AN APPROACH TO TEACHING A COMPUTER PROGRAMMING LANGUAGE

Toru Tamaki[1] **Takeshi Hagiwara**[2]
Yoshinobu Maeda[2] **Yasuo Nakamura**[2]

[1]Graduate School of Science and Technology,
Niigata University, Niigata, Japan
[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

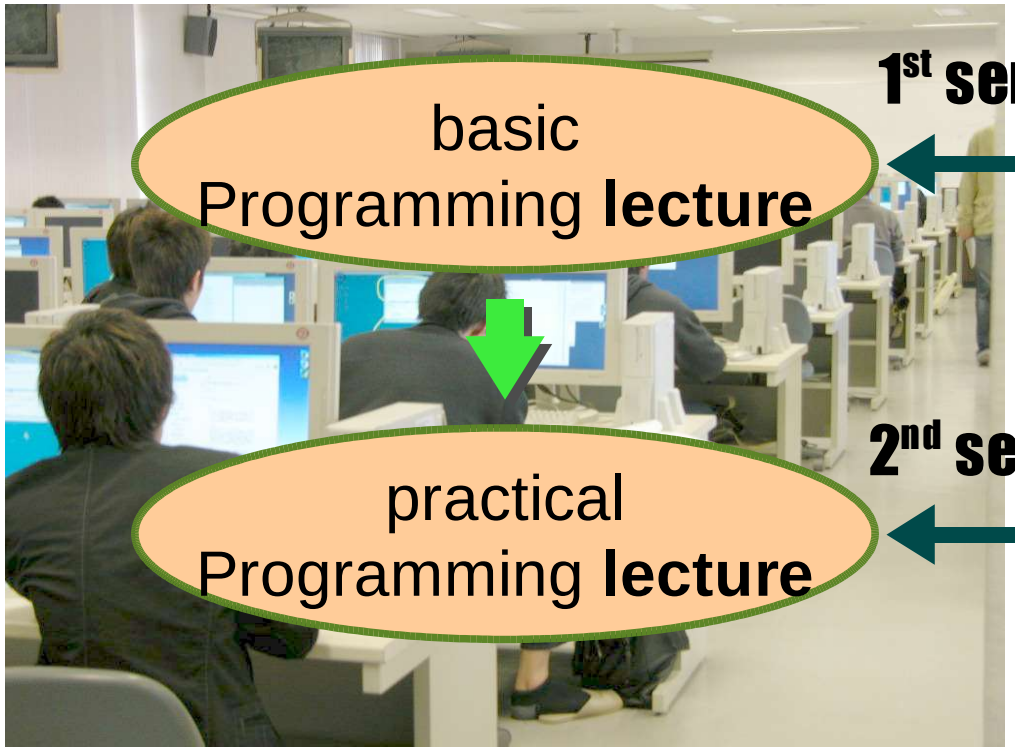
Internet

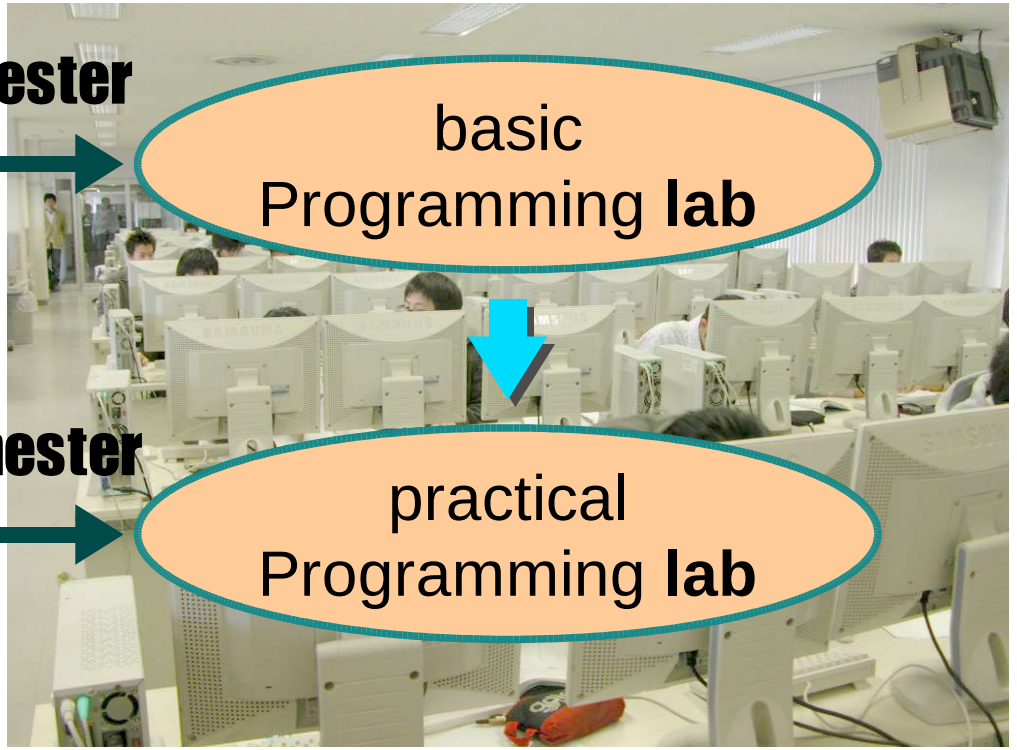
Commercial software

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





1st semester

basic Programming **lecture**

basic Programming **lab**

2nd semester

practical Programming **lecture**

practical Programming **lab**

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".

