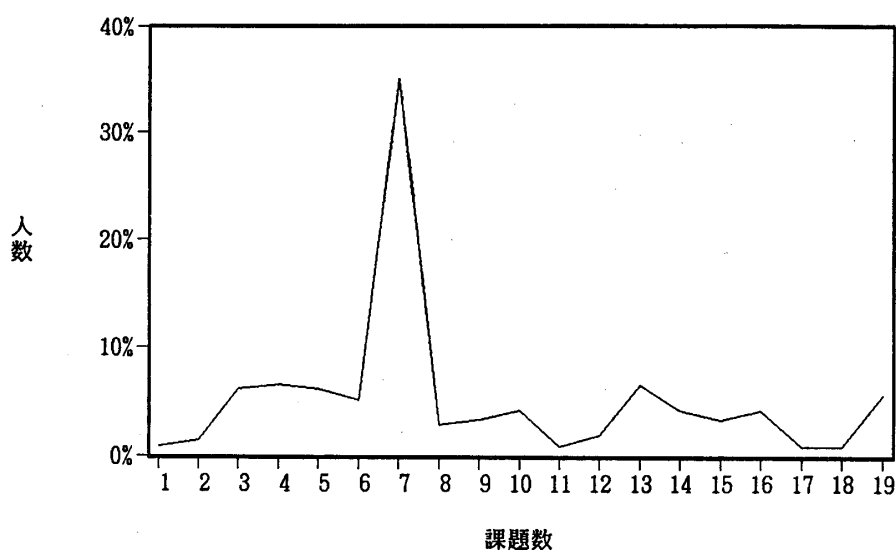


第4図 個人勉強による知識獲得の分布  
(x軸の目盛は「以上-未満」を示す)

いということである。すなわち個人勉強においては、消費時間に比例する知識量が獲得されていない。その理由として、特にプログラミングの初期の段階に当てはまることであるが、COBOLなどのコンピュータ言語の勉強を独力で進めることは非常に困難であることを指摘したい。

#### ⑤ 終了課題の分布

第5図に示す終了課題数の分布において、課題数7題までの終了者が全体の73.9%であるが、そのうち7題のみが突出して35%に達している、その他の10%未満の値と大きく異なっている。これは単位取得の条件が7題以上であることを理由とする。ただ実習締切日は、アンケート調査日より4週間ほど後であったので、ほぼ態勢は見えているにしても、その後6題以下の受講生の追込みが行われたのは当然である。その調査日時点での平均は、全体8.8題、タイプA 8.9題、タイプB 8.8題であり、図には表示していないが両者の分布はほぼ同一である。それにして



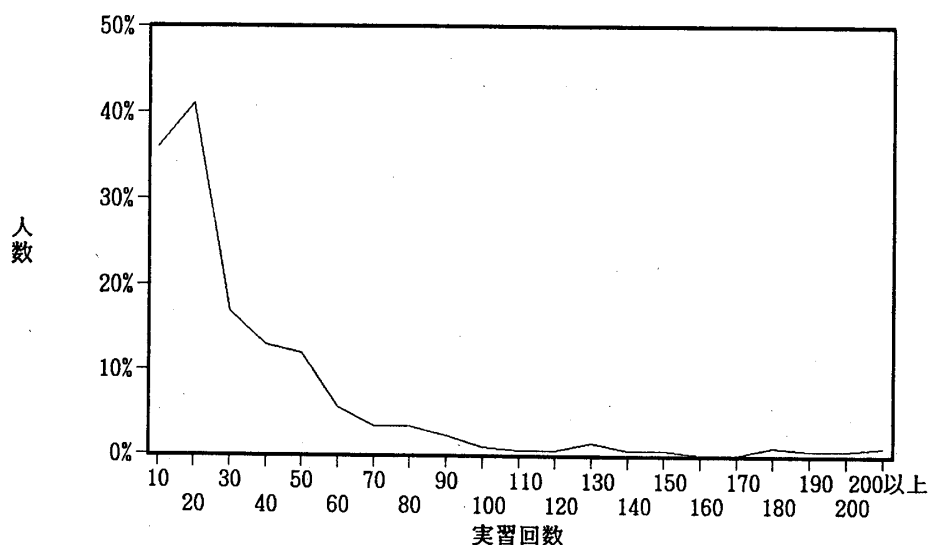
第5図 終了課題数の分布

も最低条件を満たせばよいと考える学生が多数存在するので、その条件をどこに設定するかは、特に実習においては課題負担の見極めの困難さを含む頭を悩ます問題となる。なお、単位取得の条件である7題については、流れ図方式とSP方式の2通りを重複して解題させるので、実際は14題を消化することになる。

さて、15題以上を完了した受講生が14.5%存在するが、それは最上級の課題である。それを独力で解題できれば基礎的な論理構成力は十分についたと言えるのであるが、それに該当する受講生の特性については、次の実習回数と関連づけて検討してみたい。

⑥ 実習回数の分布

第6図によれば、実習回数の分布において10回から30回の幅に全体の77%が集中し、それ以降

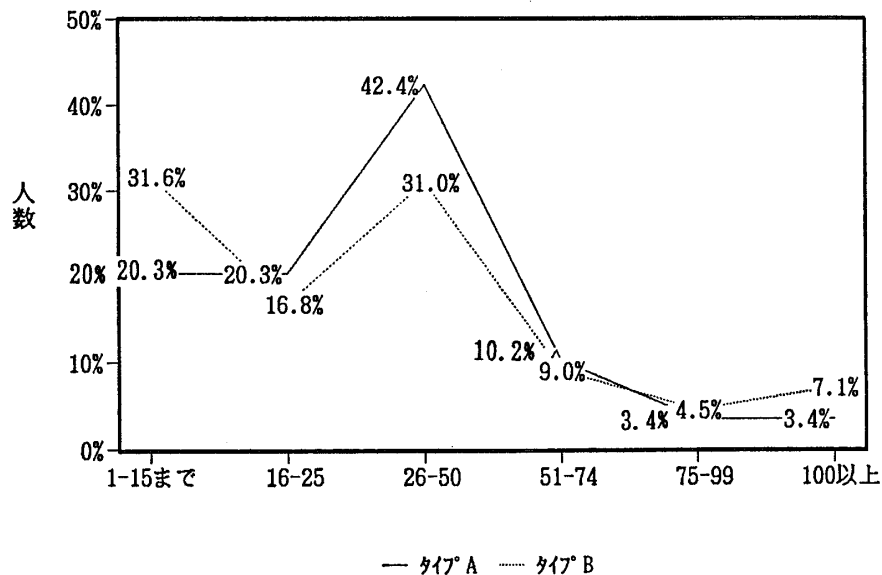


第6図 実習回数の分布 (10回幅)

