

# プログラミング実習における コンピュータを使わない教材とその効果 Teaching a Computer Programming Language without using a Computer

玉木 徹†  
Toru Tamaki

## 1. はじめに

情報関連の教育機関において、プログラミング教育は重要な位置を占めている。本工学部情報工学科でも1,2年生向けに2学期に渡ってプログラミングの講義と実習を行っており、卒研などでプログラムを作成する際の基礎的な実力を習得することを目的としている。しかし、プログラミング実習を受講する学生のほとんどが初心者であるため、次のような問題が生じている。

まず、講義や実習で使用する市販の(C言語の)テキストはすべて図1のようなサンプルプログラムから始まっている。しかし、これには初学者にとって理解できない不必要な項目が多すぎ[1]、他の言語を学ぶときにはこれとはまったく違ったサンプルプログラムを覚えなければならない。また、全てのテキストは実際にプログラムをコンパイルして実行することを前提にしている。しかし、初心者がエラーなしに最初からコンパイルできることなく、不慣れたエラーメッセージに悩まされるため、コンパイルすること自体が困難でもある[2]。

そこで、これまでにプログラミング初心者を対象としたプログラムの基礎を習得するための教材を提案してきた[3,4]。本稿では、その取組の結果について述べる。

## 2. 提案教材とこれまでの結果[3,4]

提案する初心者向けプログラミング教材[5]は、図2に示した問から始まる約20項目(式評価、型、制御構造、フローチャート等)全250問の練習問題である。2003年1年生後期15週のうち最初の4週は計算機を使用せず、この教材を使用し、残り11週は実際にプログラムを組む課題を3つ出題した。そして図3を含む15題を出題する中間テスト(持ち込みなし)とアンケートを、教材使用期間直後の4週目と約2ヶ月後の11週目に行った。

その結果、2回のテスト結果にあまり差は見られなかった。この結果の解釈は難しいが、2ヶ月後に成績が落ちなかったことで理解が定着していたということ、また成績が上昇もしなかったため、2ヶ月間では実際にプログラミングを行っても理解が進まなかったということが言える。

また二重ループやループと分岐など、制御構造が二つ以上組み合わせられた問題は成績も悪く、アンケート結

```
#include <stdio.h>
int main(int argc, char *argv[]){
    printf("Hello World!\n");
    return 0;
}
```

図1 : C言語の簡単なサンプルプログラム

```
問 「評価」とは計算することである (例 : 1+1 の評価結果は2)。次の式を評価せよ。
    1+2      22*3.3
    2-1      10.2+5.1
問 「評価」とは式の真偽を判断することである (例 : 1>0 の評価結果は真)。次の式を評価せよ。
    0<1      1.0 != 10.0
    -1>=3    3 < -1.5
```

図2 : 提案教材中の設問の例

```
配列 a[128] の要素の和を表示するように下線部を埋めよ。
int i, sum=0;
for( _____ )
    sum+=a[i];
printf("%d\n", sum);
```

図3 : 中間テスト用の問題の例

果でも苦手意識が強いことが分かった。さらに2回の中間テストと、最終的な実習や講義の成績との相関は0.5~0.6程度であり、比較的高い相関があることが確認できた。

## 3. 長期の定着度

前回の2回のテストとアンケートの間隔は2ヶ月という短期間であったが、今回は半年の間隔をあけて行った。2004年1年生後期のプログラミング実習(1期目)15週のうち最初の5週は計算機をを使用せずに提案教材[5]を使用し、残りは実際にプログラムを組む課題を出題した。そして前出と同じ中間テスト(持ち込みなし)を5週目(2004年11月)に行った。そして2回目は半年後(2005年6月)の、2005年2年生前期のプログラミング実習(2期目)中に行った。この半年の間に、学生は実際にプログラムを組む課題を8つ程度こなしている。

テストと同時に、教材の各項目について自分がどの程度理解したかを5段階で回答するアンケートを行った。この結果を図4に示す。前回のアンケート結果と同様に、1回目のアンケートでは制御構造が組み合わせる「二重ループ」や「ifとループの組合せ」などをよく理解したという学生は少なかった。しかし半年後の2回目のア