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

- ***Answer*** **:**

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

- ***% of correct answers*** **:** 74%

Students' Present Ability

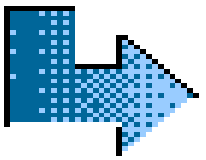
- ***Question*** **:**
  "fill the blanks to sum up all elements of the array a[128]"

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

- ***Answer*** **:**

```
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:

```
#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:

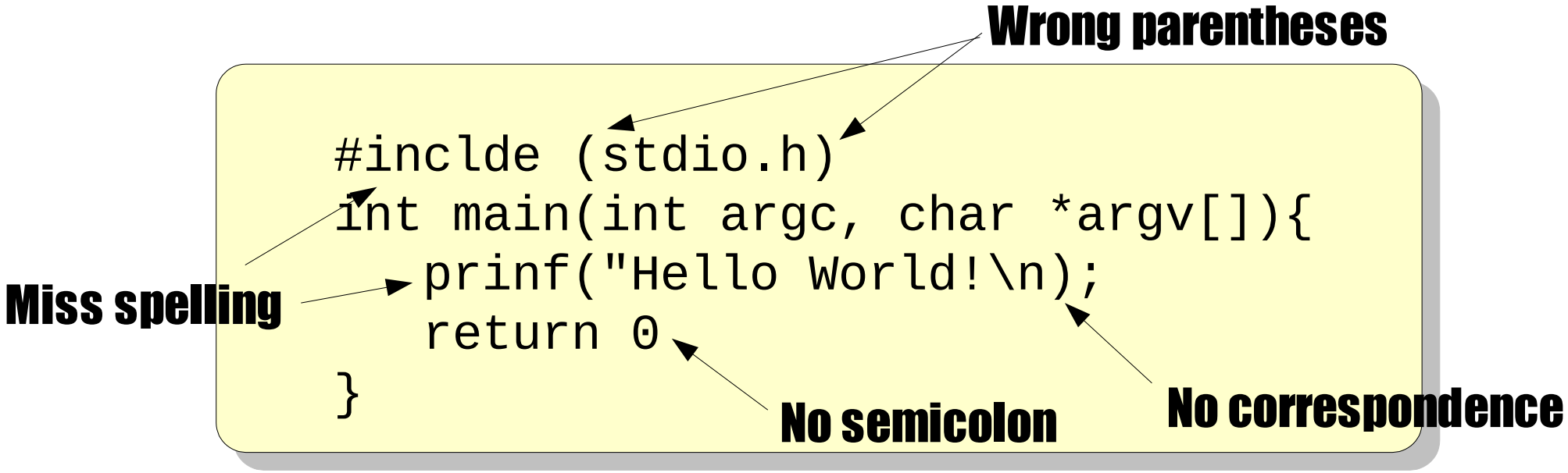
```
1:  #include <stdio.h>
2:  int main(int argc, char *argv[]){
3:     printf("Hello World!\n");
4:     return 0;
5:  }
```

