Recommendations for Bachelor- / Master Programs in Computer Science

Hans-Ulrich Heiss

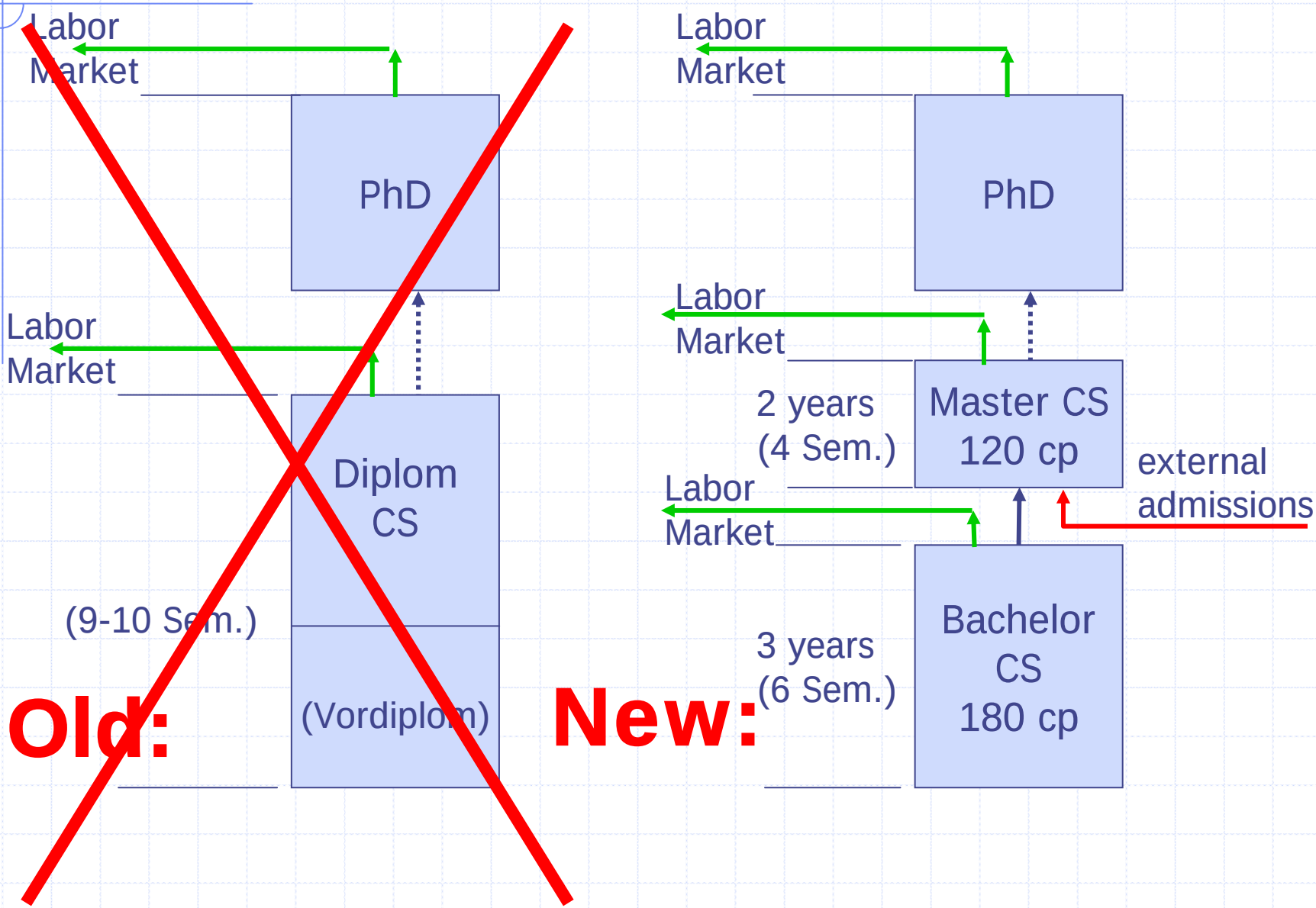
Faculty IV Electrical Engineering & Computer Science
TU Berlin

Study Commission Fakultätentag Informatik Germany

General Remarks

- German university diploma (5 year program) has a long-standing tradition of a high-quality degree
- German implementation of the Bologna Process forces universities to abolish the diploma
- Master degree should be comparable to diploma degree
- Students are expected to aim at a master degree in general

Structure


Labor Market
PhD
Labor Market
Diplom CS
(9-10 Sem.)
(Vordiplom)
Old:
Labor Market
PhD
Labor Market
2 years (4 Sem.)
Master CS 120 cp
external admissions
Labor Market
3 years (6 Sem.)
Bachelor CS 180 cp
New:

Design Space of Academic Programs restricted by....

Goals of Recommendations

- **Provide guidelines for member universities**
  - No need to reinvent the wheel
- **Restrict range of variants**
  - Define a mandatory core of subjects
- **Ensure comparability of German degrees**
  - Graduates should have a comparable competence
- **Ensure mutual acknowledgement of degrees**
  - Changing universities should be easy
- **Facilitate accreditation**
  - Following the recommendations should help to be accredited successfully

General Properties of B/M-Program

- **Scientific Orientation**
- **Consecutiveness**
  - Master program based on bachelor program's content
- **Duration**
  - Bachelor: 6 semester
  - Master: 4 semester
- **Employability**
  - Programming, Software Engineering...
  - Soft skills (team work, presentation techniques, writing,...)
- **Quality Assurance**
  - Teaching evaluation, study success and progress control, alumni program
- **Counseling**
  - Mentor program: each student is assigned a professor (mentor) as individual advisor
- **Modularization**
  - Curriculum organized in modules

Module concept

- A reasonable clustering of single courses that
  - build on each other or complement each other
  - belong to same phase of study
  - pursue a well defined competence profile
- Idea: one module – one exam – one mark
- Size of module: usually 4-9 cp
- Duration: 1-2 semesters
- Exams immediately at the end of module
- Modules may build on each other, to form sequences of specialization
- Modules are defined „supply-side“ and announced in a module catalogue (document updated annually)
- Reduction of administration overhead at central and local exam offices
- Ease at developing innovative study programs for life long learning

Calculation of „Workload“ acc. to ECTS

- ECTS: European Credit Transfer System
- Credits estimate the real work load of an average student
- 1 ECTS credit point = 30 h work
- Workload per year: max. 1800 hours:
  - 45 weeks with 40 hours
- Equivalent to 60 ECTS-cp
- per semester: 30 ECTS-cp
- Example: 2 hours weekly lecture
  - 2 presence hours * 15 weeks = 30 hours
  - Preparation and afterwork: 15 hours (1h/week)
  - Exam preparation: 15 hours total
  - Total: 60 hours workload = 2 ECTS cp

Properties of Bachelor Program

- **Teaching the scientific foundations of discipline**
  - Preparation for lifelong learning
  - (No change with regard to "old" Diplom)
- **Employability**
  - Teaching facts, methods and skills necessary for getting a qualified job
- **Challenge**
  - How to squeeze both goals into a 3-year program

Bachelor‘s General Competences

A bachelor program’s graduate…

- masters methods to analyze problems
- masters methods to build models
- has skills to solve programming problems under technical, economic and social conditions
- is aware of IT security problems and knows the means to address them
- has gained experience in some application area
- knows how to work in teams and to take over responsibility for particular tasks
- has acquired competences outside CS and is sensible for non-technical aspects
- is well prepared for lifelong learning in different areas of employment

