

## 文科系大学の情報科学教育における試みと考察

伊 藤 則 之\*

企業のコンピュータ開発部門において、ハードウェア設計を支援するソフトウェアの開発に約30年携わったあと、2011年度より本大学で情報関連科目の授業を担当する機会をいただいた。本大学は文科系大学であり、そこで学ぶ文科系学生にコンピュータ基礎や情報処理論などの情報科学を教えるのが筆者の役割である。

2011年度および2012年度前期を終了し、筆者が担当する授業においてこれまで取り入れた新たな試みについて、またその試みの効果があったのかなどについて、授業から得られたデータを分析しながら考察を行った。

こうした考察を通して、文科系大学で理科系科目を教える先生方が持つ経験や悩みを共有する機会になればとの思いから、出来るだけ具体的な事例や授業に関する分析データも紹介した。研究集会に参加いただいた先生方からは、多くの有益な意見やコメントをいただいた。今回の報告内容に基づいて、授業アンケート結果など様々なデータの詳細な分析を含む報告を別途論集に投稿する予定である。

### 1. はじめに

文科系大学において理系科目を担当する教員の悩みは、題名の中に「文系学生」や「文系学科」などの言葉が使われている教育関係論文の多さからも読み取ることが出来る。

たとえば、論文「文系における理科系科目の

教育 ―法学部の観点から―」(高等教育ジャーナル(北大), 第1号(1996), 畠山武道)では、次のような記述がある。

- ・文系学生は一般に理科系科目に関心がなく、授業に引きつけるのは非常に難しい。
- ・学生は、原理的・抽象的な話よりは、実際の・現象的な話に興味を持ちがちである。
- ・学生の眠気を吹き飛ばすだけの迫力ある講義が期待される。

また、文科系・理科系に関係なく、教師が一方的に話をして板書する講義型の授業の問題点が指摘されるようになり、教師と学生が授業の中でコミュニケーションを持つ双方向型授業やディスカッションなどを取り入れた参加型授業が行われるようになってきた。理科系科目では、講義形式の授業ながら、MITのWalter Lewis教授の物理学の授業のように、講義内容を実験で確かめることのわかりやすさも注目を集めている。

### 2. 試みおよび考察の対象となる授業内容

今回の報告の対象は、下記のような筆者の講義形式の授業とし、プログラミング演習やゼミなどは対象外とした。理由は、プログラミング演習やゼミは30人以下の少人数制であり、演習が中心のために講義形式の授業とは性質が異なるためである。

[前期]

コンピュータ基礎 I

コンピュータのハードウェアの原理を教え

\* 広島経済大学経済学部教授

る授業

□情報処理論 I

情報処理のための基本的な手法および実適用例を教える授業

[後期]

□コンピュータ基礎 II

各プログラミング言語に共通するソフトウェア開発技法を教える授業

### 3. 授業科目毎の試みと考察

#### 3.1 (1年目前期) コンピュータ基礎 I

第1回から第5回までの授業内容について、理解度確認テストを抜き打ち的に実施した。50点満点の選択式のテストであり、60名の学生が受けた。結果は、6点から47点までの間でほぼ一様に分布している。

このように理解力に大きなばらつきのある学生になるべく授業内容を理解してもらうために、プログラム実行の流れや回路の動きについての説明はアニメーション風にした。アニメーションの中には出来るだけ文字による説明も入れることにした。一度説明したあと、各学生にスライドショーとして各自のペースで再度見てもらった。

授業の最後に学生に記入してもらった5段階の理解度を毎回の授業で集計した結果、全体平均は2.79となった。3が「なんとなくわかった」という評価なので、学生全体になんとなくわかってもらえるレベルにはまだ達していない。理解度の平均が高い授業は、コンピュータの歴史やコンピュータの設計現場といった物語的な内容の場合や、たし算回路のように馴染みのある筆算をベースにした内容の場合であった。

#### 3.2 (1年目前期) 情報処理論 I

プログラミング手法の中で、プログラムの中で自分自身を呼び出す再帰プログラムというものが、これはとても重要であるが、理解が

難しい。この再帰プログラムを説明する授業では、データ構造の各データ項目に1人の学生を割り当て、各学生に同じ指令書を渡し、制御と書かれた紙が回ってきたら持っている指令書にしたがって行動することで処理が進むという実験をしてもらった。授業に参加した学生へのアンケートから、実験に参加した学生は理解出来るようになったが、参加していない学生にまだ理解が難しいことがわかった。

情報処理論 I でもコンピュータ基礎 I で行った理解度調査と同じことを行った結果、理解度の全体平均は2.95で、コンピュータ基礎 I より0.16だけ高くなっている。これは、コンピュータ基礎 I での2進数演算や論理回路という内容より、情報処理論 I での情報を処理するソフトウェア的な内容のほうが多少理解しやすいということが要因と思われる。

#### 3.3 (1年目後期) コンピュータ基礎 II

前期授業が完全に講義型の一方通行の授業であったことを反省し、後期の授業に向けた改善として、下記の取り組みを実施することとした。

- ・パワーポイント資料とは別に文字で説明した資料を配る
- ・内容は、網羅するより、重要なポイントに絞る
- ・学生にパソコン上で作業をさせる

パワーポイント資料は Hue Navi のクラスプロファイルの授業資料に登録して配布はしないこととし、文字で説明した資料を配布して復習しやすくした。

また、毎回の授業の最後の20分から30分を使ってプログラミングも教えることとした。簡単なプログラミング言語を使って短時間でスムーズに進める必要がある。そのために、MIT のメディアラボが開発した初心者向けプログラミング言語 Scratch を使うこととした。この言語では、背景や登場キャラクタをまず設定し、