60回の5.6%にかけて急激に降下して、100回から200回まで、ほんのひとにぎりの受講生が分布している。

実習回数の平均は、全体で38.8回、タイプA40.7回、タイプB37.7回であり、タイプAがタイプBを若干上回っている。実習回数の幅を広げて分布を集約し、タイプAとタイプBを区分したのが第7図である。それによれば、74回まではタイプAがタイプBを上回っていて、特に26回から50回の幅では11.4%の開きがある。それに対して75回以上においては逆にタイプBがタイプAを上回ってしまう。75回以上実習を行う学生は上級課題を完了するトップグループであるが、その人数は第4表に示すようにタイプA、Bにそれぞれ4名と18名存在する。そのような学生がタイプBに偏在することは、タイプAが優れているという仮説に反していてその知識獲得特性が異なると思われるので、仮説の修正と関連づけて後ほど検討を行う。

それにしても、数年前までは、同一課題においてその条件を変更して何度も繰り返すことによ



第7図 実習回数の分布

第4表 実習回数分布

実習回数	タイプ A		タイプ B	
	人数(人)	人数(%)	人数(人)	人数(%)
1～15まで	12	20.3	49	31.6
16～25	12	20.3	26	16.8
26～50	25	42.4	48	31.0
51～74	6	10.2	14	9.0
75～99	2	3.4	7	4.5
100以上	2	3.4	11	7.1
計	59	100.0	155	100.0



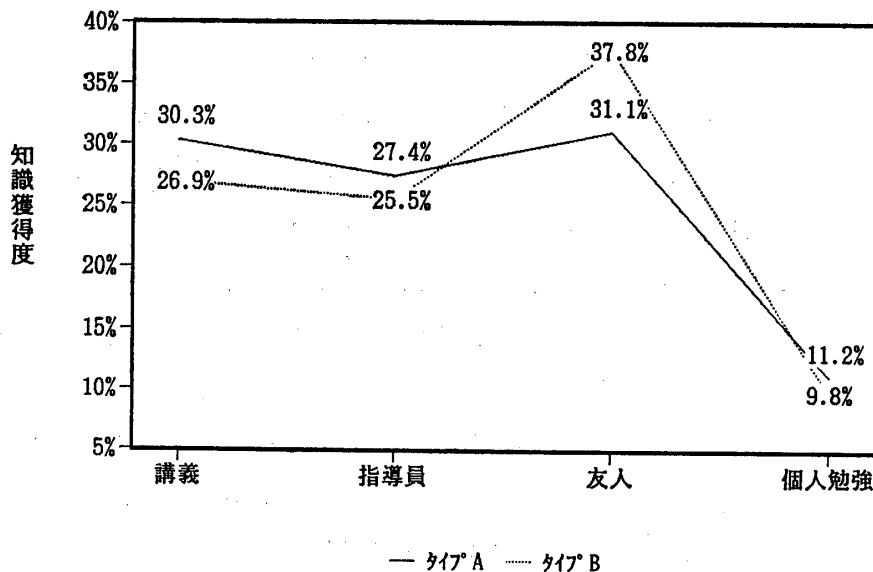
り、300回や400回を重ねる学生がある程度存在したのであるが、最近はそのような学生が少なくなかった。その理由は、上級課題を完了したトップグループの学生をこの教育ソフトから離してTSS方式によるCOBOLプログラミング実習を行わせることと、本学におけるパソコンの導入に関係がある。パソコンは、ワープロ、表計算やゲームなどにおいて幅広くかつ容易に使用することができるので、使用規則の厳格な大型コンピュータが学生に敬遠されるのである。100回未満で理解しない場合は受講生の責任であり、100回を越えていて理解できないならば教師の責任であると明言しているのであるが、未だ責任追求を受けたことがない。

(b) 知識獲得の場についての総括

これまでの考察において、知識獲得に関連づけて設定した仮説に基づいたタイプA、タイプBの対比を中心に実習効果を表わすいくつかの指標を検討してきたが、それを総括的に示すのが第3表および第8図である。それによると、講義と指導員の双方からの知識獲得割合が20%から45%であるタイプAは、それ以外であるタイプBに比較して友人や個人勉強においても平準化され、さらに課題数と実習回数においても上回っていて、タイプAの方が学習効果は大きいとする仮説は概ね正しいといえる。唯一矛盾する点は、第7図および第4表に示されるトップグループの多くがタイプBに含まれていることであった。したがって、この特別グループを一般受講者とは区別して扱うために仮説を次のように修正した。

- ① 実習回数が74回以下であってタイプAの受講生をタイプ修正Aとする。
- ② 実習回数が74回以下であってタイプAでない受講生をタイプ修正Bとする。
- ③ 実習回数が75回以上である受講生をタイプCとする。

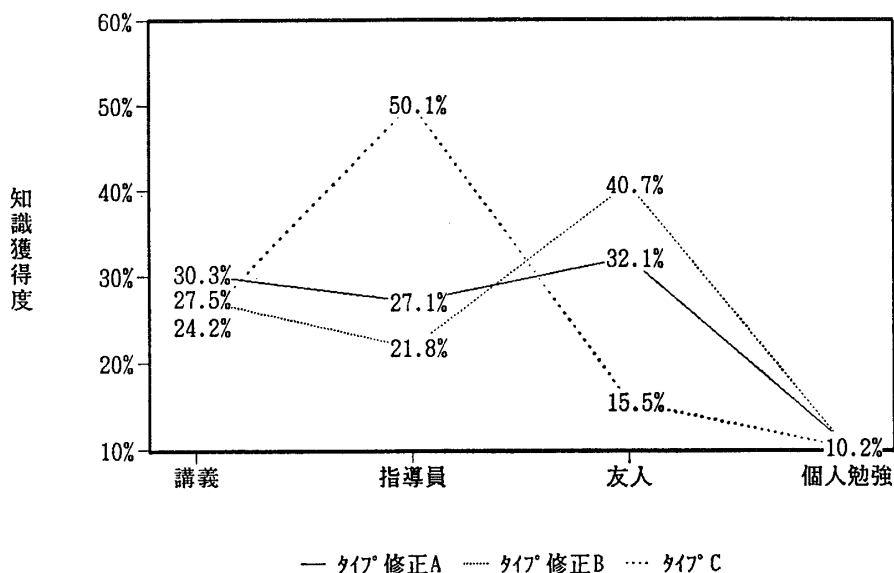
タイプ修正Aは、授業と指導員からの知識獲得の場をバランス良く利用する一般学生である。