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

Well Known, but Harmful Sample Program

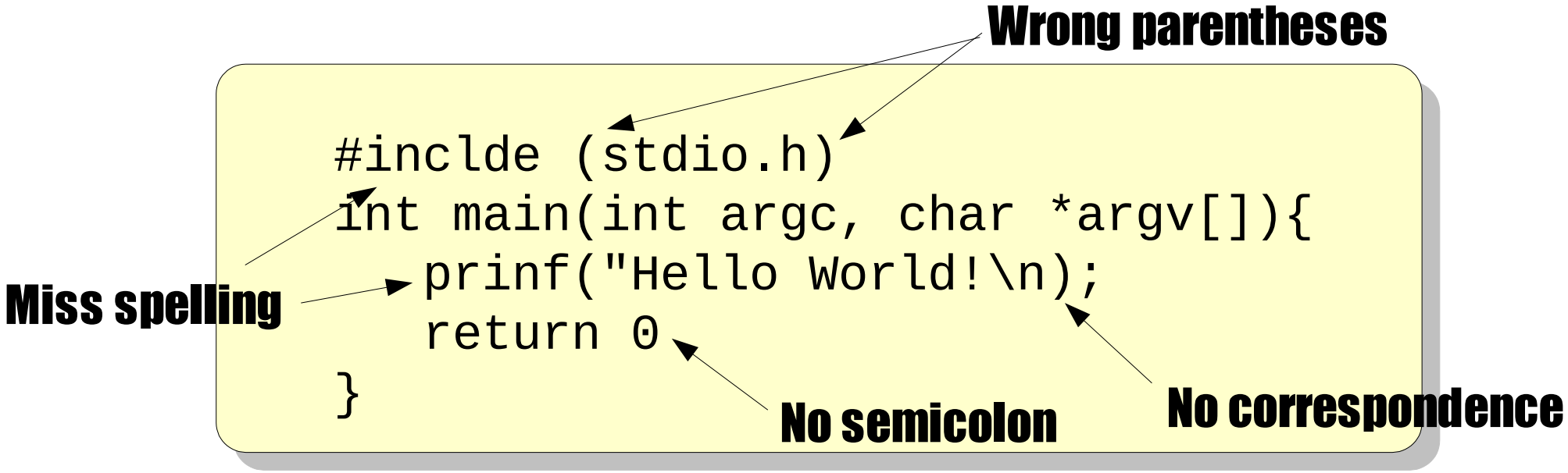
in C language:

