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: 11. SEPTEMBER 2023

Inhaltsverzeichnis

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

<table>
<tr><td colspan="6">Master-Abschlussarbeit</td></tr>
<tr><td colspan="6">Master Thesis</td></tr>
<tr><td colspan="2">Module number:</td><td>Workload (h):
900</td><td>Credits:
30</td><td colspan="2">Regular Cycle:
summer- / winter term</td></tr>
<tr><td colspan="2"></td><td>Semester number:
4. Semester</td><td>Duration (in sem.):
1</td><td colspan="2">Teaching Language:
de / en</td></tr>
<tr><td>1</td><td colspan="5">Module structure:
<table>
<tr><th></th><th>Course</th><th>form of teachin</th><th>contact-time (h)</th><th>self-study (h)</th><th>status (C/CE)</th><th>group size (TN)</th></tr>
<tr><td>a)</td><td>Master Thesis – Work Plan</td><td></td><td>30</td><td>120</td><td>C</td><td>1</td></tr>
<tr><td>b)</td><td>Master Thesis</td><td></td><td>30</td><td>720</td><td>C</td><td>1</td></tr>
</table></td></tr>
<tr><td>2</td><td colspan="5">Options within the module:
none</td></tr>
<tr><td>3</td><td colspan="5">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.</td></tr>
<tr><td>4</td><td colspan="5">Contents:
Contents of the course Master-Abschlussarbeit – Arbeitsplan:
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.
Contents of the course Master-Abschlussarbeit:
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.</td></tr>
</table>

<table>
<tr><td>5</td><td>Learning outcomes and competences:
Finishing their master thesis students show that they are able
• to solve a problem in computer sience 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.

Non-cognitive Skills:
• Commitment
• Learning competence
• Learning motivation
• Motivation
• Literacy (scientific)
• Self-monitoring</td></tr>
<tr><td>6</td><td>Assessments:
☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)
<table>
<tr><th>zu</th><th>Type of examination</th><th>Duration or scope</th><th>Weighting for the module grade</th></tr>
<tr><td>a) - b)</td><td>Thesis</td><td>30-120 pages</td><td>100%</td></tr>
</table></td></tr>
<tr><td>7</td><td>Study Achievement:
none</td></tr>
<tr><td>8</td><td>Prerequisites for participation in examinations:
none</td></tr>
<tr><td>9</td><td>Prerequisites for assigning credits:
The credit points are awarded after the module examination was passed.</td></tr>
<tr><td>10</td><td>Weighing for overall grade:
The module is weighted with 50 credits.</td></tr>
<tr><td>11</td><td>Reuse in degree courses or degree course versions :
keine</td></tr>
<tr><td>12</td><td>Module coordinator:
Studiengangsbeauftragter Informatik</td></tr>
</table>

13	**Other Notes:** 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. *Remarks of course Master-Abschlussarbeit – Arbeitsplan:* **Implementation method** In agreement with supervisor. **Learning Material, Literature** Depending on the topic. *Remarks of course Master-Abschlussarbeit:* **Implementation method** Independent studies supported by individual advice and supervision **Learning Material, Literature** Depending on the thesis topic.

Projektgruppe			
Project Group			
Module number: M.079.4041	**Workload (h):** 600	**Credits:** 20	**Regular Cycle:** summer- / winter term
	Semester number: 2	**Duration (in sem.):** 2	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Project Group	PG	240	300	C	15

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Projektgruppe:
Recommended Proficiencies
Depending on the topic.

4 **Contents:**

Contents of the course Projektgruppe:
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.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Commitment
- Team work
- Learning competence
- Learning motivation
- Motivation
- Literacy (scientific)
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

zu	**Type of examination**	**Duration or scope**	**Weighting for the module grade**
a)	Partial Module Exam		100%

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.

7 **Study Achievement:**

zu	**Type of achievement**	**Duration or Scope**	**SL / QT**
a)	Practical work		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 as 8 credits.
11	**Reuse in degree courses or degree course versions :** keine
12	**Module coordinator:** Studiengangsbeauftragter Informatik
13	**Other Notes:** *Remarks of course Projektgruppe:* **Implementation method** • 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. **Learning Material, Literature** Depending on the topic.

Seminar I			
Seminar I			
Module number: M.079.4045	**Workload (h):** 150	**Credits:** 5	**Regular Cycle:** summer- / winter term
	Semester number: 2	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	2024.7092b Seminar	S2	30	120	C	15

2	**Options within the module:** Seminars from the Master Program Computer Science.
3	**Admission requirements:** *Prerequisites of course Seminar:* **Recommended Proficiencies** Depending on the seminar topic.
4	**Contents:** *Contents of the course Seminar:* 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. 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.
5	**Learning outcomes and competences:** 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. **Non-cognitive Skills** • Commitment • Cooperation • Learning competence • Media competence • Motivation • Literacy (scientific) • Self-monitoring

6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Presentation and seminar paper	45-60 minutes, 15-30 pages	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)	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 or degree course versions :** keine
12	**Module coordinator:** Studiengangsbeauftragter Informatik
13	**Other Notes:** *Remarks of course Seminar:* **Implementation method** Seminar paper and presentation **Learning Material, Literature** Depending on the seminar topic.

Seminar II			
Seminar II			
Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4046	150	5	summer- / winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
	3	1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	2024.7092b Seminar	S2	30	120	C	15

2 **Options within the module:**

Seminars from the Master Program Computer Science.

3 **Admission requirements:**

Prerequisites of course Seminar:
Recommended Proficiencies
Depending on the seminar topic.

4 **Contents:**

Contents of the course Seminar:
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.
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.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Commitment
- Cooperation
- Learning competence
- Media competence
- Motivation
- Literacy (scientific)
- Self-monitoring

6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

zu	**Type of examination**	**Duration or scope**	**Weighting for the module grade**
a)	Presentation and seminar paper	45-60 minutes, 15-30 pages	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)	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 or degree course versions :** keine
12	**Module coordinator:** Studiengangsbeauftragter Informatik
13	**Other Notes:** *Remarks of course Seminar:* **Implementation method** Seminar paper and presentation **Learning Material, Literature** Depending on the seminar topic.

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 teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Studium Generale – Master	L6 Ex3	135	225	C	30

2 **Options within the module:**

Any courses outside of computer science may be chosen.

3 **Admission requirements:**

Prerequisites of course Studium Generale – Master:
Recommended Proficiencies
Depending on the courses chosen.

4 **Contents:**

Contents of the course Studium Generale – Master:
Depending on the courses chosen.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Commitment
- Cooperation
- Media competence
- Literacy (scientific)
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Exam in general studies		100%

The responsible lecturer announces type and duration of assessment modalities in the first three weeks of the lecture period at latest.

<table>
<tr><td>7</td><td>Study Achievement:
<table>
<tr><th>zu</th><th>Type of achievement</th><th>Duration or Scope</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Qualified participation in general studies</td><td></td><td>QP</td></tr>
</table>
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the qualified participation will be conducted.</td></tr>
<tr><td>8</td><td>Prerequisites for participation in examinations:
none</td></tr>
<tr><td>9</td><td>Prerequisites for assigning credits:
The credit points are awarded after the module examination was passed.</td></tr>
<tr><td>10</td><td>Weighing for overall grade:
The module is weighted as 4 credits.</td></tr>
<tr><td>11</td><td>Reuse in degree courses or degree course versions :
keine</td></tr>
<tr><td>12</td><td>Module coordinator:
Studiengangsbeauftragter Informatik</td></tr>
<tr><td>13</td><td>Other Notes:
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.</td></tr>
</table>

3 Wahlpflichtmodule

Advanced Algorithms

Advanced Algorithms

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4002	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	**form of teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.079.05701 Advanced Algorithms	L3 Ex2	75	105	CE	50/25

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Advanced Algorithms:
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:**

Contents of the course Advanced Algorithms:
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.

- Randomized algorithms, derandomization, examples: Randomized Rounding and others
- Online algorithms, examples: scheduling problems and others
- approximation algorithms, examples: NP-hard problems

5 **Learning outcomes and competences:**

Students apply advanced algorithmic design methods such as randomization, approximation, and online algorithms to new problems and analyze them using combinatorial and probabilistic methods.

