

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

A Novel Training Method for Programming Course

玉木 徹*

Toru Tamaki

キーワード：プログラミング, 教材, 実習

Keywords: programming, material, training

1. はじめに

コンピュータを教育する機関において、プログラミング教育は重要な位置を占めている。本学工学部情報工学科でも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る[2]。また他の言語を学ぶときには、同じことをするにも、図1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が、初心者がエラーなしに始めからコンパイルできることはまずない。そのため、不可解なエラーメッセージに悩まされることになり、コンパイルすること自体が高いハードルになってしまう[1]。

そこで、本稿では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、その実践について述べる。

2. 提案する教材

ここで提案するプログラミングの教材は、図2に示すような多数の小問である。各問には同じことを行う問題を多数掲載しているので、この教材は小中学校で行うドリルや練習帳などに似ている。この教材の利点は、実際にコンピュータを使用しないこと、どのプログラミング言語にも共通する基礎的な事柄から開始することの二点である。

一つ目の利点であるコンピュータを使用しない、つまりプログラムをテキストエディタで入力することをしない、ということは、プログラミング初心者にとって重要なことである。図1のような簡単なプログラムでさえ、多くの種類の括弧（<>, (), [], {}）を使い分けなければならない。そのような知識のない初心者にはそれ以前の訓練が必要である。

```
#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

