解　説

情報処理専門教育について

情報処理教育における実験・演習†

都　倉　信　樹††

はじめに

情報処理学会では情報処理教育の諸問題を検討しており[6]，その検討状況を本誌を通して広く会員に紹介している．その一つとして，本文は大学等における情報処理教育検討委員会の中間報告[3]に含まれている「情報系専門学科における実験演習の特質」という一節を一部修正した内容である．

1. 情報工学系学科における実験・演習の特質

どの学問分野でも単に講義をするだけでなく，実験や演習を課している．実験と演習との違いは何かの議論が必要かもしれないが，ここしばらく実験・演習を区別せず単に実験と呼ぶことにする．

1.1 実験の目的と分類

さて，実験の目的と，学習実験と研究実験とをわけて整理した記事を引用しよう[1]．

……

実験には二つの種類—学習実験と研究実験がある．実験を行う場合，簡単で易しい実験から入るが，その目的は次のようなものである．

（1） これまでに学習した知識を実験に適用して理解を深める．

（2） いろいろな法則や概念が形成されるに至った代表的実験を通して実験の論理を学ぶ．

（3） 基本的かつ代表的な測定器具の使用法の習得．

（4） データの整理法と実験結果を報告書としてまとめる方法を学ぶ．

（5） 実験室，実験器具等の整備を通して実験に対する態度，心構えを学ぶ．

† Laboratory Experiments in the Computer and Information Sciences Educational Programs by Nobuki TOKURA (Department of Information and Computer Sciences, Osaka University).

†† 大阪大学基礎工学部情報工学科

このような目的で行う実験を学習実験と呼ぶ．ここでは指導書に忠実に従って実験するわけであるが，これはちょうど，碁や将棋における定石，柔剣道における形，絵画における手本の模写といった類のものである．忠実に指導書を繰り返すということは盲目的に従うことを意味するのではなく，一つ一つの段階が何を意図するのかを絶えず考慮しながら進めることが大切である．学習実験においては実験する前から結果が分かっているのに対し，研究実験は未知の領域に踏み込むわけであり，ある程度，結果の予測はできるが必ずしも予測どおりにいくわけではないし，逆に意外な結果を生むことさえある．研究実験は基礎的な学習実験を十分に積み重ねたうえでのみ，よい結果が得られるものである．

……

ここで議論すべき実験は，大学のカリキュラム内での「実験」であり，新しい事実を発見したり確認するために研究者が行う研究実験でなく，目的からすれば学習実験ということになる．しかし，情報関係の場合には何か違う要素があるのかもしれない．実際，ここに引用した書物の題は「自然系実験」となっている．最近公開された市川教授を主査とする報告書[2]では，自然系に対する人工系という考え方を提唱している．情報関連はまさに人工系に関する面が多い．

その特性を探るためまず既存の分野の実験についてふり返ってみる．

1.2 既存分野での実験

教養部での物理実験の場合の筆者の経験から簡単にふり返ってみる：

実験テーマはかなりの数用意されており，力学，光学，電気，熱，音，時間等々，実験内容により大きく分類されており，学生は各分類からかたよらずに実験テーマを選択することが求められ

た．装置が各テーマ一つであったという理由もあったであろう（3人で一班）．学生からすれば，自分で，種々の要素を含む実験を選択するというところに楽しみもあったし，自分たちの計画を教官に告げると種々のアドバイスを得られるという点で教官とふれあうという感じもあり，自主的に実験をしているという感じがもてたものである．実験テーマが決まると，教科書には，そのテーマについての詳細な説明があり，目的，必要な装置，実験の仕方，等々が記述してあったから，その説明を読み必要なデータをとっていけばよい．熱現象を扱う実験が一部時間がかかったものの，一般にはそれほど時間は必要なく，だいたい指定された時間内におわる．そして，あとは測定データを元に，整理をし，理論計算との照合等を行って報告書をまとめて提出すればよかった．それで，実験の目的や方法はだいたい理解でき，実験がまずくて測定データが不本意なものになることはあっても，レポートを提出すれば単位はいただけた．