第8図 知識獲得の場の比較 (2タイプ)

第5表 知識獲得の場の比較 (3タイプ)

項目	タイプ修正A	タイプ修正B	タイプC	全体
講義	30.3%	27.5%	24.2%	27.9%
指導員	27.1%	21.8%	50.1%	26.0%
友人	32.1%	40.7%	15.5%	35.9%
個人勉強	10.5%	10.1%	10.2%	10.2%
課題数(題)	8.2	7.8	17.0	8.8
実習回数(回)	28.8	26.6	137.0	38.6
回答者数(人)	57	139	22	218
回答者(%)	26.1	63.8	10.1	100.0



第9図 知識獲得の場の比較 (3タイプ)

それに対してタイプ修正Bは、それに属さない一般学生である。またタイプCは、集中的な訓練を受けた特別学生であるが、それを分離し、一般学生を従来のタイプAとタイプBに分けた3種のタイプとしたのが修正された仮説である。このような区分で3つのタイプを比較したのが第5表と第9図である。それらの図表に基づいて、3種のタイプの特徴を見ることにする。

タイプCである特別学生は、全体の10.1%を占めているが、知識獲得においては第9図で見るように指導員から50.1%と圧倒的に多く、その分だけ友人からの割合が減少していることが他と著しく相違する。また、講義からの割合が24.2%であり他のタイプより若干低く、個人勉強においては他とほぼ同一である。この特別学生は、サークルの先輩でもある指導員から週に3回のマンツーマンによる実習指導と、春期および秋期休暇における集中研修を受けていて、コンピュータの勉強に費やす時間が他の学生よりもはるかに多く、通常のペースでは3年かかる知識と技術のレベルに1年間で到達する。指導員からの知識獲得割合が極端に高いのは、このような実態を反映したものである。

また、講義による知識獲得の割合が若干低いのであるが、このような学生の知識獲得の全体量は第5表の課題数や実習回数を見ても一般学生の数倍と多く、また講義そのものが一定の制限時間内での知識伝達であるために、特別学生の獲得する全体量からみるとその割合は低くなる。しかしながら、講義は知識獲得の原点であり、それから得る知識の絶対量は一般学生よりも高いことはいうまでもない。さらに友人からの獲得割合は15.5%であるが、これは同一仲間相互の意見交換や議論を通じての知識獲得であり、一般学生が友人より得るそれとは内容の異なるものである。

このような特別学生の例をみても、真の意味で実力向上を図るためには、一通りの知識が身につくまでの一定期間については上位レベルの知識保有者（すなわち、教員および指導員）の指導を集中的に受ける必要がある。

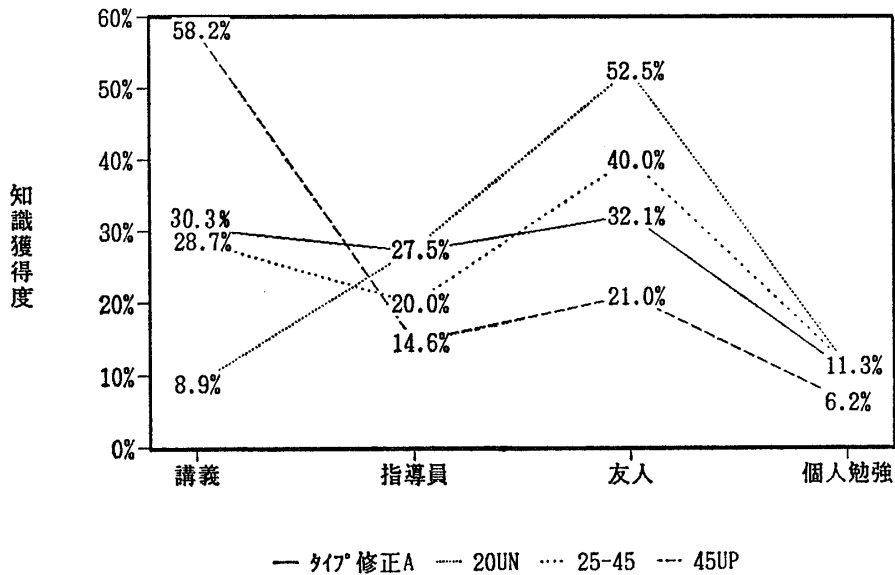
タイプ修正Aに該当する学生は、修正Bの学生と共に特別な勉強体制をとっていない一般学生である。第3表と第5表の比較で明らかのように、このグループから特別学生が4名抜けたことによる数値の変更が生じているが、実習回数の平均が40.7回から28.8回に減少した以外は大幅な変化ではない。

タイプ修正Bは、講義あるいは指導員からの知識獲得割合の一方または双方が20%から45%の幅に入らないグループであるので、平均値のみではその特性を把握することができにくい。したがって、その講義からの獲得割合を20%未満、20%から45%、45%超に分けて集計したのが第6表と第10図である。それによれば、20%未満のグループの講義からの割合は8.9%と極端に低く、その分を指導員と友人で補完し、特に友人の活用が52.5%にも達している。このグループは、おそらく社交性に優れていて仲間づくりの上手な学生であろう。20%から45%の幅のグループは、初めのグループに次いで友人に頼っている。さらに45%超のグループの場合、講義からの割合が58.2%と高いのに対して指導員、友人、個人勉強からの割合はすべて最低である。

