

コンピュータを用いた実習講義における 多人数教育の試み

北 栄 輔

<要 旨>

本論では、100人前後の学生を対象としたコンピュータを用いた2つの実習系講義において、普段の講義をどのように実施しているかについて紹介する。具体的に講義カリキュラム、受講生、講義の進め方などについて述べる。つづいて、講義実施において気をつけていること、さらには現在抱えている問題点などについて述べた後、まとめとする。

1. はじめに

本特集のテーマを伺ったとき、最初考えたことは私の担当講義は本当に多人数教育の実践といえるのだろうかということである。私が現在担当している講義では、いずれの場合も受講者は100人前後にすぎない。文系の部局では300名程度が参加している講義もある。また、私自身の経験からしても、私立大学などではもっと多数の学生を一度に教育していると思われる。それらに比べれば、私の講義というのは必ずしも多人数とはいえず、せいぜい中人数と言うところではないかと思われる。また、多人数な講義だからといって、小人数な教育と比べて、特に意識的に気をつけているわけではない。従って、普通に私が実施している講義について紹介することのできない。しかし、それでも個人的な経験をお話して何らかの参考になればと思う。

私のはじめて講義を担当したのは工学部に在籍していたときで、100人強の学生を一人で担当した。現在であれば、二名程度のTAと一緒に指導するのだと思うが、当時はTA制度がなかったので、一人で100名程度の

学生にプログラミングの実習を含む講義を行っていた。その後数年して現在の部局に移り、1クラス20名程度から120名程度の講義を半期に複数同時に担当した。

今回は、現在担当している講義のうちから、全学共通科目の「図情報とコンピュータ」、情報文化学部の特設基礎科目である「プログラミング1」について紹介したい。続いて、講義を実施するにあたって気をつけていること、また、問題点について説明し、最後にまとめとする。

2. 教育の実践1 「図情報とコンピュータ」

まず、担当科目である全学共通科目の「図情報とコンピュータ」について紹介する。

2.1 教育の目的と方法

この科目は、理系教養科目として、文系と理系に対して一コマずつ開講されている。元々、図学担当教員を中心に担当していた「図情報と計算機」という科目があり、カリキュラムの改変で今の名前に変更となった。私自身は、図学の講義経験もあるが、「図情報と計算機」さらに「図情報とコンピュータ」となってからの経験のほうが長い。

「図情報とコンピュータ」となる以前は情報リテラシー教育も含んでいたが、現在では情報リテラシー科目が別に予定されているので、それとは区別した編成が可能となっている。その結果、現在では、教育の内容として、

1. Excelを用いたデータ解析の基礎
2. Mathematicaを用いた数学、画像処理、サウンド処理

などを中心として行っている。

このような講義内容を扱っている理由は、受講学生が文系と理系にまたがっているで、両者にとって興味を持てる内容として、両者に重複した部分をコアとして配置し、それを含んで文系と理系の双方にまたがった内容を含むように編成した結果である。

講義は、毎週メディアセンターのサテライトラボで、説明・練習・演習をセットとして実施している。担当者は私とTA学生1名である。テキストは指定しないで、HPに掲示して行っている。受講者数や特徴などについて表1に示す。

表 1：方法・受講者の特徴・教材等（図情報とコンピュータ）

受講者数	1 クラスあたり80名から100名強
受講者の所属学部	理系（理，工，農，医，情文（自然）） 文系（文，経，法，教，情文（社会））
教材	独自のものをWebに掲載
TA数と業務	1名（実習補助，レポート提出確認）

表 2：カリキュラム編成（文系向）

ガイダンス	ガイダンス
Excel編	Excelによる図表作成 Excelによる最小二乗近似（回帰分析） Excelによる移動平均解析 Excelによる相関解析 Excelによる自己相関解析
Mathematica編	Mathematicaの基礎 Mathematicaによる代数計算 Mathematicaによる曲線フィット Mathematicaによるグラフィックス Mathematicaによる2次元アニメーション Mathematicaによるサウンド処理

表 3：カリキュラム編成（理系向）

ガイダンス	ガイダンス+Excelによる図表作成
Excel編	Excelによる最小二乗近似（回帰分析） Excelによる移動平均解析 Excelによる相関解析・自己相関解析
Mathematica編	Mathematicaの基礎 Mathematicaによる代数計算 Mathematicaによる線形代数 Mathematicaによる微分積分 Mathematicaによるグラフィックス Mathematicaによる2次元アニメーション Mathematicaによる3次元図形処理 Mathematicaによるサウンド処理