これらに対してプログラミング部品から必要部品を取り出してプログラミングする。実行すると、ステージ上でキャラクタがプログラム通りに動く。

こうした試みの結果、理解度の面では、コンピュータ基礎 I より全体平均で0.67だけ高くなった。これは、新たに取り入れた施策の効果も大きいものと思われるが、コンピュータ基礎 II ではソフトウェア開発に関するものであるために理科系的要素が少なくなったことも要因の一部になっている可能性もある。

### 3.4 (2年目前期) コンピュータ基礎 I および情報処理論 I

1年目前期の反省に基づき、1年目後期で実験を取り入れたことにより学生の興味を引き出すことは出来た。ただ、課題として、講義部分の理解度をさらに上げる施策が必要ということも明らかになった。

そのため、2年目では、その日のうちに授業内容を再度振り返って確認させ、どこがわからないのかを学生および筆者が把握することとした。さらに、それぞれの授業に、下記のような実験を導入することとした。

#### コンピュータ基礎 I

- ・簡易コンピュータの模擬ソフト
- ・論理設計および検証のソフト

#### 情報処理論 I

- ・ハノイの塔のゲームソフト
- ・簡易プログラミング言語
- ・アルゴリズム体験ソフト
- ・オセロゲームのソフト

コンピュータ基礎 I で適用したソフトは、簡易コンピュータを操作したり、回路設計したり出来るソフトである。情報処理論 I で適用したソフトは、情報処理のためのアルゴリズムを体験したり検証したりするためのものである。

大学が実施する授業アンケートの結果をコン

ピュータ基礎 I および情報処理論 I について、それぞれ前年度と今年度を比較した。コンピュータ基礎 I では授業アンケート結果の評価は0.22だけアップしているのが新たな試みは多少効果が出ていると考えられるが、情報処理論 I では0.04のアップのみとなっている。この原因について、今後調査して行く必要がある。

## 4. 全体的な考察と今後の取り組み

### 4.1 授業内容についての考察

プログラミングについて考察すると、下記のようにまとめることが出来る。

#### 好きになる要因

- ・音が出る、絵が動くなどの Audio Visual 性
- ・文字入力が不要なプログラミングの簡易さ
- ・パズルのように考えて解く面白さ
- ・実際に完成したときの達成感
- ・ゲーム性のある題材

#### 嫌いになる要因

- ・英語がベースのプログラミング言語で意味把握の難しさ
- ・文字入力の多さ
- ・1文字の打ち間違えでも動かない厳密性
- ・実行して表示されるものの無味乾燥さ
- ・プログラム記述の分かりにくさ

実験について考察すると、下記のようにまとめることが出来る。

#### 肯定的な要因

- ・授業で聞いたことを自らの実験で確認するわかりやすさ
- ・手を動かすことで、眠くならず、時間が忘れる
- ・ゲーム性のある実験の楽しさ

#### 懐疑的な要因

- ・実験に使うソフトの操作性の悪さ
- ・話で理解出来たものを再度実験することの意味
- ・実験しても意図した通りにならない場合

の実験の意味

- ・実験がうまく行かないときにすぐに聞けないもどかしさ

## 4.2 課題

今回のこのような試みから得られた課題としては、下記のようなものがある。

### □内容面

- ・学生の興味をもっと強く引き付けること
- ・理解出来ない学生の割合が減るような難易度の設定
- ・身近な話に置き換えたり、実際の利用場面の説明を多くすること
- ・参加度の高い学生の満足度の向上

### □運用面

- ・プログラミングや実験などを出来るだけ毎回の授業への適用
- ・全学生への手厚いサポート
- ・新たな題材の発掘（見つからない場合には、筆者自身が開発することも含めて）

### □その他

- ・プログラミング言語は文字入力不要なものを利用
- ・実験や演習と授業内容との強い関連付け

内容面において、「理解出来ない学生の割合が減るような難易度の設定」と「参加度の高い学生の満足度の向上」という2つの課題を同時に解決することはかなり難しいと思われるが、何とかして解決しなければならない課題である。

## 4.3 今後の取り組み

今後の取り組みとしては、下記のような3点

を検討している。

- 1) 学習の意欲と効果を高めるための測定難易度、理解度（テストによるもの、学生の印象）、参加度、満足度
- 2) 新たな実験教材の調査および作成（探してもない場合は自作するなど）
- 3) コンピュータサイエンスアンプラグドなどの利用（コンピュータ科学を身近なものに置き換えた体験を通してわかりやすく学習するための教育手法）

コンピュータ基礎Ⅰはハードウェアに関する内容であり、人間が登場しない典型的な理科系で科目である。昨年度と今年度の授業の中でハードウェア構成を理科系教科書的な図をそのまま使って説明した。この説明を行った今年度の授業の出席カードに、ハードウェア構成は難しいので何かに例えて説明して欲しいという要望があった。

そこで、次の回の授業では、顧客から依頼を受けた仕事を会社の組織がどのように処理して行くかに例えてハードウェア構成を再度説明したところ、その授業の出席カードには、例えて説明してもらえたのでわかりやすかったという感想があった。

このことは、人間が一切登場しないコンピュータ基礎Ⅰのような科目においても、なるべく人間社会の仕組みと対応させて身近なものとして説明することにより、文科系大学の学生も情報科学系科目を身近なものとして親しみを持つ一つの例を示しているのかもしれない。