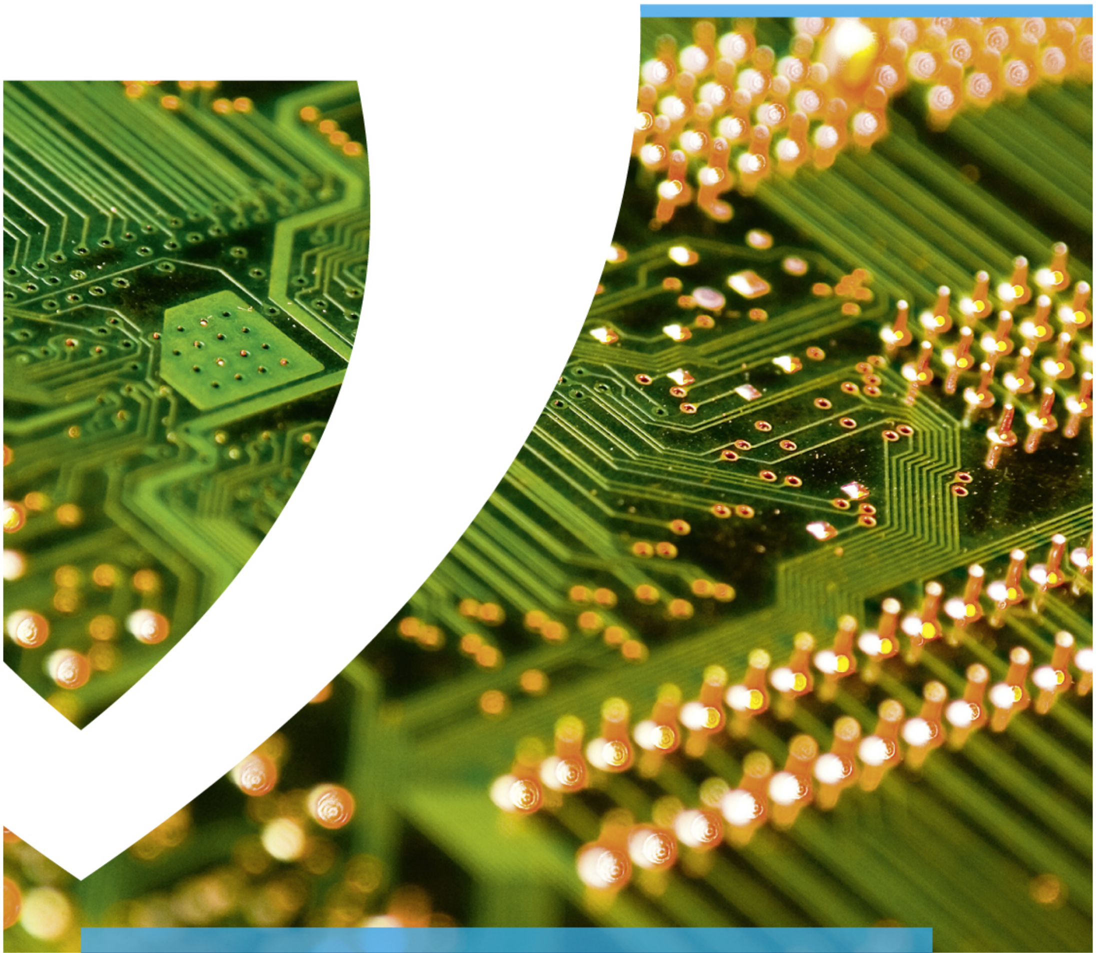
JACOBS UNIVERSITY



School of Engineering and Science

Computer Science (BSc)

Bachelor's Degree Program

Contents

1 Computer Science

1.1 Concept

Students of Computer Science at Jacobs University will receive an education that covers all modern aspects of this challenging and fascinating discipline, plus they will take courses from neighboring disciplines according to their inclinations.

The courses offered in the first year provide a general overview over a broad range of CS topics, as well as an intensive training of programming skills. Only after the first year of study students have to finalize their choice of major, a liberty uncommon at most state universities in Germany. In years two and three, students will on the one hand increasingly specialize within CS, and on the other hand benefit from the transdisciplinary openness that is the hallmark of Jacobs University and take courses from neighboring disciplines, complementing their in-depth CS training. Students, thus, learn to adapt their knowledge and expertise to a variety of tasks. This is an invaluable asset in a student's future career.

Research is an essential part of student education at Jacobs University. Enabled by a very low student to faculty ratio, students will participate in real research work even during bachelor education. Faculty is working in exciting areas of modern computer science, for example: robotics and embedded systems, motion planning, large-scale data bases, computational logic, semantic web, machine learning, computer networks, distributed systems, or visualization and computer graphics.

1.2 Cooperation with Other Universities

The faculty members of the CS Major at Jacobs University have excellent working cooperations with leading Computer Science Departments worldwide. Specifically, students with ambitious academic career plans can be efficiently recommended to renown research labs for their (mandatory) practicals.

Furthermore, the School of Engineering and Science at Jacobs University has an exchange program for second-year CS undergraduate students (or EECS students specializing in CS) with the School of Computer Science at Carnegie Mellon University.

1.3 Job Prospects and Career Options

It is almost needless to say that the job market for computer scientists has been very good in the last few years, and there is no indication that this will change in the foreseeable future. However, there is no single all-round CS education that qualifies a student for every CS career. Quite to the contrary, because of the rapid changes in the field, it is important to focus one's education in subfields of promising future relevance. Furthermore, cross-disciplinary breadth and flexibility, as well as social and work organization skills, become increasingly important. In addition, the required qualification profiles and personal attitudes differ for academic versus industrial careers. The CS program at Jacobs University responds to all of these conditions for a successful career.

The main elements to make this possible are:

- The program concentrates on the sub-disciplines that combine a "mainstream" breadth with a high potential for future innovations. It thus focuses on modern aspects of Artificial Intelligence (including robotics, machine learning, and the semantic web), and also databases and networking technologies. Furthermore, there are close affiliations with other majors, allowing the student to include into their syllabus courses in computational science, bioinformatics, mathematical modeling, and communication technology. Some of the more traditional fields (like compiler design or theory of programming languages) will be limited to short overviews.
- CS faculty was carefully recruited from modern, integrative areas of computer science, and for a record of interdisciplinary research.
- Extensive laboratory courses for practical CS training and the acquisition of social and workflow skills enable the student to work in large-scale joint project teams.
- Jacobs University is generally an exquisite place to acquire great social skills, simply because our undergraduate students come from more than 80 countries and live together in college buildings. This unique circumstance can hardly be over-rated.
- Since mathematical tools are at the core of today's CS applications and research, an early (and tough) systematic training in the relevant mathematical disciplines is ensured.
- The practical aspects of training are deepened in the mandatory 2-3 months industry or academic internship. Students are helped to find a host organization with computer science as its core competency.

As to the academic vs. industrial career decision, there are two principal career options for a student graduating in computer science. The first option is entering a non-academic position directly. In line with the expectations and demands of a majority of potential employers, Jacobs University computer science graduates will be able to start a career in industry / business / the public sector after studying for three years. Our graduates will enter their job at a young age of 21-22, so that there is ample time for job specific training at the future employer.

The alternative option is to continue on the academic track with graduate studies, possibly with a long-term goal of a higher entry qualification for industry positions. The CS major will prepare students for graduate studies, in particular for

- the Smart Systems graduate program in Computer Science at Jacobs University,
- the same subject at other institutions of higher education,
- more business oriented fields for those seeking a management career in high-tech enterprises.

At the time of writing, the fifth generation of Jacobs University EECS undergraduates have just obtained their B.Sc. degrees. The majority went on to study for a Master's degree. The best among them could promptly enlist in the world's leading institutions in their specialization areas (EPF Lausanne, ETH Zürich, Urbana-Champaign, Cornell, University of Montreal), witnessing the high quality of the training they received at Jacobs University. The ones who opted for a direct industrial career faced no difficulties in finding qualified employments, a gratifyingly large portion among them in the wider Bremen area.

