Universität Karlsruhe (TH)
Research University · founded 1825

IMI Information Management in Engineering

Computer Science for Engineers

Lecture 1

Course offerings in IMI, Organisational issues, Introduction

Prof. Dr. Dr.-Ing. Jivka Ovtcharova

Dipl. Wi.-Ing. Dan Gutu

23rd of October 2008

Universität Karlsruhe (TH)
Research University · founded 1825

IMI Information Management in Engineering

Computer Science for Engineers

Lecture 1

Course offerings in IMI, Organisational issues, Introduction

Prof. Dr. Dr.-Ing. Jivka Ovtcharova

Dipl. Wi.-Ing. Dan Gutu

23rd of October 2008

Where to find us:

Institute for Information Management in Engineering
Universität Karlsruhe (TH)

AVG Süd 1.OG.
Adenauerring 20, Geb. 50.41
D-76131 Karlsruhe

Prof. Dr. Dr.-Ing. Jivka Ovtcharova

Tel.: +49 (0) 721 608-2129

Fax: +49 (0) 721 661138

Email: jivka.ovtcharova@imi.uni-karlsruhe.de

Room 103

Dipl. Wi.-Ing. Dan Gutu

Tel: +49 (0) 721 608-6621

Fax: +49 (0) 721 661138

Email: dan.gutu@imi.uni-karlsruhe.de

Room 001

Course offering at IMI (1)

Main cources

• Computer Science for Engineers (CSE)	Prof. Dr. Dr.-Ing. J. Ovtcharova
• Virtual Engineering (VE) I+II	Prof. Dr. Dr.-Ing. J. Ovtcharova
• Product Lifecycle Management (PLM)	Prof. Dr. Dr.-Ing. J. Ovtcharova
• Simulation in the product development proces (SiPEP)	Prof. Dr. Dr.-Ing. J. Ovtcharova Prof. Dr.-Ing. A. Albers Prof. Dr.-Ing. T. Böhlke

Industry courses

• Computer Integrated Planning of New Products (RPP)	Dr.-Ing. R. Kläger, GKD
• Integration of Products, Processes and Resources in the Development of Automobiles (PPR-Integration)	Dipl.-Ing. S. Mbang, Daimler AG
• Virtual Engineering for Mechatronic Products (VEmP)	Dr.-Ing. habil. S. Rude, BMW AG
• PLM in the Manufacturing Industry (PLM-F)	Dipl.-Wirtsch.-Ing. G. Meier, Heidelberger Druckmaschinen AG

Course offering at IMI (2)

Lectures from the International Department

Information Systems I Understanding and readiness of the implementation of the Product Lifecycle Management (PLM) approach	Prof. Dr. Dr.-Ing. J. Ovtcharova
Information Systems II Knowledge in the field of Computer Aided (CAx) approach and Virtual Engineering (VE)	Prof. Dr. Dr.-Ing. J. Ovtcharova
Information Systems III Knowledge in Interface, Data and Information Technologies	Prof. Dr. Dr.-Ing. J. Ovtcharova
Methods of Simulation Knowledge in the field of Methods of Simulation in Product Development Process (PDP)	Prof. Dr. Dr.-Ing. J. Ovtcharova Prof. Dr.-Ing. A. Albers Prof. Dr.-Ing. T. Böhlke

Universität Karlsruhe (TH)
Research University · founded 1825

IMI Information Management in Engineering

Computer Science for Engineers

Lecture 1

Course offerings in IMI, Organisational issues, Introduction

Prof. Dr. Dr.-Ing. Jivka Ovtcharova

Dipl. Wi.-Ing. Dan Gutu

23rd of October 2008

- **Lectures and Exercise slides** can be found on the homepage:

  http://www.imi.uni-karlsruhe.de/280.php
- **Exercise scripts** are also available on the homepage (available chapter by chapter). *Note:* Slides may be updated after the lecture has been given!
- **Lecture:**
  - Thursday, 14:00 – 15:30 hours, SR 203, ID
- **Exercise:**
  - Thursday, 15:45 - 17:15 hours, SR 203, ID
  - Topics: Java, Data Structures, Algorithms, Development Tools (Eclipse)
- **Computer lab (in Summer Semester 2009!):**
  - See Website of the lecture
  - Certificate of completion of the computer lab is required for admission to the exam