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



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 `}'
```



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. Σ π e)
- **Repeat one topic with many exercises** to train the sense of programming.

Contents of the Drill

Exercise 1.
"*Evaluation*" is to calculate (ex: the evaluation of 1+1 is 2).
Evaluate the following statements in C.

statement:	answer:
1+2	3
22*3.3	42.6
2-1	1
10.2+5.1	15.3
10/5	2

Contents of the Drill

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.

statement:	answer:
`0 < 1`	1
`1.0 != 10.0`	1
`-1 >= 3`	0
`3 < -1.5`	0

Contents of the Drill

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

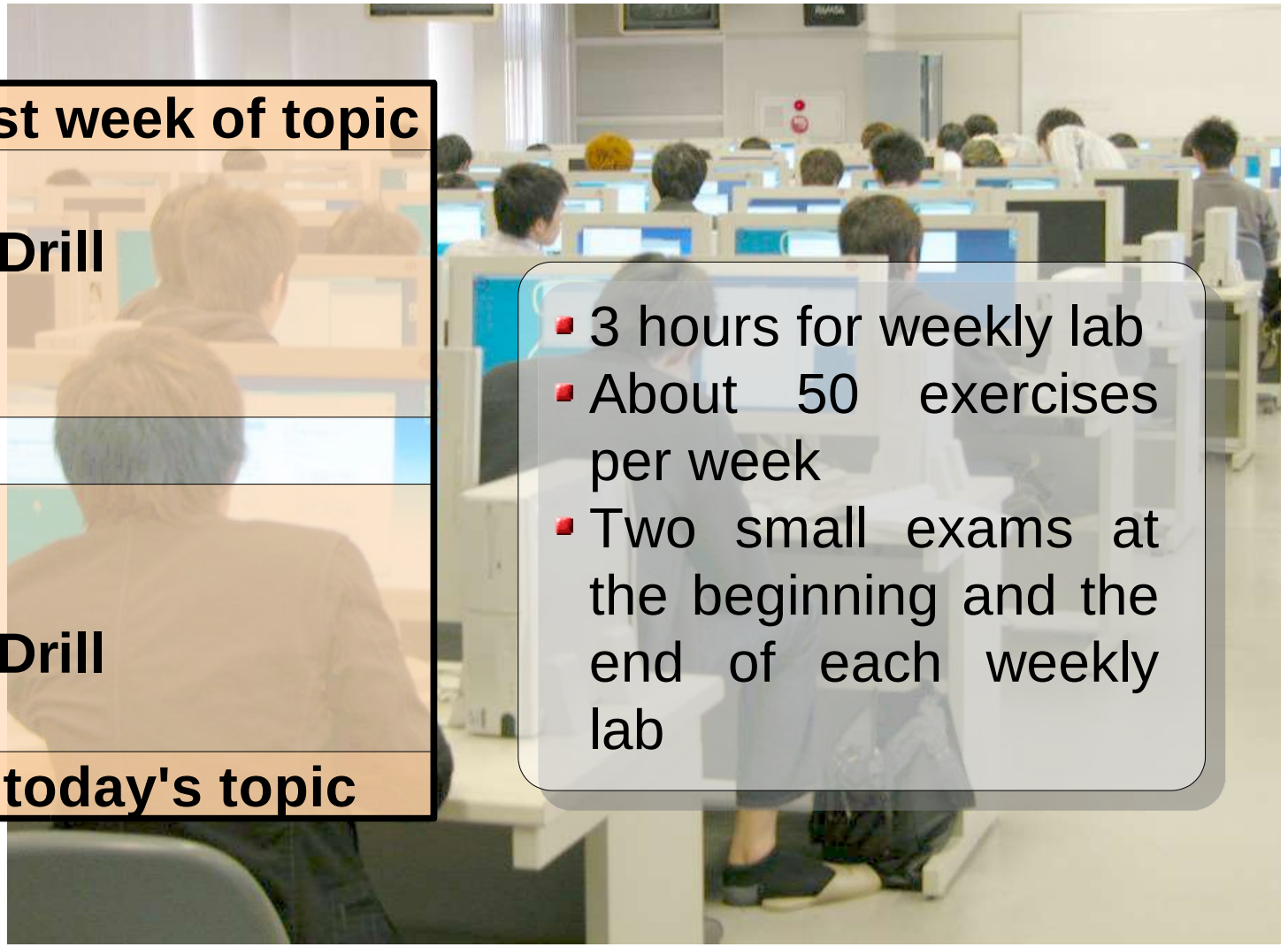
statement:	answer:
`-2+4-3.5`	`1.5`
`3*8/4`	`6`
`10/ 2/5`	`1`
`1.1 + 0.1 < 1.1`	`0`

