AN APPROACH TO TEACHING A COMPUTER PROGRAMMING LANGUAGE

Toru Tamaki\textsuperscript{1} Takeshi Hagiwara\textsuperscript{2}
Yoshinobu Maeda\textsuperscript{2} Yasuo Nakamura\textsuperscript{2}

\textsuperscript{1}Graduate School of Science and Technology, Niigata University, Niigata, Japan
\textsuperscript{2}Faculty of Engineering, Niigata University, Niigata, Japan
Programming Languages

Pascal / FORTRAN / C / C++

FORTRAN / C / C++

Perl / Ruby / PHP

Visual C++
Visual Studio .NET

Procedural language

Non-procedural language
Lisp / Prolog

Education

Science

Internet

Commercial software
Our Programming Course

30 weeks in total

- Two semesters, for a year
- 45h for lecture (class): 1.5h / week
- 90h for lab (practice): 3h / week
- C language
Students' Present Ability

About 100 students
had learned almost for a year.
allowed to see any texts, but no talking.
tasked 5 questions in 15 min.

Question:
"rewrite the mathematical
equation in C statement".

Answer:
x = \frac{a+b}{c - \frac{d}{a+2}}

x = (a+b) / (c-d/(a+2))

% of correct answers: 74%
Students' Present Ability

**Question:** "fill the blanks to sum up all elements of the array a[128]"

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

**Answer:**

```c
for(i=0;i<128;i++)
```

**% of correct answers : 41%**

- Lack of fundamental knowledge
- Necessity for training the basics
Well Known, but Harmful
Sample Program

in C language:

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

Problems :
- Required to input the program, and execute it
- Many unknown symbols and rules
- No common basics with other programming languages
Well Known, but Harmful
Sample Program

in C language:

1:   #include <stdio.h>
2:   int main(int argc, char *argv[])
3:      printf("Hello World!\n");
4:      return 0;
5:   }

Line numbers:
- Provided for making code ease to see
- Must not be written into the actual program file
- May be seen as a part of the program for beginners
Well Known, but Harmful
Sample Program

in C language:

1:    #include <stdio.h>
2:    int main(int argc, char *argv[]){
3:        printf("Hello World!\n");
4:        return 0;
5:    }

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


Well Known, but Harmful
Sample Program

in C language:

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

Wrong parentheses
Miss spelling
No semicolon
No correspondence
Well Known, but Harmful Sample Program

test.c: In function `main':
test.c:1: undefined or invalid # directive
test.c:1: `#include' expects "FILENAME" or <FILENAME>
test.c:3: possible real start of unterminated constant
test.c:3: unterminated string or character constant
test.c:5: parse error before `}'

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

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

in Pascal:    program hello(input, output);
              begin
              writeln('Hello World!');
              end.

Compatibility:
- Different rules for different languages

NO BASICS!
Proposed Materials

Programming Drill

✓ **Beginning with the basics.**
  ‧ Common to all procedural languages.
  ‧ Based on C, but applicable to Pascal/Fortran/C++/etc.

✓ **Without execution on computers.**
  ‧ Writing answers on a paper
  ‧ Computing by students' brains, not by computers

✓ **Without any knowledge about programming**, only a little mathematics in high-school level is needed.
  ‧ Fundamental mathematical functions (ex: \( \sin \), \( \cos \), \( \exp \))
  ‧ Some symbols (ex. \( \sum \pi e \))

✓ **Repeat one topic with many exercises** to train the sense of programming.
Exercise 1.
"Evaluation" is to calculate (ex: the evaluation of 1+1 is 2). Evaluate the following statements in C.

<table>
<thead>
<tr>
<th>statement</th>
<th>answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1+2</td>
<td>3</td>
</tr>
<tr>
<td>22*3.3</td>
<td>42.6</td>
</tr>
<tr>
<td>2-1</td>
<td>1</td>
</tr>
<tr>
<td>10.2+5.1</td>
<td>15.3</td>
</tr>
<tr>
<td>10/5</td>
<td>2</td>
</tr>
</tbody>
</table>
### Exercise 2.