- **When: xx.yy.2009, three hours (summer semester!)**

- **Relevant** for the exam preparation are
  - Lectures (slides)
  - Computer lab (mandatory)

- **Point assignment in the exam:**
  - approx. 100 points
  - approx. 50 points sufficient to pass
  - approx. 50% practical assignments (programming)

- No additional resources permitted

Computing lab – General Information

- **When:** Will be announced on the hompeage
- Registration over the website

- The computer lab begins in the 3rd week of lectures in the summer semester.
- The tasks will be published on the homepage of the lecture.
- The submission of all computer exercises (except sheet zero) is necessary for the lab certificate.

Computing lab – Organisation (1)

- There will be one or two exercise groups
- Each group will be supervised by a tutor; computer lab exercises should be handed in to this tutor only.
- The exercise sheets will be given out every 2 weeks.
- In the first week relevant information about the computer exercise sheet will be repeated, followed by a short programming exercise.
- Thereafter the exercise sheet can be worked on with help from the tutor.
- There is no compulsory attendance of the computer lab

Computing lab – Organisation (2)

- In the second week the solution (the working program) should be shown to the tutor. The tutor will then ask questions about the running of the program and topics covered by the exercise.
- If the solution is not correct, you have until the following class to work on it.
- Students who have not turned in the solution in a timely manner must send the tutor the source code **at least 4 days before the next class**. The tutor will read it, and ask questions about it at the next class.
- **The solution can only be handed in late once!**
- If the code is found to be copied (i.e. the questions can not be answered), the student will receive a warning.
- After **2 warnings** a re-examination of the student‘s progress in the lab will be given by the exercise instructor, which could lead to dismissal from the lab.

Computing lab – Organisation (3)

- Between the third and the fifth exercise sheet, an examination will be done by the teaching assistent
- You will be informed about it when the respective exercise sheet will be published on the website
- This test will take place during the computer lab course.

Exercise Hand-In (1)

- Hand-in:
  - Student demonstrates the program to the tutor
  - Tutor asks questions about how the program works, and other relevant information
- Hand-in date:
  - The exercise should be handed in to the tutor by the end of the second week.
  - If this is not possible (i.e. the exercise is not finished), then the source code must be sent to the tutor **at least 4 days before** the next class date.
  - The tutor will then ask questions about the handed in solution at the next lab class.

Exercise Hand-In (2)

- Corrections
  - The exercise must be corrected and handed in again if the solution is found to be wrong.
  - You may only submit two corrected exercises.
  - If it is necessary to correct another exercise, the student will fail the lab.
- If it turns out the handed in code was copied (i.e. questions can not be answered), the student will receive a warning.
- After 2 warnings the student will be re-examined by the exercise supervisor; if the student fails this they will be dismissed from the lab.