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



12:50

Exam for last week of topic

Drill

14:20

14:35

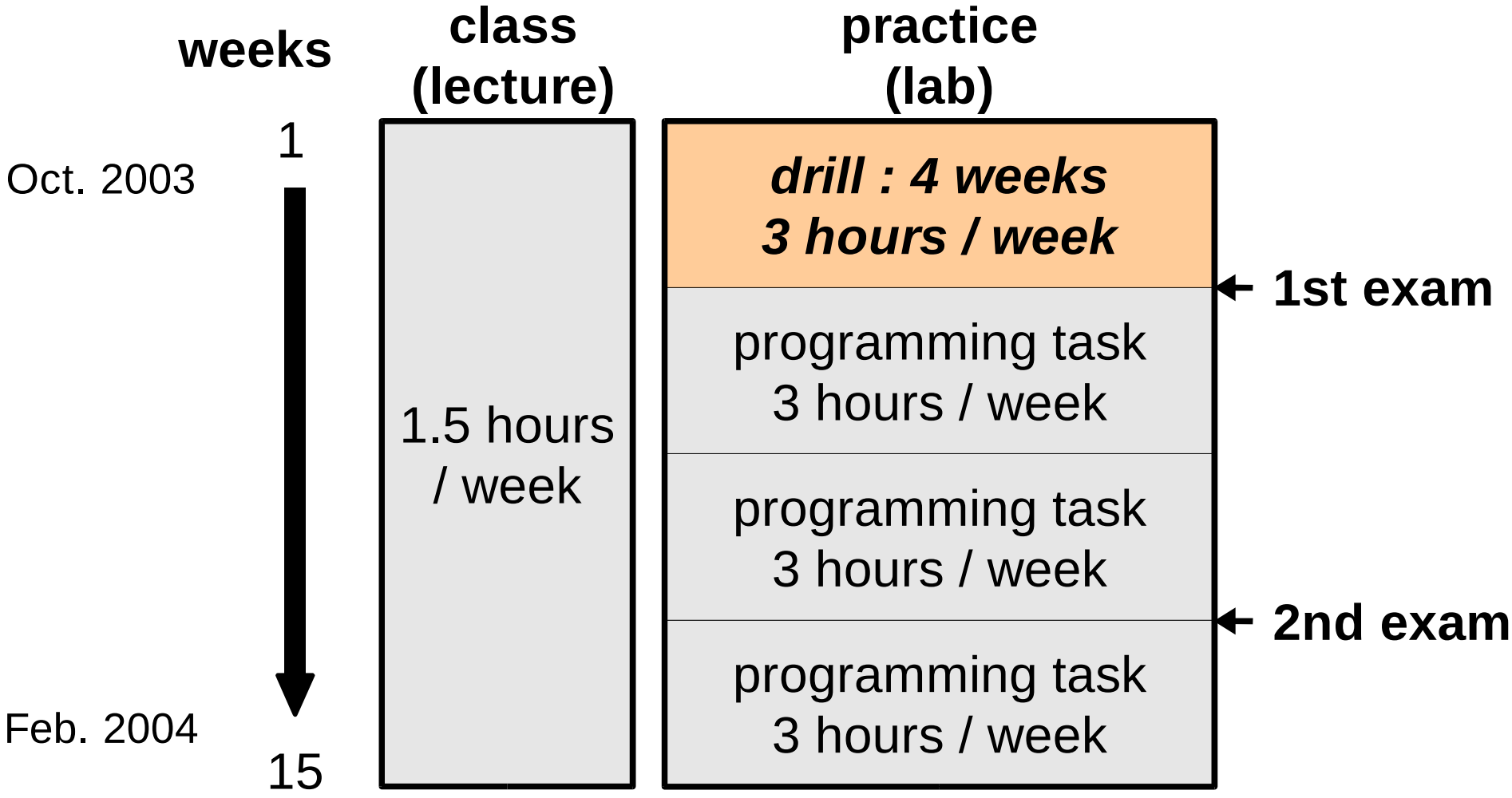
Drill

Exam for today's topic

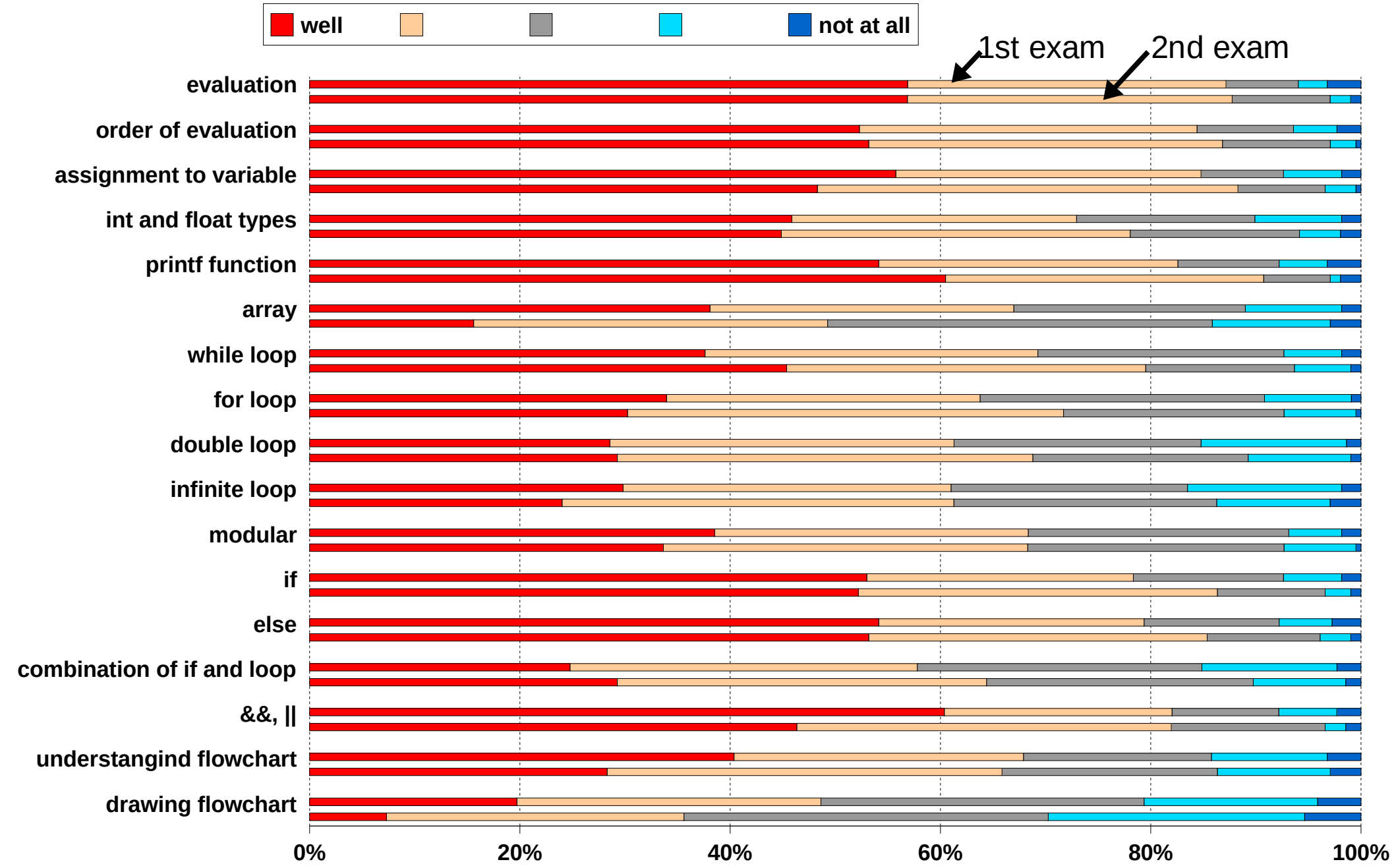
16:05

- 3 hours for weekly lab
- About 50 exercises per week
- Two small exams at the beginning and the end of each weekly lab

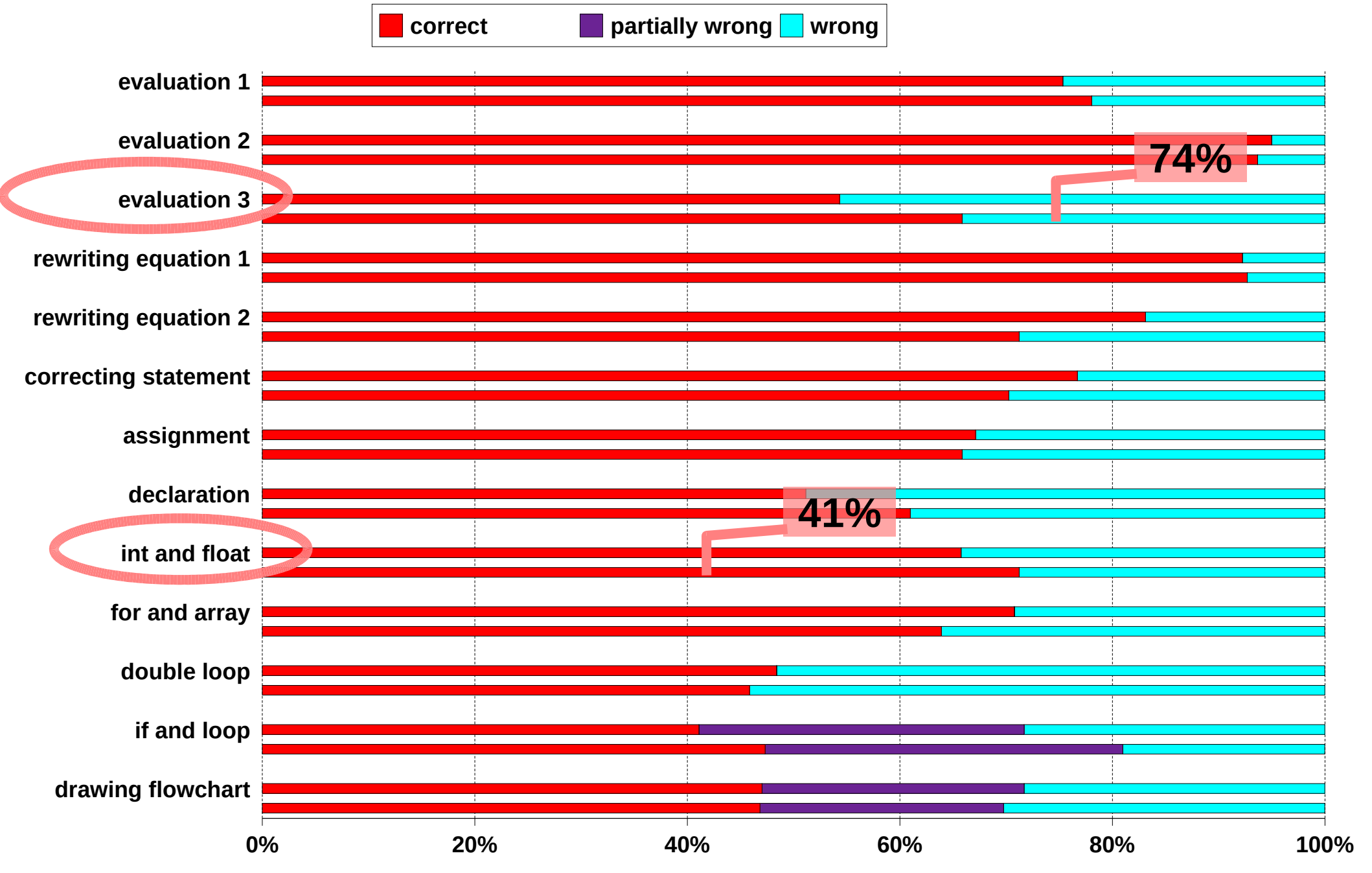
