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 2009
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 9654-526
Fax: +49 (0) 721 9654-527
Email: gutu@fzi.de
FZI
Main courses

- Computer Science for Engineers (CSE)
- Virtual Engineering (VE) I+II
- Product Lifecycle Management (PLM)
- Simulation in the product development process (SiPEP)

Industry courses

- Computer Integrated Planning of New Products (RPP)
- Integration of Products, Processes and Resources in the Development of Automobiles (PPR-Integration)
- Virtual Engineering for Mechatronic Products (VEmP)
- PLM in the Manufacturing Industry (PLM-F)
Lectures from the International Department

Information Systems I
Understanding and readiness of the implementation of the Product Lifecycle Management (PLM) approach

Information Systems II
Knowledge in the field of Computer Aided (CAx) approach and Virtual Engineering (VE)

Information Systems III
Knowledge in Interface, Data and Information Technologies

Methods of Simulation
Knowledge in the field of Methods of Simulation in Product Development Process (PDP)
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 2009
Fundamental Information

- **Lectures and Exercise slides** can be found on the homepage:
  
  http://www.imi.uni-karlsruhe.de/280.php

  *Note*: Slides may be updated after the lecture has been given!

- **Lecture**:
  - Friday, 15:45 – 17:15 hours, SR 203, ID

- **Exercise**:
  - Friday, 14:00 - 15:30 hours, SR 203, ID
  - Topics: Java, Data Structures, Algorithms, Development Tools (Eclipse)

- **Computer lab**:
  - See Website of the lecture
  - Certificate of completion of the computer lab is required for admission to the exam
• **When:** 07.04.2010, 14:00 to 17:00 hours

• **Relevant** for the exam preparation are
  - Lectures and Exercises (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
• **When**: Will be announced on the homepage

• Registration over the website

• The computer lab begins in the 3rd week of lectures.

• 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

- 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.
- The attendance of the computer lab is compulsory.
- 15 minutes late means the student is considered to be absent at that computer lab.
- A student missing twice from the lab without proper reason will be excluded from the lab.
Exercise Hand-In

• Hand-in:
  - Student demonstrates the program to the tutor
  - Tutor asks questions about how the program works, and other relevant information
  - Each exercise is evaluated with maximally 20 points
  - At least 65 points are required for exam admission

• Hand-in date:
  - The exercise should be handed in to the tutor the latest by the end of the second week.
  - The tutor will then ask questions about the handed in solution at the next lab class.
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 2009
Lecture Content

1. Introduction
   1.1. Preface
      1.1.1 Objectives and Literature
      1.1.2 Historical Development
      1.1.3 What is Computer Science?
      1.1.4 Introduction to Computer Science
   1.2. Engineering applications
      1.2.1 Application of Comp. Sci. in Engineering
      1.2.2 Application of Comp. Sci in IMI
      1.2.3 Product Lifecycle Management (PLM)
      1.2.4 Computer Aided Design (CAD)
      1.2.5 Computer Aided Engineering (CAE)
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 (1)

- Helmut Balzert: "Teaching the Basics of Computer Science“, Spektrum Akademischer Verlag, November 2004


- Waldschmidt: "Introduction to Computer Science for Engineers“, Oldenbourg Verlag, 1987


Historical Development (1)

c.a. 5000 b. Chr. Counting based on numbers (using fingers to help)

c.a. 1100 b. Chr. Suan-Pan-Procedure (Pearls on wire)
Abacus (Romans)

c.a. 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 into 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)

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

Deutsches Museum München
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

• 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** 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 context sensitive

- 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

Medical Image Processing

**Source:** IBM, CATIA V5

**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. \[\text{[Bieh00]}\].

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

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

---

1.1. Preface / 1.1.4. Introduction to Computer Science

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;
• Practical Computer Science is responsible for system software:
  - Programming languages
  - Compiler engineering
  - Operating systems
  - Programming methods
  - Computer traffic theory
  - etc.

• 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].
Technical Computer Science

- **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.

- 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.
• **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.