Universität Karlsruhe (TH)
Research University · founded 1825

IMI Information Management in Engineering

Computer Science for Engineers

Lecture 1

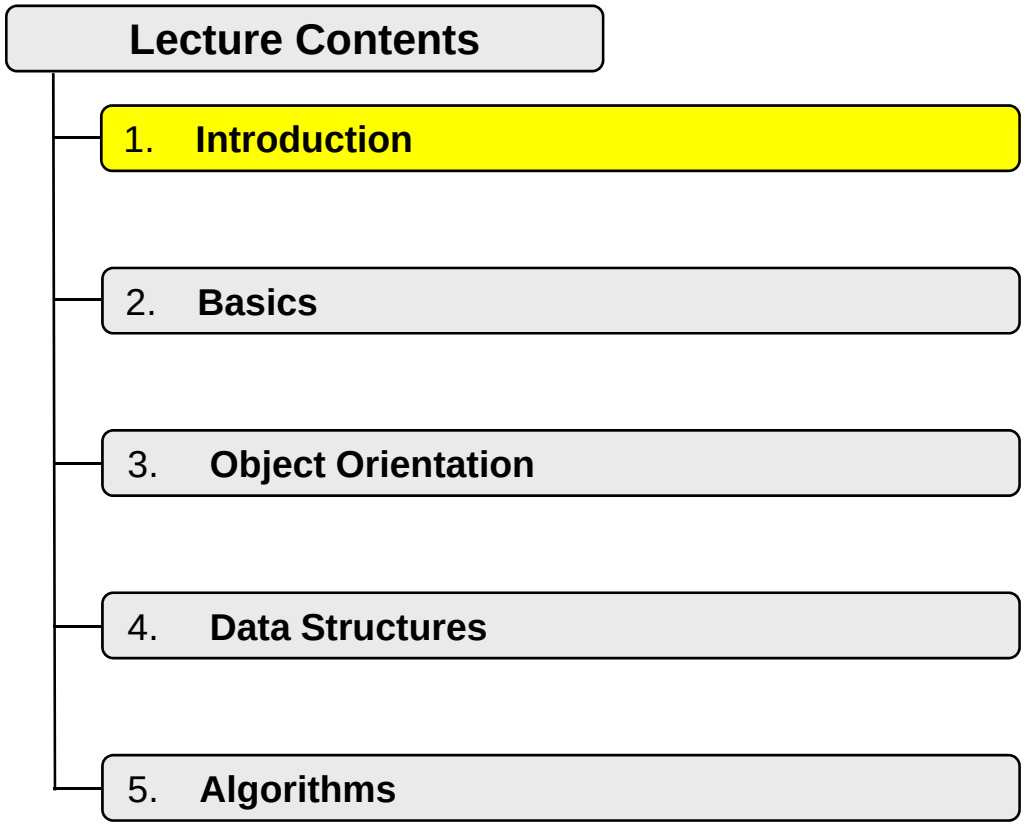
Course offerings in IMI, Organisational issues, Introduction