Non-cognitive Skills

- Learning competence
- Learning motivation

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Friedhelm Meyer auf der Heide

13	**Other Notes:** *Remarks of course Advanced Algorithms:* **Implementation method** • 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. **Learning Material, Literature** Standard textbooks, slides of the lecture, exercise sheets

Advanced Computer Architecture			
Advanced Computer Architecture			
Module number: M.079.4005	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05724 Advanced Computer Architecture	L3 Ex2	75	105	CE	50/25

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Advanced Computer Architecture:
Recommended Proficiencies
Basic knowledge in computer architecture.

4 **Contents:**

Contents of the course Advanced Computer Architecture:
The course teaches concepts and methods used in modern processor architecture to exploit the available parallelism at the levels of instructions, data and threads.

- 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 **Learning outcomes and competences:**

After attending the course, the students

- 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.

Non-cognitive Skills

- Team work
- Learning competence

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Christian Plessl, Prof. Dr. Marco Platzner
13	**Other Notes:** *Remarks of course Advanced Computer Architecture:* **Implementation method** • 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** • 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 Kauf- mann, 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:**
M.079.4006	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05700 Advanced Distributed Algorithms and Data Structures	L3 Ex2	75	105	C	30

2 **Options within the module:**

none

3	**Admission requirements:** *Prerequisites of course Advanced Distributed Algorithms and Data Structures:* **Recommended Proficiencies** Algorithms and data structures, distributed algorithms and data structures
4	**Contents:** *Contents of the course Advanced Distributed Algorithms and Data Structures:* 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.
5	**Learning outcomes and competences:** 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. **Non-cognitive Skills** • Team work • Learning competence • Literacy (scientific) • Self-monitoring
6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Christian Scheideler
13	**Other Notes:** *Remarks of course Advanced Distributed Algorithms and Data Structures:* **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):**	**Credits:**	**Regular Cycle:**
M.079.4009	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05756 Algorithms for Highly Complex Virtual Scenes	L3 Ex2	75	105	CE	40/20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Algorithms for Highly Complex Virtual Scenes:
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:**

Contents of the course Algorithms for Highly Complex Virtual Scenes:
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.

- 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

- Attitude
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Dr. Matthias Fischer
13	**Other Notes:** *Remarks of course Algorithms for Highly Complex Virtual Scenes:* **Implementation method** • 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. **Learning Material, Literature** • 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.

Approximate Computing			
Approximate Computing			
Module number: M.079.4068	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05758 Approximate Computing	L3 Ex2	75	105	CE	100/25

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Approximate Computing:
Recommended Proficiencies
Bachelor-level knowledge in digital design and computer architecture, Bachelor-level knowledge in mathematics, in particular linear algebra and probability theory

4 **Contents:**

Contents of the course Approximate Computing:
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.

- 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 **Learning outcomes and competences:**

After attending this course, the students are able

- 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

Non-cognitive Skills

- Learning competence

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Dr. Hassan Ghasemzadeh Mohammadi

13	**Other Notes:** *Remarks of course Approximate Computing:* **Implementation method** • Lecture with projector and black/white board • Interactive exercises/discussions in the lecture room • Computer-based tutorials **Learning Material, Literature** • 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.

Build It, Break It, Fix It

Build It, Break It, Fix It

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4013	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05713 Build It, Break It, Fix It	L3 Ex2	75	105	CE	20

2 **Options within the module:**

none

3	**Admission requirements:** *Prerequisites of course Build It, Break It, Fix It:* **Recommended Proficiencies** • 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:** *Contents of the course Build It, Break It, Fix It:* 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. 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. 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.
5	**Learning outcomes and competences:** After having attended this course, students will have. . . • 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 **Non-cognitive Skills** • Commitment • Cooperation • Learning competence • Self-monitoring

6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP) zu | **Type of examination** | **Duration or scope** | **Weighting for the module grade** a) | Partial Module Exam | | 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) | Practical work with subsequent discussion | | CA, QP 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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Eric Bodden
13	**Other Notes:** *Remarks of course Build It, Break It, Fix It:* **Learning Material, Literature** The lecture slides, task instructions and other materials will be uploaded to the course's PANDA page.

Combinatorial Optimization			
Combinatorial Optimization			
Module number: M.079.4074	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.0579 Combinatorial Optimization	L3 Ex2	75	105	CE	??

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Combinatorial Optimization:
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.

4 **Contents:**

Contents of the course Combinatorial Optimization:
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.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Attitude and mindset
- Self-control competence

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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.

<table>
<tr><td>7</td><td>Study Achievement:
<table>
<tr><th>zu</th><th>Type of achievement</th><th>Duration or Scope</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Written exercises</td><td></td><td>CA</td></tr>
</table>
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the course achievement will be conducted.</td></tr>
<tr><td>8</td><td>Prerequisites for participation in examinations:
Passing of course achievement</td></tr>
<tr><td>9</td><td>Prerequisites for assigning credits:
The credit points are awarded after the module examination was passed.</td></tr>
<tr><td>10</td><td>Weighing for overall grade:
The module is weighted according to the number of credits (factor 1).</td></tr>
<tr><td>11</td><td>Reuse in degree courses or degree course versions :
keine</td></tr>
<tr><td>12</td><td>Module coordinator:
Dr. Marten Maack</td></tr>
<tr><td>13</td><td>Other Notes:
Remarks of course Combinatorial Optimization:
Implementation method
• 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.

Learning Material, Literature
In addition to the usual lecture materials such as lecture notes or slides as well as exercise sheets, we refer to the following textbooks:
• 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.</td></tr>
</table>

Data-Driven Innovation and Engineering			
Data-Driven Innovation and Engineering			
Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4076	180	6	winter term

	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05722 Data-Driven Innovation and Engineering	L2 Ex3	75	105	CE	60

2 **Options within the module:**

none

3 **Admission requirements:**

none

4 **Contents:**

Contents of the course Data-Driven Innovation and Engineering:
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.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Commitment
- Empathy
- Social and ethical judgement
- Team work
- Cooperation
- Motivation
- Self-monitoring

6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr.-Ing. Roman Dumitrescu
13	**Other Notes:** *Remarks of course Data-Driven Innovation and Engineering:* **Implementation method** Lecture and exercise **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

Data Science for Dynamical Systems			
Data Science for Dynamical Systems			
Module number: M.048.92047	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number: 1.-3.Semester	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.048.92047 Data Science for Dynamical Systems	2L 2Ex, WS	60	120	C	30/30

2 **Options within the module:**

None

3 **Admission requirements:**

None

Prerequisites of course Data Science for Dynamical Systems:
None

4 **Contents:**

Contents of the course Data Science for Dynamical Systems:
This course has a modular structure and is offered in an interdisciplinary way for different degree programs and faculties. Depending on the available prior knowledge of the participants, the content will be tailored to the specific degree program. Overarching core topics include

- Basics of modelling dynamic systems using differential and difference equation models
- Data-driven identification methods for linear models on the basis of the least squares approach
- Data-driven identification methods for non-linear models (e.g., artificial neural networks)
- Learning of data-driven models utilizing a priori system knowledge
- Identification of underlying model structure equations (topology selection), e.g., by means of regularization or hypothesis tests with regard to competing objectives
- (Data-driven) model reduction
- Manipulation of the available model input data (dimensionality reduction and augmentation methods), e.g., autoencoders, principal component analysis and kernel methods
- Statistical evaluation of the available input and output data of dynamic systems as well as corresponding procedures for system excitation
- Statistical evaluation of the achieved model quality (over-fitting vs. under-fitting) by means of cross-validation

In addition to obtain new methodological knowledge, extensive programming and simulation exercises are developed using modern software programs (especially in the programming language Julia). Diverse application examples from the practice of various domains (e.g., engineering, natural sciences and economics) round off the course.