専門での電子工学科での実験もほぼ似たもので，指導書のとおりにデータをとるという作業をすれば目的を果たせるものであった．

研究者が行う研究実験なら，実験データが理論計算値と一致しないとか，不備があれば，再実験をするとか，別の条件を設定して（場合によっては，新しい方法を考案して）再実験が行われる．学習実験は装置や方法は固定されており，別の方法や装置を使うことまでは実際上行われないであろう．教官側もそこまでの検討や再実験は期待していないようである．

また，化学実験も同様で，指導書に従って，反応を起こさせたり，滴定等の測定を行い現象を観察し，レポートすればよかった．未知資料の定性分析もそれまで個別に学んだ分析手順を厳密に実行するというマニュアル誘導型の実験であった．

いずれも，上の説明にある学習実験であり，講義でならった事項を実際に実物の装置や薬品を用いて，起こるべき現象の再現を確認するという内容であったといえよう．このような学習実験では，上の(1)(2)の目的に関していえば，比較的頭で理解しやすい題材が選ばれており，その点に関していえば，あまり困難はなく，むしろ，目的の(3)の測定装置や器具になれることが実験の実質的な内容と思われるものもあった．はじめてふれるものがあり，その理解や操作法の習得にかなり時間の必要な場合があった．たとえば，シンクロスコープを扱うときは，指導書より分厚い参考書が指定された．そういう測定装置などの扱いになれることはその後，専門的な研究に入ったときに当然要求されることであった．また，測定装置の扱いは講義では扱わず，実験でのみ習得されるという意味で，重要な内容になっていたと思われる．また，指導教官にもよるが(5)でいう態度，心構えを重視しておられる先生もあった．いわゆる「しつけ」である．「しつけ」の一種の「おきて」を教えることはその専門分野のギルドの秘儀のような意味があったのかもしれない．それを通して，専門家としてのマナー，意識の向上，倫理的な面の教育を実地に行うよい機会であったのであろう．医学部の解剖室の解剖台の正面には，「屍活師　合掌」と大書してあり，それは人間の生命を扱う医師の卵に大きな感銘を与えるものであったという．ミスが重大な事故につながりうる化学実験はやはり一種の厳しさがただよう面があった．しかし，無機的，かつ，危険性のほとんどない物理実験，電気実験ではそういう機会はむしろ少なかった．

このようにみてくると，実験は学問分野によって，目的の(1)から(5)の重みもおのずと少しずつ違うことはあるようである．

以上例としてあげた物理実験，化学実験，電子工学実験，医学部の解剖実習などの実験は，いずれも実習実験であり，「すでに知られたことの再現，過去の科学者が行った実験の再体験」という性質をもつ．「観測事実に基礎をおく」という実証科学の本質的な方法論を経験するものである．学習実験の延長に，研究者が行う研究実験があることも理解できよう．

1.3 情報分野での実験・演習

では，情報分野での実験はどういう性格のものと考えればよいのだろうか．

まず，ハードウェアの実験は，電子工学科の実験と共通のものがいくつかある．それは計算機のハードウェアが電子工学の成果を利用しており，基本的な面は共通となるからである．たとえば，トランジスタの特性測定，伝送線の特性の測定などの測定的な実験は電子工学科の実験と特にかわ

らないであろう．また，電子工学科では，簡単な電子回路をハンダづけ（あるいは，ラッピング）で作らせることもあるが，それと同様の実験は情報関係の学科でもよく行われている．その場合の目的は，回路の動作原理の理解，ハンダづけやラッピングという手技になれること，いくつかの部品を知り，その取り扱いができるようになること，回路を構成すること，テスト，特性の測定などが含まれる．場合によっては手直しを経験することもあろうし，この実験を通して回路作成の手順や一種のコツを習得することもあろう．