これらの3グループは、課題数や実習回数については大差がないことがわかる。このことは、

第6表 知識獲得の場の比較（タイプBの3分割）

項 目	タイプ修正A	タイプ修正B			全 体
		20% 未 満 (UN)	20%~45%	45% 超 (UP)	
講 義	30.0%	8.9%	28.7%	58.2%	27.9%
指 導 員	27.1%	27.5%	20.0%	14.6%	26.0%
友 人	32.1%	52.5%	40.0%	21.0%	35.9%
個人勉強	10.5%	11.1%	11.3%	6.2%	10.2%
課 題 数 (題)	8.2	8.1	7.4	7.9	8.8
実 習 回 数 (回)	28.8	27.2	25.6	27.4	38.6
回 答 者 数 (人)	57	55	53	31	218
回 答 者 (%)	26.1	25.3	24.3	14.2	100.0



第10図 知識獲得の場の比較 (タイプ修正Bの3分割)

予備知識のない初心者向きに開発された実習用教育ソフトが期待通りに機能していることを示している。すなわち、実習課題の中の初級例題7題(単位取得条件)は、通常の努力をすれば誰でも正解を得られるように配慮してある。知識獲得におけるこの3グループの特性は、友人からの獲得の割合の相違にある。すなわち、講義からの割合が低いほど友人からの割合が高く、またその逆も成り立つ。

これまで知識獲得の実態把握について、設定した仮説を検証する形式で考察を進めてきたのであるが、タイプ修正Aとタイプ修正Bを比較すれば、知識獲得においては第5表と第9図に示したように前者の方がバランスが良く、課題数および実習回数においても後者を上回っていることを確認できる。既に検討したこれらの指標の分布は、仮説の修正によって前者に有利に作用しているために、より明確に前者の優位性を示すことになる。

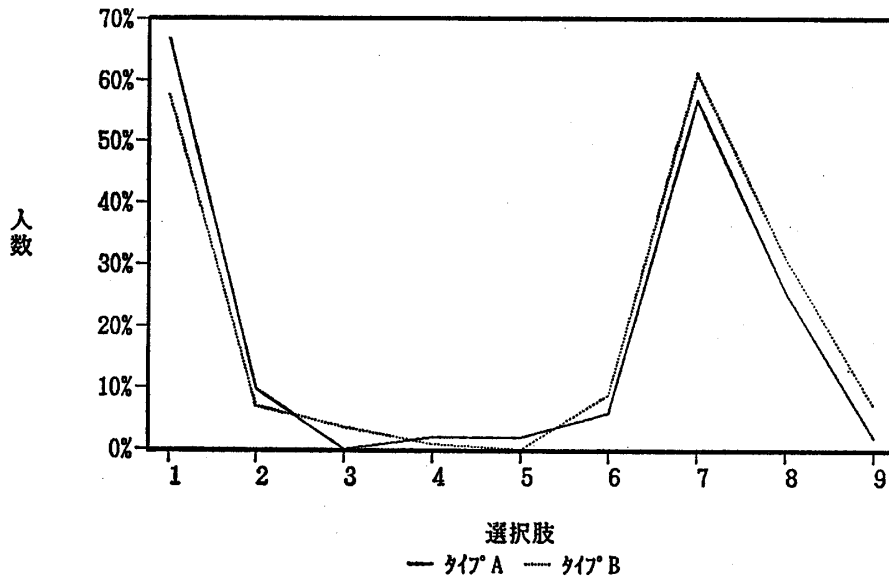
### (c) 指導員に対する受講生の評価の分析

受講生は、指導員をどのように評価したであろうか。そのことを明確にするために、指導員の説明が理解できたかどうか、さらに、指導員の態度をどのように感じたかの2点について文章で回答するよう受講生に求め、それを整理して次のような9項目にまとめた。

- 指導員の説明
- ① たいへんよく理解できた
  - ② よく理解できた
  - ③ ある程度理解できた
  - ④ あまり理解できなかった
  - ⑤ まったく理解できなかった
- 指導員の態度
- ⑥ 親切でなかった

- ⑦ 親切だった
- 指導員の判別 ⑧ 誰か判別できなかったことがある
- ⑨ 居なかったことがある

まず、指導員の説明に対する受講生の評価について、調査結果を分析しよう。第11図および第7表は、タイプA、タイプB別の回答総数を分母とする各項目ごとの人数を比率(%)で示した図表である。このグラフを一見すると、タイプAとタイプBはほぼ同一の結果であることが判別できる。それを概括すると、指導員の説明に対する評価として、理解できた(1+2+3)が、タイプA76.5%、タイプB68.1%に対して、理解できなかった(4+5)においては、タイプA4.0%、タイプB0.9%であり、双方とも圧倒的に指導員の説明が理解しやすかったことを指摘している。



第11図 指導員に対する印象

第7表 指導員に対する印象

項目	タイプ A		タイプ B	
	人数(人)	人数(%)	人数(人)	人数(%)
1	34	66.7	65	57.5
2	5	9.8	8	7.1
3	0	0.0	4	3.5
4	1	2.0	1	0.9
5	1	2.0	0	0.0
6	3	5.9	10	8.8
7	29	56.9	69	61.1
8	13	25.5	35	31.0
9	1	2.0	8	7.1
回答者	51	—	113	—

アンケート記入の際、受講生には批判的に記入するよう依頼したこともあってもっと厳しい評価を予想していたので、このような好結果が出たことに驚きの念を抱いている。

現役の学生である指導員は、2年ほど前は初学者であり、その当時の先輩から丁寧な指導を受けた経験がある。そして彼らは、その時期に課題が理解できずに苦労した記憶とそれを解決するノウハウをもっていて、初学者の心理がよくわかっている。しかしながら、理解した後はその時々サポートの有難さは急激に薄らぐものであり、その恩返しとして彼らにサポートの負担を強いてはいけなく、そのような考えは受容されにくい。指導員のサポート活動は、彼らの勉強の一環として彼らの資質を向上させるために生かされてこそ大きな意味をもつ。この年度の指導員は、組織的によくまとまり、組織力を発揮して共通の恩恵を大きく育てることに意義を感じるようになっていた。その結果、サポート活動だけではなく、学業やレクリエーションにおいても非常に活発であった。そのような背景からも、このような好結果は、指導員の真剣さを反映したものと判断してよい。

