

大学1年生へのプログラミング教育の試み

My Approach to the Lecture on Computer Programming for First Year Students in the University

伊知地 宏
Hiroshi Ichiji

ラムダ数学教育研究所
Lambda Mathematics and Education Laboratory

1 はじめに

昨今、多くの大学で大学1年生全員を対象に、コンピュータを電子文房具として使いこなせるようになるためのコンピュタリテラシーの授業が行われるようになってきている。コンピュタリテラシーの授業の充実に伴い、プログラミングの授業も様子が変わってきた。

大学におけるプログラミングの授業は、少し前まで計算機科学を専門とする学科、あるいは計算機を必ず使う必要のある学科で行われているだけだった。しかし今日では、1年生を対象にコンピュータプログラミングの授業が開設されているところも増えている。大学1年生を対象にするプログラミングの授業では、受講する学生の多くは、将来プログラミングを生業とする者でなく、ちょっとした興味やこのくらい知らないといけなかもしれないという強迫観念により受講している者である。

東京大学教養学部前期課程冬学期に行った「コンピュータプログラミング I」の私の授業でも、1年生の受講者が9割を占めており、コンピュタリテラシーの授業のおかげでコンピュータを使ったことのない人はいなかったが、これまでにプログラミングを経験したことのある人は2割弱であり、その多くはBasic、一部がC++の経験者であった。

このような進学先が決定しておらず多様な学生を対象に授業を行うためには、情報系学科でのプログラミング教育と違う視点、方法が必要になると思われる。情報系学科を目指す学生だけではないことから、プログラミング言語やプログラミング手法を学ぶための忍耐力を期待してはいけない。学生を飽きさせないためにも、早い時期にプログラミングは楽しいと思わせるような授業の進め方や題材が必要となる。

本稿では、私が昨年度に東大の授業で試みた方法について説明する。私の授業はJavaのプログラミング入門教育で、後ほど詳しく述べるが、グラフィックスを主体にしたプログラムの動きが見えるようにした。また初期の時点でオブジェクト指向の説明をしてしまい、細かな文法は後でやるようにした。結論としては、学生は12回の授業と演習課題を通して、プログラミングを知らなかったとは思えないほど成長したので、試みはかなり成功したのではないかと考えている。

本稿の構成は、第2節で従来のプログラミング教育について簡単に触れ、第3節で私が行った授業を詳細に説明し、第4節で授業に対するアンケート結果を述べ、第5節で考察を行うようになっている。

2 従来のプログラミング教育

従来の情報系学科のプログラミング入門の授業では、Javaなどのオブジェクト指向言語を教える場合でも、最初に文法の説明に十分な時間をさき、かなりしばらくしてからクラスやオブジェクトの説明に入ることが多い。さらに時間の都合でグラフィックスやアプレットについては簡単に触れるだけで、オブジェクト指向の特徴をあまり理解できないままに授業が終了してしまうように思われる。しかし、情報系学科ではこのような不足分を補う他の授業があるので、上記のことはたいした問題とはならないし、計算機科学を専門とする者にはプログラミング言語をしっかりと理解させるためにも、文法などの概念をしっかりと理解させる必要がある。

しかし、情報系学科に進学するとは限らない一般学生にプログラミング入門の授業を行う際には、プログラミングの授業を受けるのはこれが最初で最後となるかもしれない。また、学生自体が計算機科学を専門とする気がなく、プログラミングを自ら積極的に学び、さらに自ら身に付けようと思わない者も多いと予想される。

このような状況では、文法を長々と説明するとそれだけで飽きてしまう学生も多いだろうし、非常に単純な短いトイ・プログラムを書いているだけではプログラミングの面白さ、妙味を知ることもしないであろう。一般の大学1年生にプログラミングの授業を行う際には、上記の点を考慮し、出来るだけ早いうちにプログラミングの面白さを感じさせる必要があると考えている。

3 プログラミング教育への新しい試み

3.1 授業の特徴

最近の学生に良く見られる傾向として、視覚的なものに対する興味や反応に優れているところがある。その反面、論理的な思考が必要とされる数学や言葉の厳密な理解が求められる国語などを不得意とし、どちらかというと文字による理解が劣っている。またわからないこと、知らないことに対して忍耐強く取り組んで、苦勞の末に理解するというのも苦手としているようである。

このような状況と先に述べたことから、私が行うJavaプログラミング入門の授業「コンピュータプログラミングI」では、学生に早いうちにプログラミングに対する興味を持てるように、また自らプログラミングをしてみたいと思うように、以下のような方針を立てた。

- 視覚的に理解できる内容を中心にし、演習でもグラフィックス関係を主体とする。
- 文法の細かいことを長々と説明するより、プログラム例を実際に示して、プログラミングに慣れさせる。
- 文法や概念に対しては具体例の中で説明する。

3.2 授業内容

上記の方針のもと、早いうちにクラスやオブジェクトの説明をし、基礎的なことは極力簡単にすませて、早めにグラフィックスの話をしてプログラミングに興味を持たせる授業計画を立てた。またi-アプリプログラミングにも触れることを特徴とした。この結果、具体的な授業内容は以下ようになった。

月日	回数	内容
10/9	第 1 回	Java プログラミング入門 (1) + はじめての Java + 入出力と四則演算 + 本日の課題
10/16	第 2 回	Java プログラミング入門 (2) + 前回の課題の解答 + 変数と定数 + メソッド + クラス + 値とオブジェクト + 本日の課題
10/30	第 3 回	Java プログラミング入門 (3) + 前回の課題の解答 + 条件分岐 + 繰り返し + 再帰 + 本日の課題
11/6	第 4 回	Java プログラミング入門 (4) + 前回の課題の解答 + 配列 + API + 文字列: String オブジェクト + 本日の課題 + 演習の解答例
11/13	第 5 回	オブジェクト指向とアプレット + 前回の課題の解答 + 例外処理と数値の変換 + アプレットプログラミング入門 + クラス継承, オーバーロード, オーバーライド + 本日の課題
11/20	第 6 回	グラフィックス (1) + 前回の課題と演習の解答 + 図形の描画 + カラー + イベント処理 + フラクタル図形入門 + 本日の課題