2.2 講義カリキュラム

本年度の講義カリキュラムの編成例を表2、表3に示す。表2は文系向け、表3は理系向けの場合の講義編成である。このように、基本部分を両者に共通な部分としておいている。つまり、

1. ガイダンス
2. Excelの基本・最小二乗近似（回帰分析）・相関解析
3. Mathematicaの基本・代数計算・アニメーション等

しかし、その他の応用部分については両者で異なるようにしている。例えば、Mathematicaについては、理系では講義の数学を意識して教えるのに対して、文系では数学として利用するよりも、グラフィックツールのように使えることを意識して教える。これは、理系では数学との連携を強く意識するのにに対して、文系では必要性が低いために、学生の側で敬遠しがちとなるからである。

また、このカリキュラムは、あくまでも最初の講義で予定した内容である。そのときの学生の指向・スキルなどにあわせて、講義を進めながら、必要に応じて講義編成を変更していく。

2.3 講義の進め方

講義の進め方は、説明（講義）・練習・演習を1セットとして行うようにする。つまり、最初に15分程度説明をし、続いて10分程度練習問題を説明する。その後、問題を与えて自分一人で実習してもらう。この組み合わせを講義の間に2セットほど行くと、1回の講義の内容になる。

以前は、説明だけで練習をあまり実施しないことも試みに行ったが、最近はこのような方法では学生がうまくできないことが多いので、練習問題をおいて方法を詳しく説明するようにしている。もっとも、この場合は練習問題と実習問題の間に少し難易度をつける必要があり、それが問題設定の難しさとなっている。

学生が実習問題を行っている間は、学生たちの間を回り、話しかけるようにしている。最近ではそうでもないが、質問があれば手をあげなさいと言っても、手を挙げる学生は少ないので、質問はないかとこちらから尋ねるほうが質問がしやすいようだからである。また、質問はなくても、学生の進捗状況はどうか？共通した問題点は何か？などを注意して確認するようにしている。そして、時間とのタイミングを計りながらまとめて説明するようにする。

TAをお願いしているのは主に以下の2点である。第1は、私と同様に講義の間に学生の間を回って質問に対応することで、第2点は、提出されたレポートの確認である。

現在のところ定期試験を行っていないので、成績の評価については、出席と毎週のレポートに基づいて成績をつけている。

2.4 教材の準備

教材については、現在は自作したものをHPに掲示して提供している。このようなシステムをとるようになった最初のきっかけは、手頃な値段で適切なテキストがなかったこと（指定しようと思えば、最低でも2冊必要になる）、講義資料の準備が講義開始の間に合わないので、講義を行いながら手直しをしていたためである。以前は、これを印刷して配布していたこともあったが、学生から環境問題に配慮すべきという意見もあり、Webでの公開という方法をとることになった。しかし、これについては賛否両論がある。環境問題の観点からは印刷物がないほうがよいが、やはりディスプレイを見続けるというのは疲れる作業なので、印刷物を配布してほしいという要求はある程度存在している。また、講義のわかりやすさがディスプレイとビデオプロジェクタなどの性能に依存することになるので、少なくともプロジェクタなどの機器の定期的な更新が必要である。

表4：受講者の特徴・教材等（プログラミング1）

受講者数	1クラスあたり100名前後
受講者の所属学部	情報文化学部・自然情報学科（理系）と 社会システム情報学科（文系）
教材	独自のものをWebに掲載
TA数と業務	数名（実習補助，レポート提出確認）

3. 教育の実践2「プログラミング1」

次に、学部専門科目である「プログラミング1」について紹介する。

3.1 教育の目的と方法

この科目は、情報文化学部の学部専門基礎科目として開講されているもので、C言語を用いてプログラミングの基礎について学習することを目的

としている。情報文化学部の場合、他学部と異なり、いわゆる理系の学生と文系の学生が（異なる学科ではあるが）同じ学部の中に存在する文理融合教育を行っている。そして、情報文化学部の専門基礎科目である「プログラミング1」は、これら理系と文系の学生の両方が同時に受講している。全学共通科目の「図情報とコンピュータ」は理系と文系の両方に向けて開講しているが、クラスは別なのでカリキュラム編成をクラス毎に変えることで対応できた。しかし、「プログラミング1」では同じクラスに理系と文系の学生が混在するので、高校生レベルまでの素養が異なる学生たちに同時に教えることになるので、異なる方法を考えなければならない。

表5：カリキュラム編成

ガイダンス	<ul style="list-style-type: none"> ・ 講義方法の説明 ・ システムの利用方法の説明
C言語	<ul style="list-style-type: none"> ・ 基本 ・ 型宣言・配列変数 ・ 演算子 ・ 制御構造 ・ 入出力処理 ・ プリプロセッサ ・ 関数 ・ 構造体
応用プログラミング	<ul style="list-style-type: none"> ・ 数値積分 ・ その他

3.2 講義カリキュラム

本年度の講義カリキュラム編成の例を表5に示す。内容としては、いわゆるC言語のプログラミングの基本を教えた後に、少し応用的なプログラミングの実習を行うことを予定している。ただし、このカリキュラムは最初の講義で予定した内容であって、そのときの学生の進捗具合によって内容を補強したり、短縮したりして柔軟に講義内容を再構成しながら進めていく。特に応用プログラミングについては、教えないこともある。

3.3 講義の進め方

講義の進め方は、「図情報とコンピュータ」の場合とおおよそ同じで、毎週情報文化学部のコンピュータ室で、説明・練習・演習をセットとして

実施する。担当者は私と複数の学生TAである。ただし、「図情報とコンピュータ」に比べると実習問題を多く用意して実習の時間をたくさんとるようにしている。

学生が実習問題を行っている間はできるだけ学生たちの間を回り、話しかけるようにする。このこと自体は「図情報とコンピュータ」の場合と同じであるが、この講義のほうがわからなくて悩んでいる学生が多いので、より重要となる。プログラミングの学習にはある程度の経験が必要だと考えている。そのために、講義では最初から問題をとく（プログラムを作る）ために手を動かしてもらうことになる。その結果、他の講義よりも一層悩む学生は多くなる。そして、講義の中でわかった人わからなかった人について手を挙げてもらって進捗状況を確認することを頻繁に行う。その結果、多くの学生が壁にぶつかっていることがわかるので、それからが先生の出番となる。

問題をやろうとして、それから壁にぶつかるような学生は、ある程度までは理解している場合が多い（と期待できる）ので、そこからヒントを与える。ヒントの与え方はいろいろあるが、言葉によってヒントを与える方法と穴埋め問題を提示する方法をとる。

特に、穴埋め問題を提示する方法では、ポイントを空欄としたプログラムを提示するようにする。プログラムの場合、完全にできていないと正常に動作しない。（ただし、期待している結果を得るためのプログラムは一つではない。）従って、当日の講義で理解してほしいと思っている箇所は正しくできているにも関わらず、当日の講義では、必ずしも重要ではない、それ以外の部分ができているために正常な結果を得られないことはしばしばある。そこで、その回の講義では必ずしも本質的ではないところの解を明示し、講義で理解してほしい点を空欄としたプログラムを提示して穴埋め問題として提示する。そうすると、学生の側としては、プログラム全体の6割程度を答え合わせして（安心して）から、残りの部分を解いていくことになる。

3.4 教材の準備

教材については、現在は自作したものをWebとしてHPに掲示して実施している。このようなシステムをとるようになった最初のきっかけは、自分の講義の進め方、カリキュラム編成にあったテキストがなかったからだが、テキストについてはやはり市販されたものはよく練られたものが多い

ので、現在は参考書として指定して利用している。テキストを教科書として指定しなかったのにはいくつか理由があるが、以前の経験として教科書として指定すると学生は必ず購入することになる。そうすると、学生の側としては教科書通りに進められることを期待するのだが、それは必ずしもその通りにできないので、それで望む人だけが購入するように参考書としている。

4. 講義の実施方針

(1) 単純な方法から教える

問題で要求している結果を得るための方法の一つではない。プログラムを例とすれば、問題で問うている結果を得るためのプログラムはいくつかある。このうち、最初に教えるときには、できる限り簡単な、理解しやすい方法を教えるようにする。それがある程度理解されたところで、応用編やそれ以外のプログラムについて説明をする。

(2) 講義のテーマは一つが望ましい

1回の講義で教える新しい内容は一つから二つ程度のテーマに絞るようにする。たくさんのテーマを同時に教えたいのは山々だが、学生は何が重要かわからなくなる。そこで、テーマは絞るようにする。