Students with a degree in CS will find themselves at the very heart of modern developments in industry and commerce. There is hardly a field which has not been affected by the revolutionary development of information technology and micro-electronics, which has for example resulted in the ubiquitous use of computers, the omni-presence of telecommunication devices and the rapidly expanding use of the numerous network-based services offered. The pace of change will not slow down. It is predicted that the share of the electronics and information technology industries in the gross national product will further increase. Hence there is, from a national economics perspective, an urgent need for excellent CS graduates, and it appears certain that job prospects will remain excellent for at least more than a decade.

1.4 Curriculum Development Process

This curriculum has been developed from the CS specialization of the existing EECS curriculum, which has been very successfull over the last few years. The greatest challenge for curriculum design was to bind into a short 3-year program all the requirements of practical and theoretical training, social and work organization skills, as well as interdisciplinary openness, coverage of all essential CS basics, and offering a broad scope of specialization courses. This led to the following general design of the program:

- In the first year, the program offers (i) overview lectures and lab courses that expose the student to (almost) all CS aspects at a simple level, (ii) a training in standard programming languages, (iii) an essential mathematical training in calculus and linear algebra, (iv) introductory courses and labs from other natural science fields. For (i) – (iii), all courses are mandatory.
- After the first year, the student may ultimatively enroll in CS if s/he meets threshold marks in mandatory courses. A change of major is possible at this point at the latest.
- In the second year, the program offers an in-depth treatment of carefully selected mainstream topics of CS and it enables a first specialization by allowing choices. The programming and mathematical education (probability and statistics, advanced linear algebra, numerics) is continued while leaving room for transdisciplinary courses.
- During the summer break after the second year, a 2-3 month internship in industry or academic labs is mandatory.
- From the second year onward, students are offered ample opportunities to work in (paid) student assistant jobs within funded projects or as teaching assistants.
- In the third year, the program offers a choice of specialization courses that lead the student close to the frontier of research. Again, the program leaves room for transdisciplinary courses. In the second half of the third year, the student has to participate in a guided research project and produce a research report.

The curriculum will be continuously scrutinized as to whether it ensures a sound academic education (i.e., successful applications of students at graduate schools) and also fulfills the expectations of potential employers (i.e., rate to find a job in the field within a few months).

The adaptation of the curricula of the individual lectures will not be left to the instructor of record alone. In yearly reviews, the faculty members of the EECS group agree and commit on necessary changes and updates.

Specifically, the third year specialization courses are directly linked to ongoing research of faculty members. By their nature, they will continuously change, always reflecting the most recent advances in research.

2 Modules: Computer Science

For greater transparency of the logics and as guidance for the (prospective) student, we have structured the respective major programs in terms of modules. A module is defined as a combination of courses (lectures, lab units or other types of courses) interconnected by the same learning goals (Lernziel). Before listing the individual courses and describing their contents, these modules are presented and characterized by the skills and abilities that the student is expected to acquire. But irrespective of this overarching modular structure, the learning progress will be documented with credit points and grades attributed to the individual courses or lab units. This facilitates the control of the student's progress through the student as well as the university on a semester basis, while the modules may extend over a year or, in exceptional cases, even over longer periods. Only the core content of a major program is suited for modularisation. The freely choosable Home School Electives and transdisciplinary courses fall outside this structure.

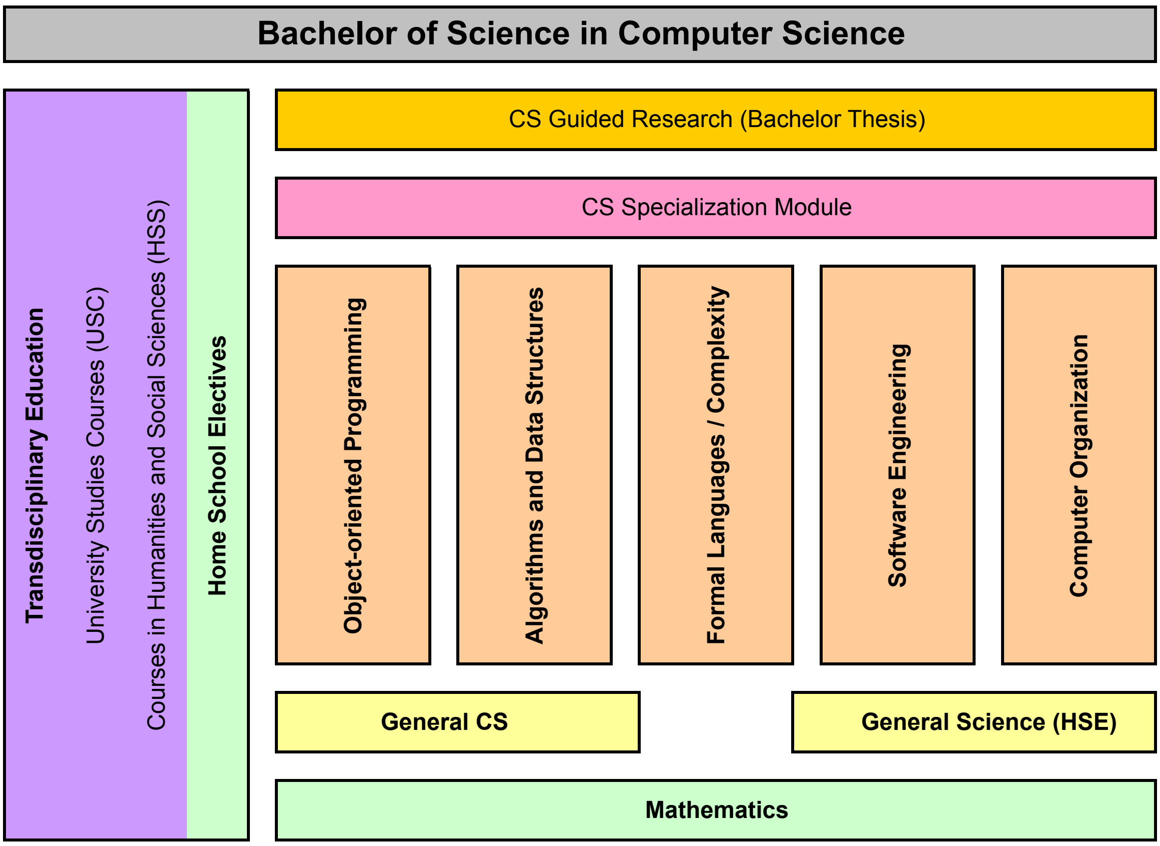


Figure 1: CS Module Structure

Subsequently the individual modules are being defined with respect to learning goals and acquired competencies. The listed course numbers constitute a reference to the individual courses and the descriptions of their contents.

2.1 General Science

Home School Electives and transdisciplinary courses are not listed as modules. In the second year, home school (Engineering and Science, ECs) courses are required which is shown in the figure, but are not separately listed since they are electives. In the 3rd year, all courses are for further specialization in a direction chosen by the student and modules there directly represent single courses. We represent them as a CS specialization module.

120130 – ESM for Computer Science

Short Name: ESM for CS
Semester: 1 – 4
Credit Points: 20 ECTS

General Information Students of Computer Science are required to take four semesters of Engineering and Science Mathematics; the courses listed below are mandatory for the CS major.