つぎに、指導員の態度に対する受講生の評価について検討しよう。ボランティアが主体である指導員活動において警戒すべきことの1つは、指導員と一般学生とのトラブルである。元来コンピュータ実習室は、種々の講座の教員および学生が同居しながら設備を共同で使用するのであるが、自身に寛容で他人に厳格であるという人間の本性も含めた種々の要因が錯綜して悪循環を呈する可能性は常に存在する。指導員と一般学生において、そのような状況を極力排除する策としてなるべく早い時期に受講生に指導員を引き合わせる場を数度設けるように配慮している。さらに、受講生にはサポートを受けたら感謝の念をもつよう指導している。さて第10図より、親切であったと答えたのがタイプA56.9%（不親切5.9%）、タイプBが61.1%（不親切8.8%）である。双方において、親切と答えた受講生が大多数であるが、少なくともこのような数字は、必ずしも人間関係能力が十分に身につけていない指導員の真剣な努力を反映していると判断する。

また、実習室における指導員の判別については、タイプA25.5%、タイプB31.0%が判別できなかったことがあると答えている。さらに指導員不在の指摘が若干あるが（タイプA2.0%、タイプB7.1%）、両者は関連しているので、前者に含めて一括して検討する。

指導員は、この年度においてはローテーションを組んでただ実習室に在室するように配慮したのであるが、講義や休憩のためにそこを離れることも多い。したがって時間帯によっては誰も居ない場合がある。ただ、実習生がセルフサービスでコンピュータ操作ができるようになるまでは、なるだけ無人の状態にはならないように配慮している。それから、指導員として活動する場合はネームプレートを付けることになっているが、それを励行していなかったり、またはそれが見分けにくいために、指導員が在室しても受講生が判別できない場合もある。その対策としては、指導員の存在をもっと明瞭にするための工夫が必要である。

これまでの分析を総合すると、指導員からサポートを受けた時間の多少にかかわらず、受講生は指導員に対してほとんど同じ評価をしていて、指導員がどのような学生にも差別なく同じような態度で接したことを示している。この結果は、当時の指導員がいかに真面目に活動したかを示している。指導員の説明が受講生に理解されるためには、教師と指導員の連係が十分にとれ、指導員が十分に活動できるためには、そのための物理的かつ精神的環境が整備されている必要があるが、この年度は、両者の歯車がうまくかみ合って好循環した事例であり、今後のモデルとなる。

## (2) 1989年度調査にみる実習の役割

### (a) 講義と実習の関係分析

最近の5年間に指導員組織の拡充策として取り組んだ目標は、先輩が後輩を組織力によって育成する自己増殖機構を有する組織を確立することであったが、そのためには、指導員そのものが互いに協調関係を継続できることと、それを積極的に支援する環境が存在することが必須条件であり、双方ともに不十分であれば、その実現は困難であることを再確認した。

各年度の実習体制を確立するための基本的要件は、指導員を希望する学生諸君が自力で組織活動ができるまでうまく育成することと、彼らが安心して活動できる基盤が実習現場に存在することである。1989年度においては、指導員組織が拡充していた前年度に比較して、その双方に環境変化が生じて前年度の体制はかなり変容した。すなわち、指導員の活動は消極的となり、組織活動の中核である勉強会では指導員相互の能力向上による知識と価値を共有することが困難となった。そのような状況であったので、実習回数は例年の半分程度に落ち込むのではないかと懸念したが、受講生の勉学意欲が高かったことと後半に新たなメンバーを加えて指導員組織を立て直したことによって最終的な実習実績は例年と同程度であった。

指導員活動にはボランティアの性格を持たせてあるので、どのようなサポート体制をとるかはその年度ごとに決定し、強制はしないようにしている。この年度は指導員による解答チェックを義務付けなかったため、セルフサービスで実習ができるようになった受講者は自由な気分で実習を行い、実習実績は予想外に伸びた。

そのような前年度との背景の相違により、この年度のアナケート調査の目的を、我々が特に興味を持った2つの点、すなわち、実習回数がテスト成績にどのように反映しているかということと、講義における基本的知識事項が実習によってどのように補強されているかを把握することにおいた。

#### ① 実習回数とテスト成績との関係分析

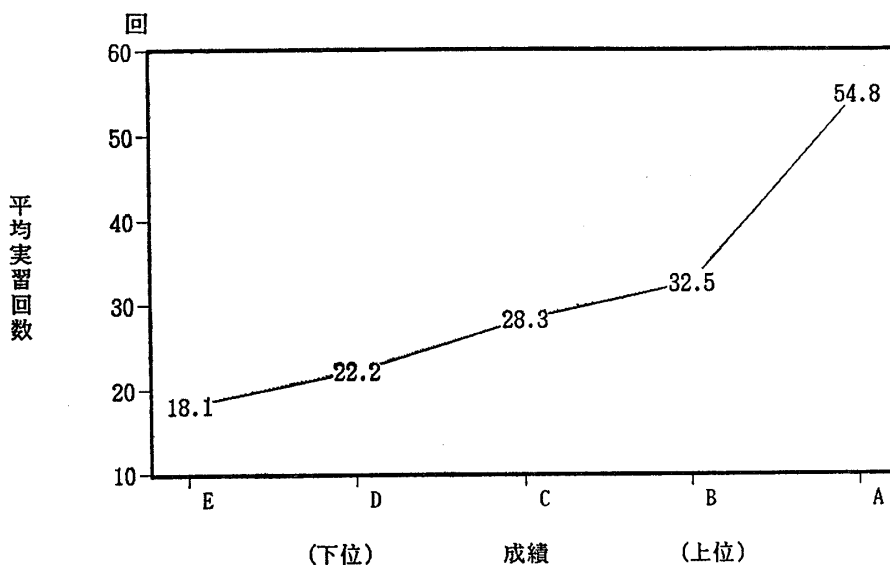
実習を経験する量が豊富になるほど知識は正確になり、論理展開能力が向上するのは自明のこ

とであるが、アンケート回答者 248 名を対象とするテストの成績と実習実績とを関連づけてそれを確認してみたい。テストの形式は、処理手順の展開とデータ領域設計に関する 2 題に対して自由に記述させる方法をとった。答案は、採点基準を設けて A から E までの 5 段階で評価した。そして各ランクごとに平均実習回数（第 8 表）を集計し、グラフにしたのが第 12 図である。