月日	回数	内容
11/27	第 7 回	グラフィックス (2) – フラクタル図形 – + 前回の課題の解答とプログラム例の追加 + シルピンスキーのギヤスケット + コッホ曲線 + 樹木曲線 + ジュリア集合 + マンデルブロー集合 + 本日の課題
12/4	第 8 回	グラフィックス (3) – アニメーション – + アニメーション入門 + マルチスレッドプログラミング + アニメーション + 本日の課題 + 演習の解答例
12/11	第 9 回	グラフィックス (4) – GUI – + 前回の課題の解答例 + GUI の部品 + レイアウト + 本日の課題
12/18	第 10 回	アルゴリズムとデータ構造 (1) – ソート – + 単純ソート + 挿入法ソート + クイックソート + ヒープソート + マージソート
1/15	第 11 回	アルゴリズムとデータ構造 (2) – 基本的なデータ構造 – + リスト + スタック + キュー + 木 + 本日の課題
1/22	第 12 回	i アプリ プログラミング + i アプリの作り方 + i アプリ用 API + サンプルプログラム

3.3 レポート 課題

課題はほぼ毎回出したが、レポート提出課題は以下の 6 回となっている。

課題提出回	課題内容
第 2 回	円錐の体積と表面積の計算
第 4 回	n 次正方行列の積の計算
第 5 回	アプレットで図形の描画
第 7 回	フラクタル図形の計算と描画
第 9 回	アニメーションまたはゲーム作成
第 11 回	後置記法の簡易電卓または S 式で表現された四則演算の計算

4 レポートとアンケートの結果

4.1 受講生の分布

受講生は以下のような構成になっている。

学年	人数	文 1	文 2	文 3	理 1	理 2	理 3
1 年生	97	2	1	3	77	11	3
2 年生	8	1	1	0	2	3	0
過年度	3	0	0	0	3	0	0
合計	108	3	2	3	82	14	3

4.2 レポートの提出状況と成績の分布

レポートは最初 3 回がやさしく、次の 2 回が難しくなっている。最後のレポートは両者の中間程度の難しさとなっている。また、100 点が私が想定していたレベルを満たすもので、それを上回るものには 100 点以上の点を付けている。

	提出数	～ 59 点	60 ～ 79 点	80 ～ 99 点	100 ～ 119 点	120 点～
第 2 回	98	2	6	52	40	-
第 4 回	88	10	11	5	62	-
第 5 回	90	3	5	8	74	-
第 7 回	72	3	4	12	39	14
第 9 回	59	2	1	5	24	27
第 11 回	52	5	2	13	32	-

4.3 アンケートの結果

大学が行った学生へのアンケートの結果は以下のようにになっている。私の授業の分は回答数 39、全体の回答数は 494 となっている。

授業の難易度					
	やさしすぎる	ちょっとやさしい	ちょうどよい	ちょっと難しい	難しすぎる
私の授業	2.6%	7.7%	10.3%	48.7%	30.8%
全体	2.8%	7.1%	30.6%	38.7%	20.6%

授業の内容					
	少なすぎる	ちょっと少ない	ちょうどよい	ちょっと多い	多すぎる
私の授業	2.6%	0.0%	35.9%	46.2%	15.4%
全体	2.0%	7.7%	44.3%	35.4%	10.3%

新しい知識・学力の獲得					
	とても役に立った	かなり役に立った	普通である	あまり役に立たなかった	ほとんど役に立たなかった
私の授業	35.9%	46.2%	10.3%	5.1%	2.6%
全体	30.8%	41.5%	22.5%	3.0%	1.8%

課題の難易度					
	やさしかった	少しやさしかった	適切だった	やや難しかった	難しかった
私の授業	0.0%	7.7%	15.4%	23.1%	53.8%
全体	5.5%	7.3%	26.1%	27.5%	26.5%

5 考察とまとめ

レポート課題の得点を見ると、グラフィックスを題材にしたときの方が概して成績が良い。特に、第9回の授業で出した課題では提出率が非常に落ちることを想定したが、予想に反して50%以上の受講生から提出があった。学生がグラフィックスの課題に対して興味を持ち、さらにプログラミング技術力も向上した現れだと思われる。それに反して記号処理の課題である第11回の授業の課題では、成績が急に悪くなっており、これは学生の問題なのか、それとも私の授業の問題なのかわかっていない。

アンケートからは、他のクラスと比べて授業の難易度が高く、教えた内容の量も多かったということがわかる。この結果は、おそらくグラフィックス関係のことを多く教えたことに起因していると考えている。しかし、新しい知識・学力の確立に関するポイントの高さから、学生は授業内容には満足していることがうかがわれる。レポート課題では、学生の多くが非常に苦しんだことが読み取れるが、第9回の授業での課題(アニメーションまたはゲーム)では独創的な解答やプログラミング能力の高さを感じるものも多々見られた。

受講前にはほとんどプログラミングについて経験のない学生が多かったのに、半数近くの学生が非常に満足できるプログラミングが出来るようになったことは、授業をした私自身が驚いている。こういった授業のやり方が本当に良いかはわからないが、今年度も同様に実験的な授業を行い、効果を確認してみたいと思っている。