Prof. Dr. Dr.-Ing. Jivka Ovtcharova

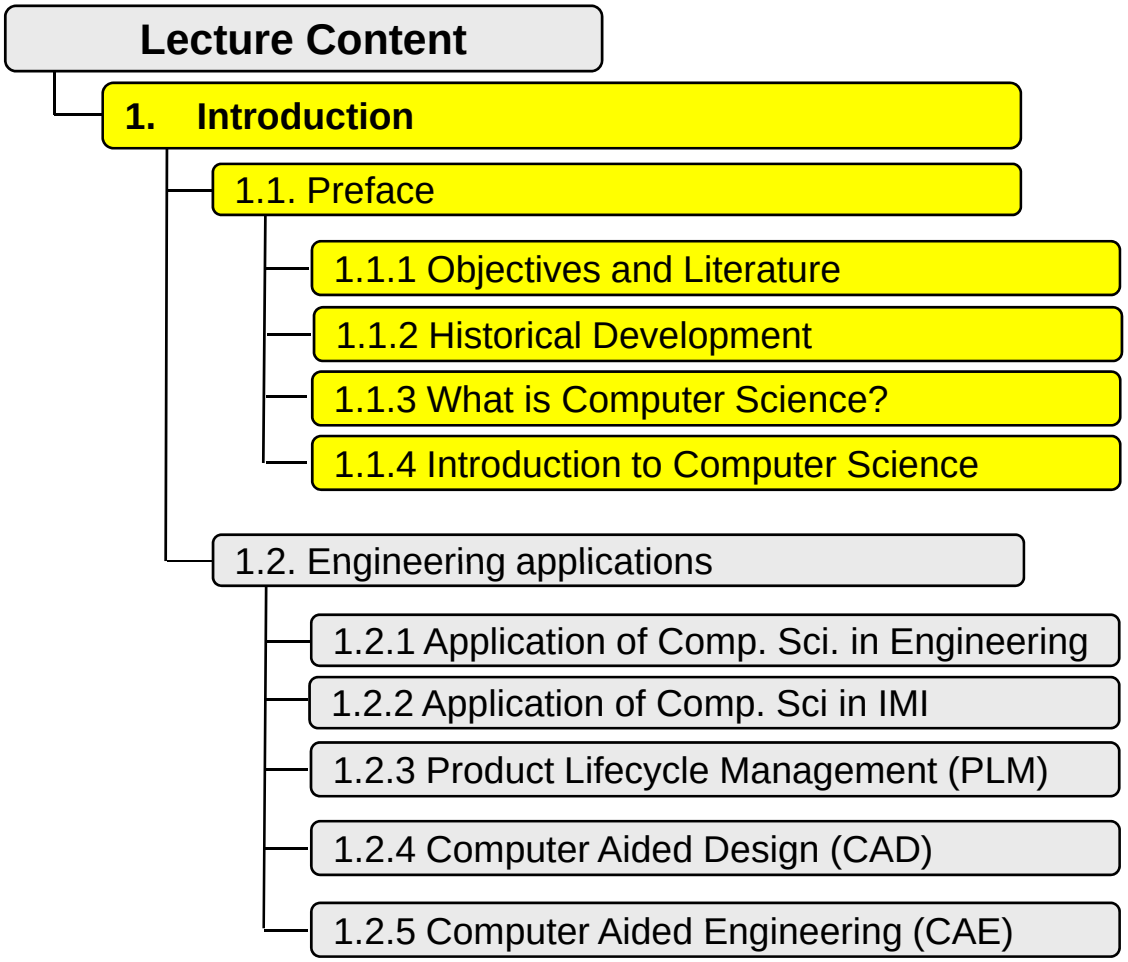
Dipl. Wi.-Ing. Dan Gutu

18th of April 2008

Computer Science for Engineers : Topics



Lecture Plan



Lecture Objectives

- Basic Knowledge of Computer Science and its Application in Engineering:
  - What is Computer Science?
  - Historical Development
  - Relevance of Computer Science to Engineering
  - Practical Applications
- Fundamental relationships of the presentation of information, processing and paradigms with which engineers must be familiar:
  - Object Orientation
  - Algorithms and Data Structures
  - Programming Languages
  - Computer Architecture
  - Software Engineering

Relevant Literature

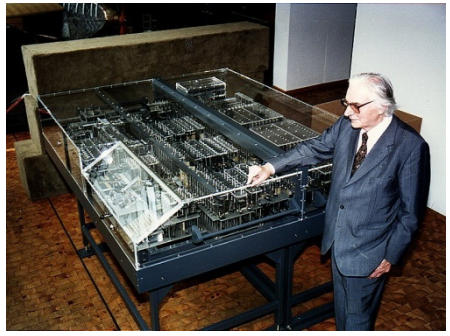
- Helmut Balzert: „Teaching the Basics of Computer Science“, Spektrum Akademischer Verlag, November 2004
- Waldschmidt, Walter: „Main Features of Computer Science“ – Band 1 & 2, Spektrum Akademischer Verlag, 1998
- Waldschmidt: „Introduction to Computer Science for Engineers“, Oldenbourg Verlag, 1987

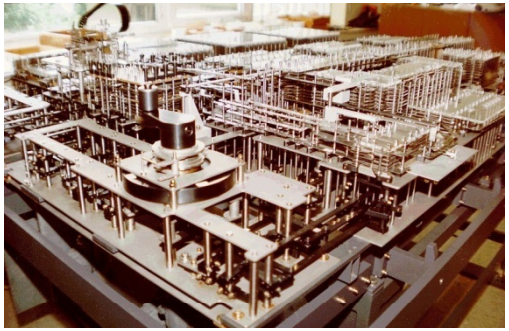
Historical Development (1)

ca. 5000 b. Chr.	Counting based on numbers (using fingers to help)
ca. 1100 b. Chr.	Suan-Pan-Procedure (Pearls on wire) Abacus (Romans)
ca. 500 a. Chr.	Hindu-Arabic counting system with 10 numbers from 0 to 9 Progress: Introduction of zero, base notation
1623	“Counting Clock” with 4 basic arithmetic operations (Schickard): first digital principle of data processing in the form of gear wheels
1833	Mechanical counting machine of Charles Babbage His suggestions for future counting machines fail due to lack of technical development
1890	Key punch method (reason: 11. American population census)

1936 Konrad Zuse develops the Z1

- In order to build the Z1 computer, in 1936 he quit his job at the Henschel Aeroplane factory and converted his parents‘ living room in to his workshop.
- The Z1 is completely mechanical (thin plates, cut out with a jigsaw).
- It was completely financed out of private means, but never reached satisfactory performance.
- Specs: 1 Hertz, 64 binary cells each with 22 Bits
  Weight: ca. 500 kg
  Power consumption: ca. 1000 Watt
  (for the electronic motor clock)
- *The Z1 was used as the model for other scientific calculators.*