<table>
<tr><td>5</td><td>Learning outcomes and competences:
After completing the course, the participants are able to
• describe and apply methods for the identification of dynamic systems,
• critically evaluate identification results,
• to understand and analyze complex data-driven modelling tasks in interdisciplinary teams, to derive target-oriented solution methods and to evaluate independently developed results.</td></tr>
<tr><td>6</td><td>Assessments:
☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)
<table>
<tr><th>zu</th><th>Type of examination</th><th>Duration or scope</th><th>Weighting for the module grade</th></tr>
<tr><td>a)</td><td>Written or Oral Examination or Presentation</td><td>120-180 min or 30-45 min or 30 min</td><td>100%</td></tr>
</table>
Within the first three weeks of the lecture period each respective lecturer will specify the manner in which the examination will be conducted.</td></tr>
<tr><td>7</td><td>Study Achievement:
none</td></tr>
<tr><td>8</td><td>Prerequisites for participation in examinations:
None</td></tr>
<tr><td>9</td><td>Prerequisites for assigning credits:
The credit points are awarded after the module examination (MAP) was passed.</td></tr>
<tr><td>10</td><td>Weighing for overall grade:
The module is weighted according to the number of credits (factor 1).</td></tr>
<tr><td>11</td><td>Reuse in degree courses or degree course versions :
Masterstudiengang Electrical Systems Engineering v3 (ESEMA v3) - Amtssprache, Masterstudiengang Informatik v3</td></tr>
<tr><td>12</td><td>Module coordinator:
Dr.-Ing. Oliver Wallscheid, Dr. Sebastian Peitz</td></tr>
</table>

13	**Other Notes:** **Implementation** Modular flipped classroom course based on digital self-learning materials (especially learning videos) in conjunction with weekly contact appointments on campus for the discussion of questions, application examples, small group work as well as discussion of homework. Interdisciplinary course for study programs of different faculties with individual curricula as well as joint, interdisciplinary project phase. The latter takes place at the end of the course in small groups incl. final presentation of the results. **Teaching Material, Literature** • Learning videos, exercise tasks, programming examples • Brunton, Steven L., and J. Nathan Kutz. Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press, 2022. • Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016. • Isermann, Rolf, and Marco Münchhof. Identification of dynamic systems: an introduction with applications. Vol. 85. Heidelberg: Springer, 2011. • Nelles, Oliver. Nonlinear dynamic system identification. Springer Berlin Heidelberg, 2001.

Data Science in Industrial Applications

Data Science in Industrial Applications

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4075	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05795 Data Science in Industrial Applications	L3 Ex2	75	105	CE	40/20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Data Science in Industrial Applications:
Recommended Proficiencies
Foundations of mathematics (linear algebra, statistics), Programming and Algorithms.

4 **Contents:**

Contents of the course Data Science in Industrial Applications:
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.
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.
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.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Team work
- Cooperation
- Learning competence

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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)	Accompanying case study		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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr.-Ing. Roman Dumitrescu
13	**Other Notes:** *Remarks of course Data Science in Industrial Applications:* **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. **Learning Material, Literature** • 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:**
M.079.4070	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	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	C	30

2 **Options within the module:**

none

3	**Admission requirements:** *Prerequisites of course Designing code analyses for large-scale software systems 1:* **Recommended Proficiencies** A mature understanding of the Java programming languages and object-oriented programming will be helpful.
4	**Contents:** *Contents of the course Designing code analyses for large-scale software systems 1:* 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 our of research. We strongly recommend attending DECA 1 before DECA 2. **Topics covered include:** • Type systems and flow-insensitive, contraint-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 Throughout, we will discuss applications to software security.
5	**Learning outcomes and competences:** After having attended this course, students will have learned. . . • 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. **Non-cognitive Skills** • Learning competence • Learning motivation

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

13	**Other Notes:** *Remarks of course Designing code analyses for large-scale software systems 1:* **Implementation method** Lectures and group exercises as well as practical programming labs using worldwide leading frameworks for static code analysis **Learning Material, Literature** • 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:**
M.079.4071	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05821 Designing code analyses for large-scale software systems 2	L3 Ex2	75	105	C	30

2 **Options within the module:**

none

3	**Admission requirements:** *Prerequisites of course Designing code analyses for large-scale software systems 2:* **Recommended Proficiencies** 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.
4	**Contents:** *Contents of the course Designing code analyses for large-scale software systems 2:* 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 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. **Topics covered include:** • 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 Throughout, we will discuss applications to software security.

5 **Learning outcomes and competences:**

After having attended this course, students will have learned. . .

- 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.

Non-cognitive Skills

- Learning competence
- Learning motivation

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12	**Module coordinator:** Prof. Dr. Eric Bodden
13	**Other Notes:** *Remarks of course Designing code analyses for large-scale software systems 2:* **Implementation method** Lectures and group exercises as well as programming exercises using widely used real-world static analysis frameworks (e.g. Soot, Phasar, FlowDroid) **Learning Material, Literature** • 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 Octeau, 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 teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.048.24001 Digital Speech Signal Processing	L2 Ex2	60	120	CE	30

2 **Options within the module:**
none

3	**Admission requirements:** *Prerequisites of course Digitale Sprachsignalverarbeitung:* **Recommended Proficiencies** Prior knowledge from the module Higher Mathematics is helpful.
4	**Contents:** *Contents of the course Digitale Sprachsignalverarbeitung:* 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. • 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	**Learning outcomes and competences:** After attending the course, the students will be able to • 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. **Non-cognitive Skills** • Commitment • Learning competence

6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Reinhold Häb-Umbach
13	**Other Notes:** *Remarks of course Digitale Sprachsignalverarbeitung:* **Implementation method** • 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: M.079.4069	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05822 Efficiency in Games	L3 Ex2	75	105	CE	25

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Efficiency in Games:
Recommended Proficiencies
Calculus and probability. Game theory knowledge is not a prerequisite, but desirable. What really matters is a good ability to understand mathematical reasoning.

4	**Contents:** *Contents of the course Efficiency in Games:* 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. **Topics** • 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. – 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, Falk theorems • Efficiency with respect to other solution concepts and social welfare definitions. • Efficiency and altruism

5 **Learning outcomes and competences:**

To define, understand and use the following:

- 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
- Falk theorems
- Altruistic players

Non-cognitive Skills

- Attitude
- Learning competence
- Literacy (scientific)
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Jun.-Prof. Dr. Gleb Polevoy
13	**Other Notes:** *Remarks of course Efficiency in Games:* **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: • 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

Explainable Artificial Intelligence			
Explainable Artificial Intelligence			
Module number: M.079.4091	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Explainable Artificial Intelligence	L2 Ex1 P2	75	105	CE	20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Explainable Artificial Intelligence:
Recommended Proficiencies
Basic knowledge in machine learning and programming

4 **Contents:**

Contents of the course Explainable Artificial Intelligence:
Explaining the predictions of machine learning models is important in an increasing number of applications. For example, bank customers would like to know why their loan was denied; machine learning engineers would like to debug and improve their models; managers would like to ensure regulatory compliance. This course aims to explain the predictions of machine learning models and introduces different explanation methods to do so. Explanation methods can be distinguished whether they are specific to a certain model or model-agnostic and whether they explain an individual prediction or the entire model.

- Introduction (e.g., importance of interpretability, evaluation of interpretability, datasets used in case studies)
- Interpretable models (e.g., linear regression, logistic regression, decision trees, decision rules)
- Global model-agnostic methods (e.g., partial dependence plots, permutation feature importance, global surrogate models)
- Local model-agnostic methods (e.g., LIME, Anchors, SHAP, counterfactual explanations)
- Model-specific methods (e.g., for neural networks)

5 **Learning outcomes and competences:**

After completing the module, students will be able to

- recognize and discuss the importance of interpretability
- explain and apply important explanation methods (e.g., interpretable models, model-agnostic methods, and model-specific methods)
- recognize characteristics of datasets, machine learning tasks, and machine learning models in application problems and argue which explanation method is appropriate for a given problem
- implement simple explanation methods from scratch
- extend and modify existing explanation methods
- discuss problems and proposed solutions with experts in the field
- read and discuss research literature in the area of XAI

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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)	Mini project		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.

11 **Reuse in degree courses or degree course versions :**

keine

12 **Module coordinator:**

Dr. Stefan Heindorf

13 **Other Notes:**

Remarks of course Explainable Artificial Intelligence:
Implementation method
Slides and blackboard writing. Important concepts and techniques will be practiced through exercises in the lecture room and tutorials, and applied in a mini-project.
Learning Material, Literature

- Slides
- Exercises
- Book: Christoph Molnar. Interpretable machine learning. 2020.
- Additional material and literature will be announced in the course.

Foundations of Cryptography

Foundations of Cryptography

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4020	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	**form of teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.079.05801 Foundations of Cryptography	L3 Ex2	75	105	C	25

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Foundations of Cryptography:
Recommended Proficiencies
Basic Knowledge in IT-Security and cryptography useful but not necessary, basic concepts of complexity theory and probability theory

4 **Contents:**

Contents of the course Foundations of Cryptography:
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.

