Informatics in Schools? – Urgently Needed!

Walter Gander

ETH Zurich and BU Hong Kong

ECSS 2012

20 – 21 November 2012, Barcelona

Informatics shapes the future!

A Quiz for Software Engineers

- George Forsythe (1968): In the past 15 years many numerical analysts have progressed from being queer people in mathematics departments to being queer people in computer science departments! [a]
- Matlab: smallest positive normalized machine number
  `realmin = 2.225073858507201e-308`
  smallest positive non-normalized machine number
  `x=realmin*eps = 4.940656458412465e-324, x/2=0`
- Exercise: Compute the smallest positive normalized machine number.

[a] What to do till the computer scientist comes. Amer. Math. Monthly 75 (1968), 454-462.

Program 1	Program 2

`function a=test1`
`a=1;`
`while a*eps>0`
`  last=a; a=a/2.0;`
`end;`
`a=last;`
	`function a=test2`
	`a=1;`
	`while a*eps>0`
	`  last=a; a=a/2.0;`
	`  fprintf('%20.15e\n',a)`
	`end;`
	`a=last;`

`>> test1` `ans =  4.940656458412465e-324`	`>> test2` `ans =  2.225073858507201e-308`

Printing changes the result?

George Forsythe wrote 1963:

Machine-held strings of binary digits can simulate a great many kinds of things, of which numbers are just one kind. For example, they can simulate automobiles on a freeway, chess pieces, electrons in a box, musical notes, Russian words, patterns on a paper, human cells, colors, electrical circuits, and so on. To think of a computer as made up essentially of numbers is simply a carryover from the successful use of mathematical analysis in studying models . . . Enough is known already of the diverse applications of computing for us to recognize the birth of a coherent body of technique, which I call computer science. [a]

[a] Educational implications of the computer revolution. Applications of Digital Computers, W. F. Freiberger and William Prager (eds.), Ginn, Boston, 1963, pp. 166-178.

Computer Development by Decades

- 1950–1960: Hardware Development
- 1960–1970: First Higher Programming Languages, Numerical Computing, (G. Forsythe)
- 1970–1980: Mainframes, Data Processing
- 1980–1990: Microprocessors, Personal Computer
- 1990–2000: Network and Communications, WWW
- 2000–today: Ubiquitous Computing [a]

[a] `http://science.discovery.com`

Computers Determine Our Life

Communication: e-mail, cell-phone, sms, social networks: facebook, twitter, LinkedIn ...

Writing: text-processing, spreadsheets, presentation tools

Reading: Google eBooks, e-Reader: Kindle, iPad, Sony Reader, Digital Book Index provides links to more than 165,000 full-text digital books `http://www.digitalbookindex.com/about.htm`

Music: iTune, e-music, MP3

Radio and Television: digital, Internet

Photography: software has replaced chemically processed films

Search for Information: libraries, archives available on-line, Wikipedia

many more examples ...

However– Schools are Sleeping!

Teaching K-12 Computer Science in the Digital Age Fails! [a]

- Computer science and the technologies it enables now lie at the heart of our economy, our daily lives, and scientific enterprise.
- The digital age has transformed the world and workforce, but education has fallen woefully behind in preparing students with the fundamental CS knowledge and skills they need for future success.
- To be a well-educated citizen as we move toward an ever-more computing-intensive world and to be prepared for the jobs of the 21st Century, students must have a deeper understanding of the fundamentals of computer science.

---

[a] ACM and CSTA released the startling findings of their computer science education standards report, 2010. Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age `http://www.acm.org/runningonempty/`

Same Failure in Other Countries

- Germany: GI [a] and BITKOM [b] observe
  - CS as an interdisciplinary technology is of special importance because it advances innovation in many other disciplines
  - We demand therefore in the curriculum of the schools CS for all students as an independent subject.
  - In high school the subjects biology, chemistry, computer science and physics have to be offered equivalently.
- Switzerland: since 2008 CS is an elective in Gymnasium

  ICTswitzerland [c] and SVIA [d] push for a mandatory subject equivalent to mathematics, physics or chemistry.