"Evaluation" is to judge a statement whether it is true: 1 or false: 0 (ex: The evaluation of 1>0 is 1, that is, true). Evaluate the following statements in C.

<table>
<thead>
<tr>
<th>statement</th>
<th>answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 &lt; 1</td>
<td>1</td>
</tr>
<tr>
<td>1.0 != 10.0</td>
<td>1</td>
</tr>
<tr>
<td>-1 &gt;= 3</td>
<td>0</td>
</tr>
<tr>
<td>3 &lt; -1.5</td>
<td>0</td>
</tr>
</tbody>
</table>
Contents of the Drill

Exercise 3.
A statement is evaluated in left-first order. Evaluate the following statements in C.

<table>
<thead>
<tr>
<th>statement</th>
<th>answer:</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2+4-3.5</td>
<td>1.5</td>
</tr>
<tr>
<td>3*8/4</td>
<td>6</td>
</tr>
<tr>
<td>10/2/5</td>
<td>1</td>
</tr>
<tr>
<td>1.1 + 0.1 &lt; 1.1</td>
<td>0</td>
</tr>
</tbody>
</table>
Topics in the Drill

• evaluation of statements
• value assignment to variable
• initialization of variable
• mathematical functions
• int and float types
• declaration
• value range
• char type
• printf function
• array
• initialization at declaration

• while loop
• double loop
• infinite loop
• if
• else
• modular
• arithmetic operators
• for loop
• flowchart

20 more topics about 250 exercises
Practical Training with Drill in Our Programming Lab

- Exam for last week of topic
- Drill
- Drill
- Exam for today's topic

- 3 hours for weekly lab
- About 50 exercises per week
- Two small exams at the beginning and the end of each weekly lab
Practical Training with Drill in Our Programming Course

1st exam

Oct. 2003

1

weeks

1.5 hours / week

class
(lecture)

practice
(lab)

1st exam

Feb. 2004

15

2nd exam

drill : 4 weeks
3 hours / week

programming task
3 hours / week

programming task
3 hours / week

programming task
3 hours / week
Questionnaires on understanding drawing flowchart

- well
- not at all

1st exam
2nd exam

- evaluation
- order of evaluation
- assignment to variable
- int and float types
- printf function
- array
- while loop
- for loop
- double loop
- infinite loop
- modular
- if
- else
- combination of if and loop
- &&, ||
- understanding flowchart
- drawing flowchart
Results of short exams

- **evaluation 3**: 74%
- **int and float**: 41%

**Correct**
- evaluation 1, evaluation 2, rewriting equation 1, rewriting equation 2, correcting statement, assignment, declaration, int and float, for and array, double loop, if and loop, drawing flowchart

**Partially Wrong**
- evaluation 3

**Wrong**
- rewriting equation 1, rewriting equation 2, correcting statement, assignment, declaration, int and float, for and array, double loop, if and loop, drawing flowchart
Correlations among grades of lecture, lab, and exams.

<table>
<thead>
<tr>
<th></th>
<th>lecture</th>
<th>lab</th>
<th>1st exam</th>
<th>2nd exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecture</td>
<td>1.00</td>
<td>0.59</td>
<td>0.48</td>
<td>0.54</td>
</tr>
<tr>
<td>lab</td>
<td>0.59</td>
<td>1.00</td>
<td>0.60</td>
<td>0.66</td>
</tr>
<tr>
<td>1st exam</td>
<td>0.48</td>
<td>0.60</td>
<td>1.00</td>
<td>0.69</td>
</tr>
<tr>
<td>2nd exam</td>
<td>0.54</td>
<td>0.66</td>
<td>0.69</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Grades:

- Lecture and lab are graded independently.
- Lecture: writing examination
- Lab: two exams (5% each) and reports of three tasks (30% each).
Scattered Diagrams
Conclusions

- Proposed a new material to train the basics of programming, and shown the result of the practical training in our programming course.
- The sort examinations on the drill have some correlations with grades of lecture and lab.
- The effect of the drill on students' ability on programming have not yet validated.