

Diverging Paths: How German University Curricula Differ From Computing Education Guidelines

Louis Kobras^{}, Marcus Soll^{}, Jörg Desel

Supplementary Material – Coding Tables

Research Questions

RQ1 *Which learning outcomes defined in the guidelines of the GI can still be found in teaching?*

RQ2 *Which learning outcomes defined in the guidelines of the GI can no longer be found in teaching?*

RQ3 *Which new learning outcomes are actually taught in higher education but are not yet addressed in the GI recommendations?*

The parts of this document map onto the research questions like this:

RQ	Part
RQ1	Part A (pp. 3)
RQ2	Part B (pp. 15)
RQ3	Parts C (pp. 37) and D (pp. 57)

For RQ3, Part **C** describes competences and contents that were identified during the study which pertained to subject areas *already existing* in the GI catalog, whereas Part **D** lists contents and competences for *entirely new* domains that are as of yet not part of the GI catalog.

The tables in Parts **A** and **B** are best-effort translations of the GI list of competences for computer science¹ provided by the authors of the main paper.

¹Zukunft, O. (2016). *Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen* (Juli 2016) [Recommendations for Bachelors and Masters programs in the academic study program computer science at higher education institutes (July 2016)]. Bonn: Gesellschaft für Informatik e.V. Available online: <https://dl.gi.de/handle/20.500.12116/2332> (accessed on 2025-09-23).

**A. Competences remaining in the GI catalog
as per RQ1**

Table S1: Competences from the GI catalog that were identified in at least four curricula. If a competence had been extended during coding only the current extended variant has been considered for this table. Such extended variants are marked with a purple background. Extensions to the original are marked in *italics*.

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
I Algorithms and Data Structures	I.1 Understand	I.1.a Explain asymptotic behavior of functions (Big O notation) and the most common complexity classes.	5
		I.1.c Explain fundamental principles of the design paradigms backtracking, greedy, divide-and-conquer, and dynamic programming as well as example algorithms. Explain fundamental data structures (fields, linked lists, binary trees, hash tables, balanced trees like AVL trees or B trees).	6
		I.1.d Explain fundamental abstract data types like stack, queue, list and corresponding data structures. Explain the difference between an abstract data type and a data structure.	5
	I.2 Apply	I.2.a Execute algorithms like breadth-/depth-first search, Dijkstra, Floyd-Warshall for exemplary inputs.	6
	I.2a Transfer	I.2a.a Develop own algorithms for small problems, e.g., recursively reversing a list. Implement an algorithm for an application scenario.	4
	I.3 Analyze	I.3.a Analyze the runtime and space complexity <i>as well as properties like real-time capability, correctness, and problem appropriateness</i> of common algorithms and datastructures and determine their complexity classes.	12

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	I.3a Evaluate	I.3a.b Select an appropriate algorithm / an appropriate data structure given some constraints.	6
	I.4 Create	—	
II Numerical Analysis	II.1 Understand	II.1.a Describe and discuss fundamental concepts of analysis (convergence, continuity, differentiability, integrability) on examples.	5
	II.2 Apply	II.2.a Confidently use methods of differential and integral calculus for functions with a single variable. Explain or verify the limits and preconditions of those methods.	7
	II.2a Transfer	II.2a.a Identify and solve analytic problems in simple applications.	4
	II.3 Analyze	—	
	II.3a Evaluate	—	
	II.4 Create	—	
III Operating Systems	III.1 Understand	III.1.a Explain basic concepts and theoretical foundations of operating systems using technical terms.	8
		III.1.b Explain the general structure of operating systems using technical terms.	6
	III.2 Apply	—	
	III.2a Transfer	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	III.3 Analyze	—	
	III.3a Evaluate	—	
	III.4 Create	—	
IV Databases and Information Systems	IV.1 Understand	—	
	IV.2 Apply	IV.2.b Develop conceptual database designs for small examples and transform them into normalized relational database schemes.	9
		IV.2.c Execute complex queries and altering statements, <i>e.g.</i> , using SQL.	9
	IV.2a Transfer	IV.2a.a Create small database applications.	5
	IV.3 Analyze	—	
	IV.3a Evaluate	—	
	IV.4 Create	—	
V Digital Logic, Digital Systems, Computer Architecture	V.1 Understand	V.1.b Explain encodings for the representation of information, especially different representations of numbers.	5
		V.1.c Explain the structure of important circuits and their functionality.	5
		V.1.f Explain the schematic construction of a modern computer, the interactions between hardware and software, and the underlying general design principles.	7

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		V.1.g Explain fundamental structures of computer systems, e.g., file paths, controlling, pipelining, memory hierarchy, and input/output.	6
	V.2 Apply	—	
	V.2a Transfer	—	
	V.3 Analyze	—	
	V.3a Evaluate	—	
	V.4 Create	—	
VI Discrete Structures, Logic, Algebra	VI.1 Understand	VI.1.c Explain syntax and semantics of propositional and first-order predicate logic.	4
	VI.2 Apply	VI.2.b Confidently handle linear maps in diverse contexts and do calculations on their matrix notation. Solve and interpret linear systems of equations..	6
	VI.2a Transfer	—	
	VI.3 Analyze	—	
	VI.3a Evaluate	—	
	VI.4 Create	—	
VII Formal Languages and Automata	VII.1 Understand	VII.1.a Explain basic concepts of describing formal languages in declarative form or using grammars. Explain the ordering of language into the Chomsky hierarchy and related automata models.	5

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	VII.2 Apply	VII.2.a Define grammars, regular expressions, and automata for formal languages and transform them into equivalent models.	5
	VII.2a Transfer	—	
	VII.3 Analyze	VII.3.a Sort formal languages into the correct levels of the Chomsky hierarchy.	4
	VII.3a Evaluate	—	
	VII.4 Create	—	
VIII Computer Science as a Discipline	VIII.1 Understand	—	
	VIII.2 Apply	—	
	VIII.2a Transfer	—	
	VIII.3 Analyze	—	
	VIII.3a Evaluate	—	
	VIII.4 Create	—	
IX Computer Science and Society	IX.1 Understand	—	
	IX.2 Apply	IX.2.a Name applications of legal constraints and formulate them as system requirements.	5
		IX.2.b Identify the components of an information system that handle sensitive data.	4
	IX.2a Transfer	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	IX.3 Analyze	—	
	IX.3a Evaluate	—	
	IX.4 Create	—	
X IT Security	X.1 Understand	—	
	X.2 Apply	—	
	X.2a Transfer	—	
	X.3 Analyze	—	
	X.3a Evaluate	—	
	X.4 Create	—	
XI Human-Computer-Interaction	XI.1 Understand	—	
	XI.2 Apply	—	
	XI.2a Transfer	—	
	XI.3 Analyze	—	
	XI.3a Evaluate	—	
	XI.4 Create	—	
XII Modeling	XII.1 Understand	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XII.2 Apply	XII.2.a Abstract tasks and system properties to their conceptual core and describe them completely and precisely with a model using formal or informal notation.	8
	XII.2a Transfer	—	
	XII.3 Analyze	—	
	XII.3a Evaluate	—	
	XII.4 Create	—	
XIII Programming Languages and Methods	XIII.1 Understand	XIII.1.b Differentiate programming paradigms like imperative, object-oriented, functional, and logic programming and explain them using language constructs.	6
	XIII.2 Apply	XIII.2.a Develop programs in diverse programming language and programming paradigms using adequate techniques to address algorithmic or datastructure-oriented tasks.	12
	XIII.2a Transfer	—	
	XIII.3 Analyze	XIII.3.a Learn on their own new programming languages representing the learned paradigms and draw relations to prior knowledge.	4
	XIII.3a Evaluate	—	
	XIII.4 Create	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
XIV Project and Team Skills	XIV.1 Understand	—	
	XIV.2 Apply	—	
	XIV.2a Transfer	—	
	XIV.3 Analyze	—	
	XIV.3a Evaluate	—	
	XIV.4 Create	—	
XV Computer Networks and Distributed Systems	XV.1 Understand	—	
	XV.2 Apply	XV.2.a Program secure and efficient network communication.	4
	XV.2a Transfer	—	
	XV.3 Analyze	—	
	XV.3a Evaluate	—	
	XV.4 Create	—	
XVI Software Engineering	XVI.1 Understand	—	
	XVI.2 Apply	XVI.2.a Use modeling (analysis, architecture, designs, patterns) for common cases.	10
	XVI.2a Transfer	XVI.2a.a Systematically plan and create smaller software systems in a team.	5