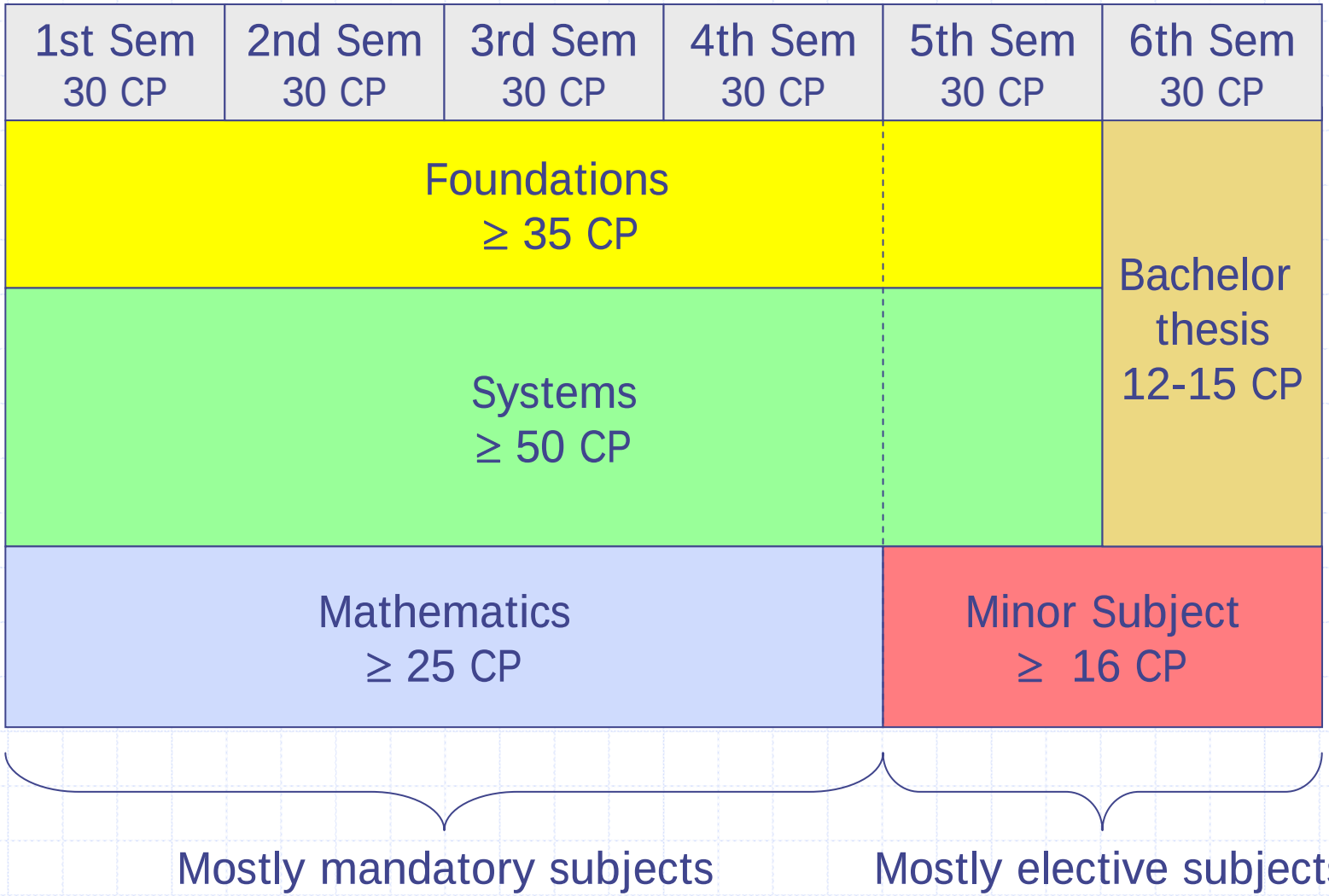
How to attain employability

- Practice in programming and software development are emphasized
- Soft skills are acquired in regular courses
- Mandatory seminars including teaching rhetoric, presentation techniques, scientific writing...
- Mandatory projects to learn to self-organize in teams, to take over responsibilities, to keep deadlines, to communicate, to document, to present results...
- Additional courses from other faculties are encouraged (rhetoric, presentation, personal working techniques, business administration...)