- 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

- Commitment
- Team work
- Learning motivation
- Literacy (scientific)
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Johannes Blömer

13	**Other Notes:** *Remarks of course Foundations of Cryptography:* **Implementation method** Lectures, exercises, reading groups **Learning Material, Literature** • Oded Gorldreich, Foundations of Cryptography I,II, • Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptogtraphy • Slides from the lectures

Foundations of Knowledge Graphs

Foundations of Knowledge Graphs

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4054	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05809 Foundations of Knowledge Graphs	L2 Ex3	75	105	CE	24

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Foundations of Knowledge Graphs:
Recommended Proficiencies
Graph theory, logics

4 **Contents:**

Contents of the course Foundations of Knowledge Graphs:
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 being 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.

- 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

5 **Learning outcomes and competences:**

The students can carry out the following after the completion of the module:

- 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,

Non-cognitive Skills

- Team work
- Learning competence
- Media competence
- Literacy (scientific)

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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:** 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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Axel-Cyrille Ngonga Ngomo
13	**Other Notes:** *Remarks of course Foundations of Knowledge Graphs:* **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. **Learning Material, Literature** Slides, homework assignments

Game Theory			
Game Theory			
Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4066	180	6	summer- / winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05744 Game Theory	L3 Ex2	75	105	CE	25

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Game Theory:
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	**Contents:** *Contents of the course Game Theory:* 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. **Topics** • 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 (im)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.), Falk 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 **Learning outcomes and competences:**

To define, understand and use the following:

- 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 falk theorems
- Cooperative games
- Transferable utility
- Simple games
- Core
- Bondareva-Shapley theorem
- The Shapley value and its axiomatic characterisation

Non-cognitive Skills

- Attitude
- Learning competence
- Literacy (scientific)
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Jun.-Prof. Dr. Gleb Polevoy
13	**Other Notes:** *Remarks of course Game Theory:* **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: • 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/ Concrete topics from their creators: • 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

Module number: M.079.4024	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	**form of teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.079.05823 High-Performance Computing	L2 Ex3	75	105	CE	40

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course High-Performance Computing:
Recommended Proficiencies

- 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-assement test

4 **Contents:**

Contents of the course High-Performance Computing:
This course teaches the foundations of high-performance computing with an emphasis on the programming of parallel computer systems and novel hardware accelerators.

- 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

5 **Learning outcomes and competences:**

After attending this course, the students are able to

- 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

Non-cognitive Skills

- Team work

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Christian Plessl

13	**Other Notes:** *Remarks of course High-Performance Computing:* **Implementation method** • Lecture with projected slides and blackboard notes • Interactive assignments in lecture room • Practical programming projects on parallel computer systems (teamwork in small groups) **Learning Material, Literature** • Lecture slides • Assignment sheets • Task descriptions and technical documentation for programming projects

Human Factors in Security and Privacy

Human Factors in Security and Privacy

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4092	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Human Factors in Security and Privacy	L3 Ex2	75	105	CE	60/20

2 **Options within the module:**

none

3 **Admission requirements:**

none

4 **Contents:**

Contents of the course Human Factors in Security and Privacy:
Humans are important actors in security. A provable secure system is only useful if it can be actually used by users, and system designers need to account for human behavior if they wish to have both security and usability. In this class, we will examine factors of usability of security and privacy through a research-based, project-driven examination. We will cover core areas of security and privacy, as well as cover methods in human interaction (HCI) that can be used to measure the usability of security and privacy. Students are expected to complete problem sets on the topic and complete a research-based project. We will also practice academic conference reviewing, and model the academic publishing process while learning how to write and present academic research.
The course includes the following contents:

- Erstens
- . . .
- 6 bis 12 Stichpunkte

5 **Learning outcomes and competences:**

Students will be able to

- read and write peer reviews of scientific papers in the area of security, privacy, and usability.
- understand and apply research methods in human factors in usable security and privacy.
- develop relevant hypotheses and research questions in the space of usable security and privacy
- design and deploy a research study and analyze the results.
- describe, support, and effectively argue a result using the best practices of scientific writing.
- understand ethical issues related to human factors research in security and privacy.
- understand the major topics and themes of usable security and privacy.
- present research results in class.

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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)	Homework assignments, project work		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.
11	**Reuse in degree courses or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Yasemin Acar
13	**Other Notes:** *Remarks of course Human Factors in Security and Privacy:* **Implementation Method** Fließtext mit ca. 200-500 Zeichen **Learning Material, Literature** • Literatur 1 • Literatur 2 • . . . • Additional literature will be announced in the course.

Information Retrieval

Information Retrieval

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4058	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	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	**Admission requirements:** *Prerequisites of course Information Retrieval:* **Recommended Proficiencies** Boolean algebra, vector spaces, stochastics
4	**Contents:** *Contents of the course Information Retrieval:* 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). • Boolean model • Indexing • Vector space model • Probabilistic model • Classification and clustering • PageRank
5	**Learning outcomes and competences:** The students possess the following abilities after the completion of this module: • 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. **Non-cognitive Skills** • Team work • Learning competence • Media competence • Literacy (scientific)
6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Axel-Cyrille Ngonga Ngomo
13	**Other Notes:** *Remarks of course Information Retrieval:* **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

Introduction to Quantum Computation			
Introduction to Quantum Computation			
Module number: M.079.4059	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05807 Introduction to Quantum Computation	L3 Ex2	75	105	C	40

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Introduction to Quantum Computation:
Recommended Proficiencies
Linear Algebra, algorithms.

4 **Contents:**

Contents of the course Introduction to Quantum Computation:
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.

- Quantum mechanics
- Quantum entanglement
- Quantum algorithms
- Quantum error correction
- Quantum information

5 **Learning outcomes and competences:**

Students are able to:

- 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

Non-cognitive Skills

- Learning competence
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

zu	Type of examination	Duration or scope	Weighting for the module grade
a)	Written or oral examination	120-180 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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Sevag Gharibian

13 **Other Notes:**

Remarks of course Introduction to Quantum Computation:
Implementation method
Slides and blackboard writing. All important concepts and techniques are further deepened with examples in exercises.
Learning Material, Literature

- 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:**
M.079.4027	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	de

1 **Module structure:**

	Course	**form of teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.079.05727 Contextual Informatics	L2 Ex3	75	105	CE	30

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Kontextuelle Informatik:
Recommended Proficiencies

- Good general knowledge
- Ability to read and analyze longer and complex texts from computer science as well as the humanities

4 **Contents:**

Contents of the course Kontextuelle Informatik:
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?
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.

- 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 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Social and ethical judgement
- Attitude
- Media competence
- Literacy (scientific)

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Dr. Harald Selke

13	**Other Notes:** *Remarks of course Kontextuelle Informatik:* **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** • 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.

Logic Programming for Artificial Intelligence

Logic Programming for Artificial Intelligence

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4031	180	6	summer- / winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05808 Logic Programming for Artificial Intelligence	L3 Ex2	75	105	CE	40/20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Logic Programming for Artificial Intelligence:
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:** *Contents of the course Logic Programming for Artificial Intelligence:* 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 quizes, but also for self-defined or domain specific languages. • 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
5	**Learning outcomes and competences:** Students learn factual knowledge about • the transformation of knowledge given as facts and rules into an executable programs • how to program in logic and in self-designed languages methodological knowledge, including • 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 transfer skills • the ability to transfer the methodolies 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 normative evaluation skills including the ability to assess • the suitability and limitations of different data and knowledgerepresentation 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 **Non-cognitive Skills** • Learning competence • Learning motivation

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Stefan Böttcher

13 **Other Notes:**

Remarks of course Logic Programming for Artificial Intelligence:
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

- 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:
M.079.4032	180	6	summer- / winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05717 Machine Learning I	L3 Ex2	75	105	CE	60/20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Machine Learning I:
Recommended Proficiencies
Basic knowledge in mathematics (linear algebra, statistics), programming and algorithms.

4 **Contents:**

Contents of the course Machine Learning I:
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.

- Introduction
- The Learning Problem
- Training versus Testing
- The Linear Model
- Non-Linear Methods
- Overfitting

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Learning competence
- Learning motivation
- Literacy (scientific)

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Eyke Hüllermeier

13 **Other Notes:**

Remarks of course Machine Learning 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

- Script
- Y.S. Abu-Mostafa, M. Magdon-Ismail, 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:**
M.079.4033	180	6	summer- / winter term

	Semester number:	Duration (in sem.):	Teaching Language:
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05810 Machine Learning II	L3 Ex2	75	105	CE	20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Machine Learning II:
Recommended Proficiencies
Basic knowledge in machine learning (as conveyed, for example, by the Machine Learning I lecture).

4 **Contents:**

Contents of the course Machine Learning II:
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.

- 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	**Learning outcomes and competences:** 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. **Non-cognitive Skills** • Learning competence • Learning motivation • Literacy (scientific)
6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Eyke Hüllermeier

13	**Other Notes:** *Remarks of course Machine Learning II:* **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** • Script • Y.S. Abu-Mostafa, M. Magdon-Ismail, 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 for Biometrics			
Machine Learning for Biometrics			
Module number: M.079.4088	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05709 Machine Learning for Biometrics	L2 Ex3	75	105	CE	50

2 **Options within the module:**
none

3 **Admission requirements:**
none

4 **Contents:**

Contents of the course Machine Learning for Biometrics:
Biometric verification is defined as the automated recognition of individuals based on their behavioral or biological characteristics. The course will give an overview of modern biometric systems and specifically address their functionality and challenges. For this purpose, various approaches of machine learning will be introduced, which aim at enabling reliable biometric recognition (e.g. by means of face recognition). At the same time, biometric applications place very specific requirements on the underlying algorithms. The course will specifically address these requirements and how they can be met algorithmically and in the algorithmic learning process. This includes the topics of privacy, fairness, explainability, uncertainties, efficiency, attacks and their automated detection.
The course includes the following content:

- Biometric systems, operation modes, and evaluation
- Recap on traditional and deep learning
- Face, Iris, and fingerprint recognition
- Soft-biometrics and privacy
- Fairness and bias in biometric systems
- Explainability and confidence in biometric systems
- Biometric sample quality
- Efficient biometric systems
- Presentation attacks and detection
- Multi-biometric fusion
- Biometric indexing

5 **Learning outcomes and competences:**

Students will be able to

- independently evaluate biometric systems,
- train biometric recognition models for different modalities,
- automatically detect biometric attacks and make systems robust against such attacks,
- explain various challenges of biometric systems and name solution strategies to counter them,
- name and explain open research questions in biometrics.

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Dr.-Ing. Philipp Terhörst
13	**Other Notes:** *Remarks of course Machine Learning for Biometrics:* **Implementation Method** First, students are given an overview of biometrics and its applications and basic functionalities. Then, required concepts of machine learning are introduced in a compact way. These will be applied and developed in context when dealing with specific biometric requirements. Parallel to the lecture, the theoretical concepts are practiced in the exercises using facial data. This is done in the form of short hand-written and implementation tasks. **Learning Material, Literature** • Anil K. Jain, Patrick Flynn, and Arun A. Ross. 2010. Handbook of Biometrics (1st. ed.). Springer Publishing Company, Incorporated. • Further literature will be announced in the lecture.

Model-Based Systems Engineering			
Model-Based Systems Engineering			
Module number: M.079.4062	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** de

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05815 Model-Based Systems Engineering	L3 Ex2	75	105	CE	??

2	**Options within the module:** none
3	**Admission requirements:** *Prerequisites of course Model-Based Systems Engineering:* **Recommended Proficiencies** Basics of Systems Engineerings
4	**Contents:** *Contents of the course Model-Based Systems Engineering:* 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. • 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, • Work in a model-based manner • Apply systems thinking • Create system architectures & derive requirements. **Non-Cognitive Competencies** • Self-monitoring • Literacy (scientific) • Learning competence • Learning motivation

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

13 **Other Notes:**

Remarks of course Model-Based Systems Engineering:
Implementation method
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.
Learning Material, Literature

- 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 Systemsengineeringhandbook: A Guide for System Life Cycle Processes and Activities(2015)
- Weilkiens, 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:**
M.079.4039	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05738 Networked Embedded Systems	L3 Ex2	75	105	CE	60/20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Networked Embedded Systems:
Recommended Proficiencies
System software and system-level programming

4 **Contents:**

Contents of the course Networked Embedded Systems:
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.

- 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 **Learning outcomes and competences:**

The learning objective is to unserstand the fundamental concepts of networded embedded systems. Students understand these concepts and are able to apply this knowledge.
Non-cognitive Skills

- Commitment
- Learning competence

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Marco Platzner
13	**Other Notes:** *Remarks of course Networked Embedded Systems:* **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):**	**Credits:**	**Regular Cycle:**
M.079.4085	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05806 Optimization methods for machine learning	L3 Ex2	75	105	CE	50

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Optimierungsverfahren für das maschinelle Lernen:
none

4	**Contents:** *Contents of the course Optimierungsverfahren für das maschinelle Lernen:* Machine learning has become an important aspect in countless applications from various disciplines. The central aspect in machine learning is the *automated identification and exploitation of patterns* in data sets. In *supervised learning*, 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 *generalizes*. The determination of the underlying parameters is in most cases realized by solving an *optimization problem*: Determine the parameters such that the error on the *training data set* is minimized. Here, the type of optimization critically depends on the machine learning approach as well as on the training data. In this lecture, we cover the basics for the training of machine learning methods in depth. The covered topics are: • Regression methods • Nonlinear optimization • The training of deep neural networks • Stochastic gradient methods • Multiobjective optimization approaches for machine learning
5	**Learning outcomes and competences:** Students learn • factual knowledge about – 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 – the configuration and implementation as well as the training of neural networks – the treatment of large data sets • transfer skills – 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 – 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 **Non-cognitive Skills** • learning competence • learning motivation

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Dr. Sebastian Peitz

13 **Other Notes:**

Remarks of course Optimierungsverfahren für das maschinelle Lernen:
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

- 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.

Post-Quantum Cryptography

Post-Quantum Cryptography

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4089	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	**form of teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	Post-Quantum Cryptography	L3 Ex2	75	105	CE	20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Post-Quantum Cryptography:
Recommended Proficiencies
Basics of cryptography and complexity theory

4 **Contents:**

Contents of the course Post-Quantum Cryptography:
IT security is largely based on modern cryptographic methods. These include many methods of so-called public-key cryptography such as the RSA and Elgamal encryption methods, the RSA signature method, and the various variants of the Digital Signature Algorithm (DSA). In 1994, Peter Shor presented an efficient algorithm for computing prime factorization of integers and for computing discrete logarithms in finite groups. Thus, all the aforementioned methods of public-key cryptography are insecure if quantum computers of sufficient size and complexity can be realized. It is therefore important to develop alternatives to classical public-key methods that, at least according to current research, cannot be broken by quantum computers. Important candidates (and some close to standardization) for such post-quantum secure methods rely on techniques of error-correcting codes and the geometry of numbers. In this lecture, we will present and discuss important candidates for post-quantum secure methods. The course includes the following contents:

- introduction to codes, lattices and discretised Gaussian distributions
- lattice and code based enryption
- lattice based signatures
- lattices and zero-knowledge proofs
- lattice based group signatures

5 **Learning outcomes and competences:**

Students will be able to

- understand and explain the difference between classical and post-quantum security.
- explain the importance of post-quantum cryptography for selected applications.
- explain and apply concepts from the field of geometry of numbers and error-correcting codes.
- explain important constructions from post-quantum cryptography and prove their security.
- explain security assumptions from post-quantum cryptography and apply them to new post-quantum primitives.

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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.

11 **Reuse in degree courses or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Johannes Blömer

13	**Other Notes:** *Remarks of course Post-Quantum Cryptography:* **Implementation Method** Basic concepts are presented in a lecture. In addition, theoretical concepts are deepened in tutorials in small groups as well as in written exercises. **Learning Material, Literature** References to current learning materials will be given in the lectures.

Privacy and Technology			
Privacy and Technology			
Module number: M.079.4087	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05705 Privacy and Technology	L2 Ex3	75	105	CE	40

2 **Options within the module:**

none

3 **Admission requirements:**

none

4 **Contents:**

Contents of the course Privacy and Technology:
This course provides students with a basic understanding of privacy risks, the most common technologies for addressing them and the human factors that shape their design.