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XVI.3 Analyze	XVI.3.a Judge the appropriateness of a process model, a notation, or a method for a categorized software system or a categorized task.	4
	XVI.3a Evaluate	—	
	XVI.4 Create	—	
XVII Probability Theory and Statistics	XVII.1 Understand	XVII.1.a Describe and explain the term probability and its mathematical implementation using an example.	5
	XVII.2 Apply	XVII.2.a Select and calculate appropriate metrics and procedures to characterize empirical data.	4
		XVII.2.b Master the handling of fundamental discrete and continuous distributions and apply them to simple contexts.	4
	XVII.2a Transfer	—	
	XVII.3 Analyze	—	
	XVII.3a Evaluate	—	
	XVII.4 Create	—	
XVIII Non-Cognitive Competences	XVIII.A Self-Control Competence	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XVIII.B Cooperation Skills	XVIII.B.a Willingness to help and cooperate	4
		XVIII.B.c Communicative Skills	9
		XVIII.B.d Willingness to discuss topics related to computer science	5
		XVIII.B.e Ability to present topics related to computer science	11
		XVIII.B.h Ability and Willingness to enter into and uphold agreements	5
	XVIII.C Learning Skills	XVIII.C.d Ability to self-organize learning processes and for self-directed learning	6
	XVIII.D Media Skills	—	
	XVIII.E Reading and Writing Skills (Scientific Writing)	XVIII.E.a Ability to research references	4
		XVIII.E.b Ability to sensibly structure topics related to computer science	5
	XVIII.F Attitude	—	
	XVIII.G Empathy	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XVIII.H	Motivational and Volitional Skills	—
	XVIII.I	Learning Motivation	—
	XVIII.J	Effort and Engagement	—

B. Competences filtered out from the GI catalog as per RQ2

Table S2: Competences from the GI model which were identified in less than 4 module catalogs and are thus not seen as relevant. If a competence has been extended during the coding process, this table only presents the extended variant, which will be highlighted in **purple**. Extensions to the original are marked in *italics*, while removed wordings are ~~struck out~~.

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
I Algorithms and Data Structures	I.1 Understand	I.1.b Explain a proof of correctness by means of loop invariants. <i>Students are able to explain concepts for formal proof of correctness, for example by means of loop invariants or diagonal argument.</i>	2
		I.1.e Explain basic examples of concurrent algorithms.	1
	I.2 Apply	I.2.b Execute typical actions (search, insert, delete) on exemplary data structures.	2
		I.2.c Describe the runtime of recursive algorithms by means of recurrence relations and transform said relations into their closed form.	1
	I.2a Transfer	—	
	I.3 Analyze	I.3.b Estimate the general algorithmic complexity of simple problems.	1
		I.3.c Compare algorithms and data structures based on their asymptotic runtime.	1
	I.3a Evaluate	I.3a.a Implement and measure the runtime of simple applications and compare the measurements against theoretical assumptions.	3
	I.4 Create	—	
II Numerical Analysis	II.1 Understand	—	

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
	II.2 Apply	II.2.b Select and use suitable approximation techniques to solve nonlinear equations.	0
		II.2.c Approximate functions through series representation.	1
	II.2a Transfer	—	
	II.3 Analyze	II.3.a Analyze the applicability and precision (problems with limited number of digits, error propagation) implementation of simple numeric algorithms.	0
		II.3.b Use libraries that implement numeric solution methods.	0
	II.3a Evaluate	—	
	II.4 Create	—	
	III Operating Systems		
	III.1 Understand	—	
	III.2 Apply	III.2.a Utilize file systems and parallel processes in programs.	0
		III.2.b Create software specific to operating systems.	1
	III.2a Transfer	III.2a.a Conceptualize and realize parallel programs including several methods of synchronization.	1
	III.3 Analyze	III.3.a Evaluate future developments in the field of operating systems.	0
		III.3.b Evaluate security concepts of operating systems.	0
	III.3a Evaluate	III.3a.a Categorize current operating systems based on their concepts.	0
	III.4 Create	—	

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
IV Databases and Information Systems	IV.1 Understand	IV.1.a Explain fundamental and theoretical concepts of relational database systems and relational query languages using technical terms.	3
		IV.1.b Differentiate different database models.	0
		IV.1.c Explain the general architecture of database management systems with technical terms.	1
		IV.1.d Discuss mechanisms of data protection and societal impact of large-scale data collection.	0
		IV.1.e Explain different types of knowledge and differentiate knowledge processing from databases.	0
	IV.2 Apply	IV.2.a Connect object-oriented software applications with a database system.	3
	IV.2a Transfer	IV.2a.b Apply models of knowledge processing to application contexts.	0
	IV.3 Analyze	IV.3.a Evaluate the utility of a database schema for a user.	0
	IV.3a Evaluate	IV.3a.a Evaluate the fit of a relational database system for a given problem.	2
	IV.4 Create	—	
V Digital Logic, Digital Systems, Computer Architecture	V.1 Understand	V.1.a Explain fundamentals of (two-element) boolean algebra and normal forms.	3
		V.1.d Explain technical implementations of circuits including their (dis-)advantages.	2

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		V.1.e Explain design methods for the register transfer level.	0
		V.1.h Explain and compare different memory technologies.	3
	V.2 Apply	V.2.a Transform numbers between different representations.	1
		V.2.b Do calculations with binary numbers.	1
		V.2.c Evaluate and apply mathematical models and methods for the design of circuits.	3
		V.2.d Write simple programs in Assembler language.	3
		V.2.e Calculate physical addresses given segmented virtual addresses and a page table.	0
	V.2a Transfer	—	
	V.3 Analyze	V.3.a Analyze circuit designs with regards to given design goals.	3
	V.3a Evaluate	V.3a.a Evaluate cost and power of computing systems with regards to given requirements.	1
	V.4 Create	—	
	VI Discrete Structures, Logic, Algebra	VI.1 Understand	
		VI.1.a Explain and give examples for fundamental algebraic structures (groups, rings, quivers, vector spaces) and their relevance for mathematics and computer science.	3