(3) 少しずつ山を登る

個人的な経験から言っても、学生にとって講義は必ずしも楽しいものではない。それでも、単位を得るためには避けて通れないので、場合によっては、いやいや受けている。そうすると、あまりに高すぎて太刀打ちできないように思えるような目標を掲げると（別の言い方をすると、太刀打ちできなく見えるように目標を掲げると）、学生としてはすぐに逃げ出してしまうように思う。特に、選択の講義ではその傾向が強いように見受けられる。そこで、少し高いけれども、手を伸ばせば届くように（見えるように）毎回の講義目標をおくようにする。目標が見えるところにあれば、学生はやってみようという気持ちにはなるように思う。

(4) 実習を入れる

本当は講義で重要なことは教えることではなく、それを学生が理解して

くれることだと思う。教える側としては理解しているかどうかをもっとも気にかかることなので、講義の中に実習を入れるようにする。そして、これによって学生の理解の程度を把握し、必要に応じてヒントを与える、または、必要に応じてカリキュラムを微調整することになる。学生の側としては、問題ができれば講義がわかったと自信を持つことができるし、わからなければ復習しなければという動機付けになると思う。ただ、最近の学生はわからないときに、復習しようとは思わないらしいので、これが大きな悩みではある。

また、学生の集中力の点からも、実習を入れることは重要である。高校時代の講義は1コマ1時間程度なので、学部低学年の学生の集中力は1時間が限界のように思われる。従って、飽きがこないように講義の中に実習を入れておくようにすると良いと思われる。

(5) 良好な関係を築く

学生の理解の程度を把握するためには、学生から本音を聞き出さなければならぬ。そのためには、学生と良好な関係を築く必要がある。つまり、学生が話しやすい状況を作らない。

5. 問題点

問題点はたくさんあるが、そのうちのいくつかについて述べる。

(1) 学生の基礎学力の把握の困難さ

第1は学生の基礎学力の程度がわからないことある。講義の内容を留意する場合 $\pm\sigma$ の範囲の学生を中心の対象としたい。これをコアとして、これより上のレベルの学生を満足させるような応用部分を加え、下のレベルの学生を引き上げる努力をするのが理想的である。しかし、講義を始める前に学生のスキルがわからないので、やってみないとカリキュラム編成ができないということなる。

(2) 学生の基礎学力やスキルのばらつきの大きさ

第2に、学生の基礎学力やスキルに幅がありすぎる場合もまた、悩みの種となる。学生のスキルが高くて、低くても、比較的まとまっていればよいのであるが、分散が大きいとコアとなるカリキュラムを編成するのが

非常に困難となる。

(3) $\pm\sigma$ の外側

第3の点として、上記2点を解決するために、講義のレベルがどちらかといえば下方にシフトしていくことがある。つまり、比較的講義に遅れがちな学生に手厚い講義となってしまう、その結果、良くできる学生がつまらなくなってしまうことになる。

(4) 学生と教員の間の良い関係

良い関係とは学生によって異なるので、こちらが、そのつもりであっても、学生の取り方は様々である。そのために、かえって一時的に悪化することもあり得る。

(5) 講義内容の更新

コンピュータを用いる講義に共通していることかもしれないが、内容の変化が非常に速いので、内容の見直しが頻繁に必要となる。講義内容の定期的なチェックにかかるロードは増大しているように思われる。

6. まとめ

今回は、多人数授業の例として、私が担当している2つの科目の授業の様子を紹介した。どちらの講義についても、基本的には学生にあわせて講義カリキュラムの編成を変更していくということで、この基本戦略は、多人数教育であっても、少人数教育であっても正直なところ大きく異なるものではない。ただ、多人数教育の場合は、その分余計に手間を掛けること、効率的に行うようなシステムチックな方法を必要とすることが重要となると考えている。

最初に、この論文についてのお話を頂いたときに、正直なところ、大変光栄に思うと同時に、本当に私がこのようなものを記述して大丈夫だろうか、各方面からおしかりを受けるのではないだろうかと心配した。しかし、このようなものでも、反面教師としてなんらかの参考になるのではないかと思い直して本拙論を記述した次第である。何らかの参考になれば幸いである。

参考文献

香取草之助『授業をどうする！』東海大学出版会

産経新聞社『理工教育を問う』

森田保男、大槻博『実践的の大学教授法』PHP

沼崎一郎『キャンパス・セクシャル・ハラスメント対応ガイド』嵯峨野書院