このあたりまでの実験は電気系の学科では共通的に行われているし，学習実験そのものといってよい．内容的にも指導書のとおりに回路を作って動作確認をすればよいというものである．

しかし，情報関係ではもう少し違った実験課題がみられる．「簡単な計算機を設計して，プロトボード（かロジックトレーナ）で試作し，動作を確認せよ」というような課題が与えられる．その指導書には，その計算機をどう作ればよいかは書いてない．ここが，上にのべた実験と違ってくるところである．伝統的な実験では指導書には「どうするか」がことこまかく指定してある．そういう手順を理解し，覚えることも目的に含まれるからである．しかし，いま述べた情報関係での実験では，「何をすべきか」という仕様はあっても，どう作ればよいかは指定されていない．ここに大きな違いがあるといえよう．これはプログラミングやソフトウェアになるとより顕著になる．プログラミングの課題は，こういうプログラムを作成せよという，仕様しか与えられていないのが普通である．

プログラムリストをすべて示して，単にタイプインして，指定した入力を入れて出力がどうなるかを記録していくというようなことは，よほどの例外的な場合しかしないであろう．

このように，アーキテクチャ関係の実験，プログラムの演習はどうも従来の実験とは違う面があるようである．

これに似たものはないだろうか？ まず，建築での設計，都市計画，工業デザイン，機械設計，美術の課題制作，音楽の課題作曲，あるいは，大学ではあまりないかも知れないが小説教室とか文学学校などでは課題を設定して創作するというのもあろう．

これらに共通している点をあげてみよう：

（1） 課題とか，設計目標を指定するがその具体的な実現部分は学生に委ねられる．

（2） いずれも講義などでだいたいの作り方などの指導を受けているが，その課題に対してどのようにやればよいかを書いた指導書があるわけではない．

（3） 過去の実験の再現，追体験ではなく，かりに結果的には類似性のあるものになるにせよ，教官側も学生側も新規性や独自性，独創性が生まれうるものであることを意識している．

（4） 実験は理論との照合をしてうまくいったと判断することがあるが，ここであげたような場合唯一の基準解があるわけでなく，解は無数にある．

（5） 一般に結果の評価は単純ではない．

（6） 一般に教育手法はあまり確立されていない．

クラスルームでの一斉授業で力がつくというよりは，よい指導者について適切な助言や批評をうけつつ徐々に学生自身が学びとっていくという方法がむいている．

これらの分野ではもはや実験という言葉は使わないであろう．設計演習，○○設計，○○実技，などの名前で呼ぶようである．プログラムの作成，あるいは，ソフトウェアの構成などの実技教育も内容的には上にあげた(1)から(6)の特性をもつであろう．また，一部の目的は共通するものの，先の学習実験とは内容的にかなり異質なものといえよう．ましてや，研究実験ともいえない．そこで，これを呼ぶのに，演習と呼ぶことにする．（「実習」の呼称を使う大学もあるということであるが，ここでは，仮にこのように呼ぶ．）

要するに，情報関連学科では，電子工学等と共通する実験，そして，プログラム等の演習が共存し，ともに重要な役割を果たしている．

1.4 プログラムの演習についての検討

前節で，ある程度プログラム演習の特性をみたがもう少し検討の必要があるかもしれない．上の(1)から(6)の性質をもつような教育を，適当な言葉を知らないが，仮に「体得」教育と呼ぶ．この種の教育はできるだけマンツーマンの体制が効果的である．なぜなら，学生によって進度がかな