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		VI.1.b Identify applications of boolean algebra in computer science, especially represent circuit elements with methods of boolean algebra.	1
		VI.1.d Explain the application of insights from elementary number theory to computer science, especially cryptography.	1
		VI.1.e Describe and explain concepts and applications of graph theory as well as the most important graph algorithms through an example.	1
	VI.2 Apply	VI.2.a Use sets, relations, and functions and operations on those in diverse contexts.	3
		VI.2.c Apply combinatorial methods to approximate enumerable problems.	0
		VI.2.d Apply modular arithmetic to solve discrete equations.	0
		VI.2.e Model and solve problems with practical relevance (planarity, coloration, shortest path, max flow, matching) with methods pertaining to graph theory.	1
	VI.2a Transfer	VI.2a.a Transform simple applications into logical propositions or predicates and analyze them using logical methods.	3
		VI.2a.b Learn abstract terms and fundamental techniques and procedures on their own.	1
	VI.3 Analyze	VI.3.a Analyze formal problems in simple contexts and apply proof techniques to verify them.	1

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
	VI.3a Evaluate	VI.3a.a Apply methods of discrete mathematics in simple contexts and evaluate their results.	0
	VI.4 Create	—	
VII Formal Languages and Automata	VII.1 Understand	VII.1.b Trace transformations between different representations.	3
		VII.1.c Explain Turing machines, computability, and non-determinism.	3
		VII.1.d Explain the computability of functions and formal languages through Turing machines and trace them for simple examples.	3
	VII.2 Apply	VII.2.b Prove equivalence of different representations.	2
		VII.2.c Apply non-determinism to achieve more effective automata.	0
		VII.2.d Judge the computability of simple examples.	3
	VII.2a Transfer	VII.2a.a Use parser generators and lexers.	1
	VII.3 Analyze	VII.3.b Evaluate and possibly optimize models.	1
	VII.3a Evaluate	—	
	VII.4 Create	—	
VIII Computer Science as a Discipline	VIII.1 Understand	VIII.1.a Describe the structure of the discipline and the typical types of research problems of its sub-disciplines and related disciplines.	2
		VIII.1.b Assign specific concepts and their presentation to sub-disciplines of computer science.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		VIII.1.c Classify information systems based on their task, functionality, and architecture.	0
		VIII.1.d List notable elements of the history of computer science and put them into historical context, especially with regards to functionality and architecture of early computing machinery and the specifics of programming languages.	0
	VIII.2 Apply	VIII.2.a Recognize and differentiate fundamental phenomena and concepts of computer science and apply them to diverse contexts, e.g., abstraction/modeling, modularization and hierarchization, syntax and semantics, recursion, non-determinism, concurrency.	2
		VIII.2.b Deconstruct a (research) problem of computer science, assign its parts to the relevant sub-disciplines, and synthesize the results.	0
	VIII.2a Transfer	VIII.2a.a Generate concrete computer science questions for tasks in a company or organization context and map these questions to their relevant sub-disciplines of computer science.	0
	VIII.3 Analyze	VIII.3.a Analyze the applicability of fundamental concepts of the sub-disciplines of computer science as strategies to solve problems and select fitting concepts.	0
	VIII.3a Evaluate	—	
	VIII.4 Create	—	

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
IX Computer Science and Society	IX.1 Understand	IX.1.a Explain fundamental concepts of data protection law; explain measures to protect personal data.	2
		IX.1.b Explain fundamental concepts of intellectual property (copyright law, patent law) and free-culture movement.	0
		IX.1.c Explain fundamental concepts of computer criminal law.	0
		IX.1.d Explain basic aspects of information economy and their resulting implications.	0
		IX.1.e Explain basic concepts of computer science professional ethics; explain the concepts Responsibility, Value, Dilemma.	3
	IX.2 Apply	IX.2.c Identify types of licenses in software systems.	0
		IX.2.d Explain protection mechanisms for an information system and the data it processes.	2
		IX.2.e Identify relevant aspects of information economy in current developments in computer science.	0
		IX.2.f State reasons for a professional code of ethics and identify dilemmas pertaining to professional ethics.	0
	IX.2a Transfer	IX.2a.a Identify aspects in events which are relevant to society as a whole that pertain to computer science.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
	IX.3 Analyze	IX.2a.b Identify challenges of professional ethics relevant to the work of computing professionals.	0
		IX.3.a Analyze interactions between information systems and relevant legal requirements.	0
		IX.3.b Analyze, evaluate, and possibly enhance requirements and user agreements of information systems with regards to legal requirements.	0
		IX.3.c Analyze, explain, and evaluate the actions of relevant actors in computing contexts which are relevant to society as a whole.	0
		IX.3.d Analyze and judge dilemmas of professional ethics.	0
	IX.3a Evaluate	IX.3a.a Analyze and judge computing aspect in events which are relevant to society as a whole.	2
		IX.3a.b Evaluate the responsibility of computer science towards society.	2
		IX.3a.c Recognize and evaluate the potential change of societal values triggered by information systems.	3
	IX.4 Create	—	
X IT Security	X.1 Understand	X.1.a Explain the semantics of security (security vs. safety) of secure information systems. Explain weaknesses and typical attacks.	1
		X.1.b Conduct a simple security analysis of a system. Understand security management systems including related processes.	1

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		X.1.c Discuss the importance of information security.	0
	X.2 Apply	X.2.c Employ concepts and techniques to increase security, especially which conservation goals can be reached with which techniques. Use typical attack vectors and tools.	2
		X.2.b Suggest improvements to the security of an information system. Record simple information system infrastructures with regards to security management.	3
	X.2a Transfer	X.2a.a Apply the mechanisms of IT security to simple scenarios.	1
	X.3 Analyze	X.3.a Question properties and limitations of security concepts, combine different concepts in a sensible manner, and evaluate the security of complex systems.	0
	X.3a Evaluate	X.3a.a Analyze situations in a company setting that pertain to IT security.	1
	X.4 Create	—	
XI Human-Computer-Interaction	XI.1 Understand	XI.1.a Describe psychological and social concepts of human-computer interaction using adequate terms and contextualize them with regards to human-computer interaction.	0
		XI.1.b Describe the genesis and importance of standards for human-computer interaction and ergonomics for the user.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
	XI.2 Apply	XI.2.a Develop small example GUIs in a user-centric design process with application of software-architectural design patterns (Event Handling, MVC) and test them with regards to relevant criteria stemming from human-computer interaction.	3
	XI.2a Transfer	XI.2a.a Develop GUIs for an information system and induced structures of human-computer interaction in a concrete application context with respect to standards of human-computer interaction that are adequate to address the given problem context and consider diverse user groups.	1
	XI.3 Analyze	XI.3.a Analyze a GUI with regards to induced structures of human-computer interaction and the underlying software design (design patterns).	1
	XI.3a Evaluate	XI.3a.a Evaluate GUIs and induced structures of human-computer interactions of an information system in a concrete application context with regards to standards of human-computer interaction and possibly develop design recommendations.	1
	XI.4 Create	—	
XII Modeling	XII.1 Understand	XII.1.a Interpret and explain the formal description of tasks and system properties in fundamental calculi, e.g., structured sets, terms, algebrae, propositional and predicate logics, as well as graphs.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XII.1.b Interpret and explain the descriptions and meanings of important specialized modeling calculi, e.g., finite automata, context-free grammars, or Petri nets, as well as the UML languages.	3
	XII.2 Apply	XII.2.b Portray diverse views on a system using appropriate models.	3
	XII.2a Transfer	XII.2a.a Describe a given problem of larger complexity using appropriate calculi on an appropriate level of abstraction while using professional tools and evaluating the results.	2
		XII.2a.b Transform general, abstract problems and goals into concrete specifications and analysis questions.	2
	XII.3 Analyze	XII.3.a Analyze formal descriptions of medium-sized tasks and identify the specified properties.	1
		XII.3.b Evaluate whether a candidate model satisfies provided requirements.	0
	XII.3a Evaluate	XII.3a.a Test, evaluate and select modeling calculi and modeling tools with practical relevance with regards to a concrete application.	1
	XII.4 Create	—	
XIII Programming Languages and Methods	XIII.1 Understand	XIII.1.a Interpret the constructs, properties, and typical programming techniques of several common programming languages and explain them in the context of programs .	3