[a] Gesellschaft für Informatik
[b] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien
[c] umbrella organization of the computer science and telecommunication sector
[d] Computer Science Teacher Association

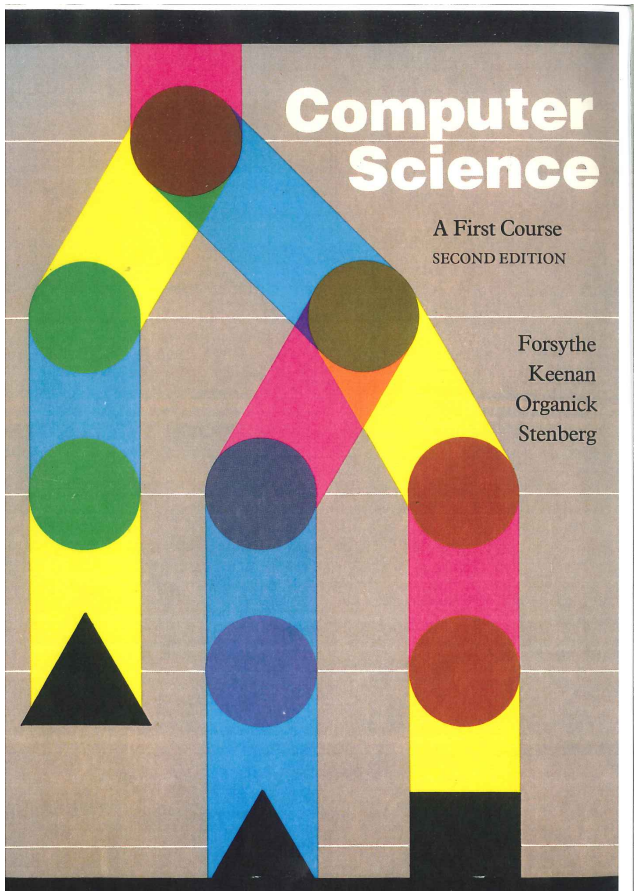
Why is CS still Missing in School Curriculum?

- mistery, not rational
- George Forsythe 1961:
  In spite of the diversity of the applications, the methods of attacking the difficult problems with computers show a great unity, and the name of Computer Science is being attached to the discipline as it emerges. [a]
- Sandra Forsythe et al. published already 1969 a textbook for schools!
  Computer Science: A First Course

[a] Engineering students must learn both computing and mathematics. J. Eng. Educ. 52 (1961), 177-188.

A Textbook of 1969/1975!

more than 40 years ago!!



Contents

How is Computer Science Perceived Today?

- Physics: natural science that involves the study of matter and its motion through space and time, along with related concepts such as energy and force.

  Curriculum in schools is well defined. Teaching fundamentals of physics as necessary part of general education is generally accepted.

  ==========

- Computer Science: blurred term in our society today. Many different opinions, definitions, concepts and interpretations.

  Often synonymously used:
  Computer Science, Informatics, ICT, Digital Literacy, New Literacies, Electracy, New Media, Media Education, ...
  Curriculum in schools is very divers. If CS is taught at all, no agreement what material to teach for general education.

Digital Literacy and Informatics

The *Informatics Europe* and *ACM Europe* Working Group proposes the definitions:

Digital Literacy: the ability to work with a computer and to be familiar with some basic applications.

Short living knowledge, important skills, technology driven

Informatics: the fundamentals of computer science including programming.

Long-lasting basic knowledge, permanent principles

Both subjects must be part of general education!

Digital Literacy (Start in first grade, familiar at age 12)

- Learning to type
- Learning to write texts using a simple word processor
- Understand nature/size of digital files for text, audio, photos, movies
- Learning to use the currently most suitable software for these activities
- Learning to well organize information by storing files in directories (copy, delete, rename) and to make backups
- Learning to present information
- Learning to search, copy and store information as digital files
- Learning to communicate via various media
- Learning to behave ethically on the web and be aware of security, fraud issues, cyber-mobbing
- Start with suitable programming environments for children

Proposed Definition of Informatics [a]

Informatics is the science of systematic, automated processing of information (representation, storage, management and communication).

- As basic science informatics studies the limitations of automation, the theory of information processing and ways to solve difficult problems efficiently (also by non-determinism and randomization).
- As engineering discipline informatics is the tool to control complexity: to design, build and operate complex systems.
- As structural science informatics supports interdisciplinary research in many fields through modeling and simulation of natural, technical, social and intellectual processes.

[a] by Juraj Hromkovic and WG

Why is Informatics Necessary for General Education?

George Forsythe 1968: *The most valuable acquisitions in a scientific or technical education are the general-purpose mental tools which remain serviceable for a lifetime. I rate natural language and mathematics as the most important of these tools, and computer science as a third ...* [a]

The question "What can be automated?" is one of the most inspiring philosophical and practical questions of contemporary civilization. [b]

Informatics has become one of leading science of the 21 century.
It influences our life and changes everything including science and education.

---

[a] What to do till the computer scientist comes. Amer. Math. Monthly 75 (1968), 454-462.

[b] Computer science and education. Proc. IFIP 68 Cong., 92-106.