り違うからである．しかし，実際には学生数との関係で種々の工夫をこらして多人数教育をしている．たとえば，牛島教授の工夫の一部の紹介である．学生の成果提出物をクラスの学生に提示し，それについて検討するという方法である．このことは完全なマンツーマン方式では得られない利点がある．なぜなら，学生は他の成果・到達点を知ることで自分の考えの足らなかった点，気のついていなかった点などを（教官の指摘によるのでなくして）自分で発見する，まさに，体得するという経験をできる．また，クラスで検討の議論をすることで，単に「課題をやらされる」という受け身の立場でなく，教官と同じく評価する立場にたつことができる．このことは非常に重要である．というのは，評価が一元的にできないものの学習の際に自分のもっていた価値判断基準だけでなく，多様な見方にふれ，評価の判断基準を磨ける．このようにして評価する眼を養える．この点を考慮し，別の観点から特性を考えるなら，一つの尺度だけで判断できないという意味では，「総合的なもの」である．解は無数にありえて，その評価は難しいと述べたが，それゆえ，総合的に価値判断のできる能力を身に付けるという重大な教育目的が明らかになってくる．たしかに，＜プログラムはかくあるべし＞という，「よいプログラム」の満たすべき条件をあげることはできる．しかし，それは抽象的な項目であって，具体的なプログラムに対して，この要求には 80 点等といえるものではない．そのような判断のできる眼を養うという意味でも体得教育になる．このような教育のもう一つの特質は，時間がかかることである．できるだけ，場数をふみ，そのつどできるだけきちんとした評価を行い，評価する判断力もつけていくことが必要である．この時間がかかるという問題についてはあとで再度考える．

ここまでの議論は情報関連以外の演習にもほぼ当てはまるであろう．次に，他と異なる点はないかの検討に移ろう．プログラム・ソフトウェアの知的財産権の保護について，現在著作権法によることとなっている．しかし，そこにいたるまで，また，その後も，果たして著作権として扱うのが適当かについて疑問の表明や異論が展開されているようである．たしかに，著作物に似た点はいろいろあって，作曲や小説を書くことが簡単に教育できないのと似たことがあることを上に述べた．しかし，著作物という枠からはみでる部分もあるから，いまだに議論が盛んなのであろう．それは単に著作物であるだけでなく，計算機上で動かすことで特定の機能を発揮するものである．そして，計算機がなくては機能を発揮できない，それを作ることはできないという特性もある．

プログラム演習での悩みは計算機の機材の問題である．

1.5 機材の悩み

大学では計算機について大きな悩みをもっている：

（1） 絶対数の不足

学生数に対して利用できる計算機が少なすぎる．

（2） 陳腐化の問題

あまりに古い計算機で教育しなければならない．

（3） ソフトウェアの問題

計算機というハードウェアがまず問題であり，これがなくては話にならないが，教育に必要なソフトウェアがおざなりにされていることも問題である．特に，従来の TSS からワークステーションのネットワークへ移行しつつある現在，それぞれのワークステーションに必要なソフトウェアを整備するには，多額のソフトウェア費用がかかる．ここに，従来あまり問題にされなかった新しい問題が発生する．せっかくかなりの数のワークステーションを確保しても，CAD 等かなり高価なソフトウェア等はとても台数分購入できず一部のワークステーションで限定的に利用するというようなこともある．この意味で良質のフリーソフトウェアが求められる．この問題はこの小文の範囲をこえるが，早急に解決を迫られる問題である．

（4） 運用面の問題

a．センタ方式の場合は利用可能時間が一般に短く，計算機ファシリティとして有効活用されていない．一般企業では高価な計算機を停止させるというようなことは許されないが，大学は毎日計算機を止めている．

b．センタ方式でない場合の運用については，計算機の管理運用のためのスタッフがいないことが問題になろう．最近，ワークステーションが低

価格化し導入するケースが多いがその運用・管理要員に苦しんでいる話はよく聞く.

(5) 保守費の問題

買い取りの場合，保守費は学科経費の硬直化をまねく.

(6) レンタル化

(2)と(5)の問題はレンタル予算であれば解決するが，レンタル化がなかなか難しい.

