Otto-Friedrich Universität Bamberg

Module Handbook

Module Handbook International Software System Science

Faculty of Information Systems and Applied Computer Sciences

According to the current version of the study and examination regulations of 24.09.2024 for the Master's degree programme International Software Systems Science at the Otto Friedrich University of Bamberg. In edffect starting winter semester 2024/25.

Valid: 07.11.2024

Notice on the validity of older versions of a module handbook:

1. date of validity

The module descriptions contained in this module handbook are valid for the first time for the semester indicated on the cover sheet.

2. transition regulations

a. Students who have already completed parts of a module according to a previously valid module handbook shall complete the module according to the previously valid version of the module handbook.

This transition regulation shall apply exclusively to the regular examination date immediately following the missed/not passed/not completed examination. At the request of the student, the examination board may, in justified cases, determine an extension of the transition period.

b. A module shall be deemed to have been completed in parts if the module examination has not been passed or missed. The same shall apply if at least one module examination has been passed, failed or missed.

Furthermore, a module shall be deemed to have been *partly completed* if the student has registered for a course assigned to the respective module in accordance with the previously applicable module handbook.

3. period of validity

This module handbook is valid for subsequent semesters *until the announcement of a changed module handbook*.

Equivalence List M.Sc. International Software Systems Science
Examination Regulations of xx.xx.2024

In the following you will find a list of modules whose name or abbreviation has been changed without a significant change to the module. If a module listed in the column "previous module" was successfully completed, the module listed in the column "new module" cannot be taken.

previous module			new module		
module abbreviation	**module name**	**valid until (semester)**	**module abbreviation**	**module name**	**valid from (semester)**
xAI-MML-M	Mathematics for Machine Learning	SS24	xAI-MML-B	Mathematics for Machine Learning	WS24/25
GdI-IFP-B	Introduction to Functional Programming	SS24	GdI-IFP-M	Introduction to Functional Programming	WS24/25

Date of the equivalence and deviations list: 18.07.2024

Modules

Index by areas of study

1) A1 Software System Science (Module group) ECTS: 36 - 54

In module group A1, at least 6 ECTS credits must be completed in each focus area. Each module can only be included once in the module group.

a) S1: Distributed and Mobile Systems (Focus Area) ECTS: 6 - 36

b) S2: Software Analysis and Verification (Focus Area) ECTS: 6 - 30

c) S3: Service-oriented Architectures (Focus Area) ECTS: 6 - 24

d) S4: Communication Systems and Protocols (Focus Area) ECTS: 6 - 36

4) A4 Master's Thesis (Module group) ECTS: 30

5) A5 International Experience (Module group) ECTS: 27

Modules amounting to 27 ECTS credits must be completed in module group A5.

a) Guided study abroad (Elective Area) ECTS: 0 - 27

Modules amounting to 0 to 27 ECTS credits can be included in the compulsory elective area a, which are completed as part of a guided study abroad program at a foreign university, provided that they differ significantly from the modules to be completed in accordance with these regulations and can be assigned to module groups A1, A2 or A3.

b) Internship in an International Context (Elective Area) ECTS: 0 - 12

For the internship a foreign or internationally acting domestic company (or research institute) may be selected. It has to offer a specific internship related to relevant topics of software systems science. The documentation of the internship requires the delivery of the following items to the degree programme representative:

- written report of 4 pages at least, reporting on the tasks and achievements, and
- a certificate issued by the hosting institution or the organizational unit that has realized the internship.

c) Foreign Languages (Elective Area) ECTS: 0 - 15

In the elective area 5c, Foreign languages, modules comprising up to 15 ECTS credits can be taken from the range offered by the University's Language Centre. Excluded are modules in English and modules in the language in which the university entrance qualification was obtained.

Details, in particular the modules available for selection and the module examinations, are described in the *Modulhandbuch des Sprachenzentrums der Otto-Friedrich-Universität Bamberg:* https://www.uni-bamberg.de/abt-studium/aufgaben/modulhandbuecher/sprachenzentrum/modulhandbuch-sprachenzentrum/

d) Further Modules from A1 and/or A2 (Elective Area) ECTS: 0 - 27

In this area, modules from A1 or A2, which were not previously completed, can be taken.
It is not possible to complete additional seminars or prejects (modules from A3).

Module AISE-Auto Automation of First- and Higher-Order Logic *Automation of First- and Higher-Order Logic*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Christoph Benzmüller	
Contents: This course provides an introduction to the theory and practice of automatic theorem proving. Interest is in the automation of classical propositional logic, first level classical logic, and higher level classical logic. The exact emphasis may vary from year to year. This also applies to the proof calculi considered in each case (tableaux, resolution, etc.), as well as the concrete implementation methodology chosen for the practical exercises.	
Learning outcomes: The students will acquire competencies regarding the development of sound and complete proof calculi for classical logic, and the application of a uniform abstract proof technique (abstract consistency) for achieving completeness results. They also acquire competencies for implementing such proof calculi with modern functional and agent-oriented programming languages. In addition, the course will explore ideas regarding the integration of machine learning techniques in automated theorem systems.	
prerequisites for the module: none	
Recommended prior knowledge: First basic knowledge in logic and first programming skills are recommended, but not mandatory (and can be worked up in an additional tutorial/exercise group parallel to the course).	**Admission requirements:** none

Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Automation of First- and Higher-Order Logic **Mode of Delivery:** Lectures and Practicals **Language:** German **Frequency:** every summer semester	**6,00 Weekly Contact Hours**
Learning outcome: The students will acquire competencies regarding the development of sound and complete proof calculi for classical logic, and the application of a uniform abstract proof technique (abstract consistency) for achieving completeness results. They also acquire competencies for implementing such proof calculi with modern functional and agent-oriented programming languages. In addition, the course will explore ideas regarding the integration of machine learning techniques in automated theorem systems.	
Contents: This course provides an introduction to the theory and practice of automatic theorem proving. Interest is in the automation of classical propositional logic, first level classical logic, and higher level classical logic.	

The exact emphasis may vary from year to year. This also applies to the proof calculi considered in each case (tableaux, resolution, etc.), as well as the concrete implementation methodology chosen for the practical exercises.	

Examination Oral examination / Duration of Examination: 30 minutes	

Module AISE-UL Universal Logic & Universal Reasoning *Universelle Logik & Universelles Schließen*	6 ECTS / 180 h
(since WS22/23) Person responsible for module: Prof. Dr. Christoph Benzmüller	
Contents: Knowledge representation and reasoning applications in computer science, AI, philosophy and math typically employ very different logic formalisms. Instead of a "single logic that serves it all" (as envisioned already by Leibniz) an entire "logic zoo" has been developed, in particular, during the last century. Logics in this zoo, e.g., include modal logics, conditional logics, deontic logics, multi-valued logics, temporal logics, dynamic logics, hybrid logics, etc. In this lecture course we will introduce, discuss and apply a meta logical approach to universal logical reasoning that addresses this logical pluralism. The core message is this: While it might not be possible to come up with a universal object logic as envisioned by Leibniz, it might in fact be possible to have a universal meta logic in which we can semantically model, analyse and apply various species from the logic zoo. Classical higher order logic (HOL) appears particularly suited to serve as such a universal meta logic, and existing reasoning tools for HOL can fruitfully be reused and applied in this context.	
Learning outcomes: The participants of this course will, in combination with a hands-on introduction to Isabelle/HOL, learn about HOL, about semantical embeddings (SSE technique) of non-classical logics in HOL, and about proof automation of these logics in Isabelle/HOL. They will conduct practical exercises regarding the application of the SSE technique in philosophy, mathematics or artificial intelligence, including, normative reasoning and machine ethics.	
Remark: The main language of instruction in this course is English. The overall workload of 180h for this module consists of: • weekly classes: 22h • tutorials: 8h • Work on assignment: 90h • Literature study 40h • preparation for and time of the final exam: 20h	
prerequisites for the module: none	
Recommended prior knowledge: Basic knowledge about classical and non-classical logics, theoretical computer science.	**Admission requirements:** non

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester Semester

Module Units	
AISE-UL: Universal Logic & Universal Reasoning (Universelle Logik & Universelles Schließen) **Mode of Delivery:** Lectures and Practicals **Lecturers:** Prof. Dr. Christoph Benzmüller	**2,00 Weekly Contact Hours**

Language: English
Frequency: every winter semester

Learning outcome:
The participants of this course will, in combination with a hands-on introduction to Isabelle/HOL, learn about HOL, about semantical embeddings (SSE technique) of non-classical logics in HOL, and about proof automation of these logics in Isabelle/HOL. They will conduct practical exercises regarding the application of the SSE technique in philosophy, mathematics or artificial intelligence, including, normative reasoning and machine ethics.

Contents:
Introduction to and discussion of tools and practical issues closely related to the topics discussed in the lecture as well as solutions of problems that come up during working on the practical assignment.

Literature:
will be announced in lecture course

Examination
Written examination, AISE-UL: Universal Logic & Universal Reasoning (Universelle Logik & Universelles Schließen)
Description:
Oral examination concerning the topics discussed in the lecture, exercises and assignment. Students may choose English or German as the language for the written assignment and oral examination. Examinations will take at the end of the summer term or at the beginning of the winter term (students may choose one of them). Students are assumed to work on an advanced modelling assignment ('schriftliche Hausarbeit') during the semester that is introduced at the beginning of the semester and uses the most important technologies (such as the See technique) discussed during the semester.

Note: Without working on the modelling assignment over the term students may run into problems during their oral examination (Kolloquium) as we discuss questions concerning topics from the lectures as well as from the assignment; questions about the assignment are based on the assignment solution modelled by the students.

Module Units

AISE-UL: Universal Logic & Universal Reasoning (Universelle Logik & Universelles Schließen) — **2,00 Weekly Contact Hours**
Mode of Delivery: Practicals
Lecturers: Prof. Dr. Christoph Benzmüller
Language: English
Frequency: every winter semester

Learning outcome:
The participants of this course will, in combination with a hands-on introduction to Isabelle/HOL, learn about HOL, about semantical embeddings (SSE technique) of non-classical logics in HOL, and about proof automation of these logics in Isabelle/HOL. They will conduct practical exercises regarding the application of

the SSE technique in philosophy, mathematics or artificial intelligence, including, normative reasoning and machine ethics. **Contents:** Knowledge representation and reasoning applications in computer science, AI, philosophy and math typically employ very different logic formalisms. Instead of a "single logic that serves it all" (as envisioned already by Leibniz) an entire "logic zoo" has been developed, in particular, during the last century. Logics in this zoo, e.g., include modal logics, conditional logics, deontic logics, multi-valued logics, temporal logics, dynamic logics, hybrid logics, etc. In this lecture course we will introduce, discuss and apply a meta logical approach to universal logical reasoning that addresses this logical pluralism. The core message is this: While it might not be possible to come up with a universal object logic as envisioned by Leibniz, it might in fact be possible to have a universal meta logic in which we can semantically model, analyse and apply various species from the logic zoo. Classical higher order logic (HOL) appears particularly suited to serve as such a universal meta logic, and existing reasoning tools for HOL can fruitfully be reused and applied in this context. **Literature:** will be announced in lecture course	

Module AlgoK-Algo Algorithms *Algorithmen*	6 ECTS / 180 h
Person responsible for module: Prof. Dr. Isolde Adler	
Contents: Algorithms and algorithmic problem solving are at the heart of computer science. This module introduces students to the design and analysis of efficient algorithms. Students learn how to quantify the efficiency of an algorithm and what algorithmic solutions are efficient. Techniques for designing efficient algorithms are taught, including efficient data structures. We begin with standard methods such as Divide-and-Conquer and Dynamic Programming. We then move on to more advanced techniques and we discuss ways of dealing with computationally intractable problems and large data sets. This is done using illustrative and fundamental problems relevant to Computer Science and AI.	
Learning outcomes: On completion of the module student should be able to: - Demonstrate an understanding of what constitutes an efficient and an inefficient solution to a computational problem, - Analyse the efficiency of algorithms, - Evaluate and justify appropriate ways to provide efficient solutions for computational problems, - Identify and apply different design principles in the design of algorithms, - Describe efficient algorithms for a range of computational problems, along with their computational complexity, - Articulate the key concepts and critically evaluate approaches in a clear and rigorous manner, - Appreciate and understand in-depth the role of proofs in the area of algorithm design, - Recognise how the methods learned can be extended and used to solve other problems.	
Remark: The workload for this module is approxmately structured as follows: • Participation in lectures and tutorials: 45 hrs • Preparing and revising the lectures and tutorials: 60 hours • Solving the worksheets: 45 hrs • Exam preparation: 30 hrs	
prerequisites for the module: none	
Recommended prior knowledge: Prerequisites: Basic knowledge of algorithms and data structures, proof techniques, mathematical skills. Good English language skills.	**Admission requirements:** none
Frequency: alle 4 Semester | **Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Algorithms **Mode of Delivery:** Lectures and Practicals **Lecturers:** Prof. Dr. Isolde Adler	**4,00 Weekly Contact Hours**

Language: English/German

Contents:

The lectures introduce the topics, providing an in-depth explanation including motivation, intuition, examples and proofs, as well as tools, techniques and applications.

The tutorials consist of hands-on problem solving, including exam-style problems.

Literature:

- Jon Kleinberg and Éva. Tardos: Algorithm Design, Pearson/Addison-Wesley 2006.
- Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani: Algorithms, McGraw-Hill, 2006
- Anany Levitin, Design and analysis of algorithms, Pearson/Addison-Wesley 2007.
- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullmanm, Data structures and algorithms, Addison-Wesley 1987
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to algorithms, 1st ed. MIT press and McGraw-Hill 1990 or 2nd ed. MIT press and McGraw-Hill 2001 or 3rd ed. MIT press and McGraw-Hill 2009.
- Kenneth H. Rosen: Discrete Mathematics and its Applications. McGraw-Hill, 2012.
- K. Houston: How to Think Like a Mathematician: A Companion to Undergraduate
  Mathematics. Cambridge University Press, 2009

Examination

No type selected

Description:

Oral exam (30 minutes) or written exam (90 minutes).