既に検討したように、実習の成果は種々の要因の相乗作用によって著しく異なるが、指導員組織がうまく機能しなかったこの年度でさえも、双方に明確な相関があることがこの図より読み取れる。すなわち、成績ランクが上がるに従って平均実習回数は増加しているが、そのなかで特に A ランクの数字の伸びが際立っている。プログラミングに必要な知識の記憶量はそれほど多くないが、それを精密かつ自在に活用する関係上、記憶の正確度は 100% を要求される。したがって、受講生の頭脳に存在する知識を反復して活用する訓練を行ってその正確度を向上させるためにコンピュータ実習を実施するのであるが、このような傾向を見るにつけ、プログラミング教育における実習の役割を再認識する。

第 8 表 成績別平均実習回数

成績 (テスト) ランク	人数 (人)	人数 (%)	実習回数
A (最上位)	7	2.8	54.8
B	60	24.2	32.5
C	95	38.3	28.3
D	67	27.0	22.2
E (最下位)	19	7.7	18.1
計	248	100.0	29.7



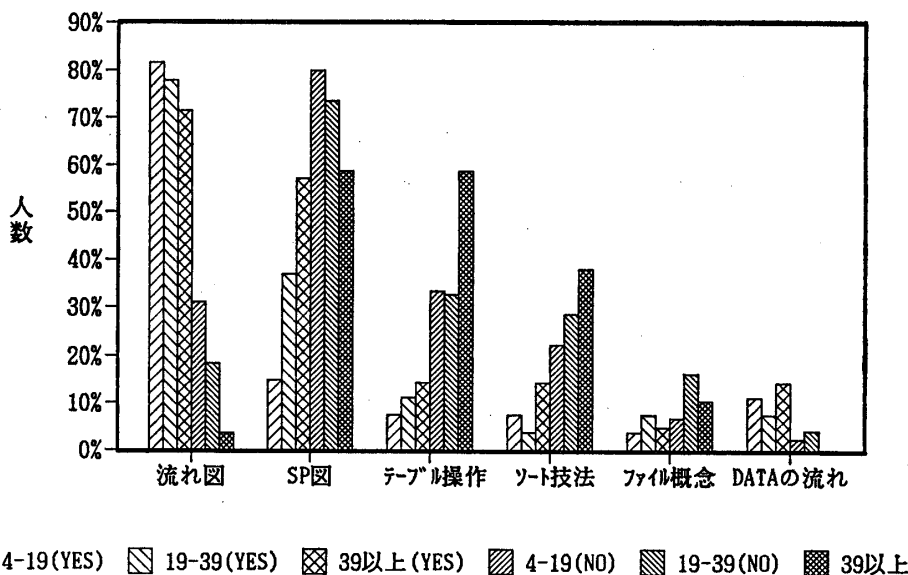
第 12 図 成績と実習回数の相関



② 実習回数と理解度との関係分析

実習科目の場合、受講生の知識獲得が順調に進んでいるか否かは実習実績に忠実に反映される。それぞれの受講生について、各課題の開始日、終了日および実習回数を自動集計しているの、そのデータを追跡することによって実習状況を個別的・総合的に掌握し、状況に応じた適切な対策を講じることができる。

さて、講義の内容が実習によってどれだけ補強され、かつ受講生の記憶に留まっているかを把握するために、1年間勉強して理解したと思う事項と、理解できなかったと思う事項をそれぞれ10個ずつ記述してもらい調査を最終講義日に行った。その回答には多くの事項が含まれているが、その中から重要と思われる次の6つの事項すなわち、流れ図、SP図、データの流れ、ファイル概念、テーブル操作、ソート技法を選定した。流れ図とSP図は、データ加工の手順をトップダウン方式で詳細に展開する方式であり、最終的にはCOBOLプログラムの手続部に具体化される部分である。データの流れとファイル概念は、加工対象であるデータの入力から出力に至る加工の各段階に必要な記憶領域の設定を行うために不可欠な知識であり、COBOLのデータ部に関係する。またテーブル操作は、柔軟で効率的なコンピュータ処理を実現するための基本技法であり、最も高いランクの課題に用いるソート技法とともに、実習により習得された能力レベルを評価するために最適の指標である。これら6つの事項は、知識体系の中核部分であるので、これらの理解度と実習回数との関係を見ることは興味深い。第13図、第9表および第10表は、これらの項目を理解した(YES)、あるいは理解しなかった(NO)と答えた受講者を実習回数幅別(3段階)に分類して表示している。以下において、それぞれの事項の特性を第13図を中心にして検討しよう。



第13図 学生が理解したと考えた事項

第9表 学生が理解したと考えた事項 (YES)

項 目	4 ~ 19 回		20 ~ 39 回		40 回 以上	
	人 数(人)	人 数(%)	人 数(人)	人 数(%)	人 数(人)	人 数(%)
流 れ 図	22	81.5	21	77.8	15	71.4
S P 図	4	14.8	10	37.0	12	57.1
テ ー ブ ル 操 作	2	7.4	3	11.1	3	14.3
ソ ー ト 技 法	2	7.4	1	3.7	3	14.3
フ ァ イ ル 概 念	1	3.7	2	7.4	1	4.8
D A T A の 流 れ	3	11.1	2	7.4	3	14.3
回 答 者 数	27	—	27	—	21	—

第10表 学生が理解しなかったと考えた事項 (NO)

項 目	4 ~ 19 回		20 ~ 39 回		40 回 以上	
	人 数(人)	人 数(%)	人 数(人)	人 数(%)	人 数(人)	人 数(%)
流 れ 図	14	31.1	9	18.4	1	3.4
S P 図	36	80.0	36	73.5	17	58.6
テ ー ブ ル 操 作	15	33.3	16	32.7	17	58.6
ソ ー ト 技 法	10	22.2	14	28.6	11	37.9
フ ァ イ ル 概 念	3	6.7	8	16.3	0	10.3
D A T A の 流 れ	1	2.2	2	4.1	0	0.0
回 答 者 数	45	—	49	—	29	—

