Education in Programming
David Gries

Dr. rer. nat., Munich Institute of Technology, 1966

Computer Science, Cornell University, Ithaca, NY

A Glimerick of Hope
David Gries, 1995

Abstract

The world is turning to C,
Though at best it is taught awkwardly,
But we don't have to mope,
There's a glimmer of hope,
In the methods of formality

President Mary of Eire
Was supposed to come to regale ya.
But matters of State
Made her cancel that date
So you're stuck with this guy from Bavaria

Well I got my degree at that place
And my ancestors came from that race
Though Great New York C
Was the Birthplace of me
And CS at Cornell is my base.

So I don't have a real PhD.
It's the Dr. Rer. Nat. that's for me.
And it's from MIT
--On your side of the sea
Munich Inst. of Technolology

2

In 1962–63, in Illinois
writing the
ALCOR-ILLINOIS
7090 ALGOL Compiler

I came to Munich to get a PhD
(and finish the compiler)

Manfred Paul

Rudiger Wiehle

Elaine and David
Gries

Instrumental in the development of
programming languages and their implementation

A early as 1952, Bauer: *Keller* principle
Influential in development of Algol 60

Educate the next generation
of computer scientists

1968 and 1969
NATO Conferences
on
Software Engineering
(Garmisch and Rome)

4

1968 and 1969 NATO Conferences
Software Engineering (Garmisch and Rome)

5

1968 and 1969 NATO Conferences
Software Engineering (Garmisch and Rome)

6

Instrumental in the development of
programming languages and their implementation

Educate the next generation
of computer scientists

1968 and 1969
NATO Conferences
on
Software Engineering
(Garmisch and Rome)

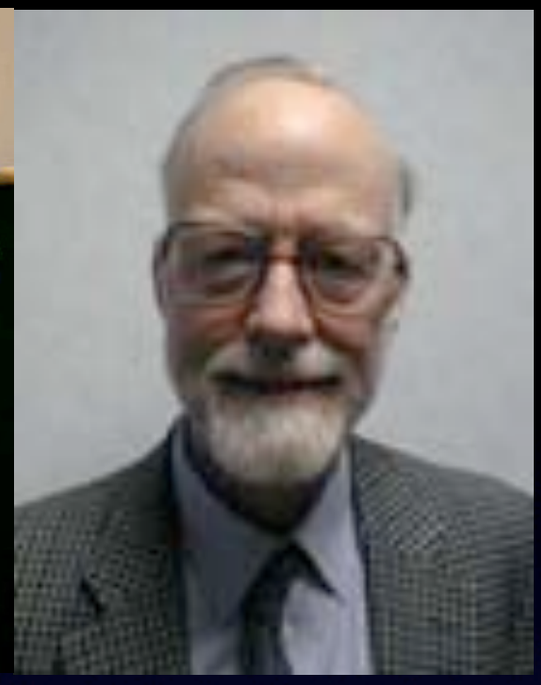
The Marktoberdorf Summer Schools
now led ably by Manfred Broy

7

The teaching of programming

simplicity
elegance
perfection
intellectual honesty

Edsger W. Dijkstra



Sir Tony Hoare

The competent programmer is fully aware of the limited size of his own skull, so he approaches the programming task in full humility, and among other things, he avoids clever tricks like the plague.

Two ways to write a program:
(1) Make it so simple that there are obviously no errors.
(2) Make it so complicated that there are no obvious errors.

8

Teaching programming to beginners, using Java

Intellectual honesty. Do *not* teach C or C++ to beginners, for these languages lack the simplicity and elegance required. In fact, these languages get in the way of programming.

Issue: teach structure/organization (OO) first? Or algorithmic aspects?

Principal: Define things before you use them.

Since almost every line of a Java program has to deal with an object or class, we are *forced* to teach OO ideas first.

Use a programming environment that does not require a method main —that is, a complete Java application.