これらの問題はすでにどこでも認識し議論されていることであるので詳論は避ける．ここでは，なぜ，情報関係の学科では(2)の陳腐化が問題であるかという点をもう少し議論しよう.

計算機は年々高速化・大容量化している．また，マウス，ディスプレイ，プリンタ等の周辺機器の変化発展も急速である．体得型のプログラミングやソフトウェアでは理屈や理論より，実地にやってみるということが重要であることはいうまでもない．10年前の計算機では，10年前の経験しかできない．「日本のソフトウェアはアメリカに10年遅れている」とか，「そんなことはない，2，3年」という人やいろいろであるが，アメリカより進んでいるという話は聞いたことはない．(日本語処理技術は別として…また，最近ある分野のソフトウェアの生産性において日本方式はアメリカにまさっているというレポートもあるが，それはすでに経験のある応用について同様のものを繰り返し作る場合のことではないかという意見もある．経験のない新しいソフトウェアや概念を生み出す点ではまだまだという意見もある．その当否はこの小文では扱えないが…) やはり，経験がそれだけ遅れれば最先端を行くことは非常に難しい．特に，ソフトウェアは使ってみた経験からまずい点を知ってまったく新しい発想を得たという経緯で発展したものが多い．その意味でつねに最も新しい機材を学生に使わせたいと教官は思うものである．計算機の処理能力だけを議論して，この使用実績からすればこれだけの能力があれば不足するはずはないという議論があるが，それは事務計算等定形的な業務の場合に成立することであって，新しい発想のでる学生を育てるためには性能と使用実績のみの議論は意味がないし，誤った議論といわざるをえない．最近は企業でも意識改革が進んで他社製品であっても新しい製品を導入し，できるだけ最新の機能にふれるようにしている．その意味でも大学は企業に格差をつけられつつある.

2. 演習をめぐる問題

ここでは，演習に関して思いつく問題点をいくつか考察しよう.

2.1 演習には時間がかかる

計算機のプログラミングの演習は講義2単位の学習に要する時間等と比較して非常に時間がかかるという現実がある．指定した時間内に完了するどころか夜遅くまで計算機のそばにいるというようなことはしばしばみられる．先にふれた従来型の実験はだいたい指定した時間内に完了していることと考え合わせるとやはり独特のものがあるのであろう．単位数，あるいは，配当時間数を見直すか，あるいは，時間・単位に見合う内容に縮小するか等が問題になろう．学生が時間を忘れて取り組んでいるのだからやらしておけばよいということで，済ましてばかりはいられないのではないか．熱心なあまり学生の能力におかまいなしに多くの課題を与えて，情報は「しんどい」という風評を作っていないだろうか．また，情報の演習は時間がかかるし，しんどいということを他分野の人にはなかなか理解してもらえない．「なぜか？」という問いにどう答えるのか？ 特に電気や電子関係の実験と同じ時間枠にはめこまれたソフトウェアの課題はかなり問題があろう．学習実験と体得的演習を同じ時間枠でやるにはかなり無理があると思われる．学科でカリキュラムの検討の際に，何を教えるかという問題とともに，どういう順序で，どういう方法で，どれくらいの時間やリソースを使って教えるのかが当然問題になる．講義や実験に関してはほぼ見積もりはあうようであるが，演習に関してはやや実態は過負荷になりがちではないだろうか？ ただ，あまりに容易な課題を与えたのでは結局力がつかない，ある程度高い壁を乗り越えてもらわねばならないと考える教官も少なくない．この点についてこれまでの経験を交換して検討をすすめる必要があると思われる.

2.2 演習には時間が相当かかるということの理由は？

いくつかの理由になりそうな点をあげてみよう.

(1) もともと体得的な性格のものであり，じ

っくり熟成を待つようなものである.

(2) 本質的に内容が多い.