流れ図においては、理解したとする回答は理解しないとするそれよりも著しく多く、つぎに検討する SP 図と比較してもそれが初学者に馴染みやすいことを示している。流れ図を理解したとする受講生が実習回数の増加に従って減少しているが、実習経験を積むごとにその理解度が増すという道理から判断すると、この結果は逆転しているように思える。SP 図の場合、実習頻度の増加に従って不理解の度合が低下していて、実習経験との相関がとれた妥当な結果になっているのに対して、流れ図の一見矛盾とも思えるこのような徴候は、流れ図の理解の容易さに原因があると判断する。すなわち受講生には、最初の初級例題（第2表参照）において流れ図の正解が与えられ、その解説を受けるので、彼らはその概念と意味内容を理解できた気分になる。しかしながら、課題と実習が進み受講生の判断部分が増大するにつれて、理解したとする自信が揺らぎ始めることがある。流れ図の回答傾向は、そのような状況を示していると思う。また、流れ図を理解しなかったとする受講生が少数存在するが、実習回数の増加とともに急減しているのは妥当な結果である。

ここで注意すべきことは、理解できたか否かという受講生の判断基準である。すなわち、理解したと答えた受講生には講義での説明を受けて概念や意味内容を漠然と理解しただけの者を含む。また理解しなかったとする回答には、実習に裏打ちされた技法の習得が完了していないものも含

んでいる。このことは他の事項にも該当するので、理解か不理解かの厳密な比較は、最上位の実習幅（39回以上）における方が信頼性が高い。

SP 図において理解できたと考える受講者は、理解できなかったとする受講者よりも少なく、全体的にみると流れ図の場合と比べて好対照をなしている。SP 図は、処理機能をトップダウン的に大機能から小機能へと細分化していく方法でプログラムの構造部と処理部を構成するための思考法に基づいて作成される一覧図であるが、一般的にはこのような思考法は高校までには経験が少ない。SP 的な思考法は、既存の概念を適用できにくいため初期学習の効果が薄く、習得するまで時間を要する。それを理解したと答えた受講者が実習の進行とともに大幅に増大していることがわかる。また、理解しなかったと答えた受講者はこの項目が最も多いのであるが、それでも実習頻度の増加につれて漸減していて、実習の着実な効果を跡づけている。

テーブル操作とソート技法は、上級の課題に含まれている。テーブル操作については、理解したと答えた受講生が実習回数とともに微増しているのに対して否との回答が圧倒的に多く、しかも最も実習回数の多い人が最も不理解を訴えているのが特徴である。またソート技法は、テーブル操作を駆使した上級課題であるが、理解しなかったと答えた受講生は、実習回数が多いほど増えている、テーブル操作と類似の特徴を示している。ソート技法は高度なテーブル操作を前提にするので、受講生の両者に対する印象は同一であろう。ソート技法には、比較的容易な技法から難解な技法まで数種類あるが、実習回数の多い受講生は複数の技法に取り組むので、その分さらに理解の困難さを自覚するのであろう。いずれにしても、実習回数が多くしかも理解していないと答えた者が、最も理解の度合いが高い場合がある。

ファイル概念とデータの流れにおいては、回答者が少なく、受講生の印象があまり強くないことを示している。これらは実習課題を解題するための前提となる事項で基本的な知識であるので講義においては強調するのであるが、プログラムのデータ部にかかわることもあって余り注目を集めていない。その理由は、教育ソフトがデータ部までを所与としていて、受講生は手続部の作成に専念すればよいためであろう。

プログラミング学習における知識体系の柱であるこれら6つの事項は、講義において概念や考え方を解説し、実習ではそれに基づいた論理展開による思考訓練をしているのであるが、本講座での実習方式の中心である手続部の作成に直接的に関係する流れ図、SP 図、テーブル操作、ソート技法の4項目が、その理解度において実習頻度と関連性をもっていることが明らかになった。このことは、実習による論理の展開およびその検証を数多く積み上げることによって知識を正確にし、それに基づいた分析力が増すことを例証している。

明確になった次の点は、学習初期における流れ図の重要性である。流れ図方式は、既存の概念や思考法で容易に理解されるために受講生がプレッシャーを受けることなく学習することができ、

SP 図に橋渡しをする前段階としての役割は大きい。それは、データの流れ（すなわち加工手順）に中心をおいてその時間的変化を追跡するために前後の関連を把握することが容易であり、比較的単純な課題に有効である。

## 5. 知識獲得の場としての CAI の検討

### (1) CAI の概要

CAI (Computer Assisted Instruction) とは、学習者がコンピュータと結合した学習端末を直接操作しながら、個別に学習するシステムである。

CAI システムは、コースウェア、ハードウェアおよびソフトウェアの3つの側面をもつ。コースウェア (courseware) は、学習者がコンピュータを介して学習を行う教材の開発および編成の側面である。学習の個別化を実現できるか否かは、コースウェアの構成次第であり、CAI の心臓部に相当する。コースウェアの方式としては、個別指導 (tutorial) 方式、練習演習 (drill and practice) 方式、模擬演習 (simulation) 方式、および問合せ方式などがある。

ハードウェアは、コンピュータおよびその他の機械装置それ自体をいう。CAI には、当初大型コンピュータやミニコンが利用されていたが、コンピュータが高価であったため、なかなか普及しなかった。それに対して、1980年以降安価なパソコンが出現し、それを利用した CAI が急激に普及してきて、複数のパソコンをネットワークにより統合した CAI が今後の主流になることは確実になってきた。

ソフトウェアは、ハードウェアに個別学習用コンピュータの機能を持たせたコンピュータ・プログラムであり、ハードウェアを稼働させるための命令の集合である。CAI ソフトウェアには、言語ソフトウェアとオーサリング・システム (authoring system) があり、双方ともコースウェアの作成と運用を行うためのツールである。言語ソフトウェアの使用にはプログラミング技術やコンピュータ関連の専門知識を必要とするのに対して、オーサリング・システムにおいては教材展開のための論理的な筋道を設定する能力や教育評価などの教材関連知識のみでコースウェアを構築することができる。ソフトウェアがコンピュータの動作指令を行う側面であるのに対して、コースウェアは、コンピュータをベースとしないメディアや学習者の行為も含むことに注意しなければならない。両者は、CAI システムを構築することによって総合される。