Example: Seminar

- Students are expected to give a presentation and to write a seminar paper
- Seminar starts with an introduction to presentation techniques, rhetoric and scientific writing
- Students are given topics, but no material
- All student presentations are video-taped and discussed in group
- Seminar papers undergo a peer-to-peer-review (similar to conferences)

General Structure of Bachelor‘s program



Mandatory Foundation Subjects

- Automata, Formal Languages, Complexity
  - Chomsky hierarchy, computability, decidability, complexity, NP-completeness,...
- Logic
  - Propositional logic, predicate logic, incompleteness, logic programming,...
- Formal Systems
  - Induction and recursion, term algebras, abstract data types,...
- Modelling
  - ER models, state and transition models, UML, petri nets, model transformation,...
- Programming
  - Basic concepts of imperative and OO languages,...
- Programming paradigms
  - OO, functional, logic and parallel programming,...
- Data Structures and Algorithms
  - Sorting and searching, hashing, trees, graph algorithms, algorithmic principles, verification, complexity analysis,...

Mandatory Systems Subjects

- Digital Systems
  - Boolean algebra, logic networks, minimization, functional components, realization of logic functions,...
- Computer organization
  - Number representation and arithmetic, assembler programming, processor architecture, instruction interpretation, pipelining, memory hierarchy, I/O,...
- Operating Systems
  - Processes, concurrency synchronization, communication, files, system calls, shells, utilities,...
- Computer Networks and Distributed Systems
  - Services and protocols, architectures, OSI reference model, Internet protocols, causality and logical time,...
- Software Technology
  - SW process models, project management, requirement analysis, specification, implementation techniques, testing, maintenance, CASE-tools,...
- Database Systems
  - Relational model and algebra, normal forms, SQL, query optimization, transactions and consistency,...
- IT Security
  - Threat and risk analysis, security policy, cryptography and protocols, authentication, access control, information flow, models,...

Elective Systems Subjects

- Artificial Intelligence
  - Knowledge repr., searching, learning, robotics, natural language processing
- Compiler construction
  - Syntax, semantics, lexical analysis, parsing, code generation, code optim.
- Human Computer Interface
  - Software ergonomics, user interfaces, usability engineering, workflows
- Simulation
  - Continuous, discrete and hybrid processes, event based simulation, applic.
- Computer Graphics
  - Raster and vector graphics, 3d-transforms, projections, illumination models
- Computer Vision
  - Pattern recognition, image processing, projective geometry, camera models
- Social Aspects
  - Information society, globalization, E-governance, regulation issues, IP
- Electrical Engineering
  - System theory, control theory, information theory, semiconductors,
- System Software
  - Low-level programming, memory management, communication networks
- Embedded Systems
  - Specification, realtime-OS, realtime scheduling, HW/SW-codesign, DSP

Mathematics Subjects

- **Mandatory:**
  - Calculus
    - Rational, real and complex numbers, series, limits, convergence, continuity, differentiation, integration (one and more variables), Fourier transform
  - Linear Algebra
    - Vector spaces, linear equation systems, linear mappings, matrices, determinants, eigenvalues
  - Discrete Structures
    - Sets, relations, graphs, terms, groups, fields, combinatorics, basics of number theory
- **Elective:**
  - Probability theory
    - Probability spaces, stochastic variables, independence, distributions and their moments, central limit theorem, stochastic processes, Markov chains
  - Statistics
    - Relevant distributions, sampling theory, test theory, confidence intervals
  - Numerics
    - Floating point arithmetic, stability, interpolation, iterative solution of LES, ordinary DES

Minor Subjects

- **Goal**
  - Understand terminology and way of thinking of another discipline
  - Be prepared for multidisciplinary work
- **Any reasonable field of application or complementary subject, e.g.**
  - Electrical Engineering
  - Mathematics
  - Business Administration
  - Economics
  - Mechanical Engineering
  - Physics
  - Linguistics
  - (Neuro)Biology
  - Philosophy
  - Sociology
  - Psychology
  - Political Sciences
  - Music