Depending on the number of participants, the exam will either be an oral exam or a written exam. The mode of examination will be communicated in the first lecture.

It is possible to contribute to your overall module grade by solving worksheets regularly and successfully, and by participating actively in the tutorials. However, it is also possible to achieve a "first" (1,0) by excelling in the exam.

Module AlgoK-Sem-M Master Seminar Algorithms and Complexity Theory *Masterseminar Algorithmen und Komplexitätstheorie*	3 ECTS / 90 h
Person responsible for module: Prof. Dr. Isolde Adler	
Contents: Selected topics in the area of Algorithms and Complexity Theory.	
Learning outcomes: Ability to develop problem solutions from independent research into the current academic literature, specifically with focus on mathematical tools; Ability to communicate complex problem solving approaches orally and in writing. Promotion of scientific curiosity and the formation of a self-confident attitude towards research and problem solving.	
prerequisites for the module: none	
Recommended prior knowledge: Discrete mathematics, in particular graph theory; mathematical proof techniques; algorithms and data structures; elementary logic and algebra; LaTeX. English language skills at level B2 (UniCert II) or above.	**Admission requirements:** none

Frequency: winter and summer semester, on demand	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Master Seminar Algorithms and Complexity Theory **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Isolde Adler **Language:** English/German **Frequency:** winter and summer semester, on demand	**2,00 Weekly Contact Hours**
Contents: Selected topics in the area of Algorithms and Complexity Theory are presented by the participants. The module will be taught in English or German. English is the default language.	
Literature: Relevant literature will be communicated at the beginning of the semester and during the first sessions.	

Examination Internship report / Duration of Examination: 30 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regular participation at the seminar. **Description:** Presentation (30 minutes) and a written report (4 months).	

Module DSG-DSAM-M Distributed Systems Architectures and Middleware *Distributed Systems Architecture and Middleware*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(since WS19/20) Person responsible for module: Prof. Dr. Guido Wirtz	

Contents:

This course introduces students to the ideas, benefits, technologies and issues related to server-centric distributed systems and middleware in general. The core topics are centered around component technologies such as Java EJBs, Business-to-Business technologies like EDI and ebXML, and Cloud Computing facilities like Google App Engine and Windows Azure. Thus the course introduces and discusses in-depth topics concerning distributed middleware and its practical use:

- Characteristics and Foundations of Distributed Systems
- Classical Middleware and Services
- Concurrency and Synchronization
- Component Technologies
- Cloud Computing, in particular platform as a service
- Business-to-Business Technologies

The selection of topics and teaching method of this course reflects the Distributed Systems Group's (DSG) dedication to integrate business and IT, theory and practice, research and teaching. You not only will be taught the classical way, but you will have hands-on experience on middleware development and middleware tools. Also, you will get the chance to discuss selected publications with your lecturers.

Learning outcomes:

Students are able to evaluate, plan, design and implement server-centric distributed systems. Students are familiar with recent approaches and standards for building and managing such systems, know about the central problems involved as well as ways to overcome these issues. Students have hands-on experience with up-to-date middleware and tools for building server-centric systems.

Remark:

The main language of instruction in this course is English.

prerequisites for the module:

Basic knowledge in software engineering and in distributed systems as introduced, e.g., in the module DSG-IDistrSys-B (or DSG-DistrSys-M).

Recommended prior knowledge: Basic knowledge in software engineering and in distributed systems as introduced, e.g., in the module DSG-IDistrSys.		**Admission requirements:** none
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Lectures Distributed Systems Architecture and Middleware **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Guido Wirtz **Language:** English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**

Learning outcome: c.f. overall module description **Contents:** c.f. overall module description **Literature:** This is a fast emerging field with new insights every year. So, up-to-date literature will be provided at the beginning of each course.	
2. Practicals Distributed Systems Architecture and Middleware **Mode of Delivery:** Practicals **Lecturers:** Scientific Staff Praktische Informatik **Language:** English/German **Frequency:** every winter semester **Learning outcome:** c.f. overall module description **Contents:** Introduction to and discussion of tools and practical issues closely related to the topics discussed in the lecture as well as solutions of problems that come up during working on the practical assignment. **Literature:** c.f. overall module description	**2,00 Weekly Contact Hours**
Examination Coursework Assignment and Colloquium / Duration of Examination: 15 minutes Duration of Coursework: 3 months **Description:** Oral examination concerning the topics discussed in the lecture, exercises and assignment. **Students may choose English or German as the language for the oral examination.** Examinations will take place at the end of the winter term or at the begin of the summer term (students may choose one of them). Students are assumed to work on a programming assignment ('schriftliche Hausarbeit') during the semester that is introduced at the beginning of the semester and uses the most important technologies discussed during the semester. **Note:** Without working on the programming assignment over the term students may run into problems during their oral examination (Kolloquium) as we discuss questions concerning topics from the lectures as well as from the assignment; questions about the assignment are based on the assignment solution programmed by the students.	