「プログラミングなんて易しいことで,単に手順から言語での表現に置き換えるだけのことだから時間はそれほど要らないはず」という議論を聞いたこともある. しかし, プログラミングの初歩では, 計算機のことを何も知らない学生を対象に考えねばならない. われわれの経験でも一回目は他の講義の時間をもらい, 2コマ連続で教官も数人が応援して, ログイン, エディタの起動, エディタの簡単な使用法, コンパイル, リンク, そして, 実行をひととおり経験させるのに走り回らねばならない. あと, 操作法はマニュアルで各自が徐々に修得するのに期待して, それ以後はプログラミングということの説明に入っていく. 言語の構文や機能の説明はそれほど難しくはないにしてもプログラミングということの説明はかなり時間も実際の経験も必要である.

学生に簡単な課題を与えられる時点になると, テスト, デバッグ等の必要性とその典型的なやりかたを説明しなければならないが, これも講義で力がつくというより演習中の自分の経験で体得していくところが大きいようである.

より,大きな課題になってくると,仕様から設計して,プログラムして,テスト,デバッグというソフトウェア開発の過程を小規模ながら経験する.

また, 演習はレポートをもって報告することになっているとすれば, そのためにまた長時間を必要とする.

先にあげた通常の実験では装置はすでにセットアップされており, 指導書どおりにデータをとれば実験が終了するのに比べて, 格段に内容が多く, また, その進め方が各人の自由というところに従来型の実験に比べてはるかに負担が大きいことを物語る.

進め方が明示されていないために, 経験のない学生は見通しがあまく取り掛かりが遅くて時間的に苦しくなるということも多い.

(3) 時間が足りなくなってくると計算機の前に長時間取り付いている学生がでてくる. そのときの学生の愚痴は, もっといつでも好きなときに自由に計算機が使えたらいいのに, というものである. もっと混む前に使えばよいが, それは言うのは簡単という類のことであり, いくら言っても実効は上がらない. このような事例があるから, 今の計算機環境でいいでしょうとは言えない. アメリカの大学を引き合いに出すことはさけるが, 理想的にはもっと現在ある計算機がいつでも使えるような運用と, 陳腐化してパソコンにも劣るような計算機を新しいものにしなければならないであろう.

パソコンが高性能になり, 自分でパソコンを持っている学生は楽に課題を仕上げることができるようになっている. 大学の計算機よりパソコンのほうが使いやすいので, 大きな計算機でなくパソコンを学生数購入して貸し出してはという意見もよく聞く.「学生が共通のパソコンを持っていたら, もっと進んだ演習の形も可能なのだが…」という教官もおられる. 機材が不自由な場合はそのためにロスタイムが増え, また, 学生の苦痛は増大する.

(4) 演習のレポートには設計, コーディング, デバッグ, レポート作成に要しただいたいの時間を記入させているが, あるテーマでの典型的な数値を紹介しよう.

設計	1～3時間
コーディング	2時間～1日
デバッグ	1日, 数日
レポート	1～2時間, 4～7時間

担当者のコメント：この課題は1年生後期（前期から Pascal をゆっくり教えている）の課題で, いわゆるクロスレファレンスを求めるプログラムの制作である. 講義でかなり丁寧に必要なことは説明しているので, 他の課題と比較して設計の時間は全員短めである. コーディングもやや短めで済んでいる. デバッグの時間は2極分化型で手早くできた学生とてこずる学生にわかれた. レポートも2極分化で, 短い人のレポートはそれなりの内容であるが, きちんと書こうとするとかなり時間もかかる. 今回は冬休み週間という休講期間をはさんでやや時間的にはゆとりのある課題であったと思われる.

配当時間外にもかなり時間をかけていることが分かる.

2.3 指導するほうも大変...