```
int i, sum=0;
for(____;____;____)
    sum+=a[i];
printf("%d\n", sum);
```

図3：調査用の問「配列a[128]の要素の和を表示するように下線部を埋めよ」

利点の二つ目は、そのような入力以前の訓練のためのものでもある。プログラミングにおいて理解すべき概念は、mainやprintfなどではなく、プログラムを形作る断片となる式の評価である。これは現在のほぼ全ての手続き型言語に共通することであり、プログラミングの基礎というべきものであるにもかかわらず、そこに重点をおいて訓練するようなカリキュラムやテキストは存在していない。

このような小さな問題を多数反復して行うことにどれだけの意味があるのかを考えるために、まず現状の学生のプログラミング能力を調査した。約100名の学生に図3を含む5問程度の問題を出題し、解答してもらった。図3はループと配列という基本的な事項を知っていれば解ける問題であるが、正解率はたった41%だった。解答した学生たちは約1年間のプログラミングの講義と実習を終える頃であり、またテキストを自由に見ながらの解答であったことを考慮すると、あまりに基礎知識がないと言うべきである。もっとも、このような簡単な問題だけで学生のプログラミング能力を判断することはできない。しかし、このような知識さえもないのに、独力でプログラムを組むことはできないのは確かである。

*新潟大学 自然科学研究科/工学部情報工学科

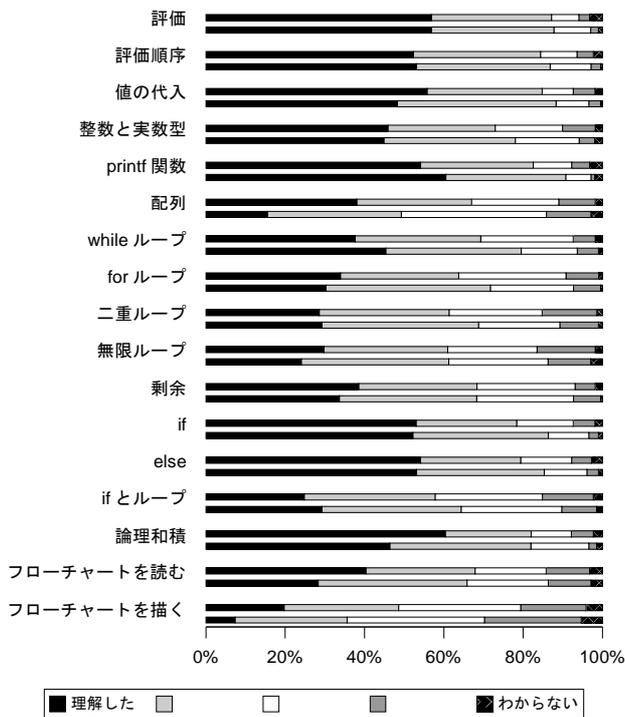


図4：理解度アンケート結果（上）ドリル期間直後（下）二つのプログラミング課題期間の後

3. 教材を使用した実習

実際に前述の教材をプログラミング実習において実践した。対象となる学生はいままでプログラミングの実習や講義を受けたことのない初心者約200人で、図3の問題を解いた100人の学生とは異なる。教材は約150ページ230問（各問には図2のように複数の小問題や問3のような問題を含む）を準備し、その内容は基本となる式評価から制御構造であるループと分岐までとした（図4の各項目がそれにあたる）。学生は毎週2コマ（3時間）の実習で50問程度ずつを机上で解き、各実習の最初と最後では復習のための小テストを行った（ドリル期間）。これを4週間行った後、ユークリッドの互余法などのプログラムを作成する3課題を各4週ずつ行った（課題期間）。

そして、教材と同程度の問題を中間テストとして、ドリル期間直後と、その後の二つの課題期間の後の二回（同じような内容で）学生に出題した。また同時にアンケートにも回答してもらった。教材の各項目について自分がどの程度理解したかというアンケートの結果を図4に示す。

図4のアンケートから分かることは、二重ループや、ifによる分岐とループの組合せなど、いくつかの制御構造が組み合わさると、学生は「難しい」と感じるということである。それは図5の中間テストの結果にも表れている。

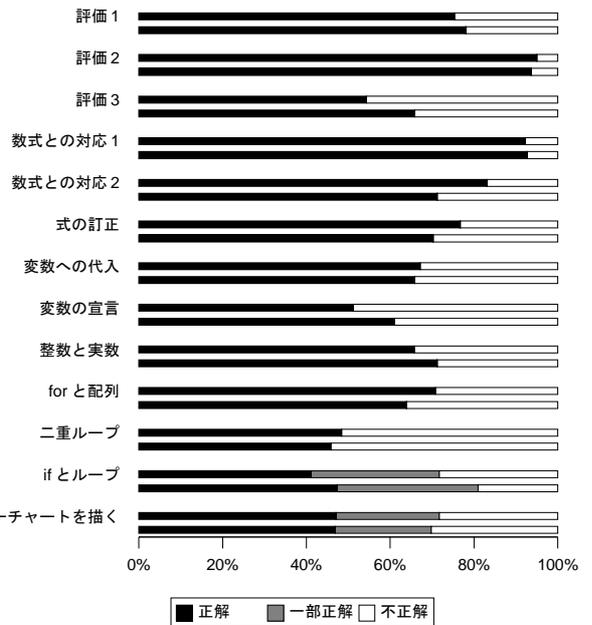


図5：中間テスト結果（上）ドリル期間直後（下）二つのプログラミング課題期間の後

果にも表れている。それらの正解率は50%以下だった。

また、図3の問題に対応するのは図5における「for と配列」についての問題である。どちらもほぼ同じ内容であるが、正解率が70%程度になっている。この中間テストのときには学生はテキストを見ずに解答し、また課題期間中はほとんど教材を使わなかったことを考えると、基礎的な事項は知識として身についたと言える。

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。作成した教材は、結果としては問題数が多く範囲が狭かった。そのため、問題を精査し、かつ範囲を関数やポインタ、構造体にも広げたほうがよいとの学生からの要望もあった。実際にプログラムを組む時点で、この教材を行ったことがどれほど効果があるのかはまだ調査を行っていない。しかし、以前は学生にサンプルプログラムを入力させる際に、括弧やダブルクォート(")が対応していなかったりセミコロンがなかったり、という簡単な事柄でもエラーとなっていたが、そのようなことで戸惑う学生は非常に少なくなった。これも教材の効果であると思われる。

参考文献

- [1] Q. H. Mahmoud, et al., "Making Computer Programming Fun and Accessible," *IEEE Computer*, Vol.37, No.2, pp.106-108, 2004.
- [2] 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.