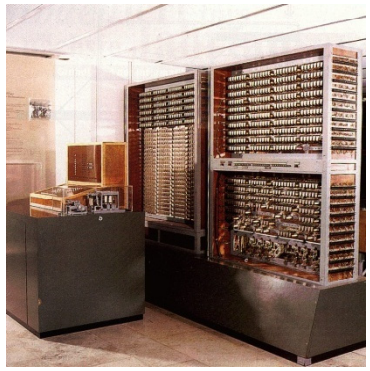




Completed Z1 1983 with Konrad Zuse - Deutsche Technikmuseum Berlin

1941 Konrad Zuse develops the Z3

- *The Z3 was the first functioning, freely programmable, binary based computer in the world.*
- Daten: 5-10 Hertz
  Weight: ca. 1000 kg
  Power Consumption: ca. 4000 Watt
  600 Relay Arithmetic Logic Unit
  1600 Relay Memory (64 Words x 22 Bit)
- Area of application: Wing calculations (Flutter problem)



Deutsches Museum München

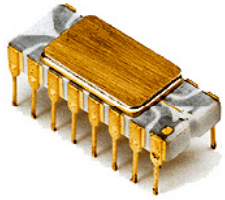
1944 H. Aiken (IBM) develops the relay computer MARK

Historical Development (4)

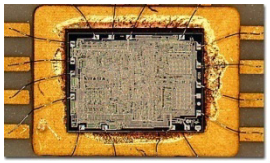
1948 Development of ENIAC (Electronical Numerical Integrator and Computer) **1st generation computer**

1957 **2nd generation computers** based on transistors

1964 **3rd generation computers** based on integrated circuits



1975 **4th generation computers**: Several thousand circuits on 1 chip



1981 First IBM PC (official name IBM 5150), which lasted 6 years without being changed

1982 **5th generation computers**: Development of parallel systems

1990 **-** today Change to **Information Society**:

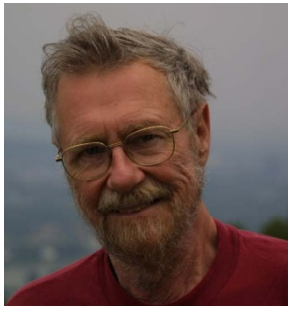
WWW, Internet/Intranet, E-Mail, E-banking, Multimedia, Virtual Reality, usw.

- Computer science has developed from mathematics at the same time the computers themselves were being developed.

Definition 1.1: **Computer Science** is the science concerned with the **structure, effectiveness, construction principles**, and the **application possibilities** of **information producing systems** as well as their **application**. [Stud05].

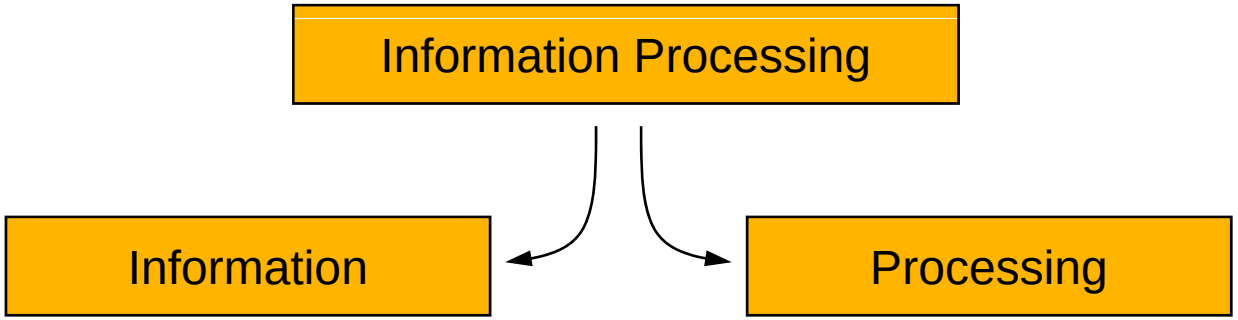
- The aim of computer science is to break away from specific conditions of **technical implementations** of existing computers as well as from **specific applications** through **abstraction and modeling**, to produce **general laws**, which determine information processing, to develop **standard solutions** and **standard development practices**. [Schn88].