Practical Training with Drill in Our Programming Course



Questionnaires on understanding



Results of short exams



Correlations

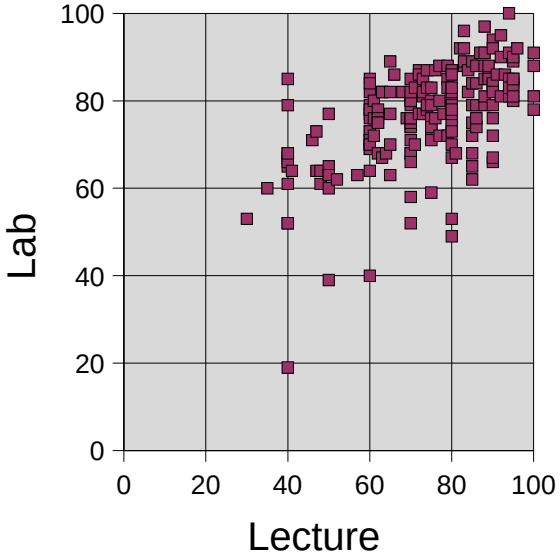
Correlations among grades of lecture, lab, and exams.

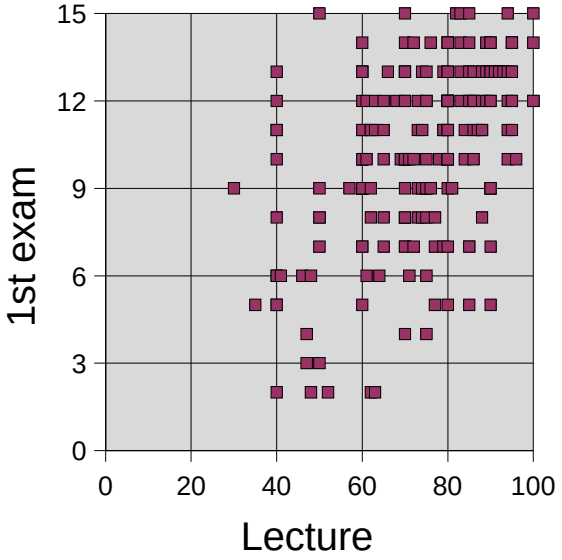
	lecture	lab	1st exam	2nd exam
lecture	1.00	0.59	0.48	0.54
lab	0.59	1.00	0.60	0.66
1st exam	0.48	0.60	1.00	0.69
2nd exam	0.54	0.66	0.69	1.00