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XIII.1.c Recognize and explain fundamental concepts of programming languages like syntax, name binding, type systems, memory structures, function calls, and parameter passing, in specific [examples (the word is missing in the German original)].	3
		XIII.1.d Explain and contrast the difference between compilation and interpretation as well as the tasks of a runtime system.	2
	XIII.2 Apply	XIII.2.b Recognize and use oft-recurring design and programming patterns.	2
	XIII.2a Transfer	XIII.2a.a Develop a software system of medium complexity and appropriate quality for a specific application using appropriate languages and methods (e.g., during the Bachelor thesis or a practical laboratory for software development).	2
	XIII.3 Analyze	—	
	XIII.3a Evaluate	XIII.3a.a Investigate and judge the fit of diverse programming paradigms and programming languages for diverse applications.	2
	XIII.4 Create	—	
XIV Project and Team Skills	XIV.1 Understand	XIV.1.a Explain fundamental terms of project management (project plan, work package, dependency between work packages).	0
		XIV.1.b Describe and explain the artifacts (specification, design, manual) and typical processes of a computer science project.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XIV.1.c Describe and explain mechanism of quality assurance.	0
	XIV.2 Apply	XIV.2.a Independently plan work packages, complete them in time, and document them.	0
		XIV.2.b Handle a repository for version management.	0
		XIV.2.c Read source code written by others, recognize design concepts applied therein and apply changes.	0
	XIV.2a Transfer	XIV.2a.a Create or substantially contribute to artifacts of software development for specific requirements of an application that is to be developed.	0
		XIV.2a.b Design sub-modules and implement them in the context of the larger software. Apply successful strategies for quality assurance, especially error handling, unit tests, and reviews, in the project context.	0
		XIV.2a.c Develop and employ self-competences like commitment, discipline, punctuality, willingness to compromise, and responsibility in a manner serving project work.	2
	XIV.3 Analyze	XIV.3.a Recognize intersection to the work packages of other team members, label problems, and independently make agreements.	0
	XIV.3a Evaluate	XIV.3a.a Judge the quality of artifacts in the context of software reviews.	0
		XIV.3a.b In the context of projects problems concerning planning and feasibility are recognized and measures suggested.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XIV.3a.c Recognize conflicts in a team and apply strategies for conflict resolution.	0
		XIV.3a.d Communicate with clients and other stakeholders.	0
	XIV.4 Create	—	
	XV Computer Networks and Distributed Systems	XV.1.a Explain the importance of layer models and the tasks and functionality of the ISO/OSI model as well as the most important service agents of each layer.	3
		XV.1.b Explain how the internet works fundamentally and in endpoint devices.	0
		XV.1.c State the concepts of the protocols TCP, IP, http, and SMTP and trace their functionality, e.g., with message sequence charts.	1
		XV.1.d Explain the specification of middleware and the meaning of transparency.	0
		XV.1.e Differentiate sockets and remote procedure calls.	0
		XV.1.f Explain requirements for important protocols, e.g., regarding synchronization and mutual exclusion, consistency, data replication, fault tolerance, and security.	3
	XV.2 Apply	XV.2.b Calculate bandwidth for diverse media.	0
		XV.2.c Calculate data rate for some protocols.	0

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XV.2.d Use distributed systems on different layers, e.g., the application and transport layers.	2
	XV.2a Transfer	XV.2a.a Program simple internet applications.	2
		XV.2a.b Decide which network technologies should be employed in the different layers for a given application problem.	0
	XV.3 Analyze	XV.3.a Evaluate protocols for network communication.	0
	XV.3a Evaluate	XV.3a.b Judge diverse architectures for distributed systems for their applicability for different requirement profiles.	3
	XV.4 Create	—	
XVI Software Engineering	XVI.1 Understand	XVI.1.a Explain fundamental concepts and techniques of large-scale software creating and software creation in teams using technical terms.	3
		XVI.1.b Contrast different processes/procedures, e.g., waterfall and iterative models.	2
		XVI.1.c Explain diverse notations, e.g., UML, for the modeling of software systems.	1
		XVI.1.d Explain tasks and typical processes of management and quality assurance of software projects.	2
	XVI.2 Apply	XVI.2.b Understand the terminology of a user using appropriate methods and condense it into a project ontology.	1

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XVI.2.c Use quality assurance, e.g., reviews, metrics, and automated tests.	9
	XVI.2a Transfer	—	
	XVI.3 Analyze	—	
	XVI.3a Evaluate	XVI.3a.a Evaluate the applicability of a design/architecture/test method for a given specification.	2
	XVI.4 Create	—	
XVII Probability Theory and Statistics	XVII.1 Understand	XVII.1.b Explain the importance of the law of large numbers and central limit theorem for probability theory and statistics.	0
		XVII.1.c Describe and explain concepts of constructing estimators (maximum likelihood estimation, confidence intervals) and hypothesis testing using a given example.	0
	XVII.2 Apply	—	
	XVII.2a Transfer	XVII.2a.a Use hypothesis tests and parameter estimation to analyze univariate data in simple contexts and interpret their results.	2
	XVII.3 Analyze	XVII.3.a Design and analyze stochastic models to describe simple contexts. Understand probabilistic methods and be able to apply them.	1
	XVII.3a Evaluate	—	
	XVII.4 Create	—	