„Computer Science is as concerned as little with the computer as Astronomy is concerned with the telescope.“
(Edsger Wybe Dijkstra, 1930 - 2002)



- It follows that in Computer Science the **computer is a tool**, to solve problems in Computer Science and other disciplines.

- Information processing (or data processing) is in general the processing of information including the storage and processing of sensory inputs from living organisms. Strictly speaking it is the **processing of digitally coded information**, which can be separated according to **fixed rules and classifications**. [Goos03].
- The term Information processing can be split into two terms „Information“ and „Processing“, which will be described on the following slides.



Data – Information - Knowledge [Stud05]

- **Data** is many characters of a language, whose purpose is to represent the processing of information.
  They contain a single syntactic dimension.

  i.e.: *„1500“*

- **Information** consists of syntax and semantic (form and content).

  i.e.: *„1500 is the number of rotations per second“*

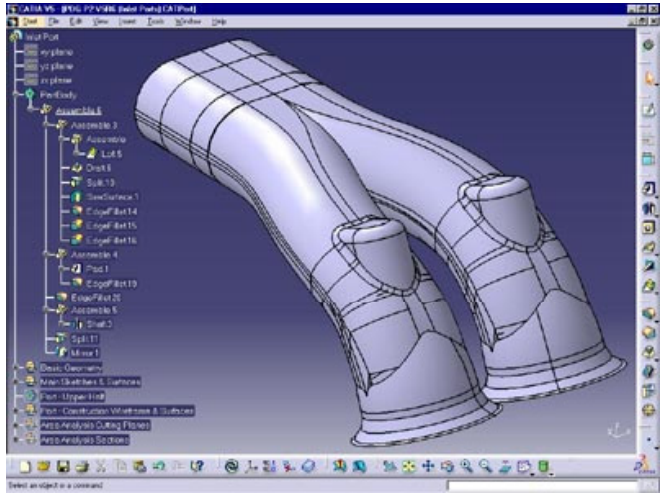
- **Knowledge** also consists of a pragmatic Dimension.
  It is also connected to a goal or purpose (operation orientated).

  i.e.: „*The idle-running speed of 1500 rpm is too high; the motor needs a break.“*

- Information is a potential, actually existing usable or used **sample of data representation**, that is relevant for an observer in a specific context.

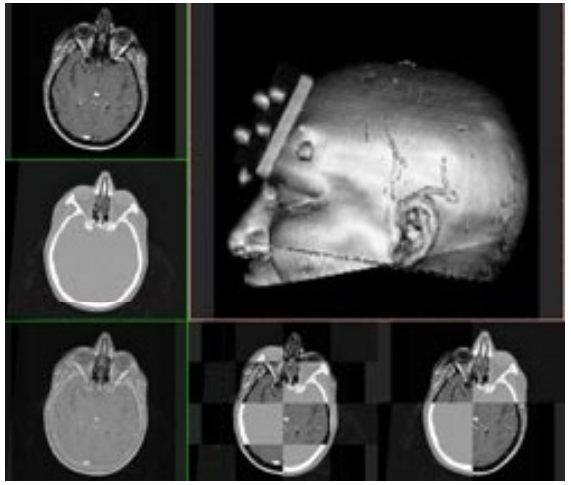
Data: ….AAB03F9390….