• **Applied Computer Science** is often used as a term for all interdisciplinary sciences with a computer science part.
1. Introduction
   1.1 Preface
      1.1.1 Objectives and Literature
      1.1.2 Historical Development
      1.1.3 What is Computer Science?
      1.1.4 Introduction to Computer Science
   1.2 Engineering applications
      1.2.1 Application of Comp. Sci. in Engineering
      1.2.2 Application of Comp. Sci in IMI
      1.2.3 Product Lifecycle Management (PLM)
      1.2.4 Computer Aided Design (CAD)
      1.2.5 Computer Aided Engineering (CAE)
Construction Office at the Beginning of the 20th Century: Construction on „Paper“

Source: Prof. Eigner, VPE 1
Product Development at the Beginning of the 21st Century: Construction in 3D - CAD

1. Introduction

1.2. Eng. Appl. / 1.2.1. Application of Computer Science in Engineering

Source: BMW AG
1. Introduction

1.2. Application in Engineering

- Integrated Product Development
- Automated Manufacturing Systems
- Conveyor Systems
- Material and Process Simulation
- Information Management in Engineering
- Mechatronics
- Robotics
- …
Simulation in the Development Process

Simulation

- Verification from calculations and measurements
- Model parameters identified
- Model improvement

Model Parameters
- Test Definitions
- Controller Data

Shifting tests in the Simulation: „From Road to Rig to Office“

Test Bench

Test Run

Source: IPEK
Aim:

Simulation of the thermal behaviour of components in the machining process for the optimisation of machine parameters, working sequence and improving production processes for cost minimisation.

Motivation:

• Complex components exhibit a high number of nodes.
• Complex components require a lot of different process steps.
• Heavy simplification of components and processes are needed to make simulation possible.

Application of high performance computers:

• Simulation of complete components
• High degree of detail in each process
• Reduce computing time
• No memory overflow

Source: wbk
Traffic Noise Reduction:

Tires / Road Noise Simulation

Modelling of tires/lane changing taking in to account age and road surface texture, drive and brake torque, friction and air displacement.

Pressure distribution across the surface of the tire in contact with the road.

Source: Continental
Application of Computer Science in IMI

- 3 Research Areas
  - Lifecycle Engineering
  - Collaborative Engineering
  - Virtual Engineering

- Application of key Virtual Reality (VR) technology

- Layout the Lifecycle Engineering Solutions Center (LESC)
Lifecycle Engineering: Product Lifecycle Management (PLM)

- MANUFACTURING PLANNING
  - CAD / CAM

- CONSTRUCTION
  - CAD / CAE

- PRODUCTION
  - ERP / PIM
  - PRODUCTION

- SALES
  - CIS

- UTILISATION
  - CIS

- DEVELOPMENT

- SALES AND UTILISATION

Legende:
- CAS Computer-Aided Styling
- CAD Computer-Aided Design
- CAE Computer-Aided Engineering
- CAM Computer-Aided Manufacturing
- VIS Visualization
- PLM Product Lifecycle Management
- PIM Production Information Management
- CIS Customer Information System
- ERP Enterprise Resource Planning
- MIS Marketing Information System
Lifecyle Engineering - Vision

- Requirements
- Specification
- "Design-in-Context"
- Prototype-reference configuration
- Integrated virtual validation
- Customer presentation
- Product optimisation

PRODUCT

- Project organisation
- Line organisation
- Supplier
- Distributor
- Workshop

PROCESS

- Development
- Testing
- Manufacturing
- Sales
- Usage
- Maintenance
- Recycling

INFORMATION

- Customer requests
- Customer relationship
- Customer feedback

CUSTOMER
Using modern information and telecommunications technologies it is possible for several participants world-wide to participate in conferences.

**Aim:**
- No physical boundaries
- Not bound to time
- Easily and economically organisable
- Automatic saving of the meeting
- "free" team composition
- Accelerated problem solution
- Provide sketches within a discussion
- Share written or electronic documents

**Example:**

New York

Digital Video Camera
Complete, computer-based and integrated modelling of a vehicle throughout the entire product life cycle, from the specification to service and recycling. The virtual vehicle allows a „purely virtual“ handling of the future vehicle for the developers, suppliers, manufacturers and clients alike, so that they are all able to judge it from the point of all its qualities and functions.
Virtual Engineering: Vision

„Build the product right the first time!“

Virtual Product

- Context-Orientated Design
- Prototype-Orientated Configuration
- „High-end“ Visualisation
- Validation
- Feedback und Optimisation

Team Collaboration

- CAx
- PLM
- VR/AR
- Web-Portale
- Telecommunications Services

Know-how

Information and Communication Technologies

Engineering Network

- Iterative Workflows
- „Workflow“ & „Workload“ Management
- Quality Gates
- Deliverables & Progress Management

Product life cycle

Product Development Process

Prof. Dr. Dr.-Ing. Jivka Ovtcharova – CSE - Lecture – Ch. 0 and 1 - WS 09/10 - Slide 48
Crash Simulation

Ergonomy Test

Assembly

Quelle: Adam Opel AG
Virtual Factory
Interactive Virtual Living Lab
Stationary widescreen projection for high-definition immersive visualisation

Collaboration Room
- Integration of different groups (End users and cluster users)
- Flexible access to the projection facilities

Mixed Reality Labor
- Mobile projections
- Haptic input/output
Pool
- Product Lifecycle Management Labor
- Simulation laboratory
LESC opening ceremony on the 25th of June 2008
Qualification for new professions: science meets industry!