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
XVIII Non-Cognitive Competences	XVIII.A Self-Control Competence	XVIII.A.a Commitment	0
		XVIII.A.b Discipline	0
		XVIII.A.c Punctuality	3
		XVIII.A.d Willingness to Compromise	0
		XVIII.A.e Taking Responsibility	2
		XVIII.A.f Patience	0
		XVIII.A.g Self Control	3
		XVIII.A.h Diligence	0
		XVIII.A.i Goal Orientation	1
		XVIII.A.j Motivation	1
		XVIII.A.k Perception	0
		XVIII.A.l Perseverance	1
	XVIII.B Cooperation Skills	XVIII.B.b Language skills	2
		XVIII.B.f Ability and willingness to pass on computer science knowledge	1
		XVIII.B.g Ability and willingness to constructive criticism	1

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XVIII.B.i Willingness to hold up agreements	1
		XVIII.B.j Willingness to accept the ideas of others	1
	XVIII.C Learning Skills	XVIII.C.a Ability and willingness to lifelong learning	0
		XVIII.C.b Ability and willingness to problem-oriented learning	1
		XVIII.C.c Ability and willingness to cooperative learning	3
	XVIII.D Media Skills	XVIII.D.a Use of problem-oriented learning and development environments	0
		XVIII.D.b Use of tools for scientific writing	0
		XVIII.D.c Use of tools for presenting research results	0
	XVIII.E Reading and Writing Skills (Scientific Writing)	XVIII.E.c Ability to delineate own and foreign ideas (avoid plagiarism)	3
	XVIII.F Attitude	XVIII.F.a Affinity for computer science problems	3
		XVIII.F.b Willingness to tackle computer science challenges	2
		XVIII.F.c See value in social and communicative skills	1
	XVIII.G Empathy	XVIII.G.a Ability to change roles and perspectives	1

Subject Area	Level of Competency	Learning Outcome	Number of Syllabi
		XVIII.G.b Ability to put oneself into the position of someone outside of the field	0
		XVIII.G.c Recognize issues of persons outside of computer science	1
	XVIII.H Motivational and Volitional Skills	XVIII.H.a Openness to new ideas and requirements	1
		XVIII.H.b Willingness to try new and unknown approaches	0
		XVIII.H.c Ability to criticize unthinking application of recipe-like approaches	1
	XVIII.I Learning Motivation	XVIII.I.a Willingness to gain new knowledge and skills pertaining to computer science	1
		XVIII.I.b Willingness to complete tasks pertaining to computer science	1
	XVIII.J Effort and Engagement	XVIII.J.a Sense of (moral) obligation to complete tasks pertaining to computer science	0
		XVIII.J.b Perseverance when dealing with tasks pertaining to computer science	1

C. New Competences as per RQ3

Table S3: Newly identified competences which were found in at least four distinct curricula. Competences and topics which address a competence in the GI catalog but are given for a different cognitive process are highlighted in red ; the related entry of the GI catalog is specified via index number (see Tables S1, S2).

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
I Algorithms and Data Structures	I.* No Level	I.*.a Fundamental terms.	8	I.1.a, I.1.c, I.1.d
		I.*.b Data structures and abstract data types.	13	I.1.d
		I.*.c Algorithm design.	10	I.1.c
		I.*.d Algorithm analysis.	12	I.2.c, I.3.a-c
		I.*.e Applications.	11	I.2.a-b, I.2a.a
		I.*.f Theoretical aspects.	4	I.1.a-b, I.2.c, I.3.b
	I.0 Remember	I.0.a Know the principle of complexity, common complexity classes, and appropriate methods to to determine the complexity of an algorithm or a data structure.	6	I.3.a-b
		I.0.b Know fundamental abstract data types and related data structures as well as example applications and related algorithms.	5	I.1.d, I.2.b

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
		I.0.c Know common paradigms for algorithm design (dynamic programming, greedy, divide-and-conquer, approximation, etc.).	4	I.1.c
	I.1 Understand	—		
	I.2 Apply	I.2.d Can employ principles of algorithm design to solve a problem.	4	I.2a.a
		I.2.e Can implement algorithms and data structures in a programming language.	6	I.3a.a
	I.2a Transfer	I.2a.b Can execute proofs of correctness for algorithms.	4	I.1.b
	I.3 Analyze	—		
	I.3a Evaluate	—		
	I.4 Create	I.4.a Can design their own algorithms to solve a given problem.	6	—
II Numerical Analysis	II.* No Level	II.*.a Fundamental concepts of analysis.	6	II.1.a
		II.*.b Series and Sequences.	10	II.2.c
		II.*.c Relations and Function.	12	I.1.a
		II.*.d Integral Calculus.	8	I.2.a
		II.*.e Differential Calculus.	10	I.2.a
		II.*.f Multivariate functions.	7	—
	II.0 Remember	II.0.a Know fundamental principles of analysis.	4	—

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	II.1 Understand	—		
	II.2 Apply	—		
	II.2a Transfer	—		
	II.3 Analyze	—		
	II.3a Evaluate	—		
	II.4 Create	—		
III Operating Systems	III.* No Level	III.*.a Workings of operating systems.	10	III.1.a
		III.*.b Specific Systems (Real Time, Embedded Systems, UNIX).	4	—
		III.*.c Interfaces between services and operating systems.	9	III.1.c
		III.*.d Historical developments and current systems.	7	—
		III.*.e Processes and process management.	12	III.1.a
	III.0 Remember	—		
	III.1 Understand	—		
	III.2 Apply	—		
	III.2a Transfer	—		
	III.3 Analyze	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	III.3a Evaluate	—		
	III.4 Create	—		
IV Databases and Information Systems	IV.* No Level	IV.*.a Data base models.	8	IV.1.b
		IV.*.b Relational models.	10	IV.1.a
		IV.*.c Handling queries.	5	—
		IV.*.d Development of database applications.	7	IV.2.a
		IV.*.e Query languages.	9	IV.2.c
		IV.*.f Fundamental database concepts.	10	IV.1.a
		IV.*.g Advanced Database concepts.	5	—
	IV.0 Remember	IV.0.a Know structure and function of multiple database models.	6	IV.1.b
	IV.1 Understand	—		
	IV.2 Apply	IV.2.d Use transactions for database interactions.	4	—
	IV.2a Transfer	—		
	IV.3 Analyze	—		
	IV.3a Evaluate	—		
	IV.4 Create	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
V Digital Logic, Digital Systems, Computer Architecture	V.* No Level	V.*.a Digital systems.	10	—
		V.*.b Modern computer architectures.	13	V.1.f
		V.*.c Different levels of abstraction.	4	—
		V.*.d Boolean algebra.	6	V.1.a
		V.*.e Micro controllers and micro processors.	5	—
		V.*.f Assembler programming.	4	V.2.d
		V.*.g Processor architectures.	7	—
		V.*.h Circuits.	6	V.1.c
		V.*.i Electrical engineering.	4	—
		V.*.j Handling Data.	8	V.2.a, V.2.b
	V.0 Remember	—		
	V.1 Understand	V.1.i Describe the functionality of common circuit elements and electrical elements.	5	V.3.a
		V.2.f Construct digital systems and programmable circuits.	4	V.3.a
	V.2a Transfer	—		
	V.3 Analyze	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	V.3a Evaluate	—		
	V.4 Create	—		
VI Discrete Structures, Logic, Algebra	VI.* No Level	VI.*.a Set theory.	11	VI.2.a
		VI.*.b Combinatorics.	8	VI.2.c
		VI.*.c Algebraic structures.	9	VI.1.a
		VI.*.d Linear algebra.	11	VI.2.b
		VI.*.e Number theory.	9	VI.1.d
		VI.*.f Logic.	10	VI.1.c
		VI.*.g Analytical geometry.	4	—
	VI.0 Remember	VI.0.a Know fundamental concepts (vectors, vector spaces, bases, scalars, functions) of linear algebra.	4	VI.2.b
	VI.1 Understand	VI.1.f Explain concepts of linear algebra (vectors and vector space, bases, scalars, functions, linear dependence, transformation).	4	VI.2.b
	VI.2 Apply	—		
	VI.2a Transfer	—		
	VI.3 Analyze	—		
	VI.3a Evaluate	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	VI.4 Create	—		
VII Formal Languages and Automata	VII.* No Level	VII.*.a Grammars.	9	VII.1.a, VII.2.a
		VII.*.b Technology for simulations.	7	—
		VII.*.c Hard problems, completeness and reduction.	5	VII.3.a
		VII.*.d Computable set, Computability, computably enumerable set.	8	VII.1.c, VIII.1.d
		VII.*.e Program description and Hoare logic.	5	—
		VII.*.f algorithmic model of computation.	4	—
		VII.*.g Automata models.	10	VII.1.a-d, VII.2.c
		VII.*.h Formalization of problems	5	—
		VII.*.i Families of languages and complexity	7	VII.3.a
	VII.0 Remember	VII.0.a Know formal syntax descriptions and methods to formalize problems.	6	—
	VII.1 Understand	—		
	VII.2 Apply	VII.2.e Create models of automata to formalize problems	5	—

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	VII.2a Transfer	—		
	VII.3 Analyze	—		
	VII.3a Evaluate	—		
	VII.4 Create	—		
VIII Computer Science as a Discipline	VIII.* No Level	VIII.*.a History of the discipline.	5	VIII.1.d
	VIII.0 Remember	—		
	VIII.1 Understand	—		
	VIII.2 Apply	—		
	VIII.2a Transfer	—		
	VIII.3 Analyze	—		
	VIII.3a Evaluate	—		
	VIII.4 Create	—		
IX Computer Science and Society	IX.* No Level	IX.*.a Social influence	5	IX.2a.a, IX.3.c, IX.3a.a-c
		IX.*.b Terms and concepts of ethics.	7	IX.1.e, IX.2.f

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
		IX.*.c Common laws.	6	IX.1.b, IX.1.c
	IX.0 Remember	—		
	IX.1 Understand	IX.1.f Describe interplay of computer systems and aspects of society.	4	IX.2a.a, IX.3.c, IX.3a.a-c
	IX.2 Apply	—		
	IX.2a Transfer	—		
	IX.3 Analyze	—		
	IX.3a Evaluate	IX.3a.d Evaluate the social effects of a technology.	4	IX.3a.c
	IX.4 Create	—		
	X IT Security	X.*.a Cryptography	4	—
		X.*.b Threads and handling threads	5	X.1.a-b, X.2.a, X.3aa.a
		X.*.c Security of operating systems	5	III.3.b
	X.0 Remember	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	X.1 Understand	—		
	X.2 Apply	—		
	X.2a Transfer	—		
	X.3 Analyze	—		
	X.3a Evaluate	—		
	X.4 Create	—		
XI Human-Computer-Interaction	X.* No Level	XI.*.a User interaction	4	XI.2.a, XI.2a.a
		XI.*.b Usability and ergonomics	5	XI.1.b
	XI.0 Remember	—		
	XI.1 Understand	—		
	XI.2 Apply	—		
	XI.2a Transfer	—		
	XI.3 Analyze	—		
	XI.3a Evaluate	—		
	XI.4 Create	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
XII Modelling	XII.* No Level	XII.*.a Creating Models	11	XII.1.a, XII.2.a-b, XII.2a.a-b, XII.3.a
		XII.*.b Model based analysis	4	XII.1.a
	XII.0 Remember	XII.0.a Can name common methods for modeling.	4	XII.1.b
	XII.1 Understand	—		
	XII.2 Apply	—		
	XII.2a Transfer	—		
	XII.3 Analyze	—		
	XII.3a Evaluate	—		
	XII.4 Create	—		
XIII Programming Languages and Methods	XIII.* No Level	XIII.*.a Programming paradigm.	13	XIII.1.b
		XIII.*.b Work in multiple languages.	5	XIII.3.a
		XIII.*.c Data handling.	9	XIII.2.a
		XIII.*.d Fundamental principles.	12	XIII.1.c
		XIII.*.e Properties of programs.	11	—

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
		XIII.*.f Syntax, semantic and pragmatism of languages.	5	XIII.1.c
		XIII.*.g Advanced concepts and abilities.	5	—
		XIII.*.h Compiler, Interpreter, Preprocessor.	5	XIII.1.d
		XIII.*.i Properties of specific languages.	8	XIII.3.a
	XIII.0 Remember	XIII.0.a Can name common programming paradigm.	6	XIII.1.b
	XIII.1 Understand	—		
	XIII.2 Apply	XIII.2.c Use a modern development environment.	5	—
	XIII.2a Transfer	—		
	XIII.3 Analyze	—		
	XIII.3a Evaluate	—		
	XIII.4 Create	XIII.4.a Create programmes to solve problems.	5	XIII.2a.a
	XIV Project and Team Skills	XIV.*.a Project management.	7	XIV.1.a
		XIV.0 Remember	—	
		XIV.1 Understand	—	
		XIV.2 Apply	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	XIV.2a Transfer	—		
	XIV.3 Analyze	—		
	XIV.3a Evaluate	—		
	XIV.4 Create	—		
XV Computer Networks and Distributed Systems	XV.* No Level	XV.*.a Fundamental of computer networks.	9	XV.1.a
		XV.*.b Network hierarchies.	6	—
		XV.*.c Common protocols.	8	XV.1.c
		XV.*.d Topology and Addressing	8	—
		XV.*.e Client-Server-Application	5	—
		XV.*.f Management of network infrastructure	8	—
		XV.*.g ISO-OSI reference model	7	XV.1.a
		XV.*.h Distributed systems	4	XV.2.c, XV.3a.a
	XV.0 Remember	XV.0.a Name the layers of the ISO-OSI reference model and the elements of the TCP/IP stack.	5	XV.1.a
		XV.0.b Know the format of IP addresses and name concepts and elements of routing.	4	—

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
		XV.0.c Name common protocols of the application layer.	5	XV.1.c
		XV.0.d Name important concepts.	4	—
	XV.1 Understand	—		
	XV.2 Apply	XV.2.e Configure and manage networks.	4	—
	XV.2a Transfer	—		
	XV.3 Analyze	—		
	XV.3a Evaluate	—		
	XV.4 Create	—		
XVI Software Engineering	XVI.* No Level	XVI.*.a Processes and tools to operate software	5	—
		XVI.*.b Design pattern.	9	XVI.2.a
		XVI.*.c Interfaces and classes.	8	—
		XVI.*.h Quality and quality management.	12	XVI.1.d, XVI.2.c
		XVI.*.e Error handling.	5	—
		XVI.*.f Processes and tools to develop software.	12	XVI.1.b
		XVI.*.g Architecture.	9	XVI.2.a