CAD-Modelling



***Source**: IBM, CATIA V5*

Medical Image Processing



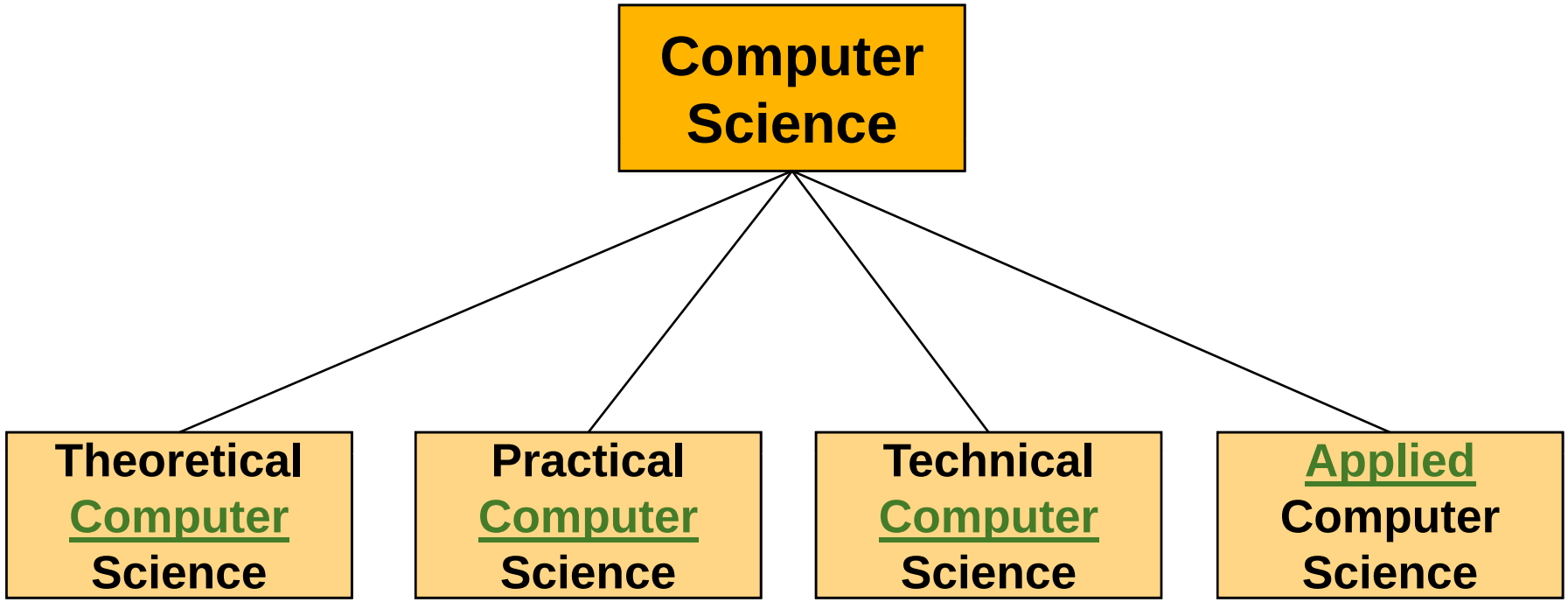
***Source:** Fraunhofer-IGD*

- In order to exactly describe the processing of information, Computer Scientists define the term „Algorithm“ as follows:

> **Definition 1.2**: An Algorithm is a precise description, according to which the execution of a certain operation of a system in a certain sequence is defined, with which it is possible to solve problems of a given type. [Bieh00].

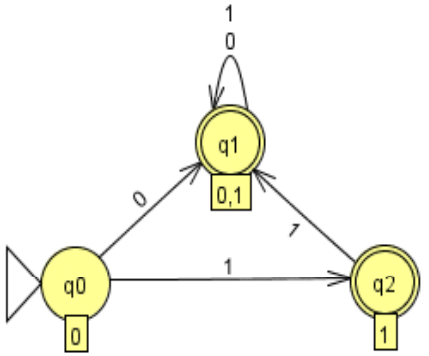
Important characteristics

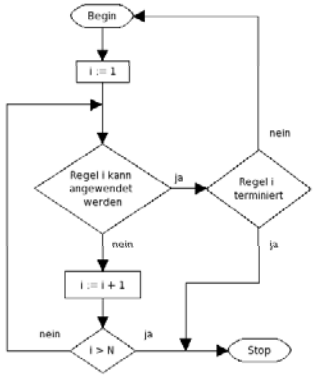
- **An algorithm must terminate.**
  Which means, it must supply an answer in a finite time.
- **An algorithm is deterministic.**
  Which means, it must for the same input data provide the same output data every time.


Computer Science
Theoretical Computer Science
Practical Computer Science
Technical Computer Science
Applied Computer Science

next

- **Theoretical Computer Science deals with theoretical basics:**
  - Automaton theory
  - Formal languages
  - Switching theory
  - Algorithm theory
  - Complexity of algorithms
  - Information theory
  - Coding theory
  - etc.