Design Engineer

Virtual Mock-Up Engineer

Process Integration Manager

Multi-project Manager
Tasks during the product development process

- Development of new application fields in the product development through the use of innovative technologies.
- Orientation towards improvement of industrial processes and optimization of the entire product life cycle.
- Extending existing local business procedures and infrastructures to allow continuous IT-supported business process.

Need for a distributed solution for the integration of
  - Data
  - Processes
  - Resources

New management and organizational concept:

Product Lifecycle Management (PLM)!

Source: Berliner Kreis, Technology Monitoring
Aims

Engineering Improvement

Guarantee of a **constant, virtual and transparent information flow** within the entire product life cycle

**Business processes**
Optimization of management and processes

**Resources**
Optimal usage

**Information**
Supply at the right place and the right time with quality and quantity which meet demands

IT and Communication technologies

**Source**: Eigner, „Produktdatenmanagement-Systeme: ein Leitfaden für Product Development und Lifecycle Management“

Engineering covers not only the construction, but all processes for product manufacturing (from the product idea to recycling).

This is achieved through control of business processes in the entire product life cycle and supply of information and resources at the right place and time, with quality and quantity which meet demands. Resources can be people, machines, technologies etc.

Using IT and communication technologies all engineering from a company can be improved.
Motivation

Product Lifecycle Phases and Tasks

- Design
- Simulation
- Process Planning

Applications

Digital Mock-Up

Digital Factory

Working Status?

Compare

Archiv

Classification

Search

Request

Test

Change

Data Status?

Version

Status

IT Systems

CAD Data

CAE Data

CAM Data

CAD System

CAE System

CAM System

Prof. Dr. Dr.-Ing. Jivka Ovtcharova – CSE - Lecture– Ch. 0 and 1 - WS 09/10 - Slide 57
• **Product elaboration:**
  - Reduction of the “lead time“ (for example though simultaneous engineering)
  - Reduction of the product manufacturing costs (for example through lean production)
  - Spatial and organizational separation of development and production
  - Strengthened engineer's team work (for example virtual teams)

• **Business management:**
  - Use of new business organizations (e.g. cooperations)
  - Frontloading in the early stages of product development

• **Customer service:**
  - Strengthened feedback and use of customer information to optimize product development

• **Specifications und Regulations:**
  - New, aggravated laws and regulations (e.g. environmental ordinances, quality regulations like ISO 9000)
Basic functionality of PLM

Product Lifecycle

- Graphically intensive working methods, Viewing, Browser
- Illustration of product data structures
- Version-based administration system
  - Illustration of life cycles
  - Access administration
- Interaction with external systems (CAx)
- Data files and queries

Web based user interface

Product data models

Administration functions

Data interfaces

- CATIA
- ProE
- UG
- SAP
- AutoCAD
- Nastran

Meta data

Product data
Management of Access Rights

PLM Access

1 User
Thomas Maier
User Access: tmaier
- assigned to the person Thomas Maier

2 User Group
Project „Delta“
Project team 1
Project team 2
Designer
Technical Staff
Project Manager

User Access: tmaier assigned to
- Project “Delta”
- Team Project team 1
- Group Designer
- Role User a
Configuration example for a car

Source: Daimler AG
- On the lowest PLM system level, the database, as well as the tools for the administration of files can be found. Usually a commercial database system runs on this level (e.g. SQL-Server or ORACLE).

- When designing a PLM system, a database concept must be chosen. This database concept establishes basic principles, according to which the data will be archived and correlation between them can be formed. There are various forms of databases, for example the
  - hierarchical database (HDBMS)
  - relational database (RDBMS)
  - Object oriented database (ODBMS)

- While hierarchical database systems are scarcely used, today most applications primarily implement the relational database concept.
Different structures (views) on the same product are used, allowing a targeted approach on the different processes and tasks.

**Construction view** contains structural relations.

**Assembly view** Represents the assembly order.

- **Cover**
- **Case**
- **Rotor**
- **Rotor cover**
- **Window fay**
- **Actuator**
- **Tank**
CAD stands for:

- **Computer Aided Drafting** – simple drawing preparation systems
- **Computer Aided Design** – efficient construction systems.