9

How to teach programming, using Java an objected-oriented language

Principal: Define things before you use them.
Teach OO concepts first.

Order of topics:

- Expressions, variables, assignments, variable decs (1-2 days)
- objects —creating them; function/procedure calls (1 day)
- class/subclass definition, with simple methods that use only function/procedure calls and return statements (1 day)
- fields, getter/setter methods, constructors, etc.

Throughout, groundwork laid for dealing with correctness concerns by requiring the use of informal but precise and thorough specifications of methods and a class invariant, which defines the fields.

10

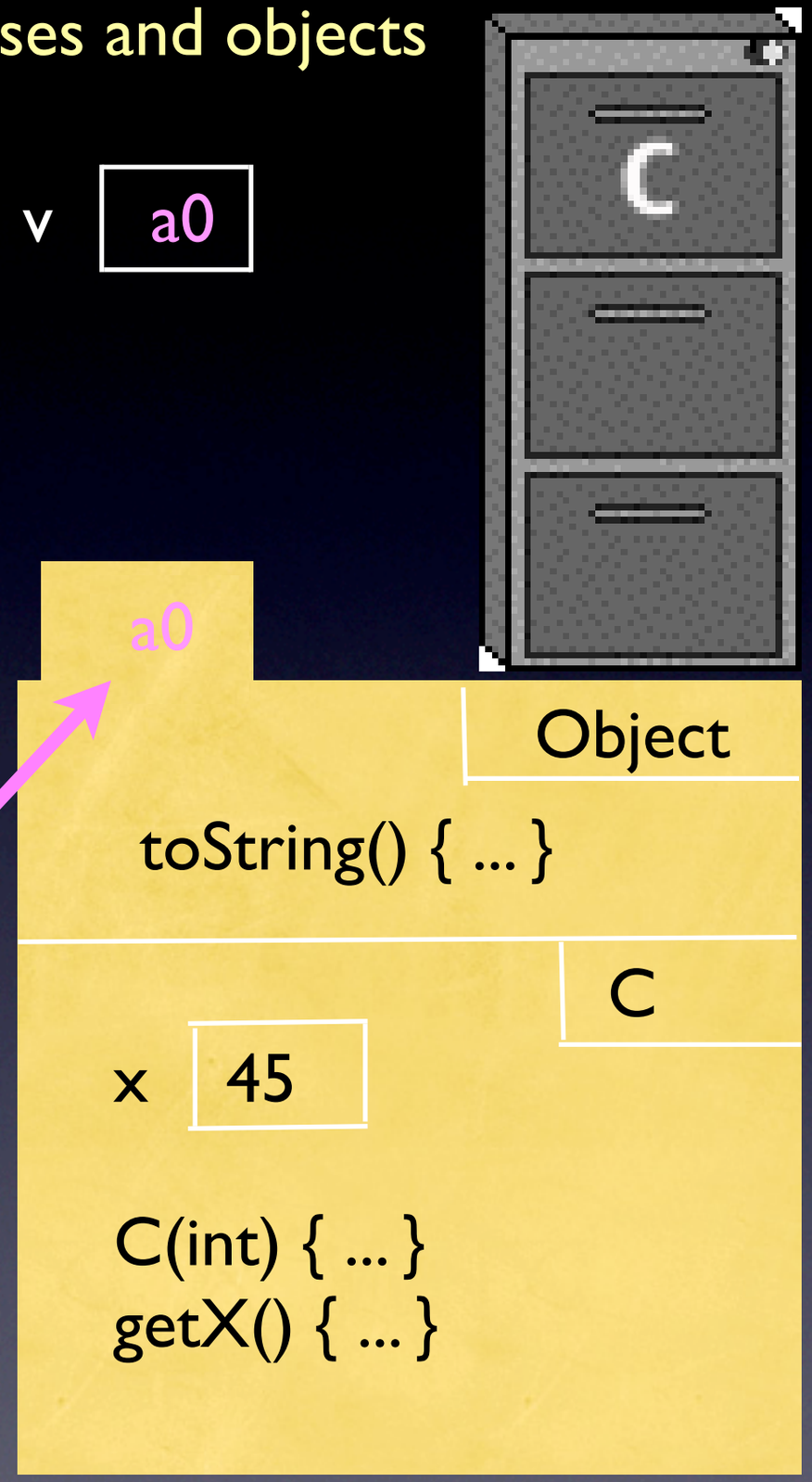
Teaching OO requires a suitable model of classes and objects

```
public class C {
   int x;
   public C(int xx) {
      x= xx;
   }
   public getX() {
      return x;
   }
}
```