Insertion in a B-Tree (pseudo code)

- insert newEntry in the appropriate leaf;
- currentNode = leaf;
- **while** (currentNode overflow)
  split the currentNode into two nodes
  on the same level, and promote median key up
  to the parent of currentNode;
  currentNode := parent of currentNode;

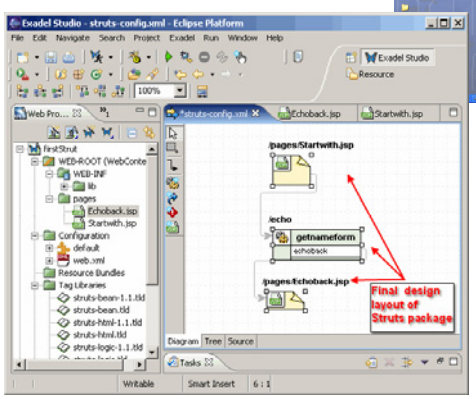
© Frederic Maire, QUT

- Knowledge of theoretical structures is important training for everyone who designs complex systems [Gumm02].

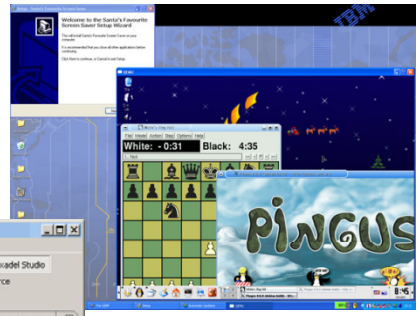
back

- **Practical Computer Science is responsible for system software:**

  - Programming languages
  - Compiler engineering
  - Operating systems
  - Programming methods
  - Computer traffic theory
  - etc.



Source: *Java Compier - Eclipse*



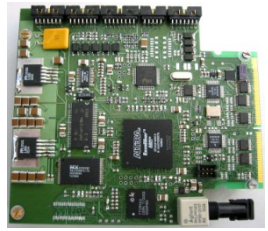
Source: *Knoppix*

Source: *Sharp*

- The interface between primitive operations, which can be performed by computer hardware, and the applications which are utilised by the user is the central task of practical computer science. [Gumm02].

back

- **Technical Computer Science is responsible for the functional architecture and logical design of digital computers and peripheral devices as well as for computer architecture and organisation:**
  - Computer architecture
  - Switching techniques
  - System and component development
  - Network technology
  - Computer organisation
  - Robotics
  - etc.



Source:
Embedded Systems
Universität Heidelberg

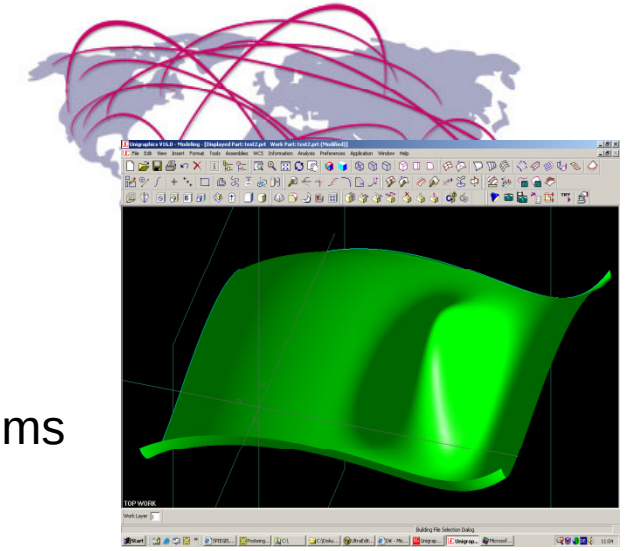
Source:
Lehrstuhl für kognitive Systeme
Universität Kiel

- To put it simply, one can say that technical computer science is responsible for the allocation of the hardware [Gumm02].
- The boundary between technical computer science and electronic engineering is not clearly defined.

back

- **Applied Computer Science uses knowledge from computer science in order to produce computers, software products and hardware for other sciences or application areas:**
  - Graphical data processing
  - Image processing
  - Data structures
  - Data organisation
  - Communication systems
  - Distributed data processing systems
  - etc.

Source: *United Videophone*



Source: *Fraunhofer-IGD*

Source*: UGS*

- Applied Computer Science is often used as a term for all interdisciplinary sciences with a computer science part.

backIntroduction to Computer Science a