Example: Change of Mathematics

- Nicholas Negroponte, MIT Media Lab[a]:
  A surgeon of 1900 would not recognize anything in todays operating room. A mathematics teacher of 1900 in todays classroom would just continue teaching the same way.

- George Forsythe: *Compared with most undergraduate subjects, mathematics courses are very easy to prepare for, because they change so slowly. The computing part of it is probably the only part that changes much. Why not devote time to learning that?* [b]

---

[a] GEP Vorlesung ETH 1992

[b] Computer science and mathematics. S1GCSE Bull. 2, 4 (Sept.-Oct. 1970), 20-23.

Change Teaching Mathematics

- Conrad Wolfram: Use Computers to modernize math
- Why teach math? Logical mind training.
- What is math?
  1. Posing the right question
  2. Real World $\rightarrow$ Math formulation
  3. Computation (no longer by hand, use computers!)
  4. Math formulation $\rightarrow$ real world, verification

Applications versus Fundamentals

- Barbara Demo: A paradox: we were better when we were worse.
  - Around 1985 no software aids to mathematics teaching available. Informatics meant introduction to programming, usually with Pascal
  - Over the years with tools like "Derive" or "Cabri" or "Octave" the need of even a limited capacity of programming in a general-purpose language disappeared. Informatics reduced to the use of specialized tools.
  - Widespread offering in schools of the ECDL strongly contributed to the vision of computer science as the usage of software tools.
- David Braben explains BBC's Rory Cellan-Jones the Raspberry PI scheme designed to give a £15 computer on a stick to every child.

  ICT↔CS:David Braben Might generate similar fascination for the youth as Amiga and Commodore 64 did in 1985

Informatics in School

Goals:

- Understand the principles and functioning of todays digital world
- Train the students in constructive problem solving

Topics

- Programming and Computational Thinking
- Algorithms and Data Structures
- Data (Representation, Management and Security)
- Computing Complexity, Limits of Automation
- Networks and Communication
- Simulation and Visualization

Programming — the Fundamental of Informatics!

George Forsythe 1959: *The automatic computer really forces that precision of thinking which is alleged to be a product of any study of mathematics.*[a]

George Forsythe 1966: *The major thing which distinguishes computer science from other disciplines is its emphasis on algorithms.*

There are few problems for which a good algorithm of probable permanent value is known... Small details are of the greatest importance... The development of excellent algorithms requires a long time, from discovery of a basic idea to the perfection of the method... A useful algorithm is a substantial contribution to knowledge. Its publication constitutes an important piece of scholarship. [b]

[a] The role of numerical analysis in an undergraduate program. Amer. Math. Monthly 66 (1959), 651-662.

[b] Algorithms for scientific computation. CACM 9 (Apr. 1966), 255-256.

Teaching a Machine

If you want to learn something, teach it. You are successful if people understand. They may say they understand even if they don't. The ultimate test if you are doing well is to teach it to a machine!

Don Knuth, Jan 14, 2012, ETH
Swiss Olympiad in Informatics

Programming a machine is part of our culture

(Berufsbildungskonferenz Nov 9, 2012, Bern)

Mauro Dell'Ambrogio, State Secretary for Education and Research, Switzerland

Computational Thinking

- Definition by Jan Cuny, Larry Snyder, and Jeannette M. Wing, Carnegie Mellon University, USA:

  Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.

- Also supported by Google:

  Computational thinking involves a set of problem-solving skills and techniques that software engineers use to write programs

Problems Solving with Computers

- analyze a task or problem, model and formalize it
- search for a way to solve it, find or design an algorithm
- program
- run the program: let the computer work, maybe correct, modify the program, interpret the results

Why is Programming IMPORTANT for General Education?

It is a

- creative and
- constructive

work like an engineer!

activity. It

- teaches precise working and
- trains computational thinking

Good Example: Slovakia

- Subject Informatics = {Digital Literacy, Informatics} interwoven
- compulsory (!) from primary school (from 2nd grade) to lower and upper secondary school
- in primary school emphasis more on digital literacy,
  in upper secondary school more on informatics
- Programming: lower secondary school LOGO, upper secondary school usually Pascal
- Main topics:
  - Digital World
  - Programming and Algorithms
  - Computer Systems
  - Informatics and Society

Switzerland the contrary of Slovakia!

- all 26 cantons have their own school system
- most of them in grades 1 to 9 neither digial literacy nor informatics in their curriculum.