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	XVI.0 Remember	XVI.0.a Name processes and procedures for software engineering.	4	XVI.1.a
		XVI.0.b List common design patterns like MVC.	4	XVI.2.a
		XVI.0.d Name concepts and methods for software engineering.	4	XVI.1.a
		XVI.0.d List quality concepts and methods for quality assurance.	5	XVI.1.d, XVI.2.c
		XVI.0.e Name constraints of concurrent programs and tools to check for non-functional constraints.	4	—
	XVI.1 Understand	—		
	XVI.2 Apply	XVI.2.d perform a requirement analysis.	4	—
		XVI.2.e Use common tools like development environments and version control systems.	6	—
		XVI.2.f Follow a given development method.	4	—
	XVI.2a Transfer	—		
	XVI.3 Analyze	—		
	XVI.3a Evaluate	—		
	XVI.4 Create	XVI.4.a Develop a software application on their own.	5	XVI.2a.a
XVII Probability Theory and Statistics	XVII.* No Level	XVII.*.a Fundamental concepts.	10	XVII.1.a

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
		XVII.*.b Estimator.	5	XVII.1.c
		XVII.*.c Conditional probability and statistical independence.	10	—
		XVII.*.d Statistical metrics.	10	XVII.2.a
		XVII.*.e Probability distributions.	11	XVII.2.b
		XVII.*.f Combinatorics.	4	—
		XVII.*.g Fundamental theorems.	4	XVII.1.b
		XVII.*.h Statistical data analysis.	4	XVII.2.a
	XVII.0 Remember	XVII.0.a Know essential terms and concepts of probability theory.	5	XVII.1.a
	XVII.1 Understand	—		
	XVII.2 Apply	—		
	XVII.2a Transfer	—		
	XVII.3 Analyze	—		
	XVII.3a Evaluate	—		
	XVII.4 Create	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
XVIII Non-Cognitive Competences	XVIII.A Self-Control Competence	XVIII.A.m Self-directed work	11	
	XVIII.B Cooperation Skills	XVIII.B.k Collaborative work	14	
	XVIII.C Learning Skills	—		
	XVIII.D Media Skills	—		
	XVIII.E Writing and Reading Skills (Scientific Writing)	XVIII.E.d Reading Comprehension	4	
	XVIII.F Attitude	—		
	XVIII.G Empathy	—		
	XVIII.H Motivational and Volitional Skills	—		

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi	GI Learning Outcome
	XVIII.I Learning Motivation	—		
	XVIII.J Effort and Engagement	—		

D. New Subject Domains as per RQ3

Table S4: Newly identified subject areas. For discussion purposes, all identified contents and competences will be listed independently of how often they were found. Contents and Competences that were found in fewer than four curricula will be **grayed out**.

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
XIX Data Science	XIX.* No Level	XIX.*.a Data and data handling	1
		XIX.*.b Methods for generating knowledge	1
	XIX.0 Remember	XIX.0.a Know fundamental data formats and data models.	1
		XIX.0.b Know statistical models.	1
		XIX.0.c Know Methods of data preparation and processing.	1
	XIX.1 Understand	XIX.1.a Explain the term and concept of Data Science.	1
		XIX.1.b Explain methods and processed to describe and interpret data.	1
	XIX.2 Apply	XIX.2.a Process data to prepare a gain of insight.	1
		XIX.2.b Describe the distribution and attributes of data.	1
		XIX.2.c Apply methods of data science for given tasks.	1
		XIX.2.d Use appropriate tools for processing data.	1
	XIX.2a Transfer	—	
	XIX.3 Analyze	—	
	XIX.3a Evaluate	—	

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XIX.4 Create	—	
XX Business Administration	XX.* No Level	XX.*.a Object of experience and object of knowledge.	1
		XX.*.b Constitutive decisions.	1
		XX.*.c Rolls found in a company.	3
		XX.*.d Business Administration as a scientific subject.	1
		XX.*.e Fundamentals of Business Administration.	3
		XX.*.f Privacy and advertisement.	1
		XX.*.g Work Systems Theory.	1
		XX.*.h Interface between Business Administration and Computer Science.	4
		XX.*.i Social environment.	2
	XX.0 Remember	XX.0.a Knows problems and methods of Business Administration.	1
		XX.0.b Know interfaces and interplay to other disciplines, esp. Engineering and Computer Science.	2
		XX.0.c Know object of experience and object of knowledge.	1
		XX.0.d Know goals of companies as well as different departments and processes of companies.	1
		XX.0.e Know principles and conditions of investment.	1

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		XX.0.f Know fundamental sources and principles of private business law.	1
		XX.0.g Know fundamental principles and methods of business processes management.	1
	XX.1 Understand	XX.1.a Explain fundamental concepts of Business Administration.	2
		XX.1.b Differentiate between entrepreneurial and market-oriented questions.	1
		XX.1.c Describe goals, concepts and methods of business management.	2
		XX.1.d Differentiate between different business structures.	2
		XX.1.e Describe processes of production and fabrication.	1
		XX.1.f Explain the function of different departments in a company.	1
		XX.1.g Describe process of investment and related considerations.	1
		XX.1.h Describe concepts and methods of business process management.	1
		XX.1.i Describe the usefulness and the influence of computer systems in a business.	2
		XX.1.j Explain important concepts of private business law.	1
	XX.2 Apply	XX.2.a Apply methods of Business Administrations to fictional and realistic scenarios.	3
		XX.2.b Make decisions of investment considering financial leverage.	2
		XX.2.c Use and administrate computer systems to support companies.	2

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XX.2a Transfer	—	
	XX.3 Analyze	XX.3.a Interpret descriptions of functional specifications.	1
		XX.3.b Analyze economic viability of a suggested project.	1
		XX.3.c Analyze models of business processes.	1
		XX.3.d Determine equipment requirements for a given process environment.	1
	XX.3a Evaluate	XX.3a.a Evaluate methods and mindsets in Business Administration.	1
		XX.3a.b Assess programs for production.	1
		XX.3a.c Assess decisions of management.	1
		XX.3a.d Assess tools regarding their advantages and disadvantages.	1
		XX.3a.e Assess the marketability of a product.	1
		XX.3a.f Assess the distribution of resources for a given context.	1
	XX.4 Create	XX.4.a Create functional specifications.	1
		XX.4.b Create and plan alternative actions for a scenario.	1
XXI Artificial Intelligence	XXI.* No Level	XXI.*.a Definition of fundamental concepts.	3
		XXI.*.b Strategies for problem solving.	1
		XXI.*.c Uncertainty.	1
		XXI.*.d Sample applications	2