CAI は、一斉授業の欠陥を克服する方式として期待され、適用されてきた。コンピュータ実習の経験によれば、30名程度を超えるクラスを1人の教師が指導するには、単一プログラムでの一斉授業にならざるをえない。そのプログラムに順応できない学習者に対しては、不適な環境となる。CAI は、そのような状況を克服するために学習者の特性を組み込んだ教育プログラムを用意し、各人それぞれの能力レベルに適合した進度と難易度をもつ個別課題の学習環境をコンピ

ュータ上に構築し、どの学習者も学習成果をあげることを目指したものである。

上述の目的を達成するためには、コースウェア、ハードウェアおよびソフトウェアに組み込まなければならない種々の条件がある。現時点でその条件を満足した CAI は存在しないが、技術革新と幾多の社会環境の整備によって理想システムにどんどん近づきつつある。いずれにしても、CAI は多くの問題点を抱えていることは事実であるが、有効であることを示す実証的データは数多く、CAI が本格的な実践段階に突入したと判断せざるをえない。

## (2) 大学教育における CAI 導入の有効性

大学における教育カリキュラムは、多数の科目により構成され、その履修方式は複雑である。また講義の受講人数や授業方法は千差万別であり、これまで主として CAI が導入・実践されてきた小・中・高校とは基本的に利用環境が異なる。しかしながら大学教育に CAI を導入し、効果をあげている事例もあり、パソコンの普及や企業内教育における CAI の利用などの社会環境からみても、現在は CAI の本格的導入期である。

CAI の目指す適用目的は、受講者を主体とした個別学習により、一斉授業の欠陥を克服することであるが、大学においては CAI と一斉授業を排他的にとらえる必要はない。CAI の導入は、学生にとって新たな知識獲得の場が提供されたことを意味する。それは、視聴覚教材や TA (Teaching Assistant) 制の導入に加えて、学習環境を豊富にする。CAI 利用の形態を多様化し、なるだけ多くの講座で柔軟に利用できるような環境整備を主な役割とする組織を確立し、CAI 利用の目的を明確にした運用を行えば、有効な活用ができるであろう。

これまで研究会や展示会などにおいて多くの CAI 事例を見聞してきたが、大学のカリキュラムに CAI を導入する観点から見ると、特にフレーム型の CAI において次のような問題があると思う。

- ① 基本的に親密感の高いテキストなどの文書情報を非効率かつ制約の多い CAI 画面に置き換えたものであり、その効果を期待することが困難である。
- ② CAI の開発者が、教育現場で生起する問題や潜在的活用者のニーズを十分に把握していない。

CAI 画面の特徴は、紙上における文章や図表が静的である欠点を克服できる動画機能をもっているため、その機能を活用した画面をふんだんに駆使した活用をすれば、プログラミング実習教育において、つぎの3つの側面に有効に利用できると思う。

まず第一は、実習課題に対する事前・事後のチェックを CAI で行うことである。コンピュータ実習に対する受講生の心理は、わずらわしい課題を最小の努力で早急に終了させることを望む場合が多く、表面的な知識の吸収に終わってしまう危険性がある。それを防止し、受講生の勉学に

関する姿勢を改めさせるためにも、特に課題に関する前提知識の確認とその理解度に関するチェックを行い、受講生がそのチェックを受けることを評価の対象にすることが重要であると考え。現在では、指導員組織がかなり拡充している年度に限り、課題の完成時における理解の確認チェックを行っているが、それをCAIに移行する実験も開始したい。

第二は、受講者の疑問に答えるためのQ & A (質問に対する回答)方式のCAIを構築する側面である。実習現場において受講生が解題活動を行う場合、彼らには、多くの疑問事項が連続的に発生する。現在、質問を主として受けるのは指導員組織である。そのような質問事項をCAIの容器に集積し、それらを整理して体系化をはかることにより、種々の表現内容をもったワンパターンでない回答のプールをつくることが有効であると考え。

第三は、基本的なプログラムを模擬的に実行させ、そのモニタリングすなわち、いくつかの指標について実行証跡を表示することに関するCAIである。それは、データ加工の基本的な流れと、その流れの原因となる各命令の動作の関係を主要指標の変化により把握することが目的である。このようなCAIは、コンピュータの動作原理をシステムティックにとらえるために重要である。

我々はCAIを今後の重要な知識獲得の場と考えているので、上掲の3点についてCAI化を検討している。現在、第二および第三の側面を実現したCAIシステムの一部を試作したので、今後、実習現場でテスト的に適用したい。

## おわりに

COBOL 訓練用ソフトを開発した10年前は、果たしてこれがどの程度COBOL 実習に適合するかに関する定見はなかった。ある時期には断念しようと思ったこともあったが、そういう危機状態を乗り越えられたのは指導員組織であった。文系学部の基礎実習において指導員を務める学生は、このソフトで育ち、その自主管理と後輩の育成をし、それを後輩に渡していく循環が現在も継続している。

しかしながら、そのような組織を長い期間にわたって運営していくことは、必ずしも容易ではない。一般的に言えることは、文系学部における知識獲得の場は、たとえば理工系学部と比較しても相対的に貧弱であり、学習環境が十分に確立していない。公式の講義のほかに半公式あるいは非公式の学習の場が数多く存続していくための環境を整備することが重要であるが、その実現のためには教職員の協力関係の保持と学生諸君の勉学意欲が不可欠であることは自明の理である。

## 《付 記》

① 本稿の作成には、加藤が全般を統括した。尾辻は、特に実習現場の実態把握と指導員に関

するアンケート調査の分析, および表計算による作表と作図を分担した。

- ② CAI に関する加藤の研究は, 1985 年度学長所管研究費の助成によって開始し, フレーム型 CAI の開発を試みた結果, 開発負担の大きさに対する期待効果がきわめて小さいため, 本格的な構築を延期してきた。本稿における考察により, コンピュータ実習に関連づけた部分的な CAI を構築し組み込むことは有効であることが判明したので, 現在の試作システムを拡充する予定である。本稿を学長所管研究の成果発表に代える。

(1990. 6. 18 完)

《注》

- (1) 加藤武信, 尾辻斗志(稿):「コンピュータ実習教育における指導員制の導入とその拡充過程」, 城西情報科学研究, 創刊号(1989年5月)。
- (2) 指導員の組織化, 育成および活動内容については, 上掲稿9~13頁を参照のこと。