Learning goals

- Working skills in differential and integral calculus, linear algebra, probability, and statistics, Fourier methods, and numerics.
- Problem solving skills
- Training in abstract reasoning and symbolic manipulation
- Ability to turn real-world problems into a concise mathematical question
- Ability to interpret mathematical statements back into the problem domain

Courses

120101 ESM 1A – Single Variable Calculus

120112 ESM 2B – Linear Algebra, Fourier, Probability

120201 ESM 3A – Advanced Linear Algebra, Stochastic Processes

120202 ESM 4A – Numerical Methods

Elective – Natural Science Module

Short Name: ModGenSES
Semester: 1 – 2
Credit Points: 15 ECTS

General Information This module includes two first year natural science courses and associated lab units. These are required for all students majoring in the School of Engineering and Science.

Learning goals

- This should offer the student an introduction into other sciences offered within the School of Engineering and Science (SES).

Courses

elective General Engineering and Science Subject I (lecture)

elective General Engineering and Science Subject II (lecture)

elective Natural Science Lab Units Subject I

elective Natural Science Lab Units Subject II

2.2 CS Major

320100 – General Computer Science

Short Name: ModGCScs
Semester: 1 – 2
Credit Points: 15 ECTS

General Information This module familiarizes students with general concepts of Computer Science. Two lecture and two accompanying lab units are provided. The lectures, based on a clear mathematical foundation, introduce abstract and concrete notions of computing machines, information, and algorithms. They also introduce basic concepts of logic, boolean circuits up to very elementary computer architectures. The lab units are more oriented towards the practical side of computer science and provide an introduction to programming as well as a hands-on experience on how to implement basic algorithms and data structures.

Learning goals

- Understanding of the mathematical foundation of computer science and core concepts such as computation or complexity
- Introduction to procedural, object-oriented and functional programming concepts
- Enabling students to solve simple programming problems

Courses

320101 General Computer Science I (lecture)

320102 General Computer Science II (lecture)

320111 Natural Science Lab Unit Programming in C I

320112 Natural Science Lab Unit Programming in C II

320200 – Algorithms and Data Structures

Short Name: ModADScs
Semester: 3
Credit Points: 5 ECTS

General Information Understanding of basic algorithms and data structures and their properties is essential in every computer science program. This module introduces these concepts from the theoretical as well as the practical point of view.

Learning goals

- Understanding of core algorithms and data structures.
- Ability to analyze algorithms or data structures in terms of their complexity.
- Ability to apply data structures and algorithms to problems based on a sound understanding of their properties.

Courses

320201 Algorithms and Data Structures (lecture)

320210 – Formal Languages and Complexity

Short Name: ModFLCcs
Semester: 3 – 4
Credit Points: 10 ECTS

General Information This module deepens the theoretical foundation of computer science by introducing the theory of formal languages, their relationships to automata in more depth. The module also covers first-order logic which is the mathematical basis of many areas in computer science.

Learning goals

- Understanding of the theoretical foundations of computer science.
- Abstract thinking skills.
- Introduction and training of prove techniques.
- Developing an understanding of the fundamental limitations of computational models.

Courses

320211 Formal Languages and Logic (lecture)

320352 Computability and Complexity (lecture)

320240 – OBJECT-ORIENTED PROGRAMMING

Short Name: ModOOcs
Semester: 2 – 3
Credit Points: 10 ECTS

General Information This module introduces object-oriented programming techniques. Concepts such as classes, inheritance, exceptions, templates are introduced using two popular programming languages. The courses combine lecture and lab sessions.

Learning goals

- Understanding of object-oriented programming and the construction of object-oriented class libraries.
- The combination of a lecture and lab sessions allows students to deepen theoretical concepts while obtaining hands-on experience.
- Students develop a self-directed working style in groups after having received the theoretical basis.

Courses

320142 Programming in C++ (lecture / lab)

320341 Programming in Java (lecture / lab)

320220 – SOFTWARE ENGINEERING

Short Name: ModSEcs
Semester: 4
Credit Points: 10 ECTS

General Information The development of large software systems requires to know a set of techniques to support the various stages of a software development project. This module introduces key software engineering topics such as process models, data modeling techniques, object-oriented design techniques and tools relevant for implementation, testing, and verification.

Learning goals

- Familiarity with fundamental software engineering techniques.
- Application of these techniques in a concrete software development project.
- Experience with software development, documentation, and testing tools.
- The combination of a lecture with a lab course allows students to deepen theoretical concepts while obtaining hands-on experience.
- Students develop a self-directed working style in groups after having received the theoretical basis.
- Students practice the documentation of lab results in reports.

Courses

320212 Software Engineering (lecture)

320222 Software Engineering Lab

320230 – Computer Organization

Short Name: ModCORGcs
Semester: 1, 4
Credit Points: 15 ECTS

General Information This module introduces fundamental knowledge about the organization of digital computers. It covers hardware concepts such as instruction sets and processor designs and fundamental principles of memory systems and system busses. The module also covers operating systems, which are complex software systems with non-sequential flows of control implementing several resource management algorithms to make effective use of the hardware components. Operating systems form a good basis to study concurrency and synchronization problems, scheduling algorithms, and resource allocation algorithms in general. The module also introduces core network programming interfaces provided by operating systems.

Learning goals

- Familiarity with computer architectures and fundamentals of basic components of digital computing systems.
- Understanding of core concepts underlying operating systems and data networks.
- Familiarity with the programming abstractions provided by operating systems.
- Ability to write simple concurrent and communicating programs.

Courses

320241 Computer Architecture (lecture)

320202 Operating Systems (lecture)

320232 Operating Systems Lab

320300 – CS Specialization Module

Short Name: ModCSSPcs
Semester: 5 – 6
Credit Points: 20 ECTS

General Information During the third year, students can specialize in the areas of their specific interests. This module gathers all CS specialization courses offered in the third year.

Learning goals

- Students specialize in their chosen fields of interest.
- Courses familiarize students with the basic knowledge and skills needed to understand and reflect state-of-the-art research and development in the choosen areas.
- Students are prepared to either enter graduate research and development programs or to acquire the knowledge necessary to successfully enter the job marked in the chosen focus areas.

Courses

320301 Computer Networks

320302 Databases and Web Applications

320331 Artificial Intelligence

320312 Distributed Systems

320322 Graphics and Visualization

320311 Robotics

320321 Image Processing

320351 Medical Image Analysis

320521 Autonomous Systems

320441 Computational Logic

320372 Machine Learning

300341 Information Theory

300362 Coding Theory

320310 – CS Guided Research Module

Short Name: ModGRcs
Semester: 5 – 6
Credit Points: 10 ECTS

General Information Guided research projects are designed to get students involved into research activities. The topics are posted by faculty members and usually related to their specific research activities. The deliverables produced by the students are a research proposal, an oral presentation of the topic and the achieved results, and the final guided research report (B.Sc. thesis).

Learning goals

- Students get involved in ongoing research activities.
- Ability to independently work on a given problem.
- Students learn to organize their work and time.
- Training of writing and presentation skills.

Courses

320371 Guided Research in Computer Science

320342 Guided Research in Computer Science + Thesis

3 Requirements for a B.Sc. in Computer Science

3.1 General Requirements

To obtain a B.Sc. degree at Jacobs University, a minimum of 180 ECTS credit points must be earned over a period of 6 semesters.

- A minimum of 140 ECTS credits must be earned in the School of Engineering and Science.
- 30 ECTS credits must be earned through transdisciplinary courses, comprised of courses in the School of Humanities and Social Sciences (SHSS) and University Study Courses (USC). Students can choose how many USCs or S HSS courses they take.
- 10 ECTS credits (4 courses) are accredited either for language courses or additional Home School electives. Students can decide whether they take language courses or not.

University requirements outside of the school of the major are type-coded "u" in the recommended course plan below.

3.2 Requirements of the Major

Students choose 140 ECTS credits in Engineering and Sciences out of the following courses:

Year 1 Level Courses	Course Numbers	ECTS Credits
Engineering and Science Mathematics I A / II B	120101, 120112	10
General Computer Science I/II	320101, 320102	10
NatSciLab Units Programming in C I/II	320111, 320112	5
Computer Architecture	320241	5
Programming in C++	320142	5
First year Engineering and Science subject I/II	n/a	10
NatSciLab Units Engineering and Science subject I/II	n/a	5

Year 2 Level Courses	Course Numbers	ECTS Credits
Engineering and Science Mathematics III A / IV A	120201, 120202	10
Algorithms and Data Structures	320201	5
Formal Languages and Logic	320211	5
Computability and Complexity	320352	5
Programming in Java	320341	5
Software Engineering / Software Engineering Lab	320212, 320222	10
Operating Systems / Operating Systems Lab	320202, 320232	10

Year 3 Level Courses	Course Numbers	ECTS Credits
CS Specialization Area Courses	n/a	20
Guided Research and BSc Thesis	320371, 320342	10

Additional Courses	Course Numbers	ECTS Credits
Two Home School electives	n/a	10

Students with special interests in certain subject areas can, with the approval of the instructor of record, choose courses offered as part of the CS graduate program as specialization area courses.

4 Recommended Course Plan

This course plan has been compiled based on the assumption of no previous knowledge when entering Jacobs University. Although not binding, it is highly recommended since it ensures an even workload, optimum efficiency and maximum congruence with the objectives of the curriculum.

Year 1 Courses	**Fall**	**C**	**T**	**Spring**	**C**	**T**
ESc Mathematics I A / II B	120101	*5*	m	120112	*5*	m
General Computer Science I/II	320101	*5*	m	320102	*5*	m
Programming in C (NatSciLab CS I/II)	320111	*2.5*	m	320112	*2.5*	m
Computer Architecture	320241	*5*	m			
Programming in C++				320142	*5*	m
First year courses in ESc subject		*5*	e		*5*	e
Associated NatSciLabs I/II		*2.5*	e		*2.5*	e
Transdisciplinary Courses		*5*	u/e		*5*	u/e
Running Total / *Semester Total*	30.0	*30.0*		60.0	*30.0*	

Year 2 Courses	**Fall**	**C**	**T**	**Spring**	**C**	**T**
ESc Mathematics III A / IV A	120201	*5*	m	120202	*5*	m
Programming in Java	320341	*5*	m			
Formal Languages and Logic	320211	*5*	m			
Algorithms and Data Structures	320201	*5*	m			
Software Engineering				320212	*5*	m
Software Engineering Lab				320222	*5*	m
Operating Systems				320202	*5*	m
Operating Systems Lab				320232	*5*	m
Computability and Complexity				320352	*5*	m
Home School Electives or Language Courses		*5*	e			
Transdisciplinary Courses		*5*	u/e			
Running Total / *Semester Total*	90.0	*30.0*		120.0	*30.0*	

Year 3 Courses	**Fall**	**C**	**T**	**Spring**	**C**	**T**
CS Specialization Area Courses		*4*5*	me			
Home School Electives or Language Courses		*1*5*	e		*2*5*	e
Guided Research and BSc Thesis CS	320371	2.5	m	320342	*7.5*	m
Transdisciplinary Courses		*5*	u/e		*10*	u/e
Running Total / *Semester Total*	152.5	*32.5*		180.0	*27.5*	

C = ECTS credit points
T = type (m=mandatory, e=elective, u=university, me=mandatory elective)

Transdisciplinary Courses are School of Humanities and Social Sciences courses or University Studies Courses

The following courses are counted as third-year specialization courses in Computer Science. Four out of 11 are mandatory and recommended to be taken in the fifth semester.

Year 3 CS Specialization Courses	Fall	C	T	Spring	C	T
Computer Networks	320301	5	me			
Databases and Web Applications	320302	5	me			
Artificial Intelligence	320331	5	me			
Graphics and Visualization	320322	5	me			
Robotics(1)	320311	5	me			
Autonomous Systems(1)	320521	5	me			
Image Processing(2)	320321	5	me			
Medical Image Analysis(2)	320351	5	me			
Computational Logic	320441	5	me			
Machine Learning	320372	5	me			
Information Theory	300341	5	me			
Distributed Systems				320312	5	e
Coding Theory				300362	5	e

(1) Robotics and Autonomous Systems alternate biannually
(2) Image Processing and Medical Image Analysis alternate biannually

4.1 Professional Skills

The SES highly recommends attending the Professional Skills seminars offered by the Career Services Center. Those seminars include soft skills development seminars and application training which will help you to cope with your studies and master your internship and job search.

All undergraduate students are required to complete an internship, normally to be accomplished between the second and third year of study. Information about the internship will be listed on the transcript. The internship must last at least two consecutive months. No credits are connected to the internship requirement. For more information on internships see:

http://www.jacobs-university.de/career-services/internship

4.2 Selection of Elective Courses

The following recommendations should be considered for the selection of elective courses.

- Students who are strong in Mathematics and interested in the more formal aspects of Computer Science should consider to select the General Mathematics and Computational Science I/II courses in the first two semesters.
- Students who like to be able to change easily to the programs Electrical Engineering and Computer Science (EECS) or Electrical and Computer Engineering (ECE) at the end of

the first year should consider to select the General Electrical Engineering I/II courses in the first two semesters and the associated NatSciLab units Electrical Engineering I/II.

5 Courses: Computer Science

5.1 1^{st} Year Courses and Labs

After the first year, the students should be conversant in the general principles of Computer Science and, since Computer Science makes use of advanced mathematical tools, with the most important mathematical concepts needed. All courses listed here are mandatory for CS students. For the inter- and transdisciplinary education, the students are furthermore required to take two electives from each of the schools and one University Studies course.

320101 – General Computer Science I

Short Name:	GenCS I
Type:	Lecture
Semester:	1
Credit Points:	5 ECTS
Prerequisites:	None
Corequisites:	None

Course contents The course covers the fundamental concepts and techniques of computer science in a bottom-up manner. Based on clear mathematical foundations (which are developed as needed) the course discusses abstract and concrete notions of computing machines, information, and algorithms, focusing on the question of representation vs. meaning in Computer Science.

To have a theoretical notion of computation, we introduce inductively defined structures, term representations, abstract interpretation via equational substitution. This is contrasted with a first concrete model of computation: Standard ML, which will also act as the primary programming language for the course. We cover a basic subset of ML that includes types, recursion, termination, lists, strings, higher-order programming, effects, and exceptions. Back on the theoretical side, we cover string codes, formal languages, Boolean expressions (syntax) and Boolean Algebras (semantics). The course introduces elementary complexity theory (big-O), applying it to analyzing the gate-complexity of Boolean Expressions (prime implicants and Quine McCluskey's algorithm).

Topics Discrete mathematics, terms, substitution, abstract interpretation, computation, recursion, termination, complexity, Standard ML, types, formal languages, boolean expressions.

320111 – Natural Science Lab Unit Programming in C I

Short Name: NatSciLabCS I
Type: Lab
Semester: 1
Credit Points: 2.5 ECTS
Prerequisites: None
Corequisites: None

Course contents This lab unit is a first introduction to programming using the programming language C. The course covers fundamental procedural programming constructs and simple algorithms in a hands-on manner.

320241 – Computer Architecture

Short Name: CSCA
Type: Lecture
Semester: 1
Credit Points: 5 ECTS
Prerequisites: None
Corequisites: None

Course contents Starting from essential logical circuits, this course introduces core components (processors, memory systems, buses) and architectures of modern computing systems.

Topics Computer architectures, processors, instruction sets, memory systems, system busses, parallel processing.

320102 – General Computer Science II

Short Name: GenCS II
Type: Lecture
Semester: 2
Credit Points: 5 ECTS
Prerequisites: 320101
Corequisites: None

Course contents The course continues the introduction of the fundamental concepts and techniques of Computer Science. Building on Boolean Algebra, it introduces Propositional Logic as a model for general logical systems (syntax, semantics, calculi). Based on elementary graph theory, combinatory circuits are introduced as basic logic computational devices. Interpreting sequences of Boolean values as representations of numbers (in positional number systems, twos-complement system), Boolean circuits are extended to numerical computational machines (presenting adders, subtracters, multipliers) and extended to basic ALUs. The course introduces very elementary computer architectures and assembly language concrete computational devices, and compares them to Turing machines to fathom the reach of computability.

In a final part of the course, two topics of general Computer Science are covered in depth, for instance "search algorithms" and "programming as search" to complement the rather horizontal (i.e. methods-oriented) organization of the course with vertically (i.e. goal-oriented) organized topics.

Topics Propositional logic, calculi, soundness, completeness, automated theorem proving, combinatory circuits, assembler turing machines, search, logic programming.

320112 – Natural Science Lab Unit Programming in C II

Short Name:	NatSciLabCS II
Type:	Lab
Semester:	2
Credit Points:	2.5 ECTS
Prerequisites:	320111
Corequisites:	None

Course contents This lab unit is a continuation of the first year CS lab unit and deepens the basic programming skills from the first lab. It covers advanced topics of C programming such as data structures, file handling, libraries, and debugging techniques.

320142 – Programming in C++

Short Name:	CSPCPP
Type:	Lecture / Lab
Semester:	2
Credit Points:	5 ECTS
Prerequisites:	320111
Corequisites:	None

Course contents The course is an introduction into object-oriented programming using the programming language C++. The unit covers the object-oriented programming constructs in C++ in a hands-on manner.

Topics C++ programming language, practical implementation of algorithms.

5.2 2nd Year Courses and Labs

In the second year, the focus areas are computer architecture, operating systems, formal languages, discrete automata, first-order logic, and software engineering.

The mathematical training is mandatory for all students and the transdisciplinary education is continued by one University Studies course.

320201 – Algorithms and Data Structures

Short Name:	CSAD
Type:	Lecture
Semester:	3
Credit Points:	5 ECTS
Prerequisites:	(320112 or 350112) and 120112
Corequisites:	None

Course contents This course introduces a basic set of data structures and algorithms that form the basis of almost all computer programs. The data structures and algorithms are analyzed in respect to their computational complexity with techniques such as worst case and amortized analysis.

Topics Fundamental data structures (lists, stacks, trees, hash tables), fundamental algorithms (sorting, searching, graph traversal).

320211 – Formal Languages and Logic

Short Name:	CSFLL
Type:	Lecture
Semester:	3
Credit Points:	5 ECTS
Prerequisites:	320102
Corequisites:	None

Course contents This course gives an introduction to the most basic themes of theoretical computer science. Formal languages and discrete automata are the foundations of programming languages and their parsing and compiling. First-order logic is the basis of artificial intelligence, program verification and advanced data base systems.

Topics Formal languages, discrete automata, first-order logic.

320341 – Programming in Java

Short Name: CSPJ
Type: Lecture / Lab
Semester: 3
Credit Points: 5 ECTS
Prerequisites: (320102 and 320112) or (350102 and 350112)
Corequisites: None

Course contents Java is an object-oriented programming language which is very widely used for the development of applications running on the Internet, and in particular electronic commerce applications. Java has some unique features such as platform independence and a very rich set of reusable class libraries. This course introduces the core language and the most important core Java packages.

Topics Java Virtual Machine, object-oriented programming in Java (types, objects, interfaces, abstract classes, etc.), Java threads, core packages (java.net, java.io, java.sql), Java web programming (servlets, JSP, beans, enterprise beans).

320202 – Operating Systems

Short Name: CSOS
Type: Lecture
Semester: 4
Credit Points: 5 ECTS
Prerequisites: 320201 and 320112 and 320241
Corequisites: None

Course contents This course provides an introduction to the concepts underlying operating systems. Students will develop an understanding how operating systems realize a virtual machine that can be used to execute multiple concurrent application programs. The course discusses resource allocation algorithms and how concurrency problems can be solved.

Topics Operating system architectures, system calls and interrupts, concurrent processes and threads, scheduling, synchronization, deadlocks, virtual memory, file systems, inter-process communication, socket programming interface.

320232 – Operating Systems Lab

Short Name: CSOSLAB
Type: Lab
Semester: 3
Credit Points: 5 ECTS
Prerequisites: 320201 and 320112
Corequisites: 320202

Course contents This lab complements the Operating Systems course. Students will gain practical experience with systems programming above and below the system call interface of operating systems. Students will learn how to write concurrent programs and gain understanding how kernel programming differs from normal application development.

320212 – Software Engineering

Short Name: CSSE
Type: Lecture
Semester: 4
Credit Points: 5 ECTS
Prerequisites: (320201 and 320142) or (320201 and 350112)
Corequisites: None

Course contents This course is an introduction to software engineering (SE) and object-oriented software design. At the core of the lecture is the notion of software quality and the methods to achieve and maintain it. Based on their pre-existing knowledge of an object-oriented programming language, students are familiarized with software architectures, design patterns and frameworks, software components and middleware, UML-based modelling, and validation by code analysis and testing. Both classical development and modern variants, in particular: Web Engineering, are covered.

Further, the course addresses the more organizational topics of project management and version control.

Topics Software quality, process models, design patterns and frameworks, components and middleware, UML, testing, tools, project management, version control.

320222 – Software Engineering Lab

Short Name: CSSELAB
Type: Lab
Semester: 4
Credit Points: 5 ECTS
Prerequisites: 320142
Corequisites: 320212

Course contents The Software Engineering Lab course ends the series of courses on software development basics, placing particular emphasis on programming-in-the-large, i.e., "multi-person construction of multi-version software". Project work encompasses team-oriented specification, implementation, documentation, and compliance tests of some non-trivial software system.

Topics Team-oriented software development project.

320352 – Computability and Complexity

Short Name: CSCC
Type: Lecture
Semester: 4
Credit Points: 5 ECTS
Prerequisites: 320211
Corequisites: None

Course contents This lecture presents one half of the core material of theoretical computer science (the other half is covered in the lecture "Formal Languages and Logic"). The question: "What problems can a computer possibly solve?", is fully answered (by characterizing those solvable problems, equivalently, through Turing machines, random access machines, recursive functions and lambda calculus). A full answer to the related question, "how much computational resources are needed for solving a given problem" is not known today. However, the basic outlines of today's theory of computational complexity will be presented up to the most famous open problem in computer science, namely the famous "P = NP" question: if a computer can guess the answer to a problem (and only needs to check its correctness), does that really help to speed up computation?

Topics Computable functions and complexity, lambda calculus, functional programming.

5.3 3^{rd} Year Courses and Labs

In the third year, the specialization lectures will bring the student "up to speed" to the frontiers of research and technology and provide the theoretical groundwork for the guided research work (see section 5.5) which has to be completed in the third year.

The mathematics training has been completed in the first two years. The transdisciplinary education is completed by two University Studies courses and two courses from the School of Humanities and Social Sciences (HSS).

320301 – Computer Networks

Short Name:	CSCN
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	320202
Corequisites:	None

Course contents The course discusses network protocols in some depth in order to enable students to understand the core issues involved in network protocol design. Fundamental algorithms and principles are explained in the context of existing IEEE / Internet protocols in order to demonstrate how they are applied in real-world scenarios. This course is recommended for all students with a strong interest in communication networks and distributed systems.

The course covers topics such as local area networks (IEEE 802), Internet protocols, routing algorithms and protocols, flow and congestion control mechanisms, data representation, application layer protocols, remote procedure calls, network security.

320302 – Databases and Web Applications

Short Name:	CSDBWA
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	320201
Corequisites:	None

Course contents This course introduces (relational) database systems in theory and practice, with special emphasis on Web-based information services. Topics addressed include database design (ER, UML), SQL, relational design theory, transaction management, security, web applications, 3-tier architectures, XML, XPath/XQuery, and an outlook on novel paradigms like NewSQL.

In the homeworks, the core of a database-enabled Web service is implemented. Students arrange themselves into small teams. Each team picks some individual application it finds exciting and agrees it with the instructor. In a guided process over the semester, the team then designs, documents, implements, and validates its chosen service based on a LAMP platform provided.

The course requires basic knowledge about algebraic expressions and laws, basic data structures like trees, object-oriented concepts, as well as basics of HTML and – for the homeworks – Linux.

Learning goals are (i) knowledge about databases and Web-based information systems and (ii) skills in designing and building database and Web services for science, engineering, and business domains.

320331 – Artificial Intelligence

Short Name:	CSAI
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	320201 and 320211
Corequisites:	None

Course contents Among the disciplines of CS, Artificial Intelligence is one of the most interdisciplinary, with connections to robotics, pattern recognition, machine learning, high-level programming, databases, software engineering and many more. To provide a modern treatment of the subject, the course focuses on approaches and techniques which enable an intelligent agent to plan, learn, and make decisions in stochastic environments.

Topics Dynamic programming and search, probabilistic reasoning, inference in Bayesian networks, Hidden Markov Models, Kalman filter, dynamic Bayesian networks, decision-theoretic expert systems, Markov decision processes, intelligent control, information entropy and decision trees, classification and regression, reinforcement learning, various applications in robotics and machine perception.

320322 – Graphics and Visualization

Short Name:	CSGV
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	320202 and 320222
Corequisites:	None

Course contents This is an introductory class into the concepts and techniques of 3D interactive computer graphics and visualization. Mathematical foundations, basic algorithms and principles, and advanced methods of real-time rendering and visualization are being taught. This course is recommended for all EECS students with an interest in data visualization and computer graphics.

Topics Geometric foundations, object representation, raster graphics, color models, shading and lighting, textures, animation and modelling, scientific visualization.

320311 – Robotics

Short Name:	CSR
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	320201 and 320222
Corequisites:	None

Course contents Robotics is a field that spans the entire range from low-level mechatronics and signal processing to high-level cooperation protocols of intelligent agents, and thus touches large portions of both CS and EE. Correspondingly, the course aims at an integrative, practically oriented education that enables students to practically combine methods he/she has encountered in various more specialized courses before.

The course is offered biannually (alternating with "Autonomous Systems").

320521 – Autonomous Systems

Short Name:	CSAS
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	320201 and 320222
Corequisites:	None

Course contents There is an increasing interest and need to generate artificial systems that can carry out complex missions in unstructured environments without permanent human supervision. Intelligent mobile robots are often used as prototype or even defining example of according autonomous systems. The investigation of autonomous systems is driven from two different perspectives. First, it is motivated by the engineering aspects of generating application oriented devices. Second, artificial autonomous systems offer new ways to investigate and constructively understand natural cognition.

The course is offered biannually (alternating with "Robotics").

320321 – Image Processing

Short Name: CSImgProc
Type: Lecture
Semester: 5
Credit Points: 5 ECTS
Prerequisites: 320222 or 300221
Corequisites: None

Course contents The course provides a foundation in the theory and applications of digital image processing. The first part will concentrate on morphological image processing, which is one of the most powerful tool sets in dealing with digital images and it is the backbone of many of today's high-performance image analysis systems. We will start on basic concepts of dilation, erosion, geodesic transformations, morphological filtering, and watershed transform, and will develop into advanced strategies for image segmentation and texture analysis. The second part of the course will concentrate on solving problems from biomedical, environmental, and industrial imaging, and will provide an overview of the broader field of image processing.

The course is offered biannually (alternating with "Medical Image Analysis").

Topics Morphological image processing, distance transformations, geodesic transformations, reconstruction based operators, image segmentation, watershed transformation, automated threshold selection, advanced image processing, motion analysis, image registration, pattern recognition, texture analysis, selected applications

320351 – Medical Image Analysis

Short Name: CSMedIA
Type: Lecture
Semester: 5
Credit Points: 5 ECTS
Prerequisites: 320222 or 300221
Corequisites: None

Course contents The course provides a foundation in the theory and methods of digital image processing with applications in medical imaging. We start with morphological image processing, which is one of the most powerful tool sets in dealing with digital images and it is the backbone of many of today's high-performance image analysis systems. After basic concepts of image-to-image transformations, morphological and Fourier filtering, and the watershed transform, we develop into advanced strategies for image segmentation, image registration, and pattern recognition. Not least, we concentrate on solving problems from diagnostic and therapeutic medical imaging, and will provide an overview of the broader field of medical image analysis. The course also addresses practical implementation aspects of specific image processing tasks. To this end, knowledge in C or C++ will be required.

The course is offered biannually (alternating with “Image Processing”)

320441 – Computational Logic

Short Name: CSCL
Type: Lecture
Semester: 5
Credit Points: 5 ECTS
Prerequisites: 320211
Corequisites: None

Course contents In this course we will cover the basics of computational logic. We will introduce the syntax and semantics of first-order logic, and discuss calculi, soundness, completeness on this system. We will cover machine-oriented inference calculi like analytic tableaux, and resolution and apply them to theorem proving and logic programming applications.

We will discuss the non-deductive reasoning modes of abduction and induction and briefly introduce computational methods for mechanizing them. Finally, we will give an introduction to knowledge representation and description logics, leading to an introduction of “semantic web” techniques.

320372 – Machine Learning

Short Name: CSML
Type: Lecture
Semester: 5
Credit Points: 5 ECTS
Prerequisites: 120112 and 120201
Corequisites: None

Course contents Machine learning (ML) is all about algorithms which are fed with (large quantities of) real-world data, and which return a compressed “model” of the data. An example is the “world model” of a robot: the input data are sensor data streams, from which the robot learns a model of its environment – needed, for instance, for navigation. Another example is a spoken language model: the input data are speech recordings, from which ML methods build a model of spoken English – useful, for instance, in automated speech recognition systems. There is a large number of formalisms in which such models can be cast, and an equally large diversity of learning algorithms. However, there is a relatively small number of fundamental challenges which are common to all of these formalisms and algorithms: most notably, the “curse of dimensionality” and the almost deadly-dangerous problem of under- vs. overfitting. This lecture introduces such fundamental concepts and illustrates them with a choice of elementary model formalisms (linear classifiers and regressors, radial basis function networks, clustering, mixtures of Gaussians, Parzen windows). Furthermore, the course also provides an intense refresher of the requisite concepts from probability theory, statistics, and linear algebra.

300341 – Information Theory

Short Name:	EEIT
Type:	Lecture
Semester:	5
Credit Points:	5 ECTS
Prerequisites:	120112 and 120201
Corequisites:	None

Course contents Information theory serves as the most important foundation for communication systems. The course provides an analytical framework for modeling and evaluating point-to-point and multi-point communication.

The key concept of information content of a signal source and information capacity of a transmission medium are precisely defined, and their relationships to data compression algorithms and error control codes are examined in detail. The course aims to install an appreciation for the fundamental capabilities and limitations of information transmission schemes and to provide the mathematical tools for applying these ideas to a broad class of communications systems. Information theory is a standard in every communications-oriented Bachelor's program.

320312 – Distributed Systems

Short Name:	CSDS
Type:	Lecture
Semester:	6
Credit Points:	5 ECTS
Prerequisites:	320301
Corequisites:	None

Course contents The first part of the course focusses on distributed file systems and generic middleware systems such as CORBA or Web Services. The second part of the course focusses on distributed algorithms that are the foundation for complex and fault-tolerant distributed systems. The material covered has been selected to provide a solid overview over the key algorithms and to develop an understanding of the issues that influence solutions for a certain problem in a distributed system.

Topics Middleware systems, distributed file systems, clock synchronization, logical and vector clocks, reliable, causal and atomic multicasts, virtual synchrony, election algorithms, voting algorithms, consistent snapshots, security.

300362 – Coding Theory

Short Name: EECT
Type: Lecture
Semester: 6
Credit Points: 5 ECTS
Prerequisites: 120112 and 120201
Corequisites: None

Course contents Error correcting codes (convolutional codes, block codes, Turbo codes, LDPC codes, etc.) play an essential role in modern digital high data-rate transmission systems. They are part of almost every modern communication and storage/recording device, like your CD player, your DSL home Internet access, and your mobile phone, to name just a few. This course will focus on theory, construction, and algorithms for error correcting codes, and will highlight the application in communication systems. For modern communications, coding knowledge is a must.

5.4 Graduate Courses in Computer Science

In the last semester, academically strong undergraduate students may participate in the following courses offered as part of the Computer Science graduate program. Enrollment in these courses requires the consent of the Instructor of Record. The credits count either as 3rd year CS specialization course credits or they may alternatively count towards a future CS graduate degree in case the student is able to fulfill the graduation requirements without the credits earned in these courses.

320523 – Advanced Autonomous Systems

Short Name: AdvAutSys
Type: Lecture
Semester: 2nd
Credit Points: 5 ECTS
Prerequisites: 320521
Corequisites: None

Course contents The Advanced Autonomous Systems course builds upon the Autonomous Systems course offered in the previous semester of that academic year. The Advanced Robotics course covers selected areas in more profound depth. The content of the course is hence devoted to most recent state of the art research on Autonomy like Cooperation or Cognitive Models.

320581 – Advanced Visualization

Short Name: AdvViz
Type: Lecture
Semester: 2nd
Credit Points: 5 ECTS
Prerequisites: 320322
Corequisites: None

Course contents Scientific visualization deals with the visualization of data with a natural spatial interpretation such as computer-generated data from numerical simulations (physics, chemistry) or measured data using scanning or sensor techniques (medicine, life sciences, geo-sciences). Volume visualization methods such as segmentation, surface extraction, and direct volume rendering for structured and unstructured gridded as well as scattered data are being taught. These include techniques for scalar field, vector field, and tensor field visualization.

Information visualization deals with the visualization of abstract data with no spatial interpretation such as graph- or network-based data (life sciences, social sciences, computer networks) or multi-dimensional data (economics, databases). Methods that tackle these visualization problems are being taught.

The course deepens, broadens, and enhances the knowledge in visualization obtained from the undergraduate course on ”Graphics and Visualization” in terms of visualization methods.

320541 – Computational Semantics of Natural Language

Short Name: CompSem
Type: Lecture
Semester: 2nd
Credit Points: 5 ECTS
Prerequisites: 320441
Corequisites: None

Course contents In this course we will cover the logical and linguistic foundation of syntactical and semantic modeling of natural language in computational linguistics (the study of natural languages from a computational perspective). We will proceed by the “method of fragments”, where fragments of natural language are studied on a syntactic level (grammar and lexicon), the semantic level (transforming syntactic structures into logical forms), and a pragmatic level (inferring material that is not explicitly realized linguistically).

We will build up a sequence of fragments of increasing coverage (covering selected salient features of language) and discuss the linguistic and logical phenomena involved in detail.

The course will be accompanied by a hands-on programming lab, where the topics of the course will be implemented in the programming language PROLOG.

320671 – Machine Vision

Short Name:	MachViz
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	120201 (ESM 3A)
Corequisites:	None

Course contents Machine vision algorithms are used in a variety of real-world applications, such as surveillance and object tracking, 3D model building (photogrammetry), and object recognition. Apart from their visual appeal, these algorithms also represent elegant applications of advanced linear algebra and optimization techniques. Topics covered in this course include image-formation and camera calibration, image homographies, segmentation, feature detection and matching, structure from motion, 3D point-cloud processing, and an introduction to object-recognition.

320402 – Advanced Computer Networks

Short Name:	AdvCompNet
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	320301
Corequisites:	None

Course contents This course covers advanced computer networking concepts such as multimedia communication and content distribution. The course covers voice communication in packet switched IP networks, related signaling and transport protocols, quality of service approaches (integrated and differentiated services), and multicast group communications. Some attention will be given to reliability and security aspects. Finally, the course covers technologies popular in backbone networks such as MPLS and new technologies of the IEEE 802 family of standards used in modern optical access networks.

320643 – Applied Machine Learning

Short Name:	AppML
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	320372
Corequisites:	None

Course contents This Spring course builds on the introductory Fall course "Machine Learning" and presents a number of machine learning techniques which are widely used in application practice: adaptive linear filters (employed virtually everywhere in signal processing and control), feedforward neural networks (employed when it comes to learn complex nonlinear input-output relationships, as e.g. in financial time series prediction or image recognition), hidden Markov models (the workhorse for speech recognition), and fuzzy logic (used by engineers to predict and control very nonlinear systems). From a math/methods side, the two main classes of supervised learning algorithms are studied: optimization by gradient descent, and by "expectation-maximization" algorithms.

320574 – Modeling Complex Systems

Short Name:	ModCompSys
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	320372
Corequisites:	None

Course contents This Spring course builds on the introductory Fall course "Machine Learning" and presents a number of machine learning techniques which aspire to model extremely complex real-world datasets and systems. Examples: robots navigating in unknown stochastic environments, image and video collections, the space shuttle, VLSI circuits, the weather. Modeling of such systems is an inherently probabilistic task: there is never enough data available to completely know the real system's state or laws – the gap must be filled by statistics. Therefore, the emphasis in this course lies on statistical modelling and other methods that allow one to cope with uncertainty. A choice from the following techniques will be covered: (i) sampling methods and representations of probability distributions, optimization by simulated annealing, (ii) Bayesian networks and graphical models, with exact, Monte Carlo, and/or variational inference techniques, (iii) input-output models used in agent modelling and control (POMDP's, input-output-OOMs), (iv) recurrent neural networks.

320491 – Advanced Graphics

Short Name:	AdvGrafx
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	320322
Corequisites:	None

Course contents Computer graphics deals with the digital synthesis and manipulation of visual content, typically embedded in a three-dimensional scene. Prominent tasks in computer graphics are geometry processing, rendering, and animation. Geometry processing is concerned with object representations such as surfaces and their modeling, rendering is concerned with simulating light transport to get physically-based photorealistic images of 3D scenes or applying a certain style to create non-photorealistic images, and animation is concerned with descriptions for objects that move or deform over time. Methods that tackle these three tasks are being taught.

The course deepens, broadens, and enhances the knowledge in 3D computer graphics obtained from the undergraduate course on "Graphics and Visualization" in terms of graphics methods.

320421 – Advanced Robotics

Short Name:	AdvRob
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	320311
Corequisites:	None

Course contents The Advanced Robotics course builds upon the Robotics course offered in the previous semester of that academic year. The Advanced Robotics course covers selected areas of robotics in more profound depth. The content of the course is hence devoted to most recent state of the art research in a field related to work of Jacobs Robotics like Underwater Robotics or 3D Perception and Mapping.

320411 – Information Architectures

Short Name:	InfArch
Type:	Lecture
Semester:	2nd
Credit Points:	5 ECTS
Prerequisites:	320302
Corequisites:	None

Course contents The title of this course can, and should, be understood in a twofold way. On conceptual level, mastering the rapidly growing volume and complexity of information in industry, science, and society requires improved modelling and design methodologies. On implementation level, existing storage, retrieval, and delivery techniques have to be revisited and new ones have to be designed in order to meet the challenges formulated conceptually. While these issues largely fall into the fields of databases, information retrieval, and Internet technology, the questions arising clearly transcend these fields and call for interdisciplinary research on more efficient and effective methods.

The course, therefore, starts with an overview of existing knowledge in the core fields and then covers selected themes in more depth. Among the candidate themes are non-standard applications such as spatio-temporal databases, array databases, parallel and distributed databases, XML databases, and NoSQL / NewSQL databases.

The goal is to make students familiar with the state of the art in Web-enabled information systems so that they will be successful database / Internet professionals in the IT industry and also have a sound knowledge base to specialize towards a scientific career in the field.

5.5 Guided Research in CS

Guided research is part of the undergraduate program in CS. The guided research courses are offered by all professors of Computer Science jointly. The professors propose a number of research projects, which will be posted on a dedicated guided research website. Students interested in particular projects should get in touch with the professor in charge of that project.

Guided research in CS will usually be organized in the form of research seminars held by the professors. These seminars are attended by undergraduate students supervised by the faculty member and by graduate students from the CS graduate programs. They serve as a forum for discussing research goals, methods and results.

320371 – Guided Research in Computer Science

Short Name:	CSGRP
Type:	Project
Semester:	5
Credit Points:	2.5 ECTS
Prerequisites:	None
Corequisites:	None

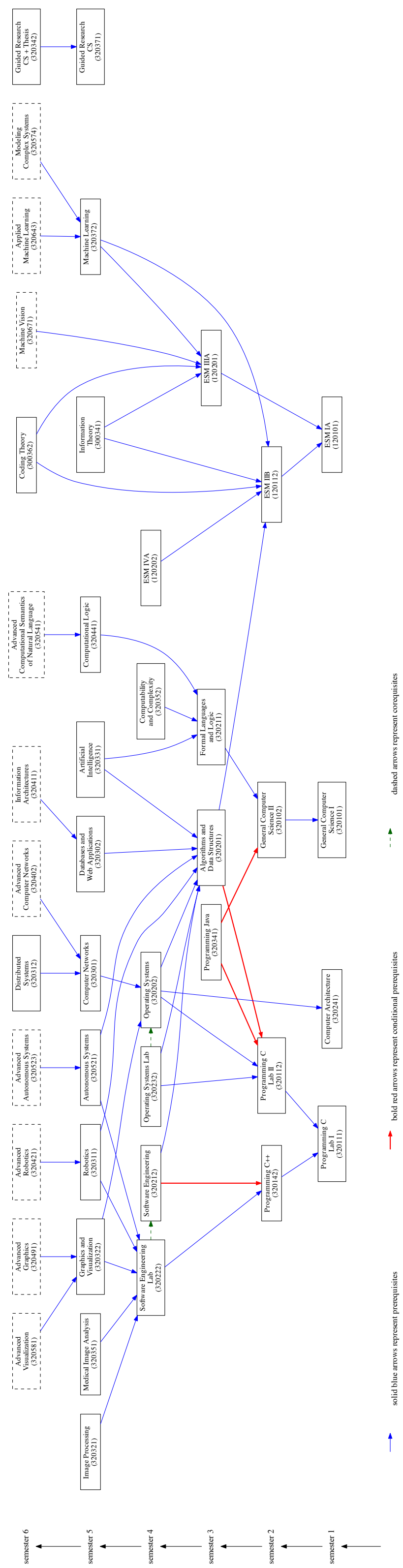
Course contents The purpose of this course is to let students choose a topic for the bachelor thesis and to work out a proposal which introduces the field of study, states the research questions/hypotheses, surveys the expected results, and sets up a work plan with timetable. The course is offered by all professors of Computer Science jointly. Topics are offered by the individual faculty members.

320342 – Guided Research in Computer Science + Thesis

Short Name:	CSGR
Type:	Project
Semester:	6
Credit Points:	7.5 ECTS
Prerequisites:	320371
Corequisites:	None

Course contents The course is offered by all professors of Computer Science jointly. The purpose of this course is to engage the students in a research project under the close supervision of a CS faculty member. Topics are offered by the individual faculty members. Upon completion of the research, the student will prepare a final report (20 pages) and present the project in a seminar during the last 2 weeks of the semester. Both the presentation and the final report will count towards the final grade.

5.6 Course Dependencies



6 Courses: Computer Science Service Courses

6.1 1^{st} Year Courses Service Courses

The 1^{st} year Computer Science service courses provide an introduction to information and communication technology (ICT) for students who want to learn the basics of Computer Science from an applied perspective. The courses also prepare students to take selected 2^{nd} and 3^{rd} year Electrical Engineering and Computer Science courses.

At the time of this writing, the General Information and Communication Technology (GenICT) service courses as well as the Natural Science Lab Units Programming in Python are mandatory in the Logistics undergraduate program.

350101 – General Information and Communication Technology I

Short Name:	GenICT I
Type:	Lecture
Semester:	1
Credit Points:	5 ECTS
Prerequisites:	None
Corequisites:	None

Course contents The course introduces fundamental concepts underlying today's information and communication technology. The course is designed to provide an applied introduction to Computer Science concepts and information and communication technology. The course is open to all students except students enrolled in the CS and EECS majors who are required to take the more formal General Computer Science I course.

The course starts with introducing basic computer science terms and concepts and different programming paradigms. Afterwards, the basic components of computer systems will be introduced as well as boolean logic, number representations, character sets their representation, and structured document formats. Finally, some key concepts of theoretical computer science such as complexity, correctness, and computability will be introduced.

Topics Algorithms and different classes of algorithms, programming models (imperative, object-oriented, functional, declarative), basic computer architecture, boolean logic and basic laws, number representation, character sets, structured document formats, data conversion, theoretical concepts (complexity, correctness, termination of algorithms, computability).

350111 – Natural Science Lab Unit Programming in Python I

Short Name: NatSciLabPy I
Type: Lab
Semester: 1
Credit Points: 2.5 ECTS
Prerequisites: None
Corequisites: None

Course contents This lab unit is a first introduction to programming using the programming language Python. The course covers fundamental programming constructs and simple algorithms in a hands-on manner.

This lab is open to all students except students enrolled in the ECE, CS, or EECS majors. These students are required to take the Natural Science Lab CS I instead. Students enrolled in the ECE/CS/EECS majors, who successfully complete the lab unit Programming in Python I, receive credits that do not count towards the 180/185 ECTS-credits required by their major.

Topics installation and first steps in python, built-in data types, operators and expressions, control flow, functions, modules, problem solving and simple data structures

350102 – General Information and Communication Technology II

Short Name: GenICT II
Type: Lecture
Semester: 1
Credit Points: 5 ECTS
Prerequisites: 350101
Corequisites: None

Course contents The course continues the applied introduction to Computer Science concepts and information and communication technology. The course is open to all students except students enrolled in the CS and EECS majors who are required to take the more formal General Computer Science II course.

The course introduces database systems, network and communication technology, software engineering concepts, and data security and data protection mechanisms.

Topics relational database model, query languages, Internet protocols, wireline and wireless communication technologies, unified modeling language, software development processes, cryptographic algorithms, key management, authentication protocols and authorization models.

350112 – Natural Science Lab Unit Programming in Python II

Short Name:	NatSciLabPy II
Type:	Lab
Semester:	1
Credit Points:	2.5 ECTS
Prerequisites:	350111
Corequisites:	None

Course contents This lab unit is a continuation of the first semester lab Programming in Python I. It covers advanced topics of Python programming such as object oriented programming, advanced data structures, file handling, debugging techniques and problem solving using frameworks.

This lab is open to all students except students enrolled in the ECE, CS, or EECS majors. These students are required to take the Natural Science Lab CS II instead. Students enrolled in the ECE/CS/EECS majors, who successfully complete the lab unit Programming in Python II, receive credits that do not count towards the 180/185 ECTS-credits required by their major.

Topics debugging techniques, object-oriented programming, advanced data structures, python frameworks

6.2 Course Dependencies Service Courses

The CS service courses can be integrated into other programs as an Information and Communication Technology (ICT) track. The following diagram shows the course dependencies and identifies the CS courses that can be taken as part of an ICT track of other study programs.