- Privacy metrics and adversary models
- Anonymous communications
- Data-perturbative privacy-enhancing technologies
- Anonymization algorithms for databases
- Homomorphic encryption and zero knowledge proofs
- Selective disclosure for identity management
- Usable privacy
- Applying privacy principles and case studies

5 **Learning outcomes and competences:**

The students

- are able to reason critically about privacy,
- gain knowledge in the evaluation of privacy risks,
- understand the design aspects of privacy-enhancing technologies,
- get familiar with the latest research in the field and
- analyze and discuss the space of solutions to a given privacy problem

Non-cognitive Skills

- Literacy (scientific)
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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)	Practical work and discussion		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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Patricia Arias Cabarcos

13	**Other Notes:** *Remarks of course Privacy and Technology:* **Implementation method** • Weekly theory lecture • Exercise and assignments to interactively discuss in the practical lectures **Learning Material, Literature** • Lecture slides, scientific literature and specific readings will be provided during the course.

Quantum Algorithms

Quantum Algorithms

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4072	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05797 Quantum Algorithms	L3 Ex2	75	105	CE	20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Quantum Algorithms:
Recommended Proficiencies
Linear Algebra, Quantum Computing

4 **Contents:**

Contents of the course Quantum Algorithms:
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.

- Quantum circuits
- Algebraic problems
- Quantum walks
- Query complexity
- Adiabatic computation

5 **Learning outcomes and competences:**

Students are able to:

- 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

Non-cognitive Skills

- Learning competence
- Self-monitoring

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Sevag Gharibian
13	**Other Notes:** *Remarks of course Quantum Algorithms:* **Implementation method** Slides and blackboard writing. All important concepts and techniques are further deepened with examples in exercises. **Learning Material, Literature** • 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: M.079.4063	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05743 Quantum Complexity Theory	L3 Ex2	75	105	C	20

2 **Options within the module:**
none

3	**Admission requirements:** *Prerequisites of course Quantum Complexity Theory:* **Recommended Proficiencies** Linear Algebra, Quantum Computing
4	**Contents:** *Contents of the course Quantum Complexity Theory:* 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. • Complexity classes BQP, QCMA, QMA • Quantum Satisfiability Problems • Quantum Interactive Proofs • Tensor Networks • Semidefinite Programming
5	**Learning outcomes and competences:** Students are able to: • 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 **Non-cognitive Skills** • Learning competence • Self-monitoring
6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Sevag Gharibian
13	**Other Notes:** *Remarks of course Quantum Complexity Theory:* **Implementation method** Slides and blackboard writing. All important concepts and techniques are further deepened with examples in exercises. **Learning Material, Literature** • 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

Quantum Information			
Quantum Information			
Module number: M.079.4090	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Quantum Information	L3 Ex2	75	105	CE	40

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Quantum Information:
Recommended Proficiencies
Linear Algebra

4 **Contents:**

Contents of the course Quantum Information:
Over the last century, Quantum mechanics has had profound impacts on both fundamental science and technology. The emerging field of Quantum Information Theory studies a paradigm for information processing empowered by quantum mechanics. This field has demonstrated that quantum information processing can outperform its classical counterpart and is a revolutionary direction to investigate future information technologies. Quantum Information Science incorporates techniques from computer science, mathematics, and physics.
Of particular interest is quantum entanglement, which is the phenomenon that occurs when a group of particles is generated or interacts in a way such that the state of each particle cannot be described independently of the others, even when the particles are separated by arbitrarily large distances. Entanglement is a primary feature of quantum mechanics not present in classical physics and it is a resource behind most modern quantum technologies, such as quantum computers. This lecture introduces the advance concepts of quantum communication and information.
The contents include:

- Entanglement of two- and many-body systems
- Quantum information processing and applications
- Measures of Entanglement, Distance and Fidelity
- Higher local dimensions (qubits vs qudits)
- Quantum channels
- Classical and quantum error correcting codes and their differences

5 **Learning outcomes and competences:**

Students learn cutting-edge concepts at the intersection of computer science and quantum mechanics. This lecture equips students with advanced, interdisciplinary technical proficiency, enabling them to pursue careers in analysis intensive industries, technology start-ups, or research and development roles in leading technology companies or academia. To achieve this, the students get familiar with the basics of quantum mechanics and the related algebra. Furthermore, they will be able to:

- understand the underlying concepts of entangled systems (two-body and many-body),
- understand the fundamental idea of maximally entangled systems, classify and characterise them for practical applications,
- describe the basic notion of higher local dimension particles (qubits vs qudits),
- apply the theory of classical and quantum error correcting codes, and study their differences,
- to work on interdisciplinary topics and, in particular, to acquire the basics of different disciplines.

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

zu	**Type of examination**	**Duration or scope**	**Weighting for the module grade**
a)	Written or oral examination	120-180 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.

11 **Reuse in degree courses or degree course versions :**

keine

12	**Module coordinator:** Dr. Zahra Raissi
13	**Other Notes:** *Remarks of course Quantum Information:* **Implementation Method** Theoretical foundations and concepts will be taught in the form of lectures and deepened in practical exercise courses, group work as well as individual homework. **Learning Material, Literature** • Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000. • F. J. MacWilliams and N. J. A. Sloane. The Theory of Error-Correcting Codes, North-Holland Mathematical Library. North-Holland, Amesterdam, 1977. ISBN 9780444851932. • Ingemar Bengtsson and Karol Zyczkowski, Geometry of quantum states: an introduction to quantum entanglement, Cambridge university press, 2006, ISBN 9780511535048. • Lecture slides • Exercises

Real World Crypto Engineering

Real World Crypto Engineering

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4067	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05819 Real World Crypto Engineering	L3 Ex2	75	105	C	40

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Real World Crypto Engineering:
Recommended Proficiencies
Knowledge in programming, IT security and basic knowledge in cryptography

4 Contents:

Contents of the course Real World Crypto Engineering:
Strong cryptography is not always sufficient to protect primary security goals. Even if strong cryptographic algorithms are used, a lot can go wrong when they are implemented. This lecture will dive into the most important protocols and cryptographic protection mechanisms (e.g., TLS, SSH, WPA) and show their basic concepts. Then, we will present prominent attacks that ultimately break the desired security goals. Based on many cases, we will learn what is essential when designing and implementing cryptographic applications.

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

- Team work
- Literacy (scientific)

6 Assessments:

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :

keine

12	**Module coordinator:** Prof. Dr.-Ing. Juraj Somorovsky
13	**Other Notes:** *Remarks of course Real World Crypto Engineering:* **Implementation method** Lectures, exercises **Learning Material, Literature** Lecture slides, scientific papers

Reconfigurable Computing

Reconfigurable Computing

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4043	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05703 Reconfigurable Computing	L2 Ex3	75	105	CE	50/20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Reconfigurable Computing:
Recommended Proficiencies
Knowledge of “Digital Design” and “Computer Architecture” is beneficial.

4 **Contents:**

Contents of the course Reconfigurable Computing:
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.

- 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 **Learning outcomes and competences:**

After attending the course, the students are able to

- 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.

Non-cognitive Skills

- Team work
- Learning competence

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Marco Platzner

13	**Other Notes:** *Remarks of course Reconfigurable Computing:* **Implementation method** • Lecture with projector and board • Interactive exercises in the lecture room • Computer-based exercises with reconfigurable systems **Learning Material, Literature** • 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

Software Quality Assurance			
Software Quality Assurance			
Module number: M.079.4048	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05805 Software Quality Assurance	L3 Ex2	75	105	CE	90/30

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Software Quality Assurance:
Recommended Proficiencies
Programming, Modeling, Model-based software development

4	**Contents:** *Contents of the course Software Quality Assurance:* 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. • Introduction to software quality assurance • Standards – Product-related Standards: ISO 9126 – Process-related Standards: ISO 9001, CMM • Constructive approaches – Patterns and styles: Design patterns, Anti-Patterns, Architectural styles – Model-driven development – Metamodeling – Domain Specific Languages – Design by contract – Research: Process constraints • Analytical approaches – Reviews, inspections – Testing: Fundamental Test Process, Black Box Testing, White Box Testing
5	**Learning outcomes and competences:** 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. **Non-cognitive Skills** • Empathy • Learning competence • Learning motivation • Motivation

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 and the the qualified participation 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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Gregor Engels

13 **Other Notes:**

Remarks of course Software Quality Assurance:
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.
Learning Material, Literature

- 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: M.079.4055	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05702 Statistical Natural Language Processing	L2 Ex3	75	105	CE	30

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Statistical Natural Language Processing:
Recommended Proficiencies
Vector spaces, grammar of natural languages, probability theory

4 **Contents:**

Contents of the course Statistical Natural Language Processing:
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.

- Text normalization
- Language modeling
- Spelling correction
- Machine Learning
- POS Tagging
- Parsing
- Distributional semantics
- Word senses
- Knowledge Extraction
- Question Answering

5 **Learning outcomes and competences:**

Students can list relevant problems and identify solution requirements for the following areas:

- Text preprocessing
- Language modelling
- Spelling correction
- Text and document classification
- Distributional Semantics
- Question Answering

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.

Non-cognitive Skills

- Team work
- Learning competence
- Media competence
- Literacy (scientific)

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Axel-Cyrille Ngonga Ngomo
13	**Other Notes:** *Remarks of course Statistical Natural Language Processing:* **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: M.048.55105	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** winter term
	Semester number: 1.-3. Semester	**Duration (in sem.):** 1	**Teaching Language:** de

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.048.24014 Statistical Signal Processing	2L 2Ex, WS	60	120	C	40/40

2 **Options within the module:**

None

3 **Admission requirements:**

None

Prerequisites of course Statistical Signal Processing:
Recommended: Undergraduate courses in signal processing and probability

<table>
<tr><td>4</td><td>Contents:

Contents of the course Statistical Signal Processing:
Short Description
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.
Contents
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.</td></tr>
<tr><td>5</td><td>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 learnt in this course to other areas.</td></tr>
<tr><td>6</td><td>Assessments:

⊠Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)
<table><tr><th>zu</th><th>Type of examination</th><th>Duration or scope</th><th>Weighting for the module grade</th></tr><tr><td>a)</td><td>Written or Oral Examination</td><td>120-180 min or 30-45 min</td><td>100%</td></tr></table></td></tr>
<tr><td>7</td><td>Study Achievement:

none</td></tr>
<tr><td>8</td><td>Prerequisites for participation in examinations:

None</td></tr>
<tr><td>9</td><td>Prerequisites for assigning credits:

The credit points are awarded after the module examination (MAP) was passed.</td></tr>
<tr><td>10</td><td>Weighing for overall grade:

The module is weighted according to the number of credits (factor 1).</td></tr>
<tr><td>11</td><td>Reuse in degree courses or degree course versions :

keine</td></tr>
<tr><td>12</td><td>Module coordinator:

Prof. Dr. Peter Schreier</td></tr>
</table>

13	**Other Notes:** *Remarks of course Statistical Signal Processing:* **Course Homepage** `http://sst.upb.de/teaching` **Implementation** Lectures and tutorials **Teaching Material, Literature** Literature references are given in the first lecture.

Topics in Pattern Recognition and Machine Learning

Topics in Pattern Recognition and Machine Learning

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 teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.048.92030 Topics in Pattern Recognition and Machine Learning	L2 Ex2	60	120	CE	20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Topics in Pattern Recognition and Machine Learning:
Recommended Proficiencies
Elementary knowledge in Probability Theory, as is taught in the course Statistical Signal Processing. Basic knowledge in statistical and machine learning.

4	**Contents:** *Contents of the course Topics in Pattern Recognition and Machine Learning:* 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: • 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 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.
5	**Learning outcomes and competences:** After completion of the module students will be able to • 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 generalizabliliy • Comprehend and analyze recent publications from the field of pattern recognition and machine learning **Non-cognitive Skills** • Commitment • Cooperation • Learning competence • Literacy (scientific)

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Reinhold Häb-Umbach

13 **Other Notes:**

Remarks of course Topics in Pattern Recognition and Machine Learning:
Implementation method

- 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

Learning Material, Literature

- 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

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 teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.048.92014 Topics in Signal Processing	L2 Ex2	60	120	CE	20

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Topics in Signal Processing:
Recommended Proficiencies
Signal and system theory, at least a basic understanding of probability and linear algebra

4 **Contents:**

Contents of the course Topics in Signal Processing:
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

- Commitment
- Cooperation
- Learning competence
- Literacy (scientific)

6	**Assessments:** ☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr. Peter Schreier
13	**Other Notes:** *Remarks of course Topics in Signal Processing:* **Implementation method** • Lecture with student participation • Student presentations **Learning Material, Literature** References will be given during first lecture.

Unsupervised Learning and Evolutionary Optimisation Using R			
Unsupervised Learning and Evolutionary Optimisation Using R			
Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4093	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	Unsupervised Learning and Evolutionary Optimisation Using R	L3 Ex2	75	105	CE	70

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Unsupervised Learning and Evolutionary Optimisation Using R:
Recommended Proficiencies

- Basic knowledge and interest in mathematics, statistics and probability theory
- Basic knowledge of programming

4 **Contents:**

Contents of the course Unsupervised Learning and Evolutionary Optimisation Using R:
The course includes the formal and applied concepts of unsupervised machine learning and its implementation in the statistical programming language R.
In particular, the following topics are covered in a theoretical and applied manner:

- Introduction to the statistical programming language R
- Data pre-processing and quality aspects of data
- (Stream) clustering techniques
- Dimensionality reduction techniques
- Basic principles of evolutionary optimisation, both single- and multi-objective
- Practical application of the methods using R in individual and group work

5 **Learning outcomes and competences:**

After completing the module, students will be able to . . .

- properly assess data quality and select suitable techniques for data pre-processing
- explain and apply core methods of unsupervised learning
- understand the basic principles of evolutionary optimisation methods
- competently apply techniques to assess the quality of optimisation procedures
- use the statistical software R for statistical data analysis, unsupervised learning and evolutionary optimisation in a competent manner
- analyse problems in a team and present practice-relevant solutions

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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)	Assignments		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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Heike Trautmann

13	**Other Notes:** *Remarks of course Unsupervised Learning and Evolutionary Optimisation Using R:* **Implementation Method** An introduction to the statistical programming language R is given compactly in the first weeks of the course. Methods of unsupervised machine learning are covered within lecture presentations interleaved with interactive exercises. Methods understanding will be further deepened in tutorials focusing both on theory as well as application-oriented tasks using R. **Learning Material, Literature** Recommended for the statistical programming language R: • Hadley Wickham & Garrett Grolemund (2023). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. 2nd ed. O'Reilly • Torsten Hothorn and Brian S. Everitt (2014). A Handbook of Statistical Analyses Using R. Chapman & Hall/CRC Press, 3rd edition, 2014. • C. Heumann, M. Schomaker, and Shalabh. Introduction to Statistics and Data Analysis With Exercises, Solutions and Applications in R. Springer, 2017. The methods sections are based on a variety of references which will be announced in the lecture.

Usable Security and Privacy

Usable Security and Privacy

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.079.4086	180	6	summer term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
		1	en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05804 Usable Security and Privacy	L2 Ex3	75	105	CE	40

2 **Options within the module:**

none

3 **Admission requirements:**

none

4	**Contents:** *Contents of the course Usable Security and Privacy:* Human factors and usability issues have traditionally played a limited role in security research and secure systems development. Usability issues have been largely disregarded by security experts due to their failure to acknowledge their significance and their insufficient knowledge to tackle them. Today there is consensus on the importance of understanding users behavior and improving usability to achieve true security. This course provides practical and research-oriented knowledge about usable security and privacy. Students will gain practical experience through focused presence exercises and work in small teams to conduct a semester-wide research project with the goal of designing and pretesting a user study on human-centered security and privacy. For that, the course will present research methods and give an introduction into HCI and usability concepts. The course will also address foundational and state-of-the-art research topics in the area, such as privacy and transparency enhancing tools, usable authentication, and developer-centered security. By reviewing relevant papers and giving presentations, the students will get familiar with the latest research in the field and gain knowledge about how to work scientifically. The course includes the following contents: • Security and privacy concepts • Foundations of cryptography • Privacy and transparency enhacing tools • HCI and usability research methods • Ethics in technology • Quantitative and qualitative data analysis • Usable authentication • Usable privacy • Developer-centered security
5	**Learning outcomes and competences:** Students will • gain an appreciation for the importance of usable security and privacy • learn about the history of the field and main research areas and challenges • are able to apply methodologies to conduct user research in security and privacy • get familiar with the latest research in the field **Non-cognitive Skills** • Literacy (scientific) • Self-monitoring • Team work

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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)	Practical work with written report and discussion		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 or degree course versions :**

keine

12 **Module coordinator:**

Prof. Dr. Patricia Arias Cabarcos

13 **Other Notes:**

Remarks of course Usable Security and Privacy:
Implementation method
Basic concepts are presented in a lecture style format. By engaging in presence exercises and conducting a research project in small groups focused on a user-study for usable security and privacy research throughout the semester, students can acquire more profound theoretical and practical knowledge.
Learning Material, Literature

- Lazar, J., Feng, J.H. and Hochheiser, H., 2017. Research methods in human-computer interaction. Morgan Kaufmann.
- Redmiles, E.M., Acar, Y., Fahl, S. and Mazurek, M.L., 2017. A summary of survey methodology best practices for security and privacy researchers.
- Slides and scientific literature references will be given during the course.