どの分野でも体得型のものの指導は易しくはないであろう. 指導法（メトード）の確立されているという場合でも, 相手のレベルや特性をよく睨

んで指導しないと効果はあがらない．学生のレベルや特性によくマッチした指導をするには少人数が望ましい．しかし，プログラム教育は需要が多いということと教官不足という状況から多人数教育をせざるをえない．この状況は TA (Teaching Assistant 院生に授業の補助をさせるなどの) 制度の導入等の方策である程度改善されるという意見もあろう．

現状では，一部の大学で TA 制度を導入しているが，一般には実現していない．そして，一人のインストラクタにすべてをまかせるか，ときに，助手を応援に付けるくらいの手しかない．そうなると，前節で述べたような学生側の苦痛とともに，教官側でも形は違うかも知れないがやはり苦痛を背負うことになる．あまりに面倒をみなければならないことが多すぎる．教官の負担はかなり大きい．「そんなことはない」という人がおられれば，実際に担当すれば，短期間のうちに納得できるであろう．レポートのチェックも容易ではない．ましてや，学生に的確なフィードバックをきちんとしている教官はまったく希少な存在ではなかろうか？

指導方法も教官によってかなり違う．体得的なものは教官の個性が出てくるのであろうか？

そして，ソフトウェア開発について正しく認識し，仮に，課題はトイプロブレムのようなものであっても，より大きなソフトウェア開発へ向けた問題意識をもって演習の教育にあたる教官が望ましいがなかなかそういう状況にはなっていないという意見を聞く．

演習・実験のカリキュラムを作った教官や最初の担当者は趣旨や目的をよく理解していても，担当者が変わっていくうちに理解がうすれていくということもあり，カリキュラムの実行段階でのフォロも必要であろう．

2.4 問題は多い，しかし...

演習の実施についてそう容易なものではないという面を述べてきた．それほど大変ならやめてもいいではないかという意見もありえよう．しかし，座学ではとうてい身につかないと考えられるから実験や演習をするのであって，その教育効果がやはり大きいことはだれしも認めるところではないだろうか？

しかも，できるだけ多くの優れた能力をもつ人材を養成することが急務とされており，講義，実験，演習を効果的に配分していく必要がある．

演習・実験が重要であることはいうまでもないであろう．最後に，理学部の数学科の教官から伺った話を紹介しよう．3年生は午前中講義があり，午後は演習である．担当の教官，あるいは，演習担当の助手の先生から問題が学生に与えられる．そして，1週間後学生は問題を説明し，解答をする．そこで，担当の先生から種々のコメントが出される．「数学は講義を聞いて知識を付けてもだめなんです．考え方，新しい問題をなんとかして解くような力を身に付けさせることが大事です．そういう意味で，少人数でのセミナ形式の演習は非常に重要です．それは，情報でも同じではないですか？」と話された．

2.5 課　題

演習の実施は幾多の困難をかかえていることを述べた．しかし，講義だけでは伝達できないことを伝え，力をつけ得る有力な方法であることに違いはない．そこで，今後情報関係の教育にあたるわれわれが課題とすべき点をあげておこう．

（1） 演習の重要性を認識し，正しい位置づけをし必要な環境作りをすること．

（2） 効果的な演習の方法の開発．

（3） 計算機を有効に利用して，よりきめ細かい指導のできるシステムを構築すること．

（4） 有効な指導方法，システムを普及させる努力．あるいは，それを促進・援助する仕組みの構築．

（5） 指導者の能力向上のための教育．

3. 議　論

この原稿について以下のような議論があった．

3.1 「プログラミングでは学習実験という形はありえないのだろうか」

これに対しては，次のような考え方はありえよう．単に，これこれのプログラムを作れという仕様を与えるのでなく，全体（あるいは，ほとんど）でき上がった形でプログラムを提供する．ソースリストでも共用ファイルでもよいが，実行させることのできるものである．そして課題としては，実行させてある種のデータをとる．たとえば，実行時間，プロフィール，動作の可視化による実行状況の把握，マンマシンインタフェース等．そのあと，ある点に着目してプログラムを変更するこ