† 新潟大学 自然科学研究科/工学部情報工学科

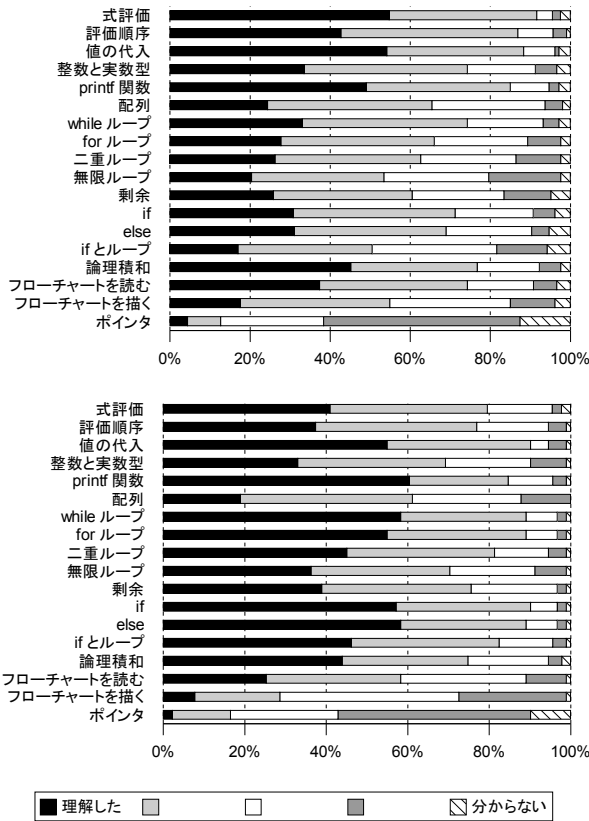


図4：理解度アンケート結果（上）教材期間直後（2004/11）（下）半年後（2005/6）

アンケートでは、これらの項目はいずれも理解したと回答した学生が増加した。これは、実際のプログラミング課題を通して制御構造に対する理解が深まった結果だといえる。一方で、「式評価」や「評価順序」などのもっとも基礎的な部分を理解したとする学生が、2回目では減少している。実際に制御構造を使ってプログラムを組む課題を行っているにもかかわらず、基礎的な式評価に苦手意識が広がったということになる。これは、制御構造を使ってはいるが、漫然とプログラムを組み立てていて、処理の流れをしっかりと把握せずにプログラミングを行っているのではないかと考えられる。

以上のことは、図5に示すテスト結果にも表れている。図3の問題に対応する「forと配列」や「ifとループ」については成績が向上しているが、「式評価」や「数式の対応」「式の訂正」など基本的な項目で、成績が落ちてしまっている。また「フローチャート」と「ポインタ」の項目も大きく成績が落ちている。

この結果から、ループや分岐などの制御構造に慣れるには、従来から行われているように実際にプログラムを組む課題を行った方がよいが、基礎的な理解を身につけるには、提案する教材のように集中した項目を（計算機を使わず自分の頭を使って）考える訓練を継続的に行うほうがよい、と言える。

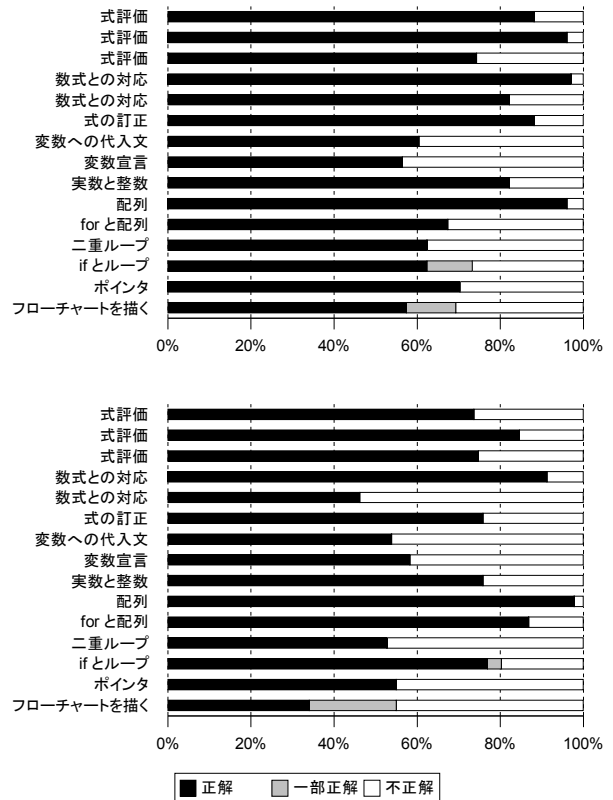


図5：中間テスト結果（上）教材期間直後（2004/11）（下）半年後（2005/6）

#### 4. おわりに

本稿ではプログラミング初心者のための教材の提案とその実践結果について報告した。前回（2003年度[3,4]）と今回（2004年度）の結果から、学生が苦手とする項目（制御構造の組合せ等）とその対応策や、プログラミング課題と提案教材の実習における効果の比較などが検討できた。今後の課題は、これらの結果に基づき教材と課題をどのように実習に採り入れるのかの方法を確立することである。

#### 参考文献

- [1] 神林靖, 「オブジェクト指向プログラミング教育への疑問」, 日本工学教育協会平成15年度工学・工業教育研究講演会論文集, pp.9-10, 2003.
- [2] Qusay H. Mahmoud, Wlodek Dobosiewicz, David Swayne, "Making Computer Programming Fun and Accessible," *IEEE Computer*, Vol.37, No.2, pp.106-108, 2004.
- [3] 玉木徹: 「プログラミング実習における新しい教材とその指導方法」, 日本工学協会平成16年度工学・工業教育研究講演会講演論文集, pp.17-18, 2004.
- [4] Toru Tamaki, Takeshi Hagiwara, Yoshinobu Maeda, Yasuto Nakamura, "An Approach to Teaching a Computer Programming Language," *Proc. of the 9th World Conference on Continuing Engineering Education*, pp.389-394, 2004.
- [5] プログラミング基礎実習問題集, <http://www.vision.ie.niigata-u.ac.jp/~tamaki/study/tamaki2005drill.pdf>