  Often voluntary efforts to educate the students in digital literacy.
- Education in grades 10 to 12 is split:
  - Vocational schools (with apprenticeship): some have serious informatics
  - Gymnasium: since 2008 optional informatics as elective
  - Business schools: mandatory digital literacy, some teach a programming language
  - Pedagogical Universities, responsible for teachers education: no digial literacy or informatics

Hong Kong P1–P6: primary school, F1-F6: secundary school

- Digital Literacy and Informatics not strictly separated (Computer Literacy)

	Computer Literacy	**Informatics**
P1-P6	optional 1-2 lessons per week	there is LOGO programming
F1-F3	optional 1-2 lessons per week	(school-based) there is Visual Basic programming, Pascal, Raptor (flowcharting)
F4-F6		software development option: one of 4 languages : Visual Basic, Java, C or Pascal database option: SQL multimedia production & web site development: script language

- Recommendation:
  ICT Curriculum and Assessment Guide (Secondary 4–6)

Austria

Digital Literacy

- primary and secundary school I: a few have optional digital literacy
- secundary schools II: mandatory digital literacy
  optional in Gymnasium

Informatics

- primary schools and secundary schools I: no informatics
- secundary schools II: Informatics-related HTL have full program of mandatory Informatics during 5 years.

  Gymnasium Oberstufe: mandatory for one year

Some Good Developments . . .

Progress in USA: CS Education Act

`www.acm.org/press-room/news-releases/2010/cs-ed-act`

Landmark progress July 30, 2010: congressional representatives from both political parties introduced legislation to strengthen CS education

- Defines CS education and its concepts to clarify the confusion of terms around K–12 CS education
- Establishes planing grants for 5 years implementation to develop CS standards curriculum, teachers certification programs and on-line courses
- Create blue-ribbon commission to review state of CS education and to address CS teacher certification crisis
- Establishes K–12 teacher preparation programs at institutions of higher education

Programming in Primary School, Pilot Project in Switzerland

Oberkulm: Start zum dreijährigen Pilotprojekt Programmieren an der Primarschule Oberkulm

Programmieren – Ein Kinderspiel?

Die Primarschule Oberkulm führt als erste Pilotschule im Kanton Aargau ein dreijähriges Informatikprojekt zum Thema Programmieren durch. Die Klassen im vierten und fünften Schuljahr, bei den Klassenlehrpersonen Beate Welsche und Ruth Trüb, starteten als erste ins Pilotprojekt.

Keine Zeit für Pause: *Nachwuchs-Informatiker tüfteln gemeinsam an Lösungen.*

moha. Das Projekt steht unter der Leitung von ETH-Professor Juraj Hromkovic, und einem Team von Studenten und Doktoranden, welche die Schüler der beiden Mittelstufenklassen instruieren. Unterstützt wird das Vorhaben von der Hasler Stiftung in Bern. Auf die Altersstufe der Schüler ausgerichtet wurden spezielle Lehrmittel ausgearbeitet. Die Beratungsstelle imedias an der Pädagogischen Hochschule FHNW begleitet das Projekt, beobachtet dessen Fortgang und wertet es aus.

Das Pilotprojekt in Oberkulm startete im [illegible] mit einer Weiterbil-

vier Studenten begleitet, welche abwechslungsweise die Assistenz übernahmen.

Intensiv und herausfordernd

[illegible]

sie kaum Zeit fanden, eine Pause zu machen. Für das Pilotprojekt wurden der Schule Oberkulm die nötige Anzahl Laptops zur Verfügung gestellt, oder die Kinder durften ihre eigenen

[illegible]

Computer programming will soon reach all Estonian schoolchildren

by **Richard Wilson** on September 4, 2012 in **News**

Estonian Tiger Leap Foundation in September 2012 launched a program called "ProgeTiiger", in the framework of which Estonian students in grades 1 to 12 will be introduced computer programming and creating web and mobile applications.

"The interest of students towards using modern technologies has grown year after year. With the "ProgeTiiger" program we create prerequisites for students to develop from consumers of software to developers of software," said Tiger Leap Foundation training sphere manager Ave Lauringson.

Estonia

Programming on Smart Phones

(Jarka Arnold and Aegidius Plüss, Pedagogical University Bern)

JTurtleLib

Java Turtle Graphics for Android

A Java Package
To Learn Programming Using Android Smartphones/Tablets

Back To The Roots: Turtle Graphics

There is no need to explain what Turtle Graphics is nor to emphasize its importance in computer programming methodology. Since the invention of Logo in 1967, turtle libraries became available for many

Why is programming fun? What delights may its practioner expect as his reward?

Fred Brooks: Mythical Man Month (1974)

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office".

Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning. The programmed computer has all the fascination of the pinball machine or the jukebox mechanism, carried to the ultimate.

Fourth is the joy of always learning, which springs from the nonrepeating nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures.