

UNIVERSITY OF PADERBORN

FACULTY FOR COMPUTER SCIENCE, ELECTRICAL ENGINEERING AND MATHEMATICS
DEPARTMENT OF COMPUTER SCIENCE

MODULE HANDBOOK
MASTER PROGRAM COMPUTER SCIENCE v3 (IMA v3), ENGLISH

DATE: 5. OKTOBER 2021

Inhaltsverzeichnis

1	Preamble and Notes	3
2	Pflichtmodule	4
3	Wahlpflichtmodule	14
4	Focus Areas	126
4.1	Algorithm Design	126
4.2	Computer Systems	127
4.3	Data Science	127
4.4	Intelligence and Data	128
4.5	Networks and Communication	129
4.6	Software Engineering	130
5	Modules in Winter Semester	131
6	Modules in Summer Semester	132
7	Modules in English	133

1 Preamble and Notes

For technical reasons, the preamble of the module handbook is maintained separately. It can be found under Module handbook computer science on the pages of the study program of the Institute of Computer Science. We kindly ask you to read this preamble. If you have any questions regarding this preamble, please contact the Computer Science Academic Advising.

Please also note that

1. this module handbook lists all modules provided for in the examination regulations, even if they are not offered in the corresponding semester.
2. this module handbook contains the data available at the time of writing. All information is without liability.

2 Pflichtmodule

Master-Abschlussarbeit						
Master Thesis						
Module number:	Workload (h):	Credits:	Regular Cycle:			
	900	30	summer- / winter term			
	Semester number:	Duration (in sem.):	Teaching Language:			
	4. Semester	1	en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Master Thesis – Work Plan		30	120	C	1
	b) Master Thesis		30	720	C	1
2	Options within the module:					
	none					
3	Admission requirements:					
	Module examinations in the major subject of at least 48 credits must have been successfully completed. At least three modules in the specialization area must be successfully completed.					
4	Contents:					
	<i>Contents of the course Master-Abschlussarbeit – Arbeitsplan:</i>					
	After having agreed on a topic, the student draws up a work plan. The work plan includes the targeted results, the techniques and methods used and important milestones.					
	<i>Contents of the course Master-Abschlussarbeit:</i>					
	The Master Thesis consists of working on a demanding subject, including a written report and an oral presentation. With the thesis the student shows her/his ability to work independently and systematically on a demanding topic which also includes developing her/his own ideas. On a state-of-the-art basis the methods of computer science should be applied systematically. Topics for master theses are published regularly on the webpages of the research groups in the Department for Computer Science.					

2 Pflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Finishing their master thesis students show that they are able</p> <ul style="list-style-type: none"> • to solve a problem in computer science within an appropriate time frame using scientifically sound Methode • to apply the techniques and methods that they learned during their studies to a new and demanding problem. <p>Non-cognitive Skills:</p> <ul style="list-style-type: none"> • Commitment • Learning competence • Learning motivation • Motivation • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">zu</th> <th style="text-align: center;">Type of examination</th> <th style="text-align: center;">Duration or scope</th> <th style="text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a) - b)</td> <td style="text-align: center;">Thesis</td> <td style="text-align: center;">30-120 pages</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table>	zu	Type of examination	Duration or scope	Weighting for the module grade	a) - b)	Thesis	30-120 pages	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a) - b)	Thesis	30-120 pages	100%						
7	<p>Study Achievement:</p> <p>none</p>								
8	<p>Prerequisites for participation in examinations:</p> <p>none</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted with 50 credits.</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Studiengangsbeauftragter Informatik</p>								

2 Pflichtmodule

13	<p>Other Notes:</p> <p>The master's thesis is an examination paper that concludes the scientific education and is intended to show that the candidate has the ability to work on a problem in computer science according to scientific methods within a certain period of time. The assignment should be designed to be equivalent to five months of full-time work. The thesis must be submitted five months after having been issued. As a rule, the thesis should not exceed 120 DIN A4 pages.</p> <p><i>Remarks of course Master-Abschlussarbeit – Arbeitsplan:</i></p> <p>Implementation method In agreement with supervisor.</p> <p>Learning Material, Literature Depending on the topic.</p> <p><i>Remarks of course Master-Abschlussarbeit:</i></p> <p>Implementation method Independent studies supported by individual advice and supervision</p> <p>Learning Material, Literature Depending on the thesis topic.</p>
----	--

Projektgruppe						
Project Group						
Module number:	Workload (h):	Credits:	Regular Cycle:			
	600	20	summer- / winter term			
	Semester number:	Duration (in sem.):	Teaching Language:			
	2	2	en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Project Group	PG	240	360	C	15
2	Options within the module:					
	none					
3	Admission requirements:					
	<i>Prerequisites of course Projektgruppe:</i>					
	Recommended Proficiencies					
	Depending on the topic.					

2 Pflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Projektgruppe:</i></p> <p>In a project group a group of usually 8-16 students works together over a period of one year (two semesters) on a research topic determined by the group organizer. Project groups introduce students to current research topics that are usually related to the group organizer's special area of interest and the team working of the project group should be a preparation for industrial practice. Topics of project groups cover the whole range of research interests of the research groups in the Department of Computer Science.</p>								
5	<p>Learning outcomes and competences:</p> <p>In project groups, participating students gain first-hand practical experience in working in a team and organizing a project; in doing so, they become prepared for daily work in their later professions. The students personally experience how to carry out extensive development processes in a team. Since the tasks are divided among the individual team members, the participating students become skilled in reporting their progress and research findings to the other group members.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Team work • Learning competence • Learning motivation • Motivation • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">zu</th> <th style="text-align: center;">Type of examination</th> <th style="text-align: center;">Duration or scope</th> <th style="text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Partial Module Exam</td> <td></td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>In the Project Group module, the successful completion of projects must be demonstrated by submitting software and documentation as a phase-related examination. A grade is awarded for the entirety of the projects worked on. The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Partial Module Exam		100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Partial Module Exam		100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">zu</th> <th style="text-align: center;">Type of achievement</th> <th style="text-align: center;">Duration or Scope</th> <th style="text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Practical work</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Practical work		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Practical work		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								

2 Pflichtmodule

9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade:</p> <p>The module is weighted as 8 credits.</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>
12	<p>Module coordinator:</p> <p>Studiengangsbeauftragter Informatik</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Projektgruppe:</i> Implementation method</p> <ul style="list-style-type: none"> • The number of participants is limited to 16 people. • Developing knowledge on the selected systematic approaches, methods and tools relevant to the research topic- usually done in an introductory seminar phase. • Logical assigning “jobs” (assigning responsibilities to the individual group members). • Discovering and promoting the participants’ special individual talents, which are either already apparent or which can be developed throughout the project - such as through seminar presentations or appropriate job assignments. • Setting up a process-oriented personnel structure, similar to the structure of an industrial design team; delegating subtasks to smaller subgroups who report their findings. • Regular progress reports made by individuals and subgroups. • Writing a highly distributed interim report and final report. <p>Learning Material, Literature Depending on the topic.</p>

Seminar I							
Seminar I							
Module number:		Workload (h):		Credits:		Regular Cycle:	
		150		5		summer- / winter term	
		Semester number:		Duration (in sem.):		Teaching Language:	
		2		1		en	
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Seminar	S2	30	120	C	15
2	Options within the module:						
	Seminars from the Master Program Computer Science.						

2 Pflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Seminar:</i></p> <p>Recommended Proficiencies Depending on the seminar topic.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Seminar:</i></p> <p>In seminars, students work independently on an individual research topic by using background literature from various sources. They describe their research topic in a presentation followed by discussion and a written report. The presentation material and the written report serve two different purposes: Whereas the presentation material supports the lectures (held within a specific time period), the written report provides students the opportunity to acquire detailed information on the reported topic at a later date.</p> <p>Seminars usually consist of 8 to 15 related subtopics, each of which is researched by one participating students. Seminar topics cover the whole spectrum of research topics of the research groups in the Department of Computer Science.</p>								
5	<p>Learning outcomes and competences:</p> <p>In seminars, students learn the techniques for independent research work on non-trivial topics and how to present these topics in a presentation and in written form. The seminar participants are encouraged to familiarize themselves with a research-oriented subfield of computer science. They learn how to plan a presentation and hold it within the determined time frame (usually 45 to 60 minutes), and to prioritize the contents of the presentation. The participants experience how an audience obtains knowledge from a presentation, and how to exchange opinions and information in discussions. Seminars also teach rhetorical skills for presentations and discussions. Participating students learn how to structure a presentation according to its contents and how to use various means to illustrate complex issues. They also learn how to handle the background literature appropriately.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Cooperation • Learning competence • Media competence • Motivation • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Presentation and seminar paper</td> <td style="text-align: center;">45-60 minutes, 15-30 pages</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Presentation and seminar paper	45-60 minutes, 15-30 pages	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Presentation and seminar paper	45-60 minutes, 15-30 pages	100%						

2 Pflichtmodule

7	Study Achievement: none
8	Prerequisites for participation in examinations: none
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).
11	Reuse in degree courses: Masterstudiengang Informatik v4
12	Module coordinator: Studiengangsbeauftragter Informatik
13	Other Notes: <i>Remarks of course Seminar:</i> Implementation method Seminar paper and presentation Learning Material, Literature Depending on the seminar topic.

Seminar II							
Seminar II							
Module number:	Workload (h): 150	Credits: 5	Regular Cycle: summer- / winter term				
	Semester number: 3	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Seminar	S2	30	120	C	15
2	Options within the module: Seminars from the Master Program Computer Science.						
3	Admission requirements: <i>Prerequisites of course Seminar:</i> Recommended Proficiencies Depending on the seminar topic.						

2 Pflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Seminar:</i></p> <p>In seminars, students work independently on an individual research topic by using background literature from various sources. They describe their research topic in a presentation followed by discussion and a written report. The presentation material and the written report serve two different purposes: Whereas the presentation material supports the lectures (held within a specific time period), the written report provides students the opportunity to acquire detailed information on the reported topic at a later date.</p> <p>Seminars usually consist of 8 to 15 related subtopics, each of which is researched by one participating students. Seminar topics cover the whole spectrum of research topics of the research groups in the Department of Computer Science.</p>								
5	<p>Learning outcomes and competences:</p> <p>In seminars, students learn the techniques for independent research work on non-trivial topics and how to present these topics in a presentation and in written form. The seminar participants are encouraged to familiarize themselves with a research-oriented subfield of computer science. They learn how to plan a presentation and hold it within the determined time frame (usually 45 to 60 minutes), and to prioritize the contents of the presentation. The participants experience how an audience obtains knowledge from a presentation, and how to exchange opinions and information in discussions. Seminars also teach rhetorical skills for presentations and discussions. Participating students learn how to structure a presentation according to its contents and how to use various means to illustrate complex issues. They also learn how to handle the background literature appropriately.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Cooperation • Learning competence • Media competence • Motivation • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">zu</th> <th style="text-align: center;">Type of examination</th> <th style="text-align: center;">Duration or scope</th> <th style="text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Presentation and seminar paper</td> <td style="text-align: center;">45-60 minutes, 15-30 pages</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Presentation and seminar paper	45-60 minutes, 15-30 pages	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Presentation and seminar paper	45-60 minutes, 15-30 pages	100%						
7	<p>Study Achievement:</p> <p>none</p>								
8	<p>Prerequisites for participation in examinations:</p> <p>none</p>								

2 Pflichtmodule

9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>
12	<p>Module coordinator:</p> <p>Studiengangsbeauftragter Informatik</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Seminar:</i></p> <p>Implementation method Seminar paper and presentation</p> <p>Learning Material, Literature Depending on the seminar topic.</p>

Studium Generale – Master							
General Studies – Master							
Module number:	Workload (h): 360	Credits: 12	Regular Cycle: summer- / winter term				
	Semester number:	Duration (in sem.): 4	Teaching Language: de / en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Studium Generale – Master	L6 Ex3	135	225	C	30
2	<p>Options within the module:</p> <p>Any courses outside of computer science may be chosen.</p>						
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Studium Generale – Master:</i></p> <p>Recommended Proficiencies Depending on the courses chosen.</p>						
4	<p>Contents:</p> <p><i>Contents of the course Studium Generale – Master:</i> Depending on the courses chosen.</p>						

2 Pflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Students expand their scientific horizons beyond the boundaries of computer science and their chosen minor. Depending on the chosen course, they have acquired competencies in communication skills, teamwork and presentation techniques.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Cooperation • Media competence • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Exam in general studies</td> <td></td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	ZU	Type of examination	Duration or scope	Weighting for the module grade	a)	Exam in general studies		100%
ZU	Type of examination	Duration or scope	Weighting for the module grade						
a)	Exam in general studies		100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Qualified participation in general studies</td> <td></td> <td style="text-align: center;">QP</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the qualified participation will be conducted.</p>	ZU	Type of achievement	Duration or Scope	SL / QT	a)	Qualified participation in general studies		QP
ZU	Type of achievement	Duration or Scope	SL / QT						
a)	Qualified participation in general studies		QP						
8	<p>Prerequisites for participation in examinations:</p> <p>none</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted as 4 credits.</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Studiengangsbeauftragter Informatik</p>								
13	<p>Other Notes:</p> <p>If no minor subject is selected, any combination of courses outside of computer science and in the scope of 12 LP must be selected. The given distribution of the LP to courses is only exemplary.</p>								

3 Wahlpflichtmodule

Advanced Algorithms						
Advanced Algorithms						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Advanced Algorithms	L3 Ex2	75	105	CE	50/25
2	Options within the module: none					
3	Admission requirements: <i>Prerequisites of course Advanced Algorithms:</i> Recommended Proficiencies Willingness and ability to learn the creative process of algorithm design and efficiency analysis using mathematical methods. Basic Knowledge of some basic algorithms and data structures and their analyses is assumed.					
4	Contents: <i>Contents of the course Advanced Algorithms:</i> This course presents advanced algorithms and algorithmic paradigms for basic problems. In particular, methods such as randomization and derandomization, as well as the concepts of approximation and online algorithms, are presented using important algorithmic problems. In all cases, proof of correctness and run-time analyzes are carried out. <ul style="list-style-type: none"> • Randomized algorithms, derandomization, examples: Randomized Rounding and others • Online algorithms, examples: scheduling problems and others • approximation algorithms, examples: NP-hard problems 					

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Students apply advanced algorithmic design methods such as randomization, approximation, and online algorithms to new problems and analyze them using combinatorial and probabilistic methods.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Learning motivation 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of achievement</th> <th style="width: 20%; text-align: center;">Duration or Scope</th> <th style="width: 25%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Prof. Dr. Friedhelm Meyer auf der Heide</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Advanced Algorithms:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture with beamer and blackboard. • Exercises in small groups. • expected student activities: active participation in exercises, homework. • Exercise sheets, solutions are presented and discussed in tutorials. • In exercises and homework, design and analysis of algorithms are practiced on selected examples. <p>Learning Material, Literature</p> <p>Standard textbooks, slides of the lecture, exercise sheets</p>
----	--

Advanced Complexity Theory							
Advanced Complexity Theory							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer- / winter term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Advanced Complexity Theory	L3 Ex2	75	105	CE	25	
2	Options within the module: none						
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Advanced Complexity Theory:</i></p> <p>Recommended Proficiencies</p> <p>Basic knowledge about complexity theory (e.g., Turing machines, NP-completeness)</p>						

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Advanced Complexity Theory:</i></p> <p>Complexity Theory deals with determining the amount of resources (e.g. runtime, memory consumption) necessary and sufficient for solving a given algorithmic problem (e.g. Travelling Salesperson Problem (TSP)) on a given machine model (e.g. Turing machine). One approach is to define complexity classes like P, NP, PSPACE, in order to classify problem complexity by means of completeness in such classes, like the famous class of NP-complete problems. This gives conditional results like “If NP is not equal P, then TSP is not solvable in polynomial time.” This branch of Complexity Theory is often referred to as Structural Complexity Theory. In contrast, proving explicit lower bounds for given problems is the topic of the so-called Concrete Complexity Theory. As nobody is currently able to prove superlinear time bounds for explicitly defined problems on general computation models like Turing machines, one considers somewhat restricted models like 1-tape Turing machines, monotone Boolean circuits, Boolean circuits with bounded depth, algebraic computation models, and several kinds of parallel computation models. This lecture surveys approaches to prove such lower bound on various such models.</p> <ul style="list-style-type: none"> • Deterministic, non-deterministic and probabilistic time and space complexity classes, hierarchies, completeness • Lower bounds for size and depth of different variants of Boolean circuits • Lower bounds for algebraic computations 								
5	<p>Learning outcomes and competences:</p> <p>The students get to know fundamental techniques in the area of complexity theory. They can decide in which complexity class the storage space and the runtime requirements of algorithmic problems can be classified. They can classify new problems into complexity classes.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Attitude • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td>CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						

3 Wahlpflichtmodule

8	<p>Prerequisites for participation in examinations: Passing of course achievement</p>
9	<p>Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade: The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses: Masterstudiengang Informatik v4</p>
12	<p>Module coordinator: Prof. Dr. Friedhelm Meyer auf der Heide</p>
13	<p>Other Notes: <i>Remarks of course Advanced Complexity Theory:</i> Implementation method</p> <ul style="list-style-type: none"> • Lecture with beamer and blackboard • Practice in small groups • Expected activities of the students: contributions to presence exercises, homework <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • C.H. Papadimitriou, Computational Complexity, Addison-Wesley • S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press • Slides of the lecture, exercise sheets

Advanced Computer Architecture							
Advanced Computer Architecture							
Module number:		Workload (h): 180		Credits: 6		Regular Cycle: winter term	
		Semester number:		Duration (in sem.): 1		Teaching Language: en	
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Advanced Computer Architecture	L3 Ex2	75	105	CE	50/25	

3 Wahlpflichtmodule

2	<p>Options within the module:</p> <p>none</p>								
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Advanced Computer Architecture:</i></p> <p>Recommended Proficiencies</p> <p>Basic knowledge in computer architecture.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Advanced Computer Architecture:</i></p> <p>The course teaches concepts and methods used in modern processor architecture to exploit the available parallelism at the levels of instructions, data and threads.</p> <ul style="list-style-type: none"> • Fundamentals of computer architectures (refresher) • Memory hierarchy design • Instruction-level parallelism • Data-level parallelism: Vector, SIMD and GPU architectures • Thread-level parallelism • Warehouse-scale computer 								
5	<p>Learning outcomes and competences:</p> <p>After attending the course, the students</p> <ul style="list-style-type: none"> • are able to explain principles of modern memory hierarchies, • to analyze different levels of parallelism, • to assess the suitability of different architectural concepts and thus • to evaluate modern developments in computer architecture. <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 25%; text-align: center;">Duration or scope</th> <th style="width: 20%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
	Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.		
8	Prerequisites for participation in examinations:		
	Passing of course achievement		
9	Prerequisites for assigning credits:		
	The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade:		
	The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses:		
	Masterstudiengang Informatik v4		
12	Module coordinator:		
	Prof. Dr. Marco Platzner		
13	Other Notes:		
	<i>Remarks of course Advanced Computer Architecture:</i>		
	Implementation method		
	<ul style="list-style-type: none"> • Lecture with projector and board • Interactive exercises in the lecture room item Computer-based exercises with simulation tools • Analysis of case studies 		
	Learning Material, Literature		
	<ul style="list-style-type: none"> • Lecture slides and exercise sheets • Exercise sheets and technical documentation for the for the computer-based exercises • Hennessey, Patterson: Computer Architecture: A Quantitative Approach (5th edition or newer), Morgan Kaufmann, 2012. • Information about alternative and additional literature as well as teaching material on the course's website and in the lecture slides 		

Advanced Distributed Algorithms and Data Structures			
Advanced Distributed Algorithms and Data Structures			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	winter term

3 Wahlpflichtmodule

	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Advanced Distributed Algorithms and Data Structures	L3 Ex2	75	105	CE	30
2	Options within the module:						
	none						
3	Admission requirements:						
	<i>Prerequisites of course Advanced Distributed Algorithms and Data Structures:</i>						
	Recommended Proficiencies						
	Algorithms and data structures, distributed algorithms and data structures						
4	Contents:						
	<p>The lecture will cover advanced topics in distributed algorithms and data structures. Topics covered in the course are access control, synchronization, consensus, information dissemination, hybrid networks, scheduling, and optimization. In addition to presenting solutions to these topics, also concrete applications will be presented.</p> <p><i>Contents of the course Advanced Distributed Algorithms and Data Structures:</i></p> <p>The lecture will cover advanced topics in distributed algorithms and data structures. Topics covered in the course are access control, synchronization, consensus, information dissemination, hybrid networks, scheduling, and optimization. In addition to presenting solutions to these topics, also concrete applications will be presented.</p>						
5	Learning outcomes and competences:						
	<p>Students get to know advanced methods and algorithms for currently very relevant distributed systems. They are able to adapt algorithms to new situations and to determine their complexity. They can implement basic distributed algorithms.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence • Literacy (scientific) • Self-monitoring 						

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Prof. Dr. Christian Scheideler		
13	Other Notes: <i>Remarks of course Advanced Distributed Algorithms and Data Structures:</i> Implementation method Lecture with tutorials and software project Learning Material, Literature Lectures notes		

Algorithms for Highly Complex Virtual Scenes			
Algorithms for Highly Complex Virtual Scenes			
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term

3 Wahlpflichtmodule

	Semester number:	Duration (in sem.):	Teaching Language:
	1	1	en
1	Module structure:		
	Course	form of teaching	contact-time (h)
	self-study (h)	status (C/CE)	group size (TN)
a)	Algorithms for Highly Complex Virtual Scenes	L3 Ex2	75
		105	CE
	40/20		
2	Options within the module: none		
3	Admission requirements: <i>Prerequisites of course Algorithms for Highly Complex Virtual Scenes:</i> Recommended Proficiencies Willingness and ability to learn the creative process of algorithm design and efficiency analysis using mathematical methods. Basic Knowledge of some basic algorithms and data structures and their analyses is assumed.		
4	Contents: <i>Contents of the course Algorithms for Highly Complex Virtual Scenes:</i> Walkthrough systems allow viewing and walking through a virtual 3D scene and finds application in architecture programs, simulations or games. The efficiency of real-time rendering algorithms is crucial for a smooth and fast visualization of the virtual 3D scene in a walkthrough system. There are different algorithmic approaches to reduce highly complex 3D geometric data and to achieve a rendering of the scene in real time. The lecture introduces algorithmic approaches in the areas of visibility culling, simplification, level of detail, image-based rendering and further approaches. <ul style="list-style-type: none"> • Introduction: walkthrough problem • Data structures: kd-tree, BSP-tree, octree, loose octree • Level of detail: adaptive LOD management, mesh simplification, progressive meshes • Visibility culling: view frustum culling, potentially visible sets (PVS), dynamic analysis of PVS, Hierarchical z-buffer, hierarchical occlusion maps, aspect graph, visibility space partition • Replacement: color cubes, randomized z-buffer, hierarchical image caching • Parallel rendering: classification and modeling, parallel rendering as a sorting problem, hybrid sort-first/sort-last rendering 		
5	Learning outcomes and competences: The students can apply fundamental techniques in the area of real time rendering of virtual 3D scenes. They can decide in which virtual 3D scene which algorithm is most appropriate. They can adapt algorithms to a new situation. Non-cognitive Skills <ul style="list-style-type: none"> • Attitude • Self-monitoring 		

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Dr. Matthias Fischer		

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Algorithms for Highly Complex Virtual Scenes:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture with beamer and blackboard • Practice in small groups • Expected activities of the students: Collaboration in presence exercises Homework • exercise sheets, sample solutions are presented in central exercises • In exercises and homework sheets and teh analysis of algorithms of selected examples are practiced. <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Standard textbooks, slides of the lecture, exercise sheets • Real-Time Rendering; Tomas Akenine-Möller, Eric Haines; AK Peters, 2002. • Level of Detail for 3D Graphics; David Luebke, Martin Reddy, Jonathan D. Cohen; Morgan Kaufmann Publishers, 2002.
----	---

Algorithms for Synthesis and Optimization of Integrated Circuits							
Algorithms for Synthesis and Optimization of Integrated Circuits							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Algorithms for Synthesis and Optimization of Integrated Circuits	L3 Ex2	75	105	CE	30
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Algorithms for Synthesis and Optimization of Integrated Circuits:</i> Recommended Proficiencies Knowledge of "Digital Design" is beneficial.						

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Algorithms for Synthesis and Optimization of Integrated Circuits:</i> The course provides the most remarkable features of digital synthesis, and explains the details of transforming hardware description languages into circuit descriptions. Besides, the major techniques for logic optimization are discussed, and then the efficient use of current design tools are exercised in practical sessions.</p> <ul style="list-style-type: none"> • Hardware modeling languages • High-level synthesis and optimization methods (i.e., scheduling and binding) • Logic Representation and optimization of two-level logic functions • Data structures for logic synthesis (Binary decision diagrams) • Representation and optimization of multiple-level logic networks (Algebraic methods, controllability and observability computation, and timing verification) • Modeling and optimization of sequential logic networks (Retiming) • Libraries and binding 								
5	<p>Learning outcomes and competences:</p> <p>After attending the course, the students are able to</p> <ul style="list-style-type: none"> • select among the available optimisation methods in design of digital circuits, • identify major problems in design of integrated circuits and recognize circuit design tradeoffs • examine current digital design tools and methods <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 25%; text-align: center;">Duration or scope</th> <th style="width: 20%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <p>none</p>								
8	<p>Prerequisites for participation in examinations:</p> <p>none</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								

3 Wahlpflichtmodule

10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>
12	<p>Module coordinator:</p> <p>Dr. Hassan Ghasemzadeh Mohammadi</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Algorithms for Synthesis and Optimization of Integrated Circuits:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture with projector and board • Interactive exercises in the lecture room • Computer-based exercises with hardware synthesis tools <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Lecture slides and exercise sheets • Exercise sheets and technical documentation for the for the computer-based exercises • Micheli, Giovanni De. Synthesis and optimization of digital circuits. McGraw-Hill Higher Education, 1994. • Information about alternative and additional literature as well as teaching material on the course's website and in the lecture slides

Approximate Computing							
Approximate Computing							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	winter term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Approximate Computing	L3 Ex2	75	105	CE	100/25
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Approximate Computing:</i></p> <p>Recommended Proficiencies Bachelor-level knowledge in digital design and computer architecture, Bachelor-level knowledge in mathematics, in particular linear algebra and probability theory</p>
4	<p>Contents:</p> <p><i>Contents of the course Approximate Computing:</i></p> <p>Approximate Computing is an emerging paradigm that trades-off computational accuracy for a significant reduction in energy, execution time, or chip area. This research-oriented course introduces to the field of Approximate Computing and its most remarkable aspects, and explains the main methods used to implement efficient computing systems by reducing accuracy. The course discusses approximations at all levels of a computing system, from applications down to hardware technologies. In exercise/tutorial sessions the efficiency of these techniques in various domains are examined, including deep learning and digital signal processing.</p> <ul style="list-style-type: none"> • Introduction and motivation for inexact computing • Approximation at the application level • Programming languages/compilers for approximate computing • Approximate microarchitectures • Synthesis of approximate circuits • Inexact arithmetic components and performance optimization via accuracy trade-offs • Approximation techniques at the technology level • Exercises/tutorial: Approximating deep learning and digital signal processing algorithms at the application and architecture levels
5	<p>Learning outcomes and competences:</p> <p>After attending this course, the students are able</p> <ul style="list-style-type: none"> • to name and explain approximation techniques at all levels of a computing system, • to identify major engineering/research problems when building approximate computing systems, • to judge the suitability of approximation techniques for different application domains, and • to apply approximation techniques to realize efficient hardware accelerators, in particular for deep learning and digital signal processing <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Dr. Hassan Ghasemzadeh Mohammadi		

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Approximate Computing:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture with projector and black/white board • Interactive exercises/discussions in the lecture room • Computer-based tutorials <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Lecture slides, exercise sheets, and tutorial assignments • Adrian Sampson, Luis Ceze, and Dan Grossman: Good-Enough Computing. IEEE Spectrum, 50(10):54-59, 2013 • Ravi Nair. Big Data Needs Approximate Computing: Technical Perspective. Communications of the ACM, 58(1): 104, 2015. • Sparsh Mittal. A Survey of Techniques for Approximate Computing. ACM Computing Surveys, 48(4), 2016. • Qiang Xu, Todd Mytkowitz, and Nam Sung Kim. Approximate Computing: A Survey. IEEE Design & Test, 33(1):8-22, 2016. • Additional resources and links to current research papers are provided in the lecture.
----	--

Architektur paralleler Rechnersysteme						
Architectures of Parallel Computer Systems						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term			
	Semester number:	Duration (in sem.): 1	Teaching Language: de			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Architecture of Parallel Computer Systems	L3 Ex2	75	105	CE	20
2	Options within the module: none					
3	Admission requirements: <i>Prerequisites of course Architektur paralleler Rechnersysteme:</i> Recommended Proficiencies Principles of computer architectures					

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Architektur paralleler Rechnersysteme:</i></p> <p>The lecture considers computer architectures of actual parallel computer systems and the usage of this systems. The focus of the lecture is on high-performance computers (supercomputers).</p> <ul style="list-style-type: none">• Users' view on Parallel Computers• Programming of Parallel Computers• Principles of Computer Architecture• Overview of actual Parallel Computer Systems• Shared Memory Systems• Cache Coherency of Scalabel Computer Systems• High Performance Interconnects• Datacenter Architectures• Energy Efficiency
5	<p>Learning outcomes and competences:</p> <ul style="list-style-type: none">• Students name and explain programming paradigms of parallel programming languages. They master basic constructions of the languages and library functions of the most important parallel programming languages and environments (e.g. OpenMP, POSIX-Threads, MPI, PGAS) and name the areas of applications.• Students are able to describe the features of actual HPC systems and processors. They describe major underlying trends (power wall, Memory wall, ILP wall) of the systems.• Students name and describe General used classifications of parallel systems. They describe the important structure elements of Operation principles of parallel computer systems. They master the theoretical behavior of scaling (Amdahl, Gustafson) and quantitative Evaluations of parallel computers.• Students name and describe architectural characteristics of scalable shared memory systems. They master different techniques to maintain memory consistency and coherency in bus based systems (Invalidation protocols, update protocols). They describe technics to increase the Performance of These systems (Multi Level caches, transient states, split Transaction busses).• Students describe mechanisms to establish synchronizations (locks, barriers) in parallel systems.• Students demonstrate knowledge and understanding of in maintaining cache coherency within scalable computer systems (hierarchical snooping, directories). They master techniques to increase the performance of these systems (e.g. latency, throughput).• Students describe techniques based on token coherency.• Students name and describe fundamental features of the architectures of cluster systems. They are able to describe the topology of communication systems and metrics for evaluation (e.g. degree, diameter, bisection). They master communication techniques of high performance interconnect networks (e.g. wormhole Routing, virtual cut-through). They master techniques to proof are deadlock are avoided.• Students describe features of existing interconnects (InfiniBand, OmniPath). <p>Non-cognitive Skills</p> <ul style="list-style-type: none">• Commitment• Learning competence

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)			
	zu	Type of examination	Duration or scope	Weighting for the module grade
	a)	Written or oral examination	90-120 min or 40 min	100%
<p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>				
7	Study Achievement:			
	zu	Type of achievement	Duration or Scope	SL / QT
	a)	Written exercises		CA
<p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>				
8	Prerequisites for participation in examinations: Passing of course achievement			
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.			
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses: Masterstudiengang Informatik v4			
12	Module coordinator: Dr. Jens Simon			
13	Other Notes: <i>Remarks of course Architektur paralleler Rechnersysteme:</i> Implementation method Presentation of slides. Exercises on available high performance computers to practise the usage of the systems and deepen the knowledge of the lecture. Learning Material, Literature Slides			

Ausgewählte Themen im Gebiet Software Engineering			
Selected Topics in Software Engineering			
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer- / winter term

3 Wahlpflichtmodule

	Semester number:	Duration (in sem.):	Teaching Language:			
		1	de / en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Software Engineering for Self-Adaptive Systems	L3 Ex2	75	105	CE	40
2	Options within the module: Exactly one of the courses included in these module is to be selected.					
3	Admission requirements: <i>Prerequisites of course Software Engineering for Self-Adaptive Systems:</i> Recommended Proficiencies Programming and modeling					
4	Contents: In this module, varying courses are offered, for example by guest lecturers. For specific contents see the course descriptions. <i>Contents of the course Software Engineering for Self-Adaptive Systems:</i> The complexity of software systems, the evolution of requirements, and the uncertainty of requirements and environments challenge the management of running systems. One promising solution are self-adaptive systems, that is, systems that are able to adapt their behavior in response to changing and uncertain requirements and environments. In this lecture we will discuss the foundations as well as methods and techniques to engineer and assure self-adaptive systems. The focus will be on the software engineering aspects and how self-adaptation impacts typical software engineering activities. A selection of methods and techniques will be exemplified on research prototypes (so-called exemplars) in the tutorials and project.					
5	Learning outcomes and competences: Non-cognitive Skills <ul style="list-style-type: none">• Attitude• Self-monitoring					
6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)					
	zu	Type of examination	Duration or scope	Weighting for the module grade		
		a) Written or oral examination	90-120 min or 40 min	100%		
	The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.					

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
	Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.		
8	Prerequisites for participation in examinations:		
	Passing of course achievement		
9	Prerequisites for assigning credits:		
	The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade:		
	The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses:		
	keine		
12	Module coordinator:		
	Studiengangsbeauftragter Informatik		
13	Other Notes:		
	<p><i>Remarks of course Software Engineering for Self-Adaptive Systems:</i></p> <p>Implementation method</p> <p>The lecture will introduce the foundations, methods, and techniques for developing and assuring self-adaptive systems, which will be exemplified in the tutorials and project to gain practical experience. This exemplification will be based on so-called exemplars originating from the research community of self-adaptive software systems (http://self-adaptive.org/exemplars/).</p> <p>Learning Material, Literature</p> <p>Lecture slides, pointers to academic papers will be given in the lecture, text books:</p> <ul style="list-style-type: none"> • Weyns, Danny (2020). An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective. Wiley • Cheng, B.H.C., de Lemos, R., Inverardi, P., Magee, J. (Eds.). (2009) Software Engineering for Self-Adaptive Systems. Springer. • de Lemos, R., Giese, H., Müller, H., Shaw, M. (Eds.) (2013). Software Engineering for Self-Adaptive Systems II. Springer. • de Lemos, R., Garlan, D., Ghezzi, C., Giese, H. (Eds.) (2017). Software Engineering for Self-Adaptive Systems III. Assurances. Springer 		

Build It, Break It, Fix It			
Build It, Break It, Fix It			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	summer term

3 Wahlpflichtmodule

	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Build It, Break It, Fix It	L3 Ex2	75	105	CE	20
2	Options within the module:						
	none						
3	Admission requirements:						
	<i>Prerequisites of course Build It, Break It, Fix It:</i>						
	Recommended Proficiencies						
	<ul style="list-style-type: none"> • Being able and willing to get into new topics independently is mandatory. • Mature coding skills in at least one popular programming language (Java, Python, C, C++) is mandatory. • Knowledge about software security requirements, secure software development practices and cryptography as well as experience in the field of software exploitation and vulnerability discovery will be helpful. 						
4	Contents:						
	<i>Contents of the course Build It, Break It, Fix It:</i>						
	<p>This course aims at teaching basic principles of secure software development in a very practical fashion. It is based on the “Break It, Build It, Fix It” security contest by Ruef et al. The contest is separated into three phases that test the applicant’s skills in the fields of building, breaking and fixing software products.</p> <p>In the “Build It” phase, students will be asked to gather in teams and develop small software projects based on a formal specification, also including security requirements. In the “Break It” phase, the developed software will be exchanged between development teams to break the implementation, i.e., find and exploit security vulnerabilities in code of other teams. Afterward, in the “Fix It” phase, teams will get the chance to fix found vulnerabilities and, hence, render their software product more secure.</p> <p>The course will contain a theoretical part in which basic strategies of secure software development and vulnerability discovery are presented. Furthermore, specific vulnerability classes and examples of their exploitation will be presented as stimulus at the beginning of the “Break It” phase. Nevertheless, the course is generally of a very practical nature and since securing a software product, as well as breaking it, demands a wide variety of skills and creativity, a high amount of motivation and self-organization is required.</p>						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>After having attended this course, students will have. . .</p> <ul style="list-style-type: none"> • gained knowledge and experience in the field of secure software development • gained knowledge and experience in the field of software exploitation as well as vulnerability discovery • learned common real world software vulnerabilities and ways of exploiting them <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Cooperation • Learning competence • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Partial Module Exam</td> <td></td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	ZU	Type of examination	Duration or scope	Weighting for the module grade	a)	Partial Module Exam		100%
ZU	Type of examination	Duration or scope	Weighting for the module grade						
a)	Partial Module Exam		100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Practical work with subsequent discussion</td> <td></td> <td style="text-align: center;">CA, QP</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	ZU	Type of achievement	Duration or Scope	SL / QT	a)	Practical work with subsequent discussion		CA, QP
ZU	Type of achievement	Duration or Scope	SL / QT						
a)	Practical work with subsequent discussion		CA, QP						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>keine</p>								
12	<p>Module coordinator:</p> <p>Prof. Dr. Eric Bodden</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Build It, Break It, Fix It:</i></p> <p>Learning Material, Literature</p> <p>The lecture slides, task instructions and other materials will be uploaded to the course's PANDA page.</p>
----	--

Clustering Algorithms						
Clustering Algorithms						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer- / winter term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a)	Clustering Algorithms	L3 Ex2	75	105	CE 25
2	Options within the module: none					
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Clustering Algorithms:</i></p> <p>Recommended Proficiencies</p> <p>Algorithms and data structures, linear algebra, probability theory</p>					
4	<p>Contents:</p> <p><i>Contents of the course Clustering Algorithms:</i></p> <p>We study an important tool to analyze data: clustering. Clustering is the process of dividing data into useful or sensible groups. A sensible division should resemble the data's natural structure. Sometimes the goal is that each cluster should contain as many items of a similar kind as possible (for example in data compression). Clustering is a very natural way to analyze and structure data. Especially in natural sciences we are working with data whose structure is unknown to us. An example is the human DNA, that humankind is trying to decode. Clustering can be a very powerful tool in such cases. We re view standard techniques for clustering like Lloyd's algorithm and agglomerative clustering. We also look at extensions and generalizations of clusterings like estimating model parameters.</p> <ul style="list-style-type: none"> • Agglomerative clustering • K-Means clustering • Geometric clustering: DBSCAN • Dimension reduction: Johnson-Littlewood and SVD • Mixture Models and EM algorithm 					

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Students know the main clustering algorithms. Depending on the problem at hand they can decide which algorithm is most appropriate. They can adapt algorithms to new situations. Students know various methods for modeling clustering problems. They can apply and adapt the methods. Depending on the modeling, they can choose appropriate algorithms and assess their quality.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">zu</th> <th style="text-align: center;">Type of examination</th> <th style="text-align: center;">Duration or scope</th> <th style="text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">zu</th> <th style="text-align: center;">Type of achievement</th> <th style="text-align: center;">Duration or Scope</th> <th style="text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td>CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Prof. Dr. Johannes Blömer</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Clustering Algorithms:</i></p> <p>Implementation method Lectures, exercises, reading groups, short presentations</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • David J.C MacKay, Information Theory, Inference and Learning Algorithms • Christopher M. Bishop, Pattern Recognition and Machine Learning • Slides from the lectures
----	---

Combinatorial Optimization						
Combinatorial Optimization						
Module number:	Workload (h):	Credits:	Regular Cycle:			
	180	6	winter term			
	Semester number:	Duration (in sem.):	Teaching Language:			
		1	en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a)	Combinatorial Optimization	L3 Ex2	75	105	CE ??
2	Options within the module:					
	none					
3	Admission requirements:					
	<p><i>Prerequisites of course Combinatorial Optimization:</i></p> <p>Recommended Proficiencies Solid basic knowledge of algorithms and data structures, complexity theory, discrete mathematics, and linear algebra as usually taught in undergraduate courses in computer science.</p>					
4	Contents:					
	<p><i>Contents of the course Combinatorial Optimization:</i></p> <p>This course gives an introduction to combinatorial optimization. Algorithmic approaches and methods are presented and analyzed, and various combinatorial optimization problems are considered. Topics of the course include e.g. matchings, network flows and linear programming.</p>					

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Participants will learn algorithmic approaches to combinatorial optimization problems as well as an understanding of the complexity of the problems and the limitations of the methods presented.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Attitude and mindset • Self-control competence 										
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>			zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade								
a)	Written or oral examination	90-120 min or 40 min	100%								
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of achievement</th> <th style="width: 20%; text-align: center;">Duration or Scope</th> <th style="width: 25%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>			zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT								
a)	Written exercises		CA								
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>										
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>										
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>										
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>										
12	<p>Module coordinator:</p> <p>Dr. Marten Maack</p>										

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Combinatorial Optimization:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture with beamer and/or blackboard or online with appropriate substitutes. • Small group exercises. • Expected student activities: Active participation in classroom exercises and homework assignments. <p>Learning Material, Literature</p> <p>In addition to the usual lecture materials such as lecture notes or slides as well as exercise sheets, we refer to the following textbooks:</p> <ul style="list-style-type: none"> • C. Papadimitriou and K. Steiglitz: Combinatorial Optimization: Algorithms and Complexity. Courier Corporation, 1998. • B. Korte, J. Vygen: Combinatorial optimization: theory and algorithms. Springer-Verlag Berlin Heidelberg, 2009. • A. Schrijver: Combinatorial Optimization: Polyhedra and Efficiency. Springer-Verlag Berlin Heidelberg, 2003.
----	---

Computational Argumentation							
Computational Argumentation							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Computational Argumentation	L3 Ex2	75	105	CE	150/50	
2	Options within the module: none						
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Computational Argumentation:</i></p> <p>Recommended Proficiencies</p> <p>Syllabus of the bachelor course “Introduction to Text Mining” and/or the master course “Statistical Natural Language Processing”. Alternatively, the contents of the lectures “Data Mining” (bachelor), “Machine Learning I”, and “Information Retrieval” (both master) are helpful. Basics of statistics.</p>						

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Computational Argumentation:</i> Argumentation is an integral part of both professional and everyday communication. Whenever a topic or question is subject to controversy, people consider arguments to form opinions, to make decisions, or to convince others of a certain stance. In the last years, the computational analysis and synthesis of natural language argumentation has become an emerging research area, due to its importance for the next generation of web search engines and intelligent personal assistance. Based on fundamental techniques from natural language processing, computational argumentation ranges from the mining of arguments from natural language text, over the assessment of argumentation quality, to the retrieval of arguments in web search.</p>										
5	<p>Learning outcomes and competences:</p> <p>The students learn both fundamentals from argumentation theory and state-of-the-art techniques from computational argumentation. The use of respective methods is practiced in assignments.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Attitude • Self-monitoring 										
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>			zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade								
a)	Written or oral examination	90-120 min or 40 min	100%								
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>			zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT								
a)	Written exercises		CA								
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>										
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>										
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>										
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>										

3 Wahlpflichtmodule

12	Module coordinator: Jun. Prof. Dr. Henning Wachsmuth
13	Other Notes: <i>Remarks of course Computational Argumentation:</i> Implementation method Concepts and methods related to the analysis and synthesis of argumentation are taught in the lecture. In the tutorials, the concepts and methods are applied within programming projects. Learning Material, Literature Material is given in the form of slide-style scripts and exercise sheets. The following book covers parts of the topics of the lecture: Manfred Stede and Jodi Schneider. 2018. Argumentation Mining. Synthesis Lectures on Human Language Technologies #13. Morgan & Claypool Publishers.

Data-Driven Innovation and Engineering							
Data-Driven Innovation and Engineering							
Module number:	Workload (h):	Credits:		Regular Cycle:			
	180	6		winter term			
	Semester number:	Duration (in sem.):		Teaching Language:			
		1		en			
1	Module structure:						
		Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a)	Data-Driven Innovation and Engineering	L2 Ex3	75	105	CE	60
2	Options within the module:						
	none						
3	Admission requirements:						
	none						
4	Contents:						
	<i>Contents of the course Data-Driven Innovation and Engineering:</i> Digitalization is changing future market services and the way they are developed. Traditional methods of strategic planning and system engineering leave potentials unexploited, while data-driven solutions capture these potentials. The lecture provides an overview of the challenges and approaches of Data-Driven Innovation and Engineering. Theoretical foundations and concepts are introduced and exemplary applications from practice are presented. The process from data acquisition and data analysis to the development of innovative market services is considered. The acquired knowledge will be deepened and implemented in the exercises.						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Students gain a comprehensive understanding of data-driven solutions in the area of Data-Driven Innovation and Engineering. Furthermore, students learn how to independently develop data-driven solutions.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Empathy • Social and ethical judgement • Team work • Cooperation • Motivation • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	ZU	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
ZU	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	ZU	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
ZU	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Prof. Dr.-Ing. Roman Dumitrescu</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Data-Driven Innovation and Engineering:</i></p> <p>Implementation method Lecture and exercise</p> <p>Learning Material, Literature Gausemeier, J.; Dumitrescu, R.; Echterfeld, J.; Pfänder, T.; Steffen, D.; Thielemann, F.: Innovationen für die Märkte von morgen – Strategische Planung von Produkten, Dienstleistungen und Geschäftsmodellen. Carl Hanser Verlag, München, 2019</p>
----	--

Data Science for Physics and Engineering							
Data Science for Physics and Engineering							
Module number:	Workload (h):	Credits:		Regular Cycle:			
M.079.4077	180	6		winter term			
	Semester number:	Duration (in sem.):		Teaching Language:			
		1		en			
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Data Science for Physics and Engineering	L3 Ex2	75	105	CE	50	
2	Options within the module:						
	none						
3	Admission requirements:						
	<i>Prerequisites of course Data Science for Physics and Engineering:</i> none						
4	Contents:						
	<p><i>Contents of the course Data Science for Physics and Engineering:</i></p> <p>Data-driven methods are revolutionizing the modeling, prediction, and control of complex systems. In this lecture series, we will cover the basis regarding <i>machine learning, engineering, and mathematical physics</i> to integrate modeling and control of dynamical systems with modern methods in data science. We also highlight many of the recent advances in <i>scientific computing</i> that enable data-driven methods to be applied to a diverse range of complex systems such as fluid dynamics, the brain, climate, epidemiology, finance, and robotics.</p> <p>The covered topics include:</p> <ul style="list-style-type: none"> • Pattern recognition and dimensionality reduction • Coordinate transformations • Machine learning and data analysis • Dynamics and control • Reduced order modeling 						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Students learn</p> <ul style="list-style-type: none"> • factual knowledge about <ul style="list-style-type: none"> – the basis of machine learning methods in the context of technical and physical systems – basic knowledge regarding prediction and control of dynamical systems – mathematical basics of data-driven surrogate modelling • methodological knowledge, including <ul style="list-style-type: none"> – various frequently used approaches for efficient data management and analysis in engineering – the interdisciplinary connection of different methods for solving complex problems • transfer skills <ul style="list-style-type: none"> – the application of the learned methods to new systems and problem classes – the suitable combination of established machine learning methods for new specific problem classes • normative evaluation skills including the ability to assess <ul style="list-style-type: none"> – the suitability of specific methods in terms of the available amount of data, the system complexity, etc. – the effort and feasibility of data-driven projects in engineering <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • learning competence • learning motivation 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <p>none</p>								
8	<p>Prerequisites for participation in examinations:</p> <p>none</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								

3 Wahlpflichtmodule

10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Elektrotechnik v4 (EMA v4), Masterstudiengang Informatik v4, Master's Program Electrical Systems Engineering (ESEMA v2)</p>
12	<p>Module coordinator:</p> <p>Dr. Sebastian Peitz</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Data Science for Physics and Engineering:</i></p> <p>Implementation method Theoretical concepts will be presented during lectures and consolidated in tutorials with exercises as well as programming tasks. Tutorials will be realized via individual work as well as joint discussions.</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Steven L. Brunton and J. Nathan Kutz. Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press, 2019.

Data Science in Industrial Applications							
Data Science in Industrial Applications							
Module number:		Workload (h):		Credits:		Regular Cycle:	
		180		6		winter term	
		Semester number:		Duration (in sem.):		Teaching Language:	
				1		en	
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Data Science in Industrial Applications	L3 Ex2	75	105	CE	40/20
2	Options within the module:						
	none						
3	Admission requirements:						
	<p><i>Prerequisites of course Data Science in Industrial Applications:</i></p> <p>Recommended Proficiencies Foundations of mathematics (linear algebra, statistics), Programming and Algorithms.</p>						

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Data Science in Industrial Applications:</i></p> <p>The increasing connectivity of machines, sensors and IT systems in context of Industry 4.0 has led to a rapid increase in available data volume. The analysis of data offers enormous potential for the automation of cognitive tasks, the optimization of processes and the further value creation from data.</p> <p>The lecture gives an overview of the challenges and approaches for the industrial application of Data Science. This includes the integration of industrial data sources at field level, the IT landscape in manufacturing companies and the setup of (Big Data) infrastructure, typical algorithms in the area of time series processing, optimization or image processing as well as the embedding in business processes.</p> <p>Theoretical and methodical foundations, concepts and tools are introduced during the lecture and applied based on a case study in workshops, team work, as well as in home exercises. Skills in team work and cooperation, self-control and project management are deepened.</p>								
5	<p>Learning outcomes and competences:</p> <p>Students understand the challenges of applying data science methods in industrial context and have an overview of typical use case examples. They are able to apply methods of signal processing, machine learning and statistics to industrial problems and plan the implementation of data acquisition, data architecture and integration into business processes.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Cooperation • Learning competence 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Accompanying case study</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Accompanying case study		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Accompanying case study		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								

3 Wahlpflichtmodule

9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>
12	<p>Module coordinator:</p> <p>Prof. Dr.-Ing. Roman Dumitrescu</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Data Science in Industrial Applications:</i></p> <p>Implementation method Lecture with slides. Basics and concepts are explained in the lecture and illustrated with examples. In the exercise, knowledge is transferred and the concepts are applied to a case study by means of workshops and implementation of an industrial analytics application in self-managed team work.</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Lecture slides and documents for the case study. • Recommended literature is given in the first lecture.

Designing code analyses for large-scale software systems 1							
Designing code analyses for large-scale software systems 1							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	winter term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Designing code analyses for large-scale software systems 1	L3 Ex2	75	105	CE	30
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Designing code analyses for large-scale software systems 1:</i></p> <p>Recommended Proficiencies</p> <p>A mature understanding of the Java programming languages and object-oriented programming will be helpful.</p>
4	<p>Contents:</p> <p><i>Contents of the course Designing code analyses for large-scale software systems 1:</i></p> <p>Static code analysis has the goal of finding programming mistakes automatically, by searching for suspicious anti-patterns in a program's code. This course will explain how to design static code analysis that are inter-procedural, i.e., consider the whole program, across procedure boundaries. Designing such analyses is challenging, as they need to handle millions of program statements efficiently and precisely. Example applications are drawn from the area of IT security. This course is part of a combination DECA 1/2. In DECA 2 we will be covering current approaches directly out of research. We strongly recommend attending DECA 1 before DECA 2.</p> <p>Topics covered include:</p> <ul style="list-style-type: none">• Type systems and flow-insensitive, constraint-based analysis• Lattices and fixed points• Intra-procedural flow-sensitive static code analysis• Interval analysis, widening and narrowing• Call-graph construction• Pointer Analysis• Inter-procedural program analysis• Call-strings approach to context-sensitive analysis• Functional approach to context-sensitive analysis• Value-based termination, VASCO• Distributive analyses using IFDS• Sensible arrangements of Flow Functions• Distributive analyses using IDE <p>Throughout, we will discuss applications to software security.</p>
5	<p>Learning outcomes and competences:</p> <p>After having attended this course, students will have learned. . .</p> <ul style="list-style-type: none">• how to make educated design decisions when designing automated code analysis for large-scale software systems,• which algorithms have which properties when using them to implement static code-analyses,• how to design real-world code analyses for practical problem cases from the area of IT security• how to interpret important terminology such as context, flow, field and object sensitivity• how to evaluate and explain the important limitations of static code analysis• which typical security code analyses exist (OWASP Top 10 etc.) and how they relate to the analysis frameworks explained in the course. <p>Non-cognitive Skills</p> <ul style="list-style-type: none">• Learning competence• Learning motivation

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Prof. Dr. Eric Bodden		

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Designing code analyses for large-scale software systems 1:</i></p> <p>Implementation method Lectures and group exercises as well as practical programming labs using worldwide leading frameworks for static code analysis</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Thomas Reps, Susan Horwitz, and Mooly Sagiv. 1995. Precise interprocedural dataflow analysis via graph reachability. POPL '95 • Shmuel Sagiv, Thomas W. Reps, and Susan Horwitz. 1995. Precise Interprocedural Dataflow Analysis with Applications to Constant Propagation. TAPSOFT '95 • Akash Lal, Thomas Reps, and Gogul Balakrishnan. 2005. Extended weighted pushdown systems. CAV 2005 • Nomair A. Naeem, Ondrej Lhoták, and Jonathan Rodriguez. 2010. Practical extensions to the IFDS algorithm. CC 2010 • Yannis Smaragdakis, Martin Bravenboer, and Ondrej Lhoták. 2011. Pick your contexts well: understanding object-sensitivity. POPL 2011 • Eric Bodden. 2012. Inter-procedural data-flow analysis with IFDS/IDE and Soot. SOAP 2012 • Rohan Padhye, Uday P. Khedker. Interprocedural Data Flow Analysis in Soot using Value Contexts. SOAP 2013
----	--

Designing code analyses for large-scale software systems 2							
Designing code analyses for large-scale software systems 2							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	summer term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Designing code analyses for large-scale software systems 2	L3 Ex2	75	105	CE	30	
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Designing code analyses for large-scale software systems 2:</i></p> <p>Recommended Proficiencies</p> <p>We strongly recommend that attendees have completed DECA 1 beforehand. A mature understanding of the Java and/or C++ programming languages and object-oriented programming will be helpful.</p>
4	<p>Contents:</p> <p><i>Contents of the course Designing code analyses for large-scale software systems 2:</i></p> <p>Static code analysis has the goal of finding programming mistakes automatically, by searching for suspicious anti-patterns in a program's code. This course will explain how to design static code analysis that are inter-procedural, i.e., consider the whole program, across procedure boundaries. Designing such analyses is challenging, as they need to handle millions of program statements efficiently and precisely. Example applications are drawn from the area of IT security.</p> <p>This course builds on the DECA 1 course. In DECA 2, we discuss novel concepts directly from research, for example so-called demand-driven analyses, which are characterized by a more precise and at the same time more efficient analysis, but also pushdown systems, which provide a allow elegant modeling and at the same time fast execution of program analyses. Last but not least, we explain current solutions to practical problems in static analysis, such as the use of reflection and native code.</p> <p>Topics covered include:</p> <ul style="list-style-type: none">• Program analysis of software product lines• Modeling call stacks and field accesses with Pushdown Systems• Modeling auxiliary analysis information with Weighted Pushdown Systems• Efficiency and precision gains through Demand-driven Program Analysis• Synchronized Pushdown Systems in the Boomerang framework• Applied Android code analysis with FlowDroid• Dealing with Reflection through TamiFlex• Hybrid static and dynamic analysis with Harvester• Learning source, sink and sanitizer definitions with SWAN and SWAN Assist• Explainable static analysis <p>Throughout, we will discuss applications to software security.</p>

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>After having attended this course, students will have learned. . .</p> <ul style="list-style-type: none"> • how to make educated design decisions when designing automated code analysis for large-scale software systems, • which algorithms have which properties when using them to implement static code-analyses, • how to design real-world code analyses for practical problem cases from the area of IT security • how to interpret important terminology such as context, flow, field and object sensitivity • how to evaluate and explain the important limitations of static code analysis • which typical security code analyses exist (OWASP Top 10 etc.) and how they relate to the analysis frameworks explained in the course. <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Learning motivation 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Klausur oder mündliche Prüfung	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Klausur oder mündliche Prüfung	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of achievement</th> <th style="width: 20%; text-align: center;">Duration or Scope</th> <th style="width: 25%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								

3 Wahlpflichtmodule

12	<p>Module coordinator: Prof. Dr. Eric Bodden</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Designing code analyses for large-scale software systems 2:</i></p> <p>Implementation method Lectures and group exercises as well as programming exercises using widely used real-world static analysis frameworks (e.g. Soot, Phasar, FlowDroid)</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Context-, Flow-, and Field-sensitive Data-flow Analysis Using Synchronized Pushdown Systems (Johannes Späth, Karim Ali, Eric Bodden), In Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages, pages 48:1–48:29, 3(POPL), 2019. • FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps (Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Ochteau, Patrick McDaniel), In Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 259–269, PLDI '14, ACM, 2014. • Codebase-Adaptive Detection of Security-Relevant Methods (Goran Piskachev, Lisa Nguyen Quang Do, Eric Bodden), In ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), 2019. • Taming Reflection: Aiding Static Analysis in the Presence of Reflection and Custom Class Loaders (Eric Bodden, Andreas Sewe, Jan Sinschek, Hela Oueslati, Mira Mezini), In ICSE '11: International Conference on Software Engineering, pages 241–250, ACM, 2011.

Digitale Sprachsignalverarbeitung							
Digital Speech Signal Processing							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term				
	Semester number:	Duration (in sem.): 1	Teaching Language: de				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Digital Speech Signal Processing	L2 Ex2	60	120	CE	30	
2	Options within the module: none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Digitale Sprachsignalverarbeitung:</i></p> <p>Recommended Proficiencies Prior knowledge from the module Higher Mathematics is helpful.</p>
4	<p>Contents:</p> <p><i>Contents of the course Digitale Sprachsignalverarbeitung:</i></p> <p>The course introduces the basic techniques and theories of digital speech signal processing. A focal point of the first part of the lecture is the topic “Listening and Speaking”, which is concerned with psychological effects of human sound perception and speech production. Subsequently, time discrete signals and systems, as well as computer based data processing are discussed. Further topics are non-parametric short-time analysis of speech signals, speech coding and IP-phones.</p> <ul style="list-style-type: none"> • Listen and talk • Generating voice: human vocal tract, source filter model, vocoder • Acoustic waves • Listen: human ear, psycho acoustics and physiology of listening, loudness, acoustic occlusion, frequency groups • Time-discrete signals and systems • Basics: Elementary signals, LTI systems • Transformations: Fourier transformation of time-discrete signals, DFT, FFT • Time-discrete filtering in frequency domain: Overlap-Add, overlap-Save • Statistical speech signal analysis • Basics in theory of probabilities • Short-run analysis of speech signals: Spectrogram, cepstrum • Estimation of speech signals • Optimal filters • LPC analysis • Spectral filtering for noise suppression: spectral subtraction, Wiener filter • Adaptive Filters: LMS adaptation algorithm, echo compensation • Speech coding • Time domain coding: signal shape coding, parametric coding, hybride coding techniques • Frequency domain coding • Amplitude quantization: uniform quantization, quantization with companders (ulaw, alaw)
5	<p>Learning outcomes and competences:</p> <p>After attending the course, the students will be able to</p> <ul style="list-style-type: none"> • analyze digital signals, e.g., audio signals, in the time or frequency domain, • represent audio signals efficiently and • implement widely-used algorithms for speech analysis and speech processing in the frequency or time domain. <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Learning competence

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement: none		
8	Prerequisites for participation in examinations: none		
9	Prerequisites for assigning credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Weighing for overall grade: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Prof. Dr. Reinhold Häb-Umbach		
13	Other Notes: <i>Remarks of course Digitale Sprachsignalverarbeitung:</i> Implementation method <ul style="list-style-type: none"> • Lectures using the blackboard and presentations, • Alternating theoretical and practical exercise classes with exercise sheets and computer and • Demonstration of real technical systems in the lecture hall. Learning Material, Literature Allocation of a script; information on textbooks ; matlab scripts		

Efficiency in Games			
Efficiency in Games			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	summer term
	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

3 Wahlpflichtmodule

1	Module structure:						
		Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a)	Efficiency in Games	L3 Ex2	180	105	CE	25
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Efficiency in Games:</i> Recommended Proficiencies This course requires a general mathematical maturity. Calculus and probability. Combinatorics. A good understanding of mathematics, notions such as definition and proof, basic set theory, linearity, sufficiency, necessity, characterisations, etc.						

4	<p>Contents:</p> <p><i>Contents of the course Efficiency in Games:</i></p> <p>The course reviews several solution concepts and efficiency measures in non-cooperative game theory and delves into the techniques for proving bounds on efficiency of solutions in strategic and extensive games. We consider several techniques and apply them to important games.</p> <p>Topics</p> <ul style="list-style-type: none"> • Non-cooperative games and solution concepts, Nash equilibria - pure and mixed, Social welfare, Efficiency (prices if anarchy and stability (PoA, PoS)), Normative approach here (rather than descriptive) • Classical network examples (routing Pigou example, a network formation game with harmonic PoS, simple scheduling, resource allocation with proportional sharing). Efficiency as a guidance to MD. • Routing games, Nonatomic selfish routing, Pigou and Braess, Atomic selfish routing, AAE, Potential function method, Existence and uniqueness of equilibrium flows • PoA bounds in selfish routing games, Reducing the PoA • Network formation games, The local connection game and its PoA, Potential games, Bounding PoS using potential function method, Application: Global connection game, Facility location and utility games • Selfish load balancing, A GT variant of minimising makespan (the maximum load), the agents being the tasks. The social welfare is makespan, instead of the utilitarian one (sum). Bounds on pure and mixed PoA in various settings. Best response dynamics. Finally, consider algorithms to compute pure equilibria. • Scalable resource allocation. Resource allocation mechanisms, assuming private utilities. We consider both Nash and competitive equilibrium. The mechanism should have a low PoA (efficient) and the players should have low-dimensional strategy spaces. <ul style="list-style-type: none"> – Proportional allocation mechanism. – Smooth market-clearing mechanisms. – Extending the Vickrey-Clarke-Groves (VCG), requiring simple strategies and a single clearing price. • Correlated and coarse correlated equilibria, Robust PoA and the smoothness theorem by Tim Roughgarden, Applications • Repeated games, Repeated PD example, Folk theorems • Efficiency with respect to other solution concepts and social welfare definitions. • Efficiency and altruism
---	--

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>To define, understand and use the following:</p> <ul style="list-style-type: none"> • Non-cooperative games • Normal(strategic) games • (Pure and mixed) Nash equilibria • Prices of anarchy and stability • Classical network examples • Routing games (atomic and non-atomic) • Potential function method • Network formation games • The local connection game • Potential games • Global connection game • Facility location and utility games • Selfish load balancing • Best response dynamics • Proportional allocation mechanism • Smooth market-clearing mechanisms • Vickrey-Clarke-Groves (VCG) • Correlated and coarse correlated equilibria • Robust PoA and the smoothness theorem • Repeated games • Folk theorems • Altruistic players <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Attitude • Learning competence • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p style="margin-top: 5px;">The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations:		
Passing of course achievement			
9	Prerequisites for assigning credits:		
The credit points are awarded after the module examination was passed.			
10	Weighing for overall grade:		
The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses:		
Masterstudiengang Informatik v4			
12	Module coordinator:		
Jun.-Prof. Dr. Gleb Polevoy			
13	Other Notes:		
<i>Remarks of course Efficiency in Games:</i>			
Implementation method			
Motivation, theory, applications, examples, exercises			
Learning Material, Literature			
The necessary material consists of the slides, lectures, tutorials and homeworks. The additional reading consists of:			
<ul style="list-style-type: none"> • Algorithmic game theory, edited by Noam Nisan, Tim Roughgarden, Eva Tardos and Vijay V. Vazirani • A Course in Game Theory by Martin J. Osborne and Ariel Rubinstein, 1994, besides the definition of extensive form games • A site to acquire a deeper understanding: https://plato.stanford.edu/ • Concrete topics from their creators: Bounding the Inefficiency of Altruism Through Social Contribution Games by Mona Rahn and Guido Schaefer, 2013 - about efficiency and altruism 			

Foundations of Cryptography			
Foundations of Cryptography			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	summer term
	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

3 Wahlpflichtmodule

1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Foundations of Cryptography	L3 Ex2	75	105	CE	25
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Foundations of Cryptography:</i> Recommended Proficiencies Basic Knowledge in IT-Security and cryptography useful but not necessary, basic concepts of complexity theory and probability theory						
4	Contents: <i>Contents of the course Foundations of Cryptography:</i> The most important primitives of modern cryptography will be presented. These include encryption schemes, digital signatures, identification protocols, and multiparty computations. In each case we will define precise security notions. Starting from precisely stated assumptions, we develop constructions that provably satisfy these security definitions. <ul style="list-style-type: none"> • Symmetric and asymmetric encryption schemes • Pseudorandom generators, one-way functions, trapdoor permutations • Hashfunctions and message authentication codes • Digital signatures, one-time signatures, random oracles • Identification protocols, Σ protocols • Secure multiparty computation 						
5	Learning outcomes and competences: Students understand fundamental concepts and methods of modern cryptography. They are able to choose appropriate cryptographic tools for various security problems. Students are able to combine and modify basic cryptographic primitives, they are able to define new security concepts, they are able to the the security of new constructions with respect to the security concepts. Non-cognitive Skills <ul style="list-style-type: none"> • Commitment • Team work • Learning motivation • Literacy (scientific) • Self-monitoring 						

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
	zu	Type of examination	Duration or scope
	a)	Written or oral examination	90-120 min or 40 min
	Weighting for the module grade 100%		
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:		
	zu	Type of achievement	Duration or Scope
	a)	Written exercises	
SL / QT CA			
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Prof. Dr. Johannes Blömer		
13	Other Notes: <i>Remarks of course Foundations of Cryptography:</i> Implementation method Lectures, exercises, reading groups Learning Material, Literature <ul style="list-style-type: none"> • Oded Gorldreich, Foundations of Cryptography I,II, • Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography • Slides from the lectures 		

Foundations of Knowledge Graphs
Foundations of Knowledge Graphs

3 Wahlpflichtmodule

Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Foundations of Knowledge Graphs	L2 Ex3	75	105	CE	24
2	Options within the module: none					
3	Admission requirements: <i>Prerequisites of course Foundations of Knowledge Graphs:</i> Recommended Proficiencies Graph theory, logics					
4	Contents: <i>Contents of the course Foundations of Knowledge Graphs:</i> Knowledge graphs are used in an increasing number of applications. Large organisations such as Google Yahoo! and the BBC rely on these technologies to organise and manage the access to the the large amounts of data they manage. This lecture aims to present approaches for building, storing, integrating and using knowledge graphs. We will begin by studying knowledge extraction techniques for unstructured data. These include named entity recognition, disambiguation and relation extraction. Technologies for storing and knowledge (e.g., triple stores) will be presented subsequently. Time-efficient and accurate approaches for knowledge integration and link prediction will be followed by a series of applications for knowledge graphs. <ul style="list-style-type: none"> • Semantic networks • Property graphs • RDF graphs • Query languages (e.g., Cypher, SPARQL) • Knowledge extraction from text • Knowledge extraction from semi-structured data • Link discovery • Machine learning approaches for link discovery • Link prediction and tensor factorization 					

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>The students can carry out the following after the completion of the module:</p> <ul style="list-style-type: none"> • Model knowledge graphs; • Describe the formal semantics of modeling languages; • Create formal ontologies and check them for consistency; • Model efficient imperative and descriptive languages; • Train and execute knowledge extraction models, <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence • Media competence • Literacy (scientific) 										
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>			zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min bzw. 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade								
a)	Written or oral examination	90-120 min bzw. 40 min	100%								
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>			zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT								
a)	Written exercises		CA								
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>										
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>										
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>										
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>										
12	<p>Module coordinator:</p> <p>Prof. Dr. Axel-Cyrille Ngonga Ngomo</p>										

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Foundations of Knowledge Graphs:</i></p> <p>Implementation method 2 SWS of lectures within which the students will be presented with novel content weekly. The lecture will be self-contained with the students being presented with the premises for understanding particular aspects of knowledge graphs as well as with the corresponding conclusions and approaches derived from these premises. 1 SWS of exercises allow the students to deal with the concepts presented in the lecture through formal analysis and programming. The 2 SWS of mini-projects ensure that the students obtain a holistic understanding of the concepts learned by applying them to a more complex task than the one addressed in the exercises.</p> <p>Learning Material, Literature Slides, homework assignments</p>
----	--

Game Theory							
Game Theory							
Module number:	Workload (h): 180	Credits: 6		Regular Cycle: summer- / winter term			
	Semester number:	Duration (in sem.): 1		Teaching Language: en			
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Game Theory	L3 Ex2	75	105	CE	25
2	Options within the module: none						
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Game Theory:</i></p> <p>Recommended Proficiencies This course requires a general mathematical maturity. Calculus and probability. Combinatorics. A good understanding of mathematics, notions such as definition and proof, basic set theory, linearity, sufficiency, necessity, characterisations, etc.</p>						

4	<p>Contents:</p> <p><i>Contents of the course Game Theory:</i></p> <p>The course introduces a person to non-cooperative and cooperative game theory. The students will receive a broad overview of the branches of game theory, and subsequently dive into non-cooperative solution concepts and efficiency measures. We consider several models and important classes of such games. The second part of the course deals with cooperative solution concepts and classes of games.</p> <p>Topics</p> <ul style="list-style-type: none"> • Game theory areas (non-cooperative, cooperative, MD (auctions, etc.), epistemic GT, evolutionary GT, logic in GT, etc.), Utility theory (Von-Neumann Morgenstern) and rationality assumption, Normative approach here (rather than descriptive), Non-cooperative game theory: Normal form games, Extensive games, Succinct representations (polymatrix games, graphical games, etc.), (In)Complete and (in)perfect information assumptions, (In)finite games • Normal (= strategic)-form games: (Pure) Nash equilibrium, Examples (of non-existence too), Efficiency (prices if anarchy and stability (PoA, PoS)), Strongly/weakly dominant strategies (1st and 2nd price) auctions example, Strongly/weakly dominated strategies, Elimination (order-dependent for weak and independent for strong), The influence of elimination on NE • (Exact) potential games, Equivalence to congestion games • Zero-sum games, (maxmin, minmax, value, exchangeability of NE strategies) • Mixed extension, Mixed NE, (Finite existence) • Properties of mixed NE, general (mixed dominance), symmetric games, constant-sum, potential, Finding mixed NE (general alg. and examples), Rationalizability • Social welfare, Prices of anarchy and stability, Examples (coordination, routing, etc.) • Correlated and coarse correlated equilibrium, Strong Nash equilibrium, Evolutionary equilibrium and evolutionary games • Extensive games, A winning strategy and proof techniques, Zermelo's algorithm, Examples: chess, checkers, chomp • SPE, Existence • (In)finitely repeated games (prisoner's dilemma, etc.), Folk theorems • Cooperative games: Non-transferable and transferable utility, General properties, Transferable utility: Simple games • Core, Bondareva-Shapley characterisation theorem • The Shapley value and its axiomatic characterisation
---	---

5	<p>Learning outcomes and competences:</p> <p>To define, understand and use the following:</p> <ul style="list-style-type: none"> • General areas of GT • Non-cooperative games • Normal(strategic) games • Nash equilibria, dominant strategies • Prices of anarchy and stability • Potential games • Congestion games • Constant-sum games • Mixed NE, existence and finding these equilibria • Rationalizability • Efficiency bounds • Correlated and coarse correlated NE • Strong NE • Evolutionary stable strategy • Extensive games • winning strategy • Zermelo algorithm • Subgame perfect equilibrium • Repeated games and folk theorems • Cooperative games • Transferable utility • Simple games • Core • Bondareva-Shapley theorem • The Shapley value and its axiomatic characterisation <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Attitude • Learning competence • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
	Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.		
8	Prerequisites for participation in examinations:		
	Passing of course achievement		
9	Prerequisites for assigning credits:		
	The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade:		
	The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses:		
	Masterstudiengang Informatik v4		
12	Module coordinator:		
	Jun.-Prof. Dr. Gleb Polevoy		
13	Other Notes:		
	<p><i>Remarks of course Game Theory:</i></p> <p>Implementation method Motivation, theory, applications, examples, exercises</p> <p>Learning Material, Literature The necessary material consists of the slides, lectures, tutorials and homeworks. The additional reading consists of:</p> <ul style="list-style-type: none"> • A Course in Game Theory by Martin J. Osborne and Ariel Rubinstein, 1994, besides the definition of extensive form games • Game Theory by Michael Maschler, Eilon Solan and Shmuel Zamir, 2013 • An Introduction to Game Theory by Martin J. Osborne, 2004, besides the definition of extensive form games • Game Theory: A Multi-Leveled Approach by Hans Peters, 2008 • Game Theory And Mechanism Design by Y. Narahari, 2014 • Algorithmic game theory, edited by Noam Nisan, Tim Roughgarden, Eva Tardos and Vijay V. Vazirani • A site to acquire a deeper understanding: https://plato.stanford.edu/ <p>Concrete topics from their creators:</p> <ul style="list-style-type: none"> • Non-Cooperative Games by John F. Nash, 1951 - about mixed Nash equilibrium • Potential Games by Dov Monderer and Lloyd S. Shapley, 1994 - about potential games 		

High-Performance Computing
High-Performance Computing

3 Wahlpflichtmodule

Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) High-Performance Computing	L2 Ex3	75	105	CE	40	
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course High-Performance Computing:</i> Recommended Proficiencies <ul style="list-style-type: none"> • Programming skills in C/C++ • Computer architecture (in particular caches, multi-core processors), for example from attending the lecture Advanced Computer Architecture • Practical experience in using and programming Linux systems • Self-assessment test 						
4	Contents: <i>Contents of the course High-Performance Computing:</i> This course teaches the foundations of high-performance computing with an emphasis on the programming of parallel computer systems and novel hardware accelerators. <ul style="list-style-type: none"> • Introduction to High-Performance Computing • Models and programming patterns for parallel computing • Programming languages and libraries for HPC • Performance analysis, optimization, and debugging • Heterogeneous computing with hardware accelerators • Case studies 						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>After attending this course, the students are able to</p> <ul style="list-style-type: none"> • name models and programming patterns for HPC and to select patterns for a given application, • name and apply the basic constructs of frequently used HPC libraries, in particular, MPI, OpenMP and OpenCL, • analyze the performance of applications by using profiling tools and use the gathered information to create a systematic optimization strategy, • apply the taught concepts and methods for parallelizing and optimizing existing applications <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td>CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Prof. Dr. Christian Plessl</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course High-Performance Computing:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture with projected slides and blackboard notes • Interactive assignments in lecture room • Practical programming projects on parallel computer systems (teamwork in small groups) <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Lecture slides • Assignment sheets • Task descriptions and technical documentation for programming projects
----	--

Information Retrieval						
Information Retrieval						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Information Retrieval	L2 Ex3	75	105	CE	30
2	Options within the module: none					
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Information Retrieval:</i></p> <p>Recommended Proficiencies Boolean algebra, vector spaces, stochastics</p>					

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Information Retrieval:</i></p> <p>The goal of this lecture is to present the foundations of search engines. We study the basic models underlying search (Boolean, vector space, probabilistic) as well as the corresponding approaches necessary to process search results efficiently (e.g., clustering, classification).</p> <ul style="list-style-type: none"> • Boolean model • Indexing • Vector space model • Probabilistic model • Classification and clustering • PageRank 								
5	<p>Learning outcomes and competences:</p> <p>The students possess the following abilities after the completion of this module:</p> <ul style="list-style-type: none"> • Students understand the basics of search engines. • They are able to describe and compare representation techniques for documents and texts in search engines. • They can choose a suitable model underlying search (Boolean, vector space, probabilistic) for a given task or develop a mixed form thereof. • They can to evaluate the efficiency of the ensuing scheme. <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence • Media competence • Literacy (scientific) 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td>CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						

3 Wahlpflichtmodule

8	<p>Prerequisites for participation in examinations: Passing of course achievement</p>
9	<p>Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade: The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses: Masterstudiengang Informatik v4</p>
12	<p>Module coordinator: Prof. Dr. Axel-Cyrille Ngonga Ngomo</p>
13	<p>Other Notes: <i>Remarks of course Information Retrieval:</i> Implementation method The weekly lectures (2SWS) cover new content on a weekly basis. In addition to the formal considerations, we will cover applications and corresponding limitations of the languages and methods presented throughout the course. The exercises (1SWS) are both theoretical and practical in nature. The learners are to show that they understood the concepts and can apply them to practical problems. The mini-project (2SWS) give the students a holistic view of how to solve complex problems using Information Retrieval technologies. Learning Material, Literature Slides and homework assignments</p>

Introduction to Quantum Computation							
Introduction to Quantum Computation							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Introduction to Quantum Computation	L3 Ex2	75	105	CE	40	
2	Options within the module: none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Introduction to Quantum Computation:</i></p> <p>Recommended Proficiencies Linear Algebra, algorithms.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Introduction to Quantum Computation:</i></p> <p>This lecture introduces the fundamental concepts of quantum computation and information from a computer science perspective. This includes an introduction to quantum mechanics, quantum entanglement, quantum algorithms, quantum error correction, and quantum information theory.</p> <ul style="list-style-type: none"> • Quantum mechanics • Quantum entanglement • Quantum algorithms • Quantum error correction • Quantum information 								
5	<p>Learning outcomes and competences:</p> <p>Students are able to:</p> <ul style="list-style-type: none"> • Describe and apply the postulates of quantum mechanics • Understand the use of entanglement as a resource • Design and analyze fundamental quantum algorithms • Apply the theory of error-correcting codes • Understand and apply basic quantum information theory concepts such as entropy <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">120-180 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	120-180 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	120-180 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of achievement</th> <th style="width: 20%; text-align: center;">Duration or Scope</th> <th style="width: 25%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						

3 Wahlpflichtmodule

8	<p>Prerequisites for participation in examinations: Passing of course achievement</p>
9	<p>Prerequisites for assigning credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>
10	<p>Weighing for overall grade: The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses: Masterstudiengang Informatik v4</p>
12	<p>Module coordinator: Jun. Prof. Dr. Sevag Gharibian</p>
13	<p>Other Notes: <i>Remarks of course Introduction to Quantum Computation:</i> Implementation method Slides and blackboard writing. All important concepts and techniques are further deepened with examples in exercises. Learning Material, Literature</p> <ul style="list-style-type: none"> • Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press • Lecture slides, exercises

Kontextuelle Informatik						
Contextual Informatics						
Module number:	Workload (h):	Credits:	Regular Cycle:			
	180	6	winter term			
	Semester number:	Duration (in sem.):	Teaching Language:			
		1	de			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a)	Contextual Informatics	L2 Ex3	75	105	CE 30
2	Options within the module:					
	none					

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Kontextuelle Informatik:</i></p> <p>Recommended Proficiencies</p> <ul style="list-style-type: none">• Good general education• Ability to read and analyze longer and complex texts from computer science as well as the humanities
4	<p>Contents:</p> <p><i>Contents of the course Kontextuelle Informatik:</i></p> <p>Computer scientists and programmers develop products based on symbolic descriptions (programs, specifications, documentation etc.) by modeling a specific domain. A number of different questions arise when such products are being developed: How can the developers adequately model the data that are to be processed as well as the processes that are to be implemented? Which consequences result from the possibility to develop interactive systems? When using the software, which role will the users play, which the software? Which underlying conditions of the usage context need to be considered?</p> <p>The lecture discusses the theoretical and conceptual foundations of computer science relevant to the further considerations. Special attention will be paid to differentiate between technical concepts and the sphere of usage. Against this background, theories of interactive systems will be explored in order to examine which role digital media play with respect to processes of the mind. When developing computer systems, relevant data and processes need to be anticipated to a certain degree and modeled as formal systems. This raises issues like the question under which conditions such a formal description can be made in an adequate way and with which consequences regarding the reliability and responsible use of computer systems in a given domain.</p> <ul style="list-style-type: none">• Basic concepts of computer science• Digital media and processes of the mind• Paradigms of replacement and support• Modeling and formalizing data and processes• Legal and ethical questions
5	<p>Learning outcomes and competences:</p> <p>Students will learn to examine the role of interactive systems based on theories. They will learn to distinguish between technical and non-technical problems and how to relate these to each other. They will be enabled to assess current technological trends and computer systems as well as the potentials of innovation in the field of digital media.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none">• Social and ethical judgement• Attitude• Media competence• Literacy (scientific)

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
<p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>			
7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
<p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Dr. Harald Selke		
13	Other Notes: <i>Remarks of course Kontextuelle Informatik:</i> Implementation method The lecture follows a flipped classroom concept in which students learn about topics based on their reading of scientific literature and presenting them in short presentations during the exercises. Building on this, the lecture then conveys connections between the literature covered in the exercises and adds further facets. Learning Material, Literature <ul style="list-style-type: none"> • Lecture slides • Wardrip-Fruin, N.; Montfort, N. (eds.): The New Media Reader. Cambridge, Ma.: MIT Press, 2003. • Additional scientific literature will be announced in the lectures. 		

3 Wahlpflichtmodule

Logic Programming for Artificial Intelligence							
Logic Programming for Artificial Intelligence							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer- / winter term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Logic Programming for Artificial Intelligence	L3 Ex2	75	105	CE	40/20	
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Logic Programming for Artificial Intelligence:</i> Recommended Proficiencies Students should have previous knowledge in programming as offered in the courses “Programmierung” and “Programming Languages” and knowledge in database query languages as offered in the course “Database Systems”.						
4	Contents: <i>Contents of the course Logic Programming for Artificial Intelligence:</i> This course views various concepts and techniques from computer science, artificial intelligence, and computational linguistics from a different perspective, i.e. the perspective of programming in logic. Programming in logic in general and the programming language Prolog in particular offer the ability to describe many concepts in logic, i.e. in a declarative way, and to have them tested and executed by an interpreter at a same time. This is in particular useful for puzzles and quizzes, but also for self-defined or domain specific languages. <ul style="list-style-type: none"> • Introduction into logic programming using the Prolog language • Constraint solvers, puzzles, and theorem provers • Interpreters for term substitution systems • Parsing programs, XML, and natural language • Semantics construction, question answering systems, and text translation • Meta interpreters, domain specific languages, and programming in “natural language” • Feature term unification and applications in computer linguistics and ecommerce 						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>Students learn factual knowledge about</p> <ul style="list-style-type: none"> • the transformation of knowledge given as facts and rules into an executable programs • how to program in logic and in self-designed languages <p>methodological knowledge, including</p> <ul style="list-style-type: none"> • the ability to define domain specific languages • the ability to implement interpreters for domain specific languages • the ability to develop small question answering systems • the ability to develop software for theorem provers or constraint solvers solving puzzles <p>transfer skills</p> <ul style="list-style-type: none"> • the ability to transfer the methodologies and skills gained to other data sources, knowledge representation formats, or calculi • the ability to transfer the parsing and semantics knowledge to domain specific languages <p>normative evaluation skills including the ability to assess</p> <ul style="list-style-type: none"> • the suitability and limitations of different data and knowledge representation formats for different tasks • the suitability of different programming paradigms for different projects • the effort and feasibility of projects aiming natural language understanding • the effort and feasibility of projects aiming at automated translation <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Learning motivation 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						

3 Wahlpflichtmodule

8	<p>Prerequisites for participation in examinations: Passing of course achievement</p>
9	<p>Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade: The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses: Masterstudiengang Informatik v4</p>
12	<p>Module coordinator: Prof. Dr. Stefan Böttcher</p>
13	<p>Other Notes: <i>Remarks of course Logic Programming for Artificial Intelligence:</i> Implementation method The theoretical concepts are explained in the lectures and consolidated in small groups during tutorials. The tutorials are carried out as practical exercises on the computer. Learning Material, Literature</p> <ul style="list-style-type: none"> • Ivan Bratko: Prolog Programming for Artificial Intelligence. Pearson Education, Newest Edition. • Links to further material will be provided in the lecture.

Machine Learning I							
Machine Learning I							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	summer- / winter term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Machine Learning I	L3 Ex2	75	105	CE	60/20
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Machine Learning I:</i></p> <p>Recommended Proficiencies Basic knowledge in mathematics (linear algebra, statistics), programming and algorithms.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Machine Learning I:</i></p> <p>Due to the ever increasing amount of data that is routinely produced in our information society, the topic of machine learning has become increasingly important in the recent years, not only as a scientific discipline but also as a key technology of modern software and intelligent systems. This lecture provides an introduction to the topic of machine learning, with a specific focus on supervised learning for classification and regression. The lecture covers theoretical foundations of generalisation as well as practical topics and concrete learning algorithms.</p> <ul style="list-style-type: none"> • Introduction • The Learning Problem • Training versus Testing • The Linear Model • Non-Linear Methods • Overfitting 								
5	<p>Learning outcomes and competences:</p> <p>The students understand the statistical foundations of generalisation, i.e., the induction of models from data, as well as practical tools for model validation. They are able to apply basic methods of supervised learning to problems of classification and regression.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Learning motivation • Literacy (scientific) 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <p>none</p>								
8	<p>Prerequisites for participation in examinations:</p> <p>none</p>								

3 Wahlpflichtmodule

9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>
12	<p>Module coordinator:</p> <p>Prof. Dr. Eyke Hüllermeier</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Machine Learning I:</i></p> <p>Implementation method Theoretical foundations and concepts of machine learning will be taught in the form of a lecture and deepened in practical exercise courses, group work as well as individual homework.</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Script • Y.S. Abu-Mostafa, M. Magdon-Ismael, H.T. Lin. Learning from Data, AMLBook, 2012. • P. Flach. Machine Learning, Cambridge Univ. Press, 2012. • E. Alpaydin. Machine Learning, Oldenbourg, 2008. • C.M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.

Machine Learning II							
Machine Learning II							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	summer- / winter term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Machine Learning II	L3 Ex2	75	105	CE	20
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Machine Learning II:</i></p> <p>Recommended Proficiencies</p> <p>Basic knowledge in machine learning (as conveyed, for example, by the Machine Learning I lecture).</p>								
4	<p>Contents:</p> <p><i>Contents of the course Machine Learning II:</i></p> <p>This lecture, which is conceived as a continuation of the Machine Learning I, covers advanced topics in contemporary machine learning research, such as reinforcement learning, online learning and bandit algorithms, multi-task learning, multi-target and structured output prediction, preference learning, learning from weak supervision, and uncertainty in machine learning. The focus of the lecture will be on methods and algorithms, though theoretical issues and applications will be addressed, too.</p> <ul style="list-style-type: none"> • From binary to multi-class classification • Ordinal and hierarchical classification • Ensemble methods • Nonlinear models and kernel machines • Multi-target prediction • Semi-supervised learning • Active learning • Online learning • Multi-armed bandits • Reinforcement learning • Preference learning and ranking 								
5	<p>Learning outcomes and competences:</p> <p>The students have an overview of methods for multi-class classification, the learning of nonlinear models, and extensions of the simple setting of supervised learning. They understand algorithmic concepts of corresponding methods and are able to apply them to real problems.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Learning motivation • Literacy (scientific) 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement: none
8	Prerequisites for participation in examinations: none
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).
11	Reuse in degree courses: Masterstudiengang Informatik v4
12	Module coordinator: Prof. Dr. Eyke Hüllermeier
13	Other Notes: <i>Remarks of course Machine Learning II:</i> Implementation method Theoretical foundations and concepts of machine learning will be taught in the form of a lecture and deepened in practical exercise courses, group work as well as individual homework. Learning Material, Literature <ul style="list-style-type: none"> • Script • Y.S. Abu-Mostafa, M. Magdon-Ismael, H.T. Lin. Learning from Data, AMLBook, 2012. • P. Flach. Machine Learning, Cambridge Univ. Press, 2012. • E. Alpaydin. Machine Learning, Oldenbourg, 2008. • C.M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.

Mobile Communication							
Mobile Communication							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	winter term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Mobile Communication	L3 Ex2	75	105	CE	20

3 Wahlpflichtmodule

2	<p>Options within the module:</p> <p>none</p>								
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Mobile Communication:</i></p> <p>Recommended Proficiencies</p> <p>Basic knowledge of computer networks is required, e.g., a Bachelor-level class “Computer networks”.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Mobile Communication:</i></p> <p>The lecture discusses foundations of mobile communication (e.g., wireless channel models) and fundamental techniques (e.g., spread spectrum communication), important protocol mechanisms (e.g., medium access in wireless systems), mobile communication systems, and MobileIP. In addition to technological and conceptual aspects, we shall also discuss approaches and methods for performance evaluation of mobile communication systems.</p> <ul style="list-style-type: none"> • Basics and physical layer: channel models, fading, Rayleigh channel, modulation, OFDM, spread spectrum • Medium access control: Aloha in Rayleigh channels, CSMA, hidden terminal, RTS/CTS, busy-tone protocols • Cellular systems: GSM, UMTS, LTE, focusing on system architecture • Wireless LAN systems: IEEE 802.11, medium access, power control, energy efficiency, Bianchi's performance analysis • Mobility in fixed networks: Mobile IP and related approaches 								
5	<p>Learning outcomes and competences:</p> <p>Participants of this class know challenges and problems arising in design and operation of mobile communication systems. They can differentiate between challenges based in physics and those arising from a particular system design; they can choose suitable protocols or design new ones. They are able to select mechanisms from different architectural layers, integrate them into a new complete architecture and justify their selection and integration decisions. They are also able to quantitatively evaluate protocol mechanisms.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 25%; text-align: center;">Duration or scope</th> <th style="width: 20%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral exam</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral exam	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral exam	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
	Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.		
8	Prerequisites for participation in examinations:		
	Passing of course achievement		
9	Prerequisites for assigning credits:		
	The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade:		
	The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses:		
	Masterstudiengang Informatik v4		
12	Module coordinator:		
	Dr. Florian Klingler		
13	Other Notes:		
	<i>Remarks of course Mobile Communication:</i>		
	Implementation method		
	Lecture with slides and blackboard; homework assignments with (among others) some programming assignments to simulate wireless systems.		
	Learning Material, Literature		
	Slide set; chapters from various textbooks: J. Schiller, Mobile Communication, Addison Wesley, 2nd edition; D. Tse und P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005.		

Model-Based Systems Engineering			
Model-Based Systems Engineering			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	summer term
	Semester number:	Duration (in sem.):	Teaching Language:
		1	de

3 Wahlpflichtmodule

1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Model-Based Systems Engineering	L3 Ex2	75	105	CE	??	
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Model-Based Systems Engineering:</i> Recommended Proficiencies Basics of Systems Engineerings						
4	Contents: <i>Contents of the course Model-Based Systems Engineering:</i> The goal of the lecture is a comprehensive understanding of Model-Based Systems Engineering (MBSE) and its components. The students are taught the essential topics of MBSE. This includes fundamentals including languages, methods and IT tools, which are also tested in practice. The benefits of MBSE (an understanding of the system by all involved actors, a basis for communication and cooperation between different disciplines but also functional areas, ...) will be conveyed to the students. Furthermore, essential analysis methods for testing system designs are covered. The focus is on multidisciplinary, software-intensive systems from the mechanical and plant engineering and automotive industries. <ul style="list-style-type: none"> • Basics of MBSE • SysML for multidisciplinary systems • CONSENS • further MBSE approaches • design patterns • MBSE Tools • analysis methods based on the system model 						
5	Learning outcomes and competences: Learning Outcomes Students will be able to, <ul style="list-style-type: none"> • Work in a model-based manner • Apply systems thinking • Create system architectures & derive requirements. Non-Cognitive Competencies <ul style="list-style-type: none"> • Self-monitoring • Literacy (scientific) • Learning competence • Learning motivation 						

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)			
	zu	Type of examination	Duration or scope	Weighting for the module grade
	a)	Written or oral examination	90-120 min or 40 min	100%
	The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement: none			
8	Prerequisites for participation in examinations: none			
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.			
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses: Masterstudiengang Informatik v4			
12	Module coordinator: Prof. Dr.-Ing. Roman Dumitrescu			

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Model-Based Systems Engineering:</i></p> <p>Implementation method</p> <p>The module consists of three parts 1. lecture with slides: basics and concepts are explained in the lecture and illustrated with examples. 2. exercises (tutorial): In the exercises, knowledge is transferred and the concepts are applied. The exercises have to be prepared by the students themselves. 3. practical course (labs): In the practical course, the application of what has been learned takes place in group work.</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Friedenthal, S.; Moore, A.; Steiner, R.: A Practical Guide to SysML. The Systems Modeling Language. Morgan Kaufmann, Waltham, 2. Auflage, 2012 • Gausemeier, J.; Rammig, J.; Schäfer, W. (Eds.): Design Methodology for Intelligent Technical Systems. Develop Intelligent Technical Systems of the Future. Springer-Verlag, 2014 • Gausemeier, J.; Dumitrescu, R.; Steffen, D.; Czaja, A.; Wiederkehr, O.; Tschirner, C.: Systems Engineering in industrial practice. Heinz Nixdorf Institute, University • Haberfellner, R., L., D. W. O., Fricke, E., & Voössnersiegfried. (2019). Systems engineering: fundamentals and applications. Cham: Springer International Publishing • IncoSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities (2015) • Weilkens, Tim: Systems Engineering with SysML/UML: Modeling, Analysis, Design (The MK/OMG Press) (English Edition) • Dumitrescu, R.; Albers, A.; Riedel, O.; Stark, R.; Gausemeier, J. (Hrsg.): Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft, Ein Beitrag zum Advanced Systems Engineering, Paderborn, 2021 – English Version: www.advanced-systems-engineering.de
----	---

Networked Embedded Systems							
Networked Embedded Systems							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	winter term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Networked Embedded Systems	L3 Ex2	75	105	CE	60/20	
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Networked Embedded Systems:</i></p> <p>Recommended Proficiencies System software and system-level programming</p>								
4	<p>Contents:</p> <p><i>Contents of the course Networked Embedded Systems:</i></p> <p>The objective of this course is gain insights into the operation and programming of embedded systems. A strong focus is on wireless sensor networks. We study the fundamentals of such sensor networks. In the scope of the exercises, we discuss selected topics in more detail.</p> <ul style="list-style-type: none"> • Design and architecture of embedded systems - Architecture of embedded systems, programming paradigms • Sensor networks - Principles and applications • Wireless communications - Concepts of modulation and encoding on the physical layer • Wireless access - Typical medium access protocols for low-power sensor nodes • Routing - Ad hoc routing and data centric communication • Cooperation and clustering - Clustering algorithms, guaranteed connectivity 								
5	<p>Learning outcomes and competences:</p> <p>The learning objective is to understand the fundamental concepts of networked embedded systems. Students understand these concepts and are able to apply this knowledge.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Commitment • Learning competence 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td>CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								

3 Wahlpflichtmodule

9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).
11	Reuse in degree courses: Masterstudiengang Informatik v4
12	Module coordinator: Dr. Florian Klingler
13	Other Notes: <i>Remarks of course Networked Embedded Systems:</i> Implementation method Lecture with practical exercises Learning Material, Literature Slides, textbooks, papers

Optimierungsverfahren für das maschinelle Lernen							
Optimization methods for machine learning							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Optimization methods for machine learning	L3 Ex2	75	105	CE	50
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Optimierungsverfahren für das maschinelle Lernen:</i> none						

<p>4</p>	<p>Contents:</p> <p><i>Contents of the course Optimierungsverfahren für das maschinelle Lernen:</i></p> <p>Machine learning has become an important aspect in countless applications from various disciplines. The central aspect in machine learning is the <i>automated identification and exploitation of patterns</i> in data sets. In <i>supervised learning</i>, a method is parameterized in such a way that it can be applied to until now unknown data (for instance for the identification of objects within an image), i.e., that it <i>generalizes</i>.</p> <p>The determination of the underlying parameters is in most cases realized by solving an <i>optimization problem</i>: Determine the parameters such that the error on the <i>training data set</i> is minimized. Here, the type of optimization critically depends on the machine learning approach as well as on the training data.</p> <p>In this lecture, we cover the basics for the training of machine learning methods in depth. The covered topics are:</p> <ul style="list-style-type: none"> • Regression methods • Nonlinear optimization • The training of deep neural networks • Stochastic gradient methods • Multiobjective optimization approaches for machine learning
<p>5</p>	<p>Learning outcomes and competences:</p> <p>Students learn</p> <ul style="list-style-type: none"> • factual knowledge about <ul style="list-style-type: none"> – the basis of various optimization methods – the specifics of using optimization methods in the context of machine learning – the required steps for training neural networks – the limitations of training machine learning methods • methodological knowledge, including <ul style="list-style-type: none"> – the configuration and implementation as well as the training of neural networks – the treatment of large data sets • transfer skills <ul style="list-style-type: none"> – the application of the learned methods to learning methods and neural network architectures – the selection of machine learning methods that are suitable for the available data • normative evaluation skills including the ability to assess <ul style="list-style-type: none"> – the suitability of specific methods in terms of the available amount of data, the system complexity, etc. – the limitations and reliability of machine learning methods <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • learning competence • learning motivation

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)			
	zu	Type of examination	Duration or scope	Weighting for the module grade
	a)	Written or oral examination	90-120 min or 40 min	100%
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.				
7	Study Achievement:			
	zu	Type of achievement	Duration or Scope	SL / QT
	a)	Programming tasks		AA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.				
8	Prerequisites for participation in examinations: Passing of course achievement			
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.			
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses: Masterstudiengang Informatik v4			
12	Module coordinator: Dr. Sebastian Peitz			
13	Other Notes: <i>Remarks of course Optimierungsverfahren für das maschinelle Lernen:</i> Implementation method Theoretical concepts will be presented during lectures and consolidated in tutorials with exercises as well as programming tasks. Tutorials will be realized via individual work as well as joint discussions. Learning Material, Literature <ul style="list-style-type: none"> • Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006. • Jerome H. Friedman, Robert Tibshirani, Trevor Hastie. The Elements of Statistical Learning. Springer, 2009. • Jorge Nocedal, Stephen J. Wright. Numerical Optimization. Springer, 2006. 			

3 Wahlpflichtmodule

Planning and Heuristic Search							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Planning and Heuristic Search	L3 Ex2	75	105	CE	40/20	
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Planning and Heuristic Search:</i> Recommended Proficiencies Basics of symbolic modeling: functions, relations, logical formulas; design and analysis of algorithms; basics of computational complexity theory: complexity classes, reducibility, completeness.						
4	Contents: <i>Contents of the course Planning and Heuristic Search:</i> The course "Planning and Heuristic Search" introduces two approaches to solve knowledge-intensive tasks. In the "Planning" part, domain tasks are represented as planning problems using e.g. state spaces or plan spaces. Several approaches to planning are discussed and analyzed. The "Heuristic Search" part generalizes the state space concept and introduces a general framework of systematic search algorithms that are guided by heuristic information on the problem domain. As an application example, planning algorithms can be implemented deterministically using heuristic search. Theoretical results will be presented and proved in both areas. Part Planning <ul style="list-style-type: none"> • Introduction to planning problems • State space representation and plan space representation • Planning algorithms • Complexity of planning problems and analysis of planning algorithms Part Heuristic Search <ul style="list-style-type: none"> • Search space representations • Informed search algorithms • Relaxed models • Formal properties of informed search algorithms 						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>The students are able to</p> <ul style="list-style-type: none"> • name and explain relevant concepts in modeling planning tasks and search tasks, • describe the basic steps of several planning algorithms and heuristic search algorithms, • identify and model simple tasks as planning tasks resp. search tasks, • describe approaches to compute valuable heuristics in such tasks, • use theoretical results as guidance in selecting formalizations and algorithms. <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Learning motivation • Literacy (scientific) • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td>100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td>CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Dr. Theodor Lettmann</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Planning and Heuristic Search:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture • Tutorials • Discussions • Homework Assignments • Prototype Implementations <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Lecture slides • Text books: <ul style="list-style-type: none"> • J. Pearl: Heuristics, Addison-Wesley (1984) • S.J. Russell, P. Norvig: Artificial Intelligence: A Modern Approach, Prentice Hall, 1995 • S. Edelkamp, S. Schrödl: Heuristic Search: Theory and Applications, Elsevier, 2012 • M. Ghallab, D. Nau, P. Traverso: Automated Planning, Morgan Kaufmann, 2004 • Exercises • List of classical and recent papers, e.g. R. Ebendt, R. Drechsler: Weighted A*Search - Unifying View and Application, J. Artificial Intelligence, pp. 1310-1342, 2009 • Online material, e.g. H. Geffner, B. Bonet: A Concise Introduction to Models and Methods for Automated Planning, doi: 10.2200/S00513ED1V01Y201306AIM022, 2013
----	--

Quantum Algorithms						
Quantum Algorithms						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Quantum Algorithms	L3 Ex2	75	105	CE	20
2	Options within the module: none					
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Quantum Algorithms:</i></p> <p>Recommended Proficiencies Linear Algebra, Quantum Computing</p>					

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Quantum Algorithms:</i> This lecture covers quantum algorithms from a computer science perspective. Topics include quantum circuits (e.g. Solovay-Kitaev theorem), quantum algorithms for algebraic problems (e.g. Hidden Subgroup problem), quantum walks, quantum query complexity, and adiabatic quantum computing.</p> <ul style="list-style-type: none"> • Quantum circuits • Algebraic problems • Quantum walks • Query complexity • Adiabatic computation 								
5	<p>Learning outcomes and competences:</p> <p>Students are able to:</p> <ul style="list-style-type: none"> • Describe universal gate sets • Develop Quantum Fourier-Transform based algorithms • Develop quantum walk-based algorithms • Apply the quantum adiabatic theorem • Give quantum query lower bounds <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written Exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written Exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written Exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								

3 Wahlpflichtmodule

9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>
12	<p>Module coordinator:</p> <p>Jun. Prof. Dr. Sevag Gharibian</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Quantum Algorithms:</i></p> <p>Implementation method Slides and blackboard writing. All important concepts and techniques are further deepened with examples in exercises.</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press • Andrew M. Childs, Wim van Dam, Quantum algorithms for algebraic problems, Reviews of Modern Physics, volume 82, 2010 • Lecture slides, exercises

Quantum Complexity Theory							
Quantum Complexity Theory							
Module number:	Workload (h):	Credits:	Regular Cycle:				
	180	6	summer term				
	Semester number:	Duration (in sem.):	Teaching Language:				
		1	en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Quantum Complexity Theory	L3 Ex2	75	105	CE	20
2	Options within the module:						
	none						

3 Wahlpflichtmodule

3	<p>Admission requirements:</p> <p><i>Prerequisites of course Quantum Complexity Theory:</i></p> <p>Recommended Proficiencies Linear Algebra, Quantum Computing</p>								
4	<p>Contents:</p> <p><i>Contents of the course Quantum Complexity Theory:</i></p> <p>This lecture provides a brief review of introductory quantum computation, and subsequently moves into quantum complexity theory. Beginning to advanced topics will be covered, including quantum analogues of P and NP (denoted BQP, QCMA, and QMA), quantum satisfiability problems, quantum interactive proofs, and tensor networks. Along the way, semidefinite programming will be introduced as an important tool.</p> <ul style="list-style-type: none"> • Complexity classes BQP, QCMA, QMA • Quantum Satisfiability Problems • Quantum Interactive Proofs • Tensor Networks • Semidefinite Programming 								
5	<p>Learning outcomes and competences:</p> <p>Students are able to:</p> <ul style="list-style-type: none"> • Describe and apply the postulates of quantum mechanics • Work with complexity classes such as BQP and QMA • Show QMA-hardness of computational problems • Apply semidefinite programming techniques • Use tensor networks to model entangled quantum states <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Learning competence • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
	Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.		
8	Prerequisites for participation in examinations:		
	Passing of course achievement		
9	Prerequisites for assigning credits:		
	The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade:		
	The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses:		
	Masterstudiengang Informatik v4		
12	Module coordinator:		
	Jun. Prof. Dr. Sevag Gharibian		
13	Other Notes:		
	<i>Remarks of course Quantum Complexity Theory:</i>		
	Implementation method		
	Slides and blackboard writing. All important concepts and techniques are further deepened with examples in exercises.		
	Learning Material, Literature		
	<ul style="list-style-type: none"> • Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press • S. Gharibian, Y. Huang, Z. Landau, S. W. Shin, Quantum Hamiltonian Complexity, Foundations and Trends in Theoretical Computer Science • Lecture slides, exercises 		

Real World Crypto Engineering			
Real World Crypto Engineering			
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term
	Semester number:	Duration (in sem.): 1	Teaching Language: en

3 Wahlpflichtmodule

1	Module structure:							
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)		
	a)	Real World Crypto Engineering	L3 Ex2	75	105	CE	40	
2	Options within the module: none							
3	Admission requirements: <i>Prerequisites of course Real World Crypto Engineering:</i> Recommended Proficiencies Basiskonntnisse in Programmierung, IT-Sicherheit und Kryptographie							
4	Contents: <i>Contents of the course Real World Crypto Engineering:</i> Starke Kryptographie ist nicht immer ausreichend, um die grundlegenden Sicherheitsziele zu schützen. Auch wenn starke kryptographische Algorithmen verwendet werden, kann bei deren Einsatz viel schief gehen. In dieser Vorlesung werden wir auf die wichtigsten Protokolle und kryptographische Schutzmechanismen eingehen (z.B. TLS, SSH, WPA) und werden ihre Basiskonzepte kennenlernen. Anschließend werden wir prominente Angriffe vorstellen, die die gewünschten Sicherheitsziele komplett gebrochen haben. Basierend auf vielen Fällen werden wir lernen, was beim Design und bei der Implementierung von kryptographischen Anwendungen wichtig ist.							
5	Learning outcomes and competences: Upon successful completion, students have a comprehensive understanding of the technical aspects of applied cryptographic algorithms. They have recognized that cryptography alone is not sufficient to solve security-related problems. They have an overview of current cryptographic attacks and know how to practically prevent them. Non-cognitive Skills <ul style="list-style-type: none">• Team work• Literacy (scientific)							
6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)							
	zu	Type of examination	Duration or scope		Weighting for the module grade			
	a)	Written or oral examination	90-120 min or 40 min		100%			
	The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.							

3 Wahlpflichtmodule

7	Study Achievement:			
	zu	Type of achievement	Duration or Scope	SL / QT
	a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.				
8	Prerequisites for participation in examinations:			
Passing of course achievement				
9	Prerequisites for assigning credits:			
The credit points are awarded after the module examination was passed.				
10	Weighing for overall grade:			
The module is weighted according to the number of credits (factor 1).				
11	Reuse in degree courses:			
Masterstudiengang Informatik v4				
12	Module coordinator:			
Prof. Dr.-Ing. Juraj Somorovsky				
13	Other Notes:			
<i>Remarks of course Real World Crypto Engineering:</i>				
Implementation method				
Lectures, exercises				
Learning Material, Literature				
Lecture slides, scientific papers				

Reconfigurable Computing						
Reconfigurable Computing						
Module number:	Workload (h):	Credits:	Regular Cycle:			
	180	6	winter term			
	Semester number:	Duration (in sem.):	Teaching Language:			
		1	en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Reconfigurable Computing	L2 Ex3	75	105	CE	50/20

3 Wahlpflichtmodule

2	<p>Options within the module:</p> <p>none</p>								
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Reconfigurable Computing:</i></p> <p>Recommended Proficiencies</p> <p>Knowledge of “Digital Design” and “Computer Architecture” is beneficial.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Reconfigurable Computing:</i></p> <p>This lecture provides an understanding of architectures and design methods for reconfigurable hardware systems and presents applications in the areas of high performance computing and embedded systems.</p> <ul style="list-style-type: none"> • Introduction: evolution of programmable logic devices, market economics • Architectures: FPGA architectures, reconfigurable devices, reconfigurable systems • Design methods: CAD for FPGAs, high-level languages and compilers, system-level design • Applications: custom computing machines, embedded systems 								
5	<p>Learning outcomes and competences:</p> <p>After attending the course, the students are able to</p> <ul style="list-style-type: none"> • explain the architectures of reconfigurable hardware devices, • name and analyze the main design methods and • judge the suitability of reconfigurable hardware for different application domains. <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 25%; text-align: center;">Duration or scope</th> <th style="width: 20%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of achievement</th> <th style="width: 25%; text-align: center;">Duration or Scope</th> <th style="width: 20%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						

3 Wahlpflichtmodule

8	<p>Prerequisites for participation in examinations: Passing of course achievement</p>
9	<p>Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade: The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses: Masterstudiengang Informatik v4</p>
12	<p>Module coordinator: Prof. Dr. Marco Platzner</p>
13	<p>Other Notes: <i>Remarks of course Reconfigurable Computing:</i> Implementation method</p> <ul style="list-style-type: none"> • Lecture with projector and board • Interactive exercises in the lecture room • Computer-based exercises with reconfigurable systems <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Lecture slides and exercise sheets • Exercise sheets and technical documentation for the for the computer-based exercises • S. Hauck and A. DeHon (editors): Reconfigurable Computing, Volume 1: The Theory and Practice of FPGA-Based Computation, Morgan Kaufmann, 2008 • Information about alternative and additional literature as well as teaching material on the course's website and in the lecture slides

Routing and Data Management in Networks						
Routing and Data Management in Networks						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Routing and Data Management in Networks	L3 Ex2	75	105	CE	40/20

3 Wahlpflichtmodule

2	<p>Options within the module:</p> <p>none</p>								
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Routing and Data Management in Networks:</i></p> <p>Recommended Proficiencies</p> <p>Algorithm design, theoretical correctness and efficiency proofs, tools from combinatorics and probability theory.</p>								
4	<p>Contents:</p> <p><i>Contents of the course Routing and Data Management in Networks:</i></p> <p>Routing and data management are fundamental tasks to be solved in order to ensure efficient use of large networks, e.g. the Internet, peer-to-peer systems, or wireless mobile ad-hoc networks. This lecture deals with algorithms and their analysis for routing and data management in such systems and describes, in particular, methods for dealing with their dynamics (movement of nodes, joining and exiting nodes). In particular, local, distributed algorithms, often as online algorithms, are considered.</p> <ul style="list-style-type: none"> • Offline and online routing strategies • Scheduling strategies • Data management strategies 								
5	<p>Learning outcomes and competences:</p> <p>The students get to know fundamental techniques in the area of routing and data management of large networks. They can decide in which situation which data management, scheduling, or routing algorithm is most appropriate. They can adapt algorithms to a new situation.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Attitude • Self-monitoring 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 25%; text-align: center;">Duration or scope</th> <th style="width: 20%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min bzw. 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min bzw. 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations:		
Passing of course achievement			
9	Prerequisites for assigning credits:		
The credit points are awarded after the module examination was passed.			
10	Weighing for overall grade:		
The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses:		
Masterstudiengang Informatik v4			
12	Module coordinator:		
Prof. Dr. Friedhelm Meyer auf der Heide			
13	Other Notes:		
<i>Remarks of course Routing and Data Management in Networks:</i>			
Implementation method			
<ul style="list-style-type: none"> • Lecture with beamer and blackboard • Practice in small groups • Expected activities of the students: Solving homework exercises, contributing to the tutorials 			
Learning Material, Literature			
<ul style="list-style-type: none"> • Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Frank Thomson Leighton, M. Kaufmann Publishers, 1992. • Research papers, script, slide set of the lecture, exercise sheets 			

Software Quality Assurance			
Software Quality Assurance			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	summer term
	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

3 Wahlpflichtmodule

1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a)	Software Quality Assurance	L3 Ex2	75	105	CE	90/30
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course Software Quality Assurance:</i> Recommended Proficiencies Programming, Modeling, Model-based software development						
4	Contents: <i>Contents of the course Software Quality Assurance:</i> The aim of the lecture is to cover approaches, technologies and strategies related to quality assurance for software systems. These include on the one hand constructive approaches such as design patterns, anti-patterns, domain-specific languages, model driven development, model quality analysis, and architectural styles, and on the other hand analytic approaches such as static reviewing techniques and dynamic testing techniques.. Furthermore, approaches for the improvement of the software development process and international standards like ISO 9001, 9126, CMM etc. are covered. <ul style="list-style-type: none"> • Introduction to software quality assurance • Standards <ul style="list-style-type: none"> – Product-related Standards: ISO 9126 – Process-related Standards: ISO 9001, CMM • Constructive approaches <ul style="list-style-type: none"> – Patterns and styles: Design patterns, Anti-Patterns, Architectural styles – Model-driven development – Metamodeling – Domain Specific Languages – Design by contract – Research: Process constraints • Analytical approaches <ul style="list-style-type: none"> – Reviews, inspections – Testing: Fundamental Test Process, Black Box Testing, White Box Testing 						

3 Wahlpflichtmodule

5	<p>Learning outcomes and competences:</p> <p>The students are able to explain quality characteristics of software development processes, software models as well as software systems. They have understood constructive and analytical techniques used to ensure quality properties, and they are able to apply them. They can describe standards for measuring process and product quality. They are able to understand new research approaches in the area of process and product quality.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Empathy • Learning competence • Learning motivation • Motivation 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	ZU	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
ZU	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ZU</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement and the the qualified participation will be conducted.</p>	ZU	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
ZU	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>								
11	<p>Reuse in degree courses:</p> <p>Masterstudiengang Informatik v4</p>								
12	<p>Module coordinator:</p> <p>Prof. Dr. Gregor Engels</p>								

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course Software Quality Assurance:</i></p> <p>Implementation method Partially slides and partially board writing. All essential concepts and techniques will be repeatedly applied in examples during the tutorial. In a lab part, the techniques will be employed using tools, particularly testing tools.</p> <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Daniel Galin: Software Quality Assurance: From Theory to Implementation, Pearson / Addison Wesley, 2004 • Slides, Exercises
----	---

Statistical Natural Language Processing							
Statistical Natural Language Processing							
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term				
	Semester number:	Duration (in sem.): 1	Teaching Language: en				
1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) Statistical Natural Language Processing	L2 Ex3	75	105	CE	30	
2	Options within the module: none						
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Statistical Natural Language Processing:</i></p> <p>Recommended Proficiencies Vector spaces, grammar of natural languages, probability theory</p>						

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Statistical Natural Language Processing:</i></p> <p>The goal of this lecture is to present students with the foundational tools and methods necessary to implement natural language processing pipelines. The course includes content pertaining to text preprocessing, parsing, distributional semantics, dedicated machine learning approaches and applications such as question answering.</p> <ul style="list-style-type: none"> • Text normalization • Language modeling • Spelling correction • Machine Learning • POS Tagging • Parsing • Distributional semantics • Word senses • Knowledge Extraction • Question Answering 								
5	<p>Learning outcomes and competences:</p> <p>Students can list relevant problems and identify solution requirements for the following areas:</p> <ul style="list-style-type: none"> • Text preprocessing • Language modelling • Spelling correction • Text and document classification • Distributional Semantics • Question Answering <p>They are aware of basic techniques in these areas, can identify limitations and shortcomings of these techniques when applied to concrete problem situations, and develop modifications of these techniques for specific areas. They can evaluate such modifications qualitatively and quantitatively.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Learning competence • Media competence • Literacy (scientific) 								
6	<p>Assessments:</p> <p> <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations:		
Passing of course achievement			
9	Prerequisites for assigning credits:		
The credit points are awarded after the module examination was passed.			
10	Weighing for overall grade:		
The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses:		
Masterstudiengang Informatik v4			
12	Module coordinator:		
Prof. Dr. Axel-Cyrille Ngonga Ngomo			
13	Other Notes:		
<i>Remarks of course Statistical Natural Language Processing:</i>			
Implementation method			
The weekly lectures (2SWS) cover new content on a weekly basis. In addition to the formal considerations, we will cover applications and corresponding limitations of the methods presented throughout the course. The exercises (1SWS) are both theoretical and practical in nature. The learners are to show that they understood the concepts and can apply them to practical problems. The mini-project (2SWS) give the students a holistic view of how to solve complex problems using Semantic Web technologies.			
Learning Material, Literature			
Slides and homework assignments			

Statistical Signal Processing			
Statistical Signal Processing			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	winter term
	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

3 Wahlpflichtmodule

1	Module structure:							
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)		
	a)	Statistical Signal Processing	L2 Ex2	60	120	CE	60/20	
2	Options within the module: none							
3	Admission requirements: <i>Prerequisites of course Statistical Signal Processing:</i> Recommended Proficiencies Undergraduate courses in signal processing and probability							
4	Contents: <i>Contents of the course Statistical Signal Processing:</i> Statistical signal processing comprises the techniques that engineers and statisticians use to draw inference from imperfect and incomplete measurements. This course covers a selection of topics from the major domains of detection, estimation, and time series analysis. Topics that may be covered in this course include correlation analysis, linear minimum mean-squared error estimation, performance bounds for parameter estimation, Neyman-Pearson detectors, wide-sense stationary, nonstationary and cyclostationary time series, and complex-valued random signals.							
5	Learning outcomes and competences: After attending this course, students will be familiar with the basic principles of statistical signal processing. They will understand how to apply statistical signal processing techniques to relevant fields in electrical engineering (such as communications). Students will develop confidence in their ability to solve mathematical problems of analysis and design. They will be able to apply the principles they have learned in this course to other areas. Non-cognitive Skills <ul style="list-style-type: none">• Commitment• Cooperation• Learning competence							
6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)							
	zu	Type of examination	Duration or scope	Weighting for the module grade				
	a)	Written or oral examination	90-120 min or 40 min	100%				
	The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.							

3 Wahlpflichtmodule

7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations:		
Passing of course achievement			
9	Prerequisites for assigning credits:		
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Weighing for overall grade:		
The module is weighted according to the number of credits (factor 1).			
11	Reuse in degree courses:		
Masterstudiengang Informatik v4			
12	Module coordinator:		
Prof. Dr. Peter Schreier			
13	Other Notes:		
<i>Remarks of course Statistical Signal Processing:</i>			
Implementation method			
Lectures and exercises (including some computer simulations)			
Learning Material, Literature			
Handouts and tutorial questions, literature references will be given in the first lecture.			

Strategische Produktplanung im Zeitalter der Digitalisierung						
Strategic product planning in the age of digitalization						
Module number:	Workload (h):	Credits:	Regular Cycle:			
	180	6	winter term			
	Semester number:	Duration (in sem.):	Teaching Language:			
		1	de			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Strategic Product Planning in the Age of Digitalization	L2 Ex2 P2	90	90	CE	35

3 Wahlpflichtmodule

2	<p>Options within the module:</p> <p>none</p>								
3	<p>Admission requirements:</p> <p>none</p>								
4	<p>Contents:</p> <p><i>Contents of the course Strategische Produktplanung im Zeitalter der Digitalisierung:</i> In the age of digitalization, fascinating opportunities for new products and services are opening up more than ever before. The only question is, will these new products be successful in the market? The goal of the lecture is a comprehensive overview of methods and tools of strategic planning in the age of digitization. Students are taught the essential topics of strategic planning. This includes basics, methods and IT tools. The findings from the lecture will be explained and deepened in an accompanying exercise and case study. The lecture focuses on intelligent technical systems with a high proportion of software.</p>								
5	<p>Learning outcomes and competences:</p> <p>The aim is to understand basic concepts of strategic planning and to be able to apply selected methods. In particular, potential identification, product identification and business planning are investigated in depth.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Cooperation • Motivation • Self-monitoring 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/>Final module exam (MAP) <input type="checkbox"/>Module exam (MP) <input type="checkbox"/>Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of examination</th> <th style="width: 20%; text-align: center;">Duration or scope</th> <th style="width: 25%; text-align: center;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td style="text-align: center;">90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 45%; text-align: center;">Type of achievement</th> <th style="width: 20%; text-align: center;">Duration or Scope</th> <th style="width: 25%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								

3 Wahlpflichtmodule

9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>
10	<p>Weighing for overall grade:</p> <p>The module is weighted according to the number of credits (factor 1).</p>
11	<p>Reuse in degree courses:</p> <p>keine</p>
12	<p>Module coordinator:</p> <p>Prof. Dr.-Ing. Roman Dumitrescu</p>
13	<p>Other Notes:</p> <p><i>Remarks of course Strategische Produktplanung im Zeitalter der Digitalisierung:</i></p> <p>Implementation method Lecture with exercise and accompanying case study</p> <p>Learning Material, Literature Slides and exercises; literature references will be given in the first lecture.</p>

Topics in Pattern Recognition and Machine Learning						
Topics in Pattern Recognition and Machine Learning						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Topics in Pattern Recognition and Machine Learning	L2 Ex2	60	120	CE	20
2	Options within the module: none					
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Topics in Pattern Recognition and Machine Learning:</i></p> <p>Recommended Proficiencies Elementary knowledge in Probability Theory, as is taught in the course Statistical Signal Processing. Basic knowledge in statistical and machine learning.</p>					

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Topics in Pattern Recognition and Machine Learning:</i></p> <p>The course on “Topics in Pattern Recognition and Machine Learning” first briefly summarizes the main concepts of statistical pattern recognition and machine learning. Next selected topics will be presented in detail. The choice of topics depends on current research activities and thus may change over time. Examples of such topics to be studied in detail include:</p> <ul style="list-style-type: none">• Model estimation in the presence of hidden variables, in order to reveal suspected latent structure buried in the data• Specific classification tasks, such as automatic speech recognition• Fundamentals of statistical pattern recognition: Bayes rule, learning of class-conditional densities, linear models for classification and regression• EM Algorithm and extensions thereof• Models with discrete or continuous latent variables; GMM, NMF• Bias-Variance dilemma and the tradeoff between degree of detail and generalizability of models• Graphical models• Sequential data and Hidden Markov models and their application in speech recognition• Recent publications in pattern recognition and machine learning <p>While the first part of the course will follow a regular lecture format, the second part will include active student participation. Students will be asked to read, analyze and present recently published papers from the pattern recognition and machine learning literature. This will often also include the implementation of proposed algorithms in Matlab.</p>
5	<p>Learning outcomes and competences:</p> <p>After completion of the module students will be able to</p> <ul style="list-style-type: none">• Choose an appropriate classifier for a given classification problem and be able to learn the parameters of the classifier from training data• Choose an appropriate regression method for function approximation and learn its parameters from training data• Search for latent variables and structure in given data• Make an informative choice for the model order to find a good compromise between degree of detail and generalizability• Comprehend and analyze recent publications from the field of pattern recognition and machine learning <p>Non-cognitive Skills</p> <ul style="list-style-type: none">• Commitment• Cooperation• Learning competence• Literacy (scientific)

3 Wahlpflichtmodule

6	Assessments:	<input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
	zu	Type of examination	Duration or scope	Weighting for the module grade
	a)	Written or oral examination	90-120 min bzw. 40 min	100%
	The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:	none		
8	Prerequisites for participation in examinations:	none		
9	Prerequisites for assigning credits:	The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade:	The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses:	Masterstudiengang Informatik v4		
12	Module coordinator:	Prof. Dr. Reinhold Häb-Umbach		
13	Other Notes:	<p><i>Remarks of course Topics in Pattern Recognition and Machine Learning:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lectures predominantly using the blackboard or overhead projector, occasional presentations of (powerpoint) slides, • Exercise classes with exercise sheets and demonstrations on computer • Instructions how to read and analyze scientific publications in this field • Autonomous analysis of publications and presentation of results and gained insight <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, 2001 • K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990 • C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006 		

Topics in Signal Processing
Topics in Signal Processing

3 Wahlpflichtmodule

Module number:	Workload (h): 180	Credits: 6	Regular Cycle: winter term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Topics in Signal Processing	L2 Ex2	60	120	CE	20
2	Options within the module: none					
3	Admission requirements: <i>Prerequisites of course Topics in Signal Processing:</i> Recommended Proficiencies Signal and system theory, at least a basic understanding of probability and linear algebra					
4	Contents: <i>Contents of the course Topics in Signal Processing:</i> This course covers a selection of current topics in signal processing. One part of this course will follow a regular lecture format, while the other part will require student participation. The course will first review relevant aspects of linear algebra and probability theory. Then students will learn how to read, analyze, and present recent papers from the signal processing literature.					
5	Learning outcomes and competences: In this course, students will familiarize themselves with some current research topics in signal processing. They will learn to read and understand scientific publications and to critically evaluate results. Non-cognitive Skills <ul style="list-style-type: none"> • Commitment • Cooperation • Learning competence • Literacy (scientific) 					

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement: none		
8	Prerequisites for participation in examinations: none		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Prof. Dr. Peter Schreier		
13	Other Notes: <i>Remarks of course Topics in Signal Processing:</i> Implementation method <ul style="list-style-type: none"> • Lecture with student participation • Student presentations Learning Material, Literature References will be given during first lecture.		

VLSI Testing			
VLSI Testing			
Module number:	Workload (h):	Credits:	Regular Cycle:
	180	6	winter term
	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

3 Wahlpflichtmodule

1	Module structure:						
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)	
	a) VLSI Testing	L2 Ex2	60	120	CE	40/20	
2	Options within the module: none						
3	Admission requirements: <i>Prerequisites of course VLSI Testing:</i> Recommended Proficiencies Digital Design						
4	Contents: <i>Contents of the course VLSI Testing:</i> The course focuses on techniques for detecting hardware defects in micro-electronic circuits. Algorithms for test data generation and test response evaluation as well as hardware structures for design for test (DFT) and on-chip test implementation (BIST) are presented. In detail the following topics are covered: <ul style="list-style-type: none">• Fault models• Testability measures and design for test (DFT)• Logic and fault simulation• Automatic test pattern generation (ATPG)• Built-in self-test (BIST), in particular test data compression and test response compaction• Memory test						
5	Learning outcomes and competences: After attending the course, the students will be able to <ul style="list-style-type: none">• describe fault models, DFT techniques, and test tools,• explain and apply the underlying models and algorithms for fault simulation and test generation,• analyze systems with respect to their testability and to derive appropriate test strategies. Non-cognitive Skills <ul style="list-style-type: none">• Commitment• Cooperation• Learning competence						

3 Wahlpflichtmodule

6	Assessments: <input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)		
zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	90-120 min or 40 min	100%
The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.			
7	Study Achievement:		
zu	Type of achievement	Duration or Scope	SL / QT
a)	Written exercises		CA
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.			
8	Prerequisites for participation in examinations: Passing of course achievement		
9	Prerequisites for assigning credits: The credit points are awarded after the module examination was passed.		
10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).		
11	Reuse in degree courses: Masterstudiengang Informatik v4		
12	Module coordinator: Prof. Dr. Sybille Hellebrand		

3 Wahlpflichtmodule

13	<p>Other Notes:</p> <p><i>Remarks of course VLSI Testing:</i></p> <p>Implementation method</p> <ul style="list-style-type: none"> • Lecture based on slide presentation, extensions on blackboard • Exercises in small groups based on exercise sheets with students presenting their own solutions • Hands-on exercises using various software tools <p>Learning Material, Literature</p> <ul style="list-style-type: none"> • Handouts of lecture slides • M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits, Boston, Dordrecht, London: Kluwer Academic Publishers, 2000 • L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann Series in Systems on Silicon, ISBN: 0123705975 • Additional links to books and other material available in PANDA
----	---

Web Security						
Web Security						
Module number:	Workload (h): 180	Credits: 6	Regular Cycle: summer term			
	Semester number:	Duration (in sem.): 1	Teaching Language: en			
1	Module structure:					
	Course	form of teaching	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
	a) Web Security	L3 Ex2	75	105	CE	40
2	Options within the module: none					
3	<p>Admission requirements:</p> <p><i>Prerequisites of course Web Security:</i></p> <p>Recommended Proficiencies Knowledge in programming, IT security and basic knowledge in cryptography</p>					

3 Wahlpflichtmodule

4	<p>Contents:</p> <p><i>Contents of the course Web Security:</i></p> <p>Modern web applications and web services usually consist of multiple layers. They are based on different (often complex) technologies that are constantly being developed. Their complexity is often the reason for new types of attacks that can be observed on the web every day.</p> <p>In this lecture, we will focus on the most important technologies and learn what you have to consider while securing your web applications. We will introduce prominent and widespread attacks and show how to prevent them. These range from typical attacks from the OWASP Top 10 list, such as XSS or SQL Injection, to attacks on web services and Single Sign-On standards (e.g., on SAML and OpenID Connect). Based on many cases, we will learn what is important in the design and implementation of secure web applications.</p>								
5	<p>Learning outcomes and competences:</p> <p>After successful completion, students have a comprehensive understanding of the technical aspects of web applications, web services, and various authentication mechanisms. They have learned that the web technologies used today are complex and that their complexity poses many security problems. Students have an overview of current web attacks and know how to prevent them practically.</p> <p>Non-cognitive Skills</p> <ul style="list-style-type: none"> • Team work • Literacy (scientific) 								
6	<p>Assessments:</p> <p><input checked="" type="checkbox"/> Final module exam (MAP) <input type="checkbox"/> Module exam (MP) <input type="checkbox"/> Partial module exams (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of examination</th> <th style="width: 20%;">Duration or scope</th> <th style="width: 25%;">Weighting for the module grade</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written or oral examination</td> <td>90-120 min or 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.</p>	zu	Type of examination	Duration or scope	Weighting for the module grade	a)	Written or oral examination	90-120 min or 40 min	100%
zu	Type of examination	Duration or scope	Weighting for the module grade						
a)	Written or oral examination	90-120 min or 40 min	100%						
7	<p>Study Achievement:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Type of achievement</th> <th style="width: 20%;">Duration or Scope</th> <th style="width: 25%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Written exercises</td> <td></td> <td style="text-align: center;">CA</td> </tr> </tbody> </table> <p>Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</p>	zu	Type of achievement	Duration or Scope	SL / QT	a)	Written exercises		CA
zu	Type of achievement	Duration or Scope	SL / QT						
a)	Written exercises		CA						
8	<p>Prerequisites for participation in examinations:</p> <p>Passing of course achievement</p>								
9	<p>Prerequisites for assigning credits:</p> <p>The credit points are awarded after the module examination was passed.</p>								

3 Wahlpflichtmodule

10	Weighing for overall grade: The module is weighted according to the number of credits (factor 1).
11	Reuse in degree courses: Masterstudiengang Informatik v4
12	Module coordinator: Prof. Dr.-Ing. Juraj Somorovsky
13	Other Notes: <i>Remarks of course Web Security:</i> Implementation method Lecture with exercises Learning Material, Literature <ul style="list-style-type: none">• Lecture slides• Scientific papers

4 Focus Areas

4.1 Algorithm Design

Coordination

Prof. Dr. rer. nat. Johannes Blömer

Included Modules

- Advanced Algorithms
- Advanced Complexity Theory
- Advanced Distributed Algorithms and Data Structures
- Algorithms for Highly Complex Virtual Scenes
- Clustering Algorithms
- Combinatorial Optimization
- Efficiency in Games
- Foundations of Cryptography
- Game Theory
- Introduction to Quantum Computation
- Quantum Algorithms
- Quantum Complexity Theory
- Real World Crypto Engineering
- Routing and Data Management in Networks
- Web Security

Description

In this focus area students can concentrate on studying the

- important techniques for the design of efficient algorithms
- application areas for the design of efficient algorithms, i.e. computer graphics, networks, big data, ...
- limits for the design of efficient algorithms, i.e. complexity theory
- constructive use of complexity theory in cryptography and security
- connection between efficient algorithms and verification and software design

4.2 Computer Systems

Coordination

Prof. Dr. Marco Platzner

Included Modules

- Advanced Computer Architecture
- Algorithms for Synthesis and Optimization of Integrated Circuits
- Approximate Computing
- Architektur paralleler Rechnersysteme
- High-Performance Computing
- Reconfigurable Computing
- VLSI Testing

Description

The focus area “Computer systems” goes into technical depths of various aspects of modern computer systems. Main topics are the analysis and evaluation of computer architectures, systematic methods for design and optimisation of computer systems, in particular the interplay of hardware and software, and programming models and methods for parallel and specialised computer architectures, which are increasingly gaining importance.

4.3 Data Science

Coordination

Prof. Dr. Axel Ngonga

Included Modules

- Advanced Algorithms
- Advanced Distributed Algorithms and Data Structures
- Data Science in Industrial Applications
- Digitale Sprachsignalverarbeitung
- Foundations of Knowledge Graphs
- High-Performance Computing
- Information Retrieval
- Machine Learning I
- Machine Learning II
- Statistical Natural Language Processing
- Topics in Pattern Recognition and Machine Learning

4 Focus Areas

- Topics in Signal Processing

Description

Data science is a young scientific discipline in the intersection of computer science, statistics, mathematics, and engineering, which has quickly developed into one of the most impactful areas in the current research landscape. It is a main driving factor of the digitalization and “datafication” of a large portion of our society, including companies, research organizations, and even private homes and people. In science and research, it is often viewed as a “fourth paradigm”, next to the empirical, theoretical, and computational approach. Broadly speaking, the major goal of data science is to develop methodological and algorithmic foundations as well as computer systems for automating the extraction of useful knowledge and insight from data.

The focus area “data science” will provide the students with solid theoretical foundations as well as practical skills that constitute the profile of a modern data scientist. To this end, courses will be offered in three main directions: Mathematical and algorithmic foundations, data analytics, software and systems. Here, the students will learn how to acquire, archive, compress, and aggregate large amounts of heterogeneous data (text, image, audio and video, etc.), and how to analyze such data using methods from statistics, machine learning, and data mining. Moreover, they will be familiarized with relevant programming languages, software engineering techniques, and scalable information processing architectures. Finally, the students will broaden their practical experience and develop soft skills by specializing in application areas such as Industrial Data Science, Digital Humanities, Business Analytics and Cybersecurity.

4.4 Intelligence and Data

Coordination

Prof. Dr. Axel Ngonga

Included Modules

- Clustering Algorithms
- Computational Argumentation
- Data Science for Physics and Engineering
- Foundations of Knowledge Graphs
- Information Retrieval
- Logic Programming for Artificial Intelligence
- Machine Learning I
- Machine Learning II
- Optimization Methods for Machine Learning
- Planning and Heuristic Search
- Statistical Natural Language Processing
- Statistical Signal Processing

Description

Intelligent systems are computer systems the behavior of which is controlled by methods and algorithms from artificial intelligence (AI). Systems of that kind are becoming increasingly important, not only on a scientific level but also in a social context: Autonomous or semi-autonomous systems such as service robots, self-driving cars or medical diagnosis systems will have a deep impact on our future private and professional life. In addition to methodological advances and improved hardware, the “data explosion” can be seen as a main driving factor for the rapid development of AI-systems during the last decade: Thanks to the availability of massive amounts of data or sensory feedback from their environment, intelligent systems are able to automatically improve their behavior through adaptation and learning.

This focus area covers important aspects of intelligent systems design and conveys corresponding theoretical and methodological foundations. This includes lectures on machine learning and data analysis, data management, computer graphical and visual data analysis, as well as swarm intelligence and robotics.

4.5 Networks and Communication

Coordination

Dr. rer. nat. Florian Klingler

Included Modules

- Advanced Distributed Algorithms and Data Structures
- Mobile Communication
- Networked Embedded Systems
- Routing and Data Management in Networks

Description

The focus area “Networks and Communication” teaches architectures, methods and systems of modern communication technology. To this end, we investigate methods of various abstraction levels, starting from the lowest level physical transmissions up to and including application design in distributed environments. Different types of systems are considered, ranging from conventional mobile communication over ad hoc networks and vehicular communication systems to networking in data centers and architectures for the future Internet at large. In doing so, we strive to build the bridge to aspects of distributed systems design. Questions on architecture design and options for protocol designs are complemented by the evaluation of such systems. To answer those questions, we introduce experimental and statistical performance evaluation techniques.

4.6 Software Engineering

Coordination

Prof. Dr. Eric Bodden

Included Modules

- Build It, Break It, Fix It
- Data-Driven Innovation and Engineering
- Data Science in Industrial Applications
- Designing code analyses for large-scale software systems 1
- Designing code analyses for large-scale software systems 2
- High-Performance Computing
- Kontextuelle Informatik
- Logic Programming for Artificial Intelligence
- Model-Based Systems Engineering
- Selected Topics in Software Engineering
- Software Quality Assurance

Description

In this focus area, students can concentrate on studying concepts, languages, methods, techniques, and tools for the systematic development of software systems. These comprise

- constructive techniques for developing functional and non-functional aspects of a system,
- formal and informal analytical techniques to ensure high quality of a system,
- systematic techniques to enable situation-specific process models

5 Modules in Winter Semester

• Advanced Algorithms	14
• Advanced Complexity Theory	16
• Advanced Computer Architecture	18
• Advanced Distributed Algorithms and Data Structures	20
• Algorithms for Highly Complex Virtual Scenes	22
• Approximate Computing	27
• Clustering Algorithms	37
• Combinatorial Optimization	39
• Kontextuelle Informatik	76
• M.079.4077 Data Science for Physics and Engineering	45
• Data Science in Industrial Applications	47
• Data-Driven Innovation and Engineering	43
• Designing code analyses for large-scale software systems 1	49
• Foundations of Knowledge Graphs	63
• Game Theory	66
• Studium Generale – Master	12
• High-Performance Computing	69
• Information Retrieval	72
• Logic Programming for Artificial Intelligence	78
• Machine Learning I	81
• Machine Learning II	83
• Master-Abschlussarbeit	4
• Mobile Communication	85
• Networked Embedded Systems	90
• Planning and Heuristic Search	94
• Projektgruppe	6
• Real World Crypto Engineering	101
• Reconfigurable Computing	103
• Ausgewählte Themen im Gebiet Software Engineering	32
• Seminar I	8
• Seminar II	10
• Statistical Natural Language Processing	110
• Statistical Signal Processing	112
• Strategische Produktplanung im Zeitalter der Digitalisierung	114
• Topics in Pattern Recognition and Machine Learning	116
• Topics in Signal Processing	118
• VLSI Testing	120

6 Modules in Summer Semester

• Advanced Complexity Theory	16
• Algorithms for Synthesis and Optimization of Integrated Circuits	25
• Architektur paralleler Rechnersysteme	30
• Build It, Break It, Fix It	34
• Clustering Algorithms	37
• Computational Argumentation	41
• Designing code analyses for large-scale software systems 2	52
• Digitale Sprachsignalverarbeitung	55
• Efficiency in Games	57
• Foundations of Cryptography	61
• Game Theory	66
• Studium Generale – Master	12
• Introduction to Quantum Computation	74
• Logic Programming for Artificial Intelligence	78
• Machine Learning I	81
• Machine Learning II	83
• Master-Abschlussarbeit	4
• Model-Based Systems Engineering	87
• Optimierungsverfahren für das maschinelle Lernen	92
• Projektgruppe	6
• Quantum Algorithms	97
• Quantum Complexity Theory	99
• Routing and Data Management in Networks	105
• Ausgewählte Themen im Gebiet Software Engineering	32
• Seminar I	8
• Seminar II	10
• Software Quality Assurance	107
• Web Security	123

7 Modules in English

• Advanced Algorithms	14
• Advanced Complexity Theory	16
• Advanced Computer Architecture	18
• Advanced Distributed Algorithms and Data Structures	20
• Algorithms for Highly Complex Virtual Scenes	22
• Algorithms for Synthesis and Optimization of Integrated Circuits	25
• Approximate Computing	27
• Build It, Break It, Fix It	34
• Clustering Algorithms	37
• Combinatorial Optimization	39
• Computational Argumentation	41
• M.079.4077 Data Science for Physics and Engineering	45
• Data Science in Industrial Applications	47
• Data-Driven Innovation and Engineering	43
• Designing code analyses for large-scale software systems 1	49
• Designing code analyses for large-scale software systems 2	52
• Efficiency in Games	57
• Foundations of Cryptography	61
• Foundations of Knowledge Graphs	63
• Game Theory	66
• General Studies – Master	12
• High-Performance Computing	69
• Information Retrieval	72
• Introduction to Quantum Computation	74
• Logic Programming for Artificial Intelligence	78
• Machine Learning I	81
• Machine Learning II	83
• Master Thesis	4
• Mobile Communication	85
• Networked Embedded Systems	90
• Optimization methods for machine learning	92
• Planning and Heuristic Search	94
• Project Group	6
• Quantum Algorithms	97
• Quantum Complexity Theory	99
• Real World Crypto Engineering	101
• Reconfigurable Computing	103
• Routing and Data Management in Networks	105
• Selected Topics in Software Engineering	32
• Seminar I	8

7 Modules in English

• Seminar II	10
• Software Quality Assurance	107
• Statistical Natural Language Processing	110
• Statistical Signal Processing	112
• Topics in Pattern Recognition and Machine Learning	116
• Topics in Signal Processing	118
• VLSI Testing	120
• Web Security	123

Erzeugt am 5. Oktober 2021 um 12:05.