とを求める．そして，比較データをとる．こういう形で一からプログラムを作るのでなく，一部の変更を求めるという程度なら学習実験的な感じではないだろうか？

なお，このような方法をもう少しプログラミング的にするならば，保守作業の経験もできよう．また，基本的なモジュールを自由に使えるように用意してそれを使ったやや大きなプログラムを作らせるのもよいであろう．前項の(2)にあげたように，効果的な方法を見つけていかねばならない．

3.2 「グループ実験・演習か個人実験・演習か？」

これも実験・演習の実施に際して大きな問題になろう．教育効果，機材等の利用可能資源の制限等を考慮して決めねばならないであろう．教育効果というとき，即，個人でということではない．大きなテーマをうまく分担して協力してやっていくということも求められる．ただ，安易にやるとグループ中の一人に他の学生が依存してしまう等の問題点も指摘される．

3.3 「講義との連携が難しいが...」

理想的には講義で基礎的なことを説明して，それを演習によって確認し，あるいは，さらに発展深化させ，単に知識でなく体得することが望ましいと思われる．しかし，実際には時間割の関係で講義と演習をうまく連携を取るのはなかなか難しい．そのため，講義と演習を一つの科目として作り，中でうまく進行させるようにするような工夫もあろう．講義と演習をまとめた内容という考え方の授業科目を実施している大学もある．たとえば，プログラミング科目は講義と演習をほぼ交互にやっている．そのために，教授または助教授が講義を担当し，それを助ける形で，講師や助手が1，2名がついている．これ以外の方法もあろう．

3.4 よい教材がない

これも深刻な問題である．適当な処理系やソフトウェアがない（あるいは，この世に存在しても手が届かない）という状況でまず行き詰まる．なんとか環境があってもどう指導してよいかが上にのべてきたようになかなか難しい．効果的な教育法についての情報，教材（教科書，教育用のソフトウェア等）が望まれる．

おわりに

情報系の実験と演習という点についていくつかの面から検討した．情報という新しい分野での実験・演習の特質がある程度整理されたかと思われるが，伝統的な分野の実験に比べ，時間や計算機，指導者等はるかに多くのリソースを必要とすること等が浮かび上がってきたのではないだろうか？

以上，速成の不十分な検討であり，論点の漏れ，重みつけの不適切，表現の不適切等が心配される．さらに議論を積み重ねて検討すべきことも少なくないと思われる．ご意見をいただければ幸いである．

なお，最終報告[4]には，いくつかの大学の実験・演習の実例が紹介されており，また参考文献 5)の木村先生の訳者コメントでは，実験科目の重要性を指摘し，デニング報告の付録Ⅱから具体的な実験課題を拾い出して紹介されている．いずれも実験演習の内容について参考になると思われる．

最後に本文をまとめるに当たって議論していただいた方々，特に，CS 分科会のメンバ，そして，査読者に感謝いたします．

参考文献

1) 宮代彰一：「自然系実験」放送大学教育振興会 (1985).
2) 文部省高等教育局　工学教育の振興に関する調査研究協力者会議：変革期の工学教育(平成元年12月).
3) 情報処理学会　大学等における情報処理教育検討委員会：平成元年度教育改革の推進に関する研究委託中間報告書 (1990年3月).
4) 情報処理学会　大学等における情報処理教育検討委員会：平成元年度教育改革の推進に関する研究委託最終報告書 (1991年春発行予定).
5) Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J. and Young, P. R. (木村　泉訳)：学問としての計算機分野，情報処理，Vol. 31, No. 10, pp. 1351-1372 (Oct. 1990).
6) 野口正一，中森眞理雄：大学等における情報処理教育の諸問題，平成元年度の調査研究を中心として，情報処理，Vol. 31, No. 10, pp. 1373-1389 (Oct. 1990).

（平成3年4月1日受付）

著者紹介（p. 1092 参照）