Allows one to dispense with terms like *pointer* and *reference*, which students don't understand and which complicate

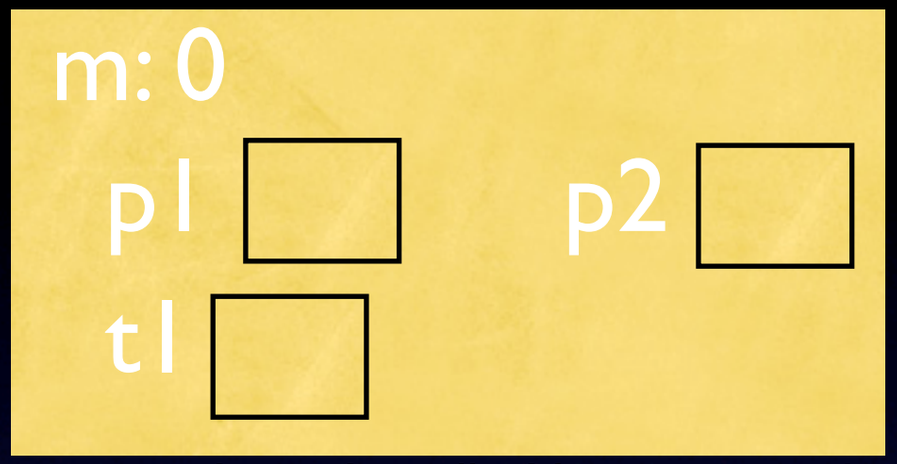
Principal: Introduce and discuss concepts at the appropriate level of abstraction

Principal: introduce names for items to be discussed.



How to teach programming, using Java
an objected-oriented language

Algorithmic aspects
Principals: Emphasize correctness concerns
Give a good model of execution
Teach programming, not programs



frame for call of function m with pars p1, p2 and local var t1

1. Deeper investigation into procedures/functions
2. If-statements, local variables
3. Model of execution, including a frame for a call
4. Stepwise refinement
5. Recursion. Emphasize difference between the model of execution and understanding a recursive function (proof of correctness)
6. Loops: taught in terms of loop invariants, right from the beginning

Pedagogy

How to teach programming, using Java an objected-oriented language

1. On first assignments, require *mastery*. Everyone gets 100/100
   - No proper specs on methods? Fix/resubmit
   - No suitable class invariant? Fix/resubmit
   - Test cases not satisfactory? Fix/resubmit
   - Bug? Find it, fix/resubmit

Allows beginning students to make mistakes, usually based on misunderstanding, without harming their grade.

Allows students who think they already know how to program to overcome bad habits.

2. Give each student at least one 1/2 hour one-on-one session with instructor, TA, or senior-level consultant.

The only way to teach good programming practices and give the students an idea of program development.

13

On this 40th anniversary celebration

Thanks to those affiliated with the dept in some way who helped shape my life and helped me contribute in a small way






J.Stoer R.Bayer W.Niegel M.Broy
W.Brauer S.Braun R.Bulirsh P.Duessen J.Eickel G.Goos
U.Hill P. Kandzia H.Langmaack H.Partsch F.Peischl
P.Pepper C.Reinsch G.Schmidt G.Seegmuller T.Strohlein
M.Wirsing C.Zenger

14