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		XXI.*.e Ethical questions.	1
		XXI.*.f Agents.	1
		XXI.*.g Learning strategies.	2
	XXI.0 Remember	XXI.0.a Name goals and applications of Artificial Intelligence.	1
		XXI.0.b Name fundamental technologies.	1
	XXI.1 Understand	XXI.1.a Explain methods of Artificial Intelligence and their areas of application.	1
		XXI.1.b Explain fundamental terms and taxonomies.	1
	XXI.2 Apply	XXI.2.a Implement selected technologies.	1
		XXI.2.b Use different strategies for learning.	1
	XXI.2a Transfer	XXI.2a.a Apply methods on practical examples.	1
	XXI.3 Analyze	—	
	XXI.3a Evaluate	XXI.3a.a Evaluate the usage of different learning strategies for a given scenario.	1
		XXI.3a.b Access applications in different practical examples.	1
	XXI.4 Create	—	
XXII Signal Processing	XXII.* No Level	XXII.*.a Elementary signals and LTI systems	2

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		XXII.*.b Spectral analysis	1
		XXII.*.c Example applications	1
		XXII.*.d Transformations	1
		XXII.*.e Discretization of signals	2
		XXII.*.f Filters	1
		XXII.*.g Recurrence relation	1
		XXII.*.h Signals as random processes	1
		XXII.0 Remember —	
	XXII.1 Under-stand	XXII.1.a Represent Signals as periodical functions.	1
		XXII.1.b Explain the frequency properties of a signal.	1
		XXII.1.c Explain frequency band limit.	1
		XXII.1.d Explain sample theory, sampling and filtering.	1
		XXII.1.e Explain transformation of signals.	1
		XXII.1.f Explain LTI systems.	1
		XXII.2 Apply	
		XXII.2.a Filter signals.	1
		XXII.2.b Determine properties of signals.	1

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		XXII.2.c Sample signals.	1
		XXII.2.d Apply signal transformations (Laplace-, z-, Fourier-transformation).	1
		XXII.2.e Characterize the behavior of linear systems.	1
	XXII.2a Transfer	XXII.2a.a Determine statistical properties of signals.	1
	XXII.3 Analyze	XXII.3.a Interpret statistical properties of signals.	1
	XXII.3a Evaluate	—	
	XXII.4 Create	—	
	XXIII Research Methods	XXIII.* No Level	
		XXIII.*.a Epistemology.	1
		XXIII.*.b Philosophy of science.	3
		XXIII.*.c Fundamentals of research methods.	3
		XXIII.*.d Criteria of scientific studies (e.g, validity, reliability).	1
		XXIII.0 Remem-ber	
		XXIII.0.a State the process of publication in science.	1
		XXIII.0.b List (discipline specific) scientific methods and tools.	2
		XXIII.0.c Name fundamental epistemological foundations of science.	1
		XXIII.0.d Name scientific questions of discipline.	1

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	XXIII.1 Understand	XXIII.1.a Explain scientific methods and concepts outside of ones own subject area.	1
		XXIII.1.b Explain measurement as part of scientific insight gain.	1
		XXIII.1.c Contrast the content from different sources.	1
	XXIII.2 Apply	XXIII.2.a Employ research methods of natural and social science.	3
		XXIII.2.b Adhere to scientific standards.	1
		XXIII.2.c Use different presentation formats.	1
	XXIII.2a Transfer	XXIII.2a.a Use methods of project management for science projects.	1
	XXIII.3 Analyze	XXIII.3.a write protocols of laboratory experiments.	1
	XXIII.3a Evaluate	XXIII.3a.a Reflect on different learning methods.	1
		XXIII.3a.b Judge whether scientific standards were met.	1
		XXIII.3a.c Judge the usage of scientific methods.	1
		XXIII.3a.d do a final inspection of a scientific project.	1
		XXIII.3a.e Evaluate the usage of different presentation formats.	1
	XXIII.4 Create	XXIII.4.a Develop their own learning methods.	1
		XXIII.4.b Develop a rigorous referencing process.	1
		XXIII.4.c Write scientific papers.	1

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
XXIV Robotics	XXIV.* Ungeordet	XXIV.*.a Foundations in physics and math	1
		XXIV.*.b Path planning	1
		XXIV.*.c PID controller	1
	XXIV.0 Remember	—	
	XXIV.1 Understand	—	
	XXIV.2 Apply	XXIV.2.a Use components of intelligent systems.	1
		XXIV.2.b Implement path planning algorithms.	1
	XXIV.2a Transfer	—	
	XXIV.3a Evaluate	—	
	XXIV.4 Create	—	
ICVIII Computer Science (General)	ICVIII.* No Level	ICVIII.*.a Information vs. Knowledge, Representation vs. Meaning.	2
		ICVIII.*.b Technical terms.	3
		ICVIII.*.c Think like computing specialist.	2
		ICVIII.*.d Data oriented approach.	1
		ICVIII.*.e Digital automatization.	1

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		ICVIII.*.f Fundamental knowledge of computers.	4
		ICVIII.*.g Synthesis and Analysis.	1
		ICVIII.*.h Information management.	1
	ICVIII.0 Remember	ICVIII.0.a Name influences of software and hardware.	1
	ICVIII.1 Understand	ICVIII.1.a Contextualize their own activities with the broader context of computer science.	1
	ICVIII.2 Apply	ICVIII.2.a set documents in \LaTeX .	1
	ICVIII.2a Transfer	ICVIII.2a.a Transfer concepts onto new problems.	1
	ICVIII.3 Analyze	—	
	ICVIII.3a Evaluate	—	
	ICVIII.4 Create	—	
ICIX Mathematics (General)	ICIX.* No Level	ICIX.*.a Interplay between Mathematics and Computer Science.	2
		ICIX.*.b Numeracy.	1
		ICIX.*.c Mathematical way of thinking / expression.	6
		ICIX.*.d Mathematical tools for solving formal problems.	1

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
		ICIX.*.e Mathematical modeling.	1
		ICIX.*.f Methods for proofs	11
ICIX.0	Remember	ICIX.0.a Name fundamental mathematical concepts and terms.	2
		ICIX.0.b Name interplay between Mathematics and Computer Science.	2
		ICIX.0.c Name fundamental methods for proofs.	2
ICIX.1	Understand	ICIX.1.a Understand and use mathematical notation and terms.	8
		ICIX.1.b Understand and use proofs and methods for proofs.	6
		ICIX.1.c Grasp mathematical thinking.	3
		ICIX.1.d Explain fundamental methods in mathematics.	3
ICIX.2	Apply	ICIX.2.a Purposefully use fundamental mathematical concepts and structures.	6
		ICIX.2.b describe problems mathematically precise.	8
		ICIX.2.c Use software tools to work on mathematical problems.	1
		ICIX.2.d Use methods for proofs.	7
ICIX.2a	Transfer	ICIX.2a.a Use mathematical methods as a tool to tackle problems and studies in computer science.	3

Subject Area	Level of Competence	Learning Outcome	Number of Syllabi
	ICIX.3 Analyze	ICIX.3.a Grasp new assignments.	1
		ICIX.3.b Familiarize themselves with new formal systems.	1
		ICIX.3.c Formalize plain language problem statements.	1
	ICIX.3a Evaluate	ICIX.3a.a validate formal proofs.	2
		ICIX.3a.b evaluate the usefulness of methods and concepts for a given problem..	2
		ICIX.3a.c critically evaluate the results of a method.	1
	ICIX.4 Create	ICIX.4.a construct proofs.	1