Example: Bachelor at TU Berlin

<table>
<tr><td>CP</td><td colspan="5">Bachelor’s Program in Computer Science</td></tr>
<tr><td>1st Sem. 29 CP</td><td>Digital Systems (6 CP)</td><td colspan="2">Algorithmic and Functional Solution of Discrete Problems (9 CP)</td><td>Found. and Algebraic Structures (8 CP)</td><td>Linear Algebra (6 CP)</td></tr>
<tr><td>2nd Sem. 29 CP</td><td>Computer Organization (6 CP)</td><td colspan="2">Data Structures and Algorithms in Imperative Style (9 CP)</td><td>Automata and Complexity (6 CP)</td><td>Calculus I (8 CP)</td></tr>
<tr><td>3rd Sem. 32 CP</td><td>System Programming (6 CP)</td><td rowspan="2">Software Engineering (12 CP) Including Project</td><td>Practical Program Development (6 CP)</td><td>Logic and Calculi (6 CP)</td><td>Calculus II (8 CP)</td></tr>
<tr><td>4th Sem. 30 CP</td><td>Networks and Distributed Systems (6 CP)</td><td>Database Systems (6 CP)</td><td>Specification and Semantics (6 CP)</td><td>Stochastics (6 CP)</td></tr>
<tr><td>5th Sem. 30 CP</td><td colspan="2" rowspan="2">Computer Science Electives (21-24 CP) Software Technology or Communication Technology</td><td colspan="2">Minor Studies (12-15 CP)</td><td>Management (6 CP)</td></tr>
<tr><td>6th Sem. 30 CP</td><td colspan="2">Bachelor’s Thesis (12 CP)</td><td>Social Aspects of CS (6 CP)</td></tr>
</table>

Properies of a Master‘s Program

- Research orientation
- Specialization and depth
- Integration into research activities of faculty
- Providing knowledge and skills for academic work (e.g. Master‘s thesis)

Master‘s General Competences

A master program’s graduate

- has reached a higher maturity and self-assurance in solving CS problems
- has acquired advanced knowledge in one specialization area
- has the necessary breadth and depth to get quickly used to new areas
- is not only able to apply the acquired problem solving skills in research and development, but also to challenge and develop them further, if necessary
- has acquired technical, intellectual and social skills to be prepared for management and leadership

Example: Master at TU Berlin

<table>
<tr><th>CP</th><th colspan="3">Master’s Program in Computer Science (Basic Structure)</th></tr>
<tr><td>1st
30 CP</td><td rowspan="3">Major Studies (54 - 60 CP)
including at least 30 CP
in the specialization area
◆ System Engineering
◆ Dependable Systems
◆ Intelligent Systems
◆ Communication Systems</td><td rowspan="3">Minor Studies
(18 - 24 CP)</td><td rowspan="3">General Studies
(12 - 18 CP)</td></tr>
<tr><td>2nd
30 CP</td></tr>
<tr><td>3rd
30 CP</td></tr>
<tr><td>4th
30 CP</td><td colspan="3">Master’s Thesis
(30 CP)</td></tr>
</table>

Specialization areas in the Master‘s Program

- **System Engineering**
  - Software Engineering, Programming Language Design, Compiler Construction, Computer Organization, Design Automation, Operating Systems, Performance Evaluation, Information Systems, Computer&Law, Information Economy, System Analysis, Enterprise Architecture, Net Business Processing,...
- **Dependable Systems**
  - Component-Based Modeling, Specification Tools, Semantics and Calculi, Security&Trust, Realtime Systems, Computer and Network Security, Correctness, Testing, Fault-tolerance,...
- **Intelligent Systems**
  - Neural Information Processing, Bio-Informatics, Intelligent Data Analysis, Computer Graphics, Computer Vision, Image Analysis, Robotics, Artificial Intelligence, Agent Oriented Systems,...
- **Communication-based Systems**
  - Communication Networks, Protocol Design, Performance Evaluation, Mobile Communication, Ubiquituous Communication & Ambient Intelligence, Next Generation Networks, (Open) Distributed Systems, Service Delivery Platforms,...

Next Steps

- Get feedback from member universities about compliance with recommendations
- Further elaborate recommendations in terms of outcome orientation
- Exchange ideas and concepts with other European countries

- Recommendations available at the homepage of Fakultätentag Informatik
  - in German

www.ft-informatik.de/fileadmin/dokumente/2005/bachelor_master_empfehlungen.pdf

  - in English

www.ft-informatik.de/fileadmin/dokumente/2005/bachelor_master_recommendations.pdf

Alternatives in discussion