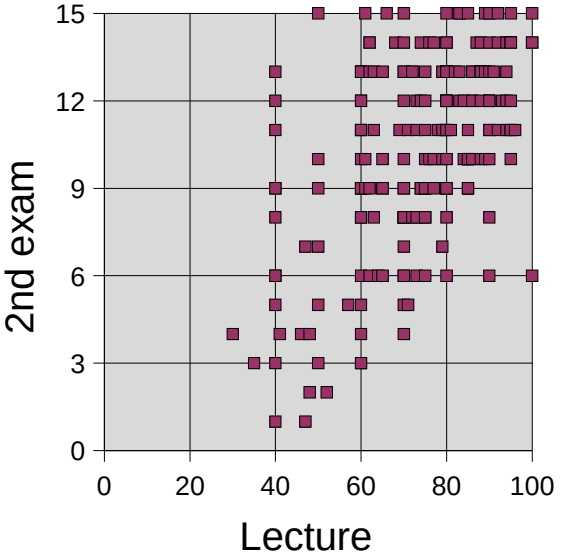
Grades :

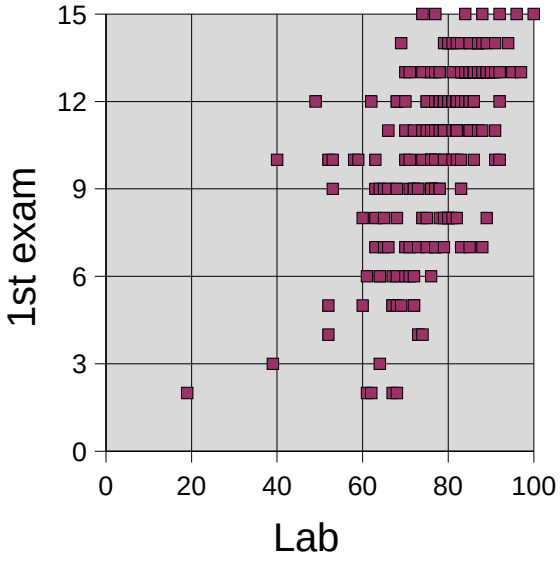
- Lecture and lab are graded independently.
- Lecture : writing examination
- Lab : two exams (5% each) and reports of three tasks (30% each).

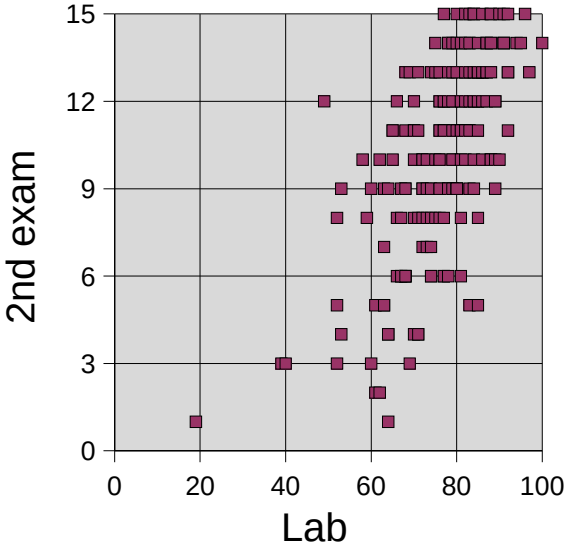
Scattered Diagrams

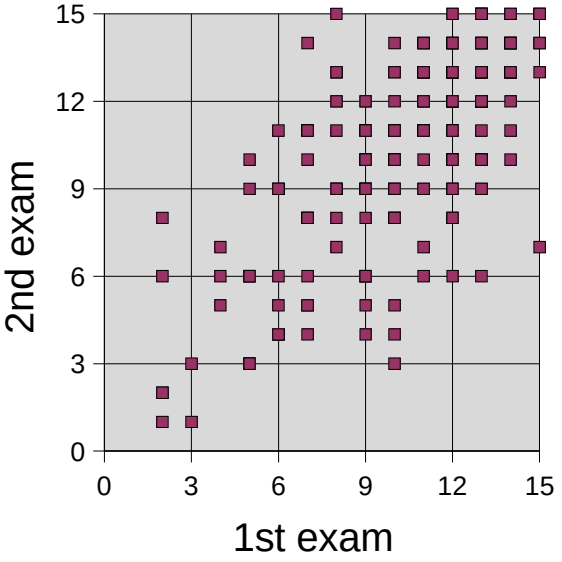












Conclusions

- 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.