VLSI-Testing

VLSI-Testing

Module number:	**Workload (h):**	**Credits:**	**Regular Cycle:**
M.048.55104	180	6	winter term
	Semester number:	**Duration (in sem.):**	**Teaching Language:**
	1.-3. Semester	1	de / en

1 **Module structure:**

	Course	**form of teachin**	**contact-time (h)**	**self-study (h)**	**status (C/CE)**	**group size (TN)**
a)	L.048.25005 VLSI Testing	2L 2Ex, WS	60	120	C	40/40

2 **Options within the module:**

None

3 **Admission requirements:**

None

Prerequisites of course VLSI Testing:
Recommended: Digital Design

4 **Contents:**

Contents of the course VLSI Testing:
Short Description
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.
Contents
In detail the following topics are covered:

- 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

<table>
<tr><td>5</td><td>Learning outcomes and competences:

Domain competence:
After attending the course, the students will be able

• to describe fault models, DFT techniques, and test tools,
• to explain and apply the underlying models and algorithms for fault simulation and test generation,
• to analyze systems with respect to their testability and to derive appropriate test strategies.

Key qualifications:
The students

• are able to apply the practiced strategies for problem solving across varying disciplines,
• have experience in presenting their solutions to their fellow students, and
• know how to improve their competences by private study.</td></tr>
<tr><td>6</td><td>Assessments:

☒Final module exam (MAP) □Module exam (MP) □Partial module exams (MTP)

<table><tr><th>zu</th><th>Type of examination</th><th>Duration or scope</th><th>Weighting for the module grade</th></tr><tr><td>a)</td><td>Written or Oral Examination or Presentation</td><td>120-180 min or 30-45 min or 30 min</td><td>100%</td></tr></table></td></tr>
<tr><td>7</td><td>Study Achievement:

none</td></tr>
<tr><td>8</td><td>Prerequisites for participation in examinations:

None</td></tr>
<tr><td>9</td><td>Prerequisites for assigning credits:

The credit points are awarded after the module examination (MAP) was passed.</td></tr>
<tr><td>10</td><td>Weighing for overall grade:

The module is weighted according to the number of credits (factor 1).</td></tr>
<tr><td>11</td><td>Reuse in degree courses or degree course versions :

keine</td></tr>
<tr><td>12</td><td>Module coordinator:

Prof. Dr. Sybille Hellebrand</td></tr>
</table>

13	**Other Notes:** *Remarks of course VLSI Testing:* **Course Homepage** `https://ei.uni-paderborn.de/en/electrical-engineering/date/teaching/electrical-engineering/overview` **Implementation** • 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 **Teaching Material, Literature** Additional material can be found in panda • Michael L. Bushnell, Vishwani D. Agrawal, „Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits,“Boston, Dordrecht, London: Kluwer Academic Publishers, 2000 • Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, „VLSI Test Principles and Architectures: Design for Testability,“ Morgan Kaufmann Series in Systems on Silicon, ISBN: 0123705975

Web Security			
Web Security			
Module number: M.079.4073	**Workload (h):** 180	**Credits:** 6	**Regular Cycle:** summer term
	Semester number:	**Duration (in sem.):** 1	**Teaching Language:** en

1 **Module structure:**

	Course	form of teachin	contact-time (h)	self-study (h)	status (C/CE)	group size (TN)
a)	L.079.05820 Web Security	L3 Ex2	75	105	C	40

2 **Options within the module:**

none

3 **Admission requirements:**

Prerequisites of course Web Security:
Recommended Proficiencies
Knowledge in programming, IT security and basic knowledge in cryptography

4 **Contents:**

Contents of the course Web Security:
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.
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.

5 **Learning outcomes and competences:**

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.
Non-cognitive Skills

- Team work
- Literacy (scientific)

6 **Assessments:**

☒Final module exam (MAP) □Module exam (MP) □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 or degree course versions :** keine
12	**Module coordinator:** Prof. Dr.-Ing. Juraj Somorovsky
13	**Other Notes:** *Remarks of course Web Security:* **Implementation method** Lecture with exercises **Learning Material, Literature** • 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
- Post-Quantum Cryptography
- Quantum Algorithms
- Quantum Complexity Theory
- Quantum Information
- Routing and Data Management in Networks

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
- Approximate Computing
- High-Performance Computing
- Human Factors in Security and Privacy
- Reconfigurable Computing
- Usable Security and Privacy
- 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-Cyrille Ngonga Ngomo

Included Modules

- Advanced Algorithms
- Advanced Distributed Algorithms and Data Structures
- Data Science for Dynamical Systems
- Data Science for Physics and Engineering
- Data Science in Industrial Applications
- Digitale Sprachsignalverarbeitung
- Explainable Artificial Intelligence
- 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
- Topics in Signal Processing
- Unsupervised Learning and Evolutionary Optimisation Using R

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-Cyrille Ngonga Ngomo

Included Modules

- Clustering Algorithms
- Data Science for Dynamical Systems
- Data Science for Physics and Engineering
- Explainable Artificial Intelligence
- Foundations of Knowledge Graphs
- Information Retrieval
- Logic Programming for Artificial Intelligence
- Machine Learning I

- Machine Learning II
- Machine Learning for Biometrics
- Optimization Methods for Machine Learning
- Statistical Natural Language Processing
- Statistical Signal Processing
- Unsupervised Learning and Evolutionary Optimisation Using R

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

Prof. Dr. Marco Platzner

Included Modules

- Advanced Distributed Algorithms and Data Structures
- Networked Embedded Systems
- Routing and Data Management in Networks
- Web Security

Description

The focus area “Networks and Communication” teaches architectures, methods and systems of modern communication technology. To this end, we investigate methods of various abstractions 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 Security

Coordination

Prof. Dr. Eric Bodden

Included Modules

- Advanced Distributed Algorithms and Data Structures
- Designing code analyses for large-scale software systems 1
- Designing code analyses for large-scale software systems 2
- Foundations of Cryptography
- Human Factors in Security and Privacy
- Introduction to Quantum Computation
- Machine Learning for Biometrics
- Post-Quantum Cryptography
- Privacy and Technology
- Quantum Complexity Theory
- Quantum Information
- Real World Crypto Engineering
- Usable Security and Privacy
- Web Security

Description

In all areas of life, digital technologies, such as the (Industrial) Internet of Things, Cyber-Physical Systems, Digital Automotives, Digital Health or Industry 4.0, offer immense innovation potential. However, increasing digitization requires new approaches to safely exploit this potential. In order to be able to tackle this challenge, there is a great need in industry, research and teaching for well-trained computer science experts with in-depth knowledge of IT security. In the specialization area “IT Security”, solid theoretical basic knowledge is taught in combination with practical skills. The course covers technical skills from the field of IT security (e.g. software security, formal verification, basics of modern cryptography and communication security), in which typical security vulnerabilities and attack techniques are presented and countermeasures and their effectiveness are examined.

Since security cannot be seen independently of concrete applications and different applications have different security requirements, specialized competencies in modern application fields with special security requirements (e.g., communication protocols in the mobile and automotive fields) as well as supplementary qualifications in the areas of algorithms and quantum computing are also covered.

4.7 Software Engineering

Coordination

Prof. Dr. Yasemin Acar

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
- Human Factors in Security and Privacy
- Kontextuelle Informatik
- Logic Programming for Artificial Intelligence
- Model-Based Systems 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

6 Modules in Summer Semester

7 Modules in English

Erzeugt am 11. September 2023 um 18:53.