Module DT-DB42-M Database Systems - The question to or the better answer than 42? *Datenbanksysteme - Die Frage zu oder die bessere Antwort auf 42?*		3 ECTS / 90 h
(since WS24/25) Person responsible for module: Prof. Dr. Maximilian Schüle		
Contents: In this seminar, we will study the challenges of modern database systems. We will discuss the topic along with very recent publications about database systems for machine learning and knowledge discovery		
Learning outcomes: Ability to write publications independently		
prerequisites for the module: none		
Recommended prior knowledge: none		**Admission requirements:** none
Frequency: winter and summer semester, on demand	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Database Systems - The question to or the better answer than 42? **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Maximilian Schüle **Language:** German/English **Frequency:** winter and summer semester, on demand	**2,00 Weekly Contact Hours**
Learning outcome: Ability to write publications independently	
Contents: In this seminar, we will study the challenges of modern database systems. We will discuss the topic along with very recent publications about database systems for machine learning and knowledge discovery	

Examination Internship report / Duration of Examination: 30 minutes Duration of Coursework: 14 days	

Module DT-DBCPU-M Database Systems for modern CPU *Datenbanksysteme für moderne CPU*		6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Maximilian Schüle		
Contents: This lecture covers the implementation of database systems, including how to leverage modern hardware architectures, for example vector intrinsics (AVX-512) and CUDA programming for GPU. Diese Vorlesung behandelt die Implementierung von Datenbanksystemen, einschließlich der Nutzung moderner Hardware-Architekturen, z.B. Vektorinstruktionen (AVX-512) und CUDA-Programmierung für die GPU.		
Learning outcomes: Understand the concepts of database systems and be able to implement database systems, also for modern hardware		
prerequisites for the module: none		
Recommended prior knowledge: MOBI-DBS-B		**Admission requirements:** none
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Database Systems for modern CPU **Mode of Delivery:** Lectures and Practicals **Lecturers:** Prof. Dr. Maximilian Schüle **Language:** English **Frequency:** every summer semester	**4,00 Weekly Contact Hours**
Learning outcome: Understand the concepts of database systems and be able to implement database systems, also for modern hardware	
Contents: This lecture covers the implementation of database systems, including how to leverage modern hardware architectures, for example vector intrinsics (AVX-512) and CUDA programming for GPU. Diese Vorlesung behandelt die Implementierung von Datenbanksystemen, einschließlich der Nutzung moderner Hardware-Architekturen, z.B. Vektorinstruktionen (AVX-512) und CUDA-Programmierung für die GPU.	
Literature: • Theo Härder, Erhard Rahm.Datenbanksysteme: Konzepte und Techniken der Implementierung.Springer, Berlin; 2nd ed. • Hector Garcia-Molina, Jeff Ullman, Jennifer Widom. *Database Systems: The Complete Book* • D. E. Knuth.The Art of Computer Programming Volume III	

• Joseph M. Hellerstein, Michael Stonebraker, James Hamilton.Architecture of a Database System • Franz Faerber, Alfons Kemper, Per-Åke Larson, Justin J. Levandoski, Thomas Neumann, Andrew Pavlo.Main Memory Database Systems	

Examination Written examination / Duration of Examination: 90 minutes	

Module DT-Proj-M Project: Data Engineering *Projekt: Data Engineering*		6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Maximilian Schüle		
Contents: Data Engineering projects		
Learning outcomes: In-depth scientific work as part of a data engineering project		
prerequisites for the module: none		
Recommended prior knowledge: none		**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Project: Data Engineering **Mode of Delivery:** **Lecturers:** Prof. Dr. Maximilian Schüle **Language:** English **Frequency:** every semester	**4,00 Weekly Contact Hours**
Learning outcome: In-depth scientific work as part of a data engineering project	
Contents: Data Engineering projects	

Examination Colloquium, Coursework Assignment / Duration of Examination: 30 minutes Duration of Coursework: 3 months	

Module EESYS-ADAML-M Applied Data Analytics and Machine Learning in R *Applied Data Analytics and Machine Learning in R*	6 ECTS / 180 h

(since SS21)
Person responsible for module: Prof. Dr. Thorsten Staake

Contents:
This course provides the theoretical foundation and conveys hands-on skills in the fields of data analytics and machine learning using the statistics software GNU R. It uses real-word datasets from the realm of energy efficiency and consumer behavior and conveys the subject matter through real-world examples and practical challenges.

Following a refresher in descriptive statistic, the course covers

- an introduction to the statistics software GNU R,
- the design of field experiments and the use of Information Systems to collect behavioral data,
- techniques to formulate, solve, and interpret linear and logistic regression analyses,
- techniques to formulate, solve, and interpret clustering analyses,
- setting up, training, and evaluating machine learning algorithms, including KNN, regression, and support vector machines, and
- ethical issues and data privacy regulations.