One understands that CAD is computer aid in development and construction and refers strictly to graphically intense production and manipulation of an object.
View on the assembly axle driving shaft

Perspective representation of the assembly „axle driving shaft“
Assembly Modelling in 3D

Local coordinate systems

Assembly representation

Explosion representation
CAD systems can be fundamentally differentiated according to whether their model space is a 2D or 3D system.

**2D CAD systems:**

Component geometry is illustrated in a two dimensional co-ordinate system by one or two dimensional elements such as points, lines, curves and surfaces. Commonly used systems include:

- AutoCAD LT from Autodesk
3D CAD Systems:

The model of a construction unit / building group is illustrated using a three dimensional co-ordinate system and consists of one, two or three dimensional model elements such as points, lines, two and three dimensional curves, surfaces and solid primitives (cube, cylinder, ball, torus etc.).

Common 3D systems are:
- **Catia** from Dassault Systemes
- **UG NX (6)** from Siemens PLM
- **Pro/Engineer** from Parametric Technology Corporation
3D CAD Systems

- 3D CAD systems allow the **production of solid models**, e.g. of construction units / building groups, and offer **extended modelling techniques** such as:
  - Parametric design
  - Feature based design.

- **Modelling in 3D CAD requires a modified way of thinking and working for co-workers**

- The solid model can be used in the next phases of the product development process e.g.:
  - Programming manufacturing machines (CNC)
  - Calculation und Simulation: e.g. finite element methods (FEM), multiple body systems (MBS)
  - Assembly planning and Virtual Mock-Up (VMU).
2D and 3D approach

2D drawing

Prototype

Manufacturing

Computation

Prototype

CNC-Model

CNC-Program

Computation model

FEM-Model

Virtual Mock Up

VMU-Model

Interfaces

Model

3D model

1. Introduction

1.2 Eng. Appl. / 1.2.4. Computer Aided Design (CAD)
Product Informationen in the Product Model

Product definition
- i.e. over
- Reference: Valve housing
- Identification number: 1234509876
- Classification number: VE-0815-4711

Product representation
- i.e. as
- CSG-Structure
- B-Rep-Structure
- Feature-Structure

Product presentation
- i.e. As an exploded representation, parts list or technical design

Source: DiK, TU Darmstadt
• The Product Structure contains the outline of product geometry in module structures. As such, single modules are divided into assemblies and single parts.

• On the figure above, the product structure of a robot arm is represented. At the highest level of the product structure, the **product „robot arm“** is located. The single parts and a sub-assembly („Produkt3“) are located in the product structure.

• The structuring of the product is done in compliance with the conditions between the parts and sub-assemblies.
Crash Simulation (1)

Crash-Test with a DMU

Source: Torsten Kuhlen „Virtuelle Realität in der Automobilindustrie“
Crash types:

- Frontal crash
- Side crash
- Rear-end collision
- Crash with a tree
- Protection from cargo
- Seat belt
- Pedestrian crash.

Source: Adam Opel GmbH
• System & Software Engineering
  - Software Architecture
    ▪ SOA: Service Oriented Architecture (techn.) - approach for interlinking of engineering applications
    ▪ Distributed applications – programs, that run on resources that are spatially distributed in different locations: servers in America, database in Europe

• Software Engineering
  - Development/Customizing of PLM systems
    ▪ Ex. requirements modelling
    ▪ Working in a interdisciplinary teams (Computer Scientists + Mechanical Engineers)
  - Mechatronic
    ▪ Mechatronic = Mechanic + Electronic + Software
      • Development of mechatronic systems
- Data structures
  - Product structures
    - Representation of a product structure in software
  - Data exchange
    - Data formats / structures for data exchange (ex. XML-based solutions)
  - Scene graphs (3D-visualisation)
    - Treelike structures for representing a 3D scene in software.
- Simulation Algorithms
  - Finite Elemente Method (FEM),
  - Multibody Simulation
  - Virtual factory: material flow,
  - Kinematic Simulation (parts movement)
- Cryptography
  - Information encryption for data exchange,
    Security aspects in product data exchange

```c
int function foobar(int i){
  for (i;i<42;++i) {
    do.(something);
    dosomething(else);
  }
  return else;
}
```
[Balz05] Helmut Balzert: „Lehrbuch Grundlagen der Informatik“, Spektrum Akademischer Verlag, 2005


[Stud05] Rudi Studer: „Grundlagen der Informatik 1, SS 2005“, Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Uni-Karlsruhe, 2005


Sources

[RNKS05] http://www-rnks.informatik.tu-cottbus.de/
[InfA05] http://www.informatik.uni-augsburg.de/
[Bogd05] http://www.uni-koblenz.de/~bogdan/TCP_IP_Presentation.ppt
[MatM05] http://www.mathematik.uni-marburg.de/