Learning outcomes:
After a successful participation in this course, participants can

- translate new business and research questions that can be answered using empirical methods into suitable experimental designs,
- plan and conduct corresponding experiments,
- choose suitable methods from the set of methods presented in class to analyze the data,
- explain their design choices, the choice of methods, and the steps of the analyses,
- apply the methods correctly and efficiently using the statics software R,
- adjust the methods if needed to solve new and specific problems based on an understanding of the necessary theories,
- interpret the outcome of such analyses and identify the strengths and limitations of the approaches, and
- reflect upon data protection, privacy and ethical issues related to powerful techniques for data acquisition and analytics.

Remark:
The lecture will be held as a self-paced, video-based online lecture.

The tutorials take place once per week as in-classroom events.

The online lecture includes instructional videos (scripted, i.e., with subtitles), reading material, exemplary data sets, and a multitude of online and offline tasks. It also includes an online discussion forum.

The online lecture is supported by three classroom lectures (in addition to the classroom tutorials):

1. Classroom lecture: The introductory event includes a course overview and motivation. Moreover, credentials to access the online resources will be announced. Date: First week of the semester.

2. Classroom lecture: This intermediate session includes a review of the concepts covered so far. It should help participants to self-assess their learning progress. Date: Announced in the first week of the semester.
3. Classroom lecture: Exam preparation and Q&A. Date: Last week of the semester.

An introduction to the statistics software GNU R will be given as in-classroom event during the tutorials at the beginning of the semester.

prerequisites for the module:
none

Recommended prior knowledge:	**Admission requirements:**
This course requires a basic understanding of statistics (e.g., from a bachelor-level course). A statistics repetition and is part of the online material of the course and the of the first tutorials and should be complemented in self-study if necessary. Basic familiarity with a programming language.	none

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units

1. Lectures Data Analytics in Energy Informatics **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Thorsten Staake **Language:** German/English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: The video-based online lecture is divided into two parts. Part 1 conveys the statistical basics required for the module, including, for example, properties of random distributions and descriptive and injunctive statistics. This part serves as refresher of bachelor-level statistics and thereby enables students with no statistics-knowledge beyond a basic introductory course to participate. Part 2 covers the methods outlined in “Module EESYS-DAE-M” subsection “Contents”. It includes both, the theory behind the concepts and their application using R. Both, Part 1 and Part 2 use datasets and examples from industry and research and provides many hands-on examples. In order to deepen the understanding and to ease the transfer of the methods to new problems and settings, mini-tasks and small exercises are part of the online lecture.	
Literature: Reading material will be announced in class.	

2. Practicals Data Analytics in Energy Informatics **Mode of Delivery:** Practicals **Language:** German/English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: In the classroom tutorial, participants apply the methods, tools, and theories conveyed in the lecture to exemplary problems and to new challenges. This includes solving smaller tasks (e.g., acing case studies, working on concrete	

data problems) on paper and using the statistics software GNU R. Tasks are addressed individually or in small teams. The tutorials can also cover new content, especially when its immediate application supports the learning process. Selected tutorials contain a self-assessment of the learning progress. An introduction to GNU R is given in the first sessions.	

Examination Written examination / Duration of Examination: 90 minutes **Description:** The examination covers subject matter taught in the lectures and tutorials. The examination can also cover transfers of the subject matter to new problems and settings. Students can achieve up to 90 points. Through the voluntary completion of coursework (“bonus exercises”) during the semester, participants can collect up to 12 additional points that are counted towards the exam, given that the exam is passed also without points from bonus exercises. Bonus exercises can take the form of written assignments, presentations, or smaller software projects. Points from bonus exercises are only valid in the semester they have been earned in and in the immediately following semester. In the first week of the course, the publishing dates of bonus exercise tasks, the submission deadlines, and the points per bonus exercise will be announced. It is possible to pass the exam with a grade of 1.0 also without points from bonus exercises. Exam questions are stated in English, answers can be given in German or English.	

Module EESYS-ES-M Energy Efficient Systems *Energieeffiziente Systeme*		6 ECTS / 180 h
(since WS19/20) Person responsible for module: Prof. Dr. Thorsten Staake		
Contents: The course covers the design and application of Information Systems that help increase energy efficiency and reduce greenhouse gas emissions. It is directed to computer science and Information Systems students that want to apply their skills to challenges in the fields of energy, mobility, production, and sustainable consumption/consumer behavior. The course introduces methods and theories from behavioral economics, operations management, and simulation analysis that help to understand, analyze, and shape both, industry processes and consumer behavior in the field of sustainability. Also covered are cost/benefit considerations on a micro- and macro-level (including, for example, rebound effects) and a discussion on the economic and societal implications of the subject matter. The course includes an introduction to physics and energy engineering to allow students with very limited knowledge in these fields to participate successfully.		
Learning outcomes: Successful participants of this course shall acquire the skills to • explain the physical and technical principals covered in this course and apply them to new problems, • explain the components, influencing factors, requirements and challenges related to electric mobility and describe the contribution that Information Systems can make to solve the challenges; moreover, successful participants shall be able to set up data-based simulations to derive important characteristic variables related to electric vehicles, such as electric reachability, peak loads to electric grids, etc., • outline, assess, and conceptually model the potential of Information Systems and the effects to heating and room climate applications, • explain in detail the characteristics of and implications from environmental business Information Systems, • explain the discussed behavioral theories (e.g., the prospect theory), make use of them when building Information Systems that support decision making and behavioral change, and be able to evaluate the effectiveness of such systems, and • evaluate the effects of the tools and methods introduced, including their micro- and macro-economic effects, and critically assess the techniques used to perform such evaluations. Moreover, successful participants shall be able to apply the acquired skills to new challenges and adjust and extend them as needed. Finally, the participants shall realize the scope for design and the potential that results from their IT studies to favorably shape a sustainable and socially desirable development of our society.		
prerequisites for the module: none		
Recommended prior knowledge: none		**Admission requirements:** none
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Lectures Energy Efficient Systems **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Thorsten Staake **Language:** German/English **Frequency:** every summer semester **Contents:** The lecture covers the topics mentioned in “Module EESYS-ES-M”, subsection “Contents”. It uses traditional lecture elements, discussions, exercises, and group work to support participants in reaching the learning objectives. Special emphasis is placed on working on cases and on discussions of studies and scientific publications. Methods, tools, and theories are introduced with references to practical challenges and are applied to exemplary problems. For selected topics, the lecture relies on flipped classroom elements for which participants need to acquire knowledge in advance (e.g., through reading tasks), which is then critically reflected and extended in the classroom sessions. **Literature:** Weiterführende Unterlagen werden in der Veranstaltung bekanntgegeben.	**2,00 Weekly Contact Hours**
2. Practicals Energy Efficient Systems **Mode of Delivery:** Practicals **Language:** German/English **Frequency:** every summer semester **Contents:** The first tutorials convey basics in physics and electrical engineering in order to also allow students who did not take related modules to participate in this course. Subsequently, participants apply the methods, tools, and theories conveyed in the lecture to exemplary problems and to new challenges. Tutorials include small tasks, case studies, and reviews of scientific publications that are addressed individually or in small teams. The tutorials can also cover new content, especially when its immediate application supports the learning process. Selected tutorials contain a self-assessment of the learning progress. **Literature:** Reading material will be announced in class.	**2,00 Weekly Contact Hours**

Examination Written examination / Duration of Examination: 90 minutes **Description:** The examination covers subject matter taught in the lectures and tutorials. The examination can also cover transfers of the subject matter to new problems and settings. Students can achieve up to 90 points. Through the voluntary completion of coursework (“bonus exercises”) during the semester, participants can collect up to 12 additional points that are counted	

towards the exam, given that the exam is passed also without points from bonus exercises. Bonus exercises can take the form of written assignments, presentations, or smaller software projects. Points from bonus exercises are only valid in the semester they have been earned in and in the immediately following semester. In the first week of the course, the publishing dates of bonus exercise tasks, the submission deadlines, and the points per bonus exercise will be announced. It is possible to pass the exam with a grade of 1.0 also without points from bonus exercises. Exam questions are stated in English, answers can be given in German or English.	

Module GdI-FPRS-M Functional Programming of Reactive Systems *Functional Programming of Reactive Systems*	6 ECTS / 180 h
(since WS23/24) Person responsible for module: Prof. Ph.D. Michael Mendler	

Contents:

Based on an existing basic knowledge of functional programming (FP), the aim of this module is to develop advanced skills in the use of FP languages to structure and solve algorithmic problems in designing interactive and concurrent systems. We will study advanced programming abstractions specifically developed for the functional modelling of synchronous reactive systems. Following the methodological structure of the introductory course GDI-IFP, this advanced course, too, combines both practical programming with a focused discussion of pertinent underlying mathematical concepts. Though we use Haskell as our main language we may also look at other FP languages such as F#, ML or OCAML where appropriate.

Learning outcomes:

At the end of this course students should

- be familiar with advanced FP programming concepts and their application (e.g., class mechanism, type families, higher-rank polymorphism, monad and arrow abstractions, lenses, continuation-style programming, stream programming, concurrency abstractions)
- be able to use these advanced language concepts to solve complex algorithmic problems efficiently, in particular involving the use of memory, concurrency and interaction
- be able use the Haskell stack build tool and understand the mechanisms of package management
- appreciate the importance of functional abstraction for conciseness and efficiency of programming complex applications
- be familiar with the second-order polymorphic lambda calculus (Hindley-Milner predicative let-polymorphism, impredicative System F) as an operational semantics behind (eager, lazy) functional programming
- be able to explain the encoding of recursive data structures in type theory
- have an elementary understanding of the execution model of functional languages and transformation to operational code through defunctionalisation and abstract machines.
- by able to use FP (specifically Haskell) as a development tool for the design of new programming languages

Remark:

The workload for this module splits up roughly like this:

- participation in lectures and tutorials: 45 hrs
- preparation of classes and tutorials as well literature research: 60 hrs
- solving (ungraded) programming exercises and participation in lab sessions: 45 hrs
- exam preparation: 30 hrs

prerequisites for the module:

none

Recommended prior knowledge:	**Admission requirements:**
Elementary programming skills in a functional programming language, such as from module GdI-IFP-B; Basic knowledge in the use of	none

temporal and modal logic specification formalisms such as from GdI-MTL-B. English language skills at Level B2 (UniCert II) or above. Module Introduction to Functional Programming (GdI-IFP-M) - recommended		
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units

1. Advanced Functional Programming — **2,00 Weekly Contact Hours**

Mode of Delivery: Lectures
Lecturers: Prof. Ph.D. Michael Mendler
Language: English/German
Frequency: every summer semester

Contents:
Through class presentations and direct interactions with the students the lecturer introduces the topics of the course in detail, poses exercises and suggests literature for self-study.

Literature:

- S. Marlow: The Haskell 2010 Language Report. https://www.haskell.org/onlinereport/haskell2010/
- V. Zsók, Z. Horváth, R. Plasmeijer: Central European Functional Programming School. Springer 2012.
- S. Marlow: Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming, O'Reilly 2013.
- B. O'Sullivan, J. Goerzen, D. Stewart: Real World Haskell. O'Reilly 2009.
- Ch. Okasaki: Purely Functional Data Structures, CUP 1998
- F. Rabhi, G. Lapalme: Algorithms - A Functional Approach.
- D. Syme, A. Granicz, A. Cisternino: Expert F#4.0, Apress 2015.
- B. Pierce: Types and Programming Languages. MIT Press 2002. (esp. Chapters 23+25)
- H. Barendregt, W. Dekkers, R. Statman: Lambda Calculus with Types. CUP 2013.

2. Functional Programming of Reactive Systems — **2,00 Weekly Contact Hours**

Mode of Delivery: Practicals
Lecturers: Prof. Ph.D. Michael Mendler
Language: English/German
Frequency: every summer semester

Contents:
The tutorials deepen the students' understanding of the theoretical concepts and constructions covered in the lectures through practical exercises. Participants are given the opportunity to discuss their solutions to homework question sheets and sample solutions are presented by the tutors or lecturer for selected exercises. The tutorials also provide exam preparation.

Literature:

The literature will be announced in class. Here are some general pointers on FP languages and synchronous programming. • S. Marlow: The Haskell 2010 Language Report. https://www.haskell.org/onlinereport/haskell2010/ • V. Zsók, Z. Horváth, R. Plasmeijer: Central European Functional Programming School. Springer 2012. • S. Marlow: Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming, O'Reilly 2013. • D. Syme, A. Granicz, A. Cisternino: Expert F#4.0, Apress 2015. • H. Barendregt, W. Dekkers, R. Statman: Lambda Calculus with Types. CUP 2013. • Benveniste, A. et al: The Synchronous Languages 12 years later. Proc. IEEE, Vol 91(1), January 2003. • Berry, G.: SCADE: Synchronous design and validation of embedded control software. In: Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems. Proc. GM R&D Workshop, Bangalore, January 2007. pp. 19-33. • Potop-Butucaru et. al: The Synchronous Hypothesis and Synchronous Languages. In Richard Zurawski. *Embedded Systems Design and Verification*, CRC Press, pp.6-1-6-27, 2009.	
Examination Written examination / Duration of Examination: 90 minutes **Description:** The examination language is English. The form of examination is either oral (30 minutes) or written (90 minutes) depending on the number of participants. The form of examination will be determined at the beginning of the semester and announced in class.	

Examination Oral examination / Duration of Examination: 30 minutes **Description:** The examination language is English. The form of examination is either oral (30 minutes) or written (90 minutes) depending on the number of participants. The form of examination will be determined at the beginning of the semester and announced in class.	

Module GdI-IFP-M Introduction to Functional Programming *Introduction to Functional Programming*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Ph.D. Michael Mendler	
Contents: The aim of this module is to provide an introduction to functional programming using Haskell. This course develops both elementary practical programming skills and discusses the typed lambda calculus and its role as an operational semantics for functional programming, stressing the importance of types and type checking for static program analysis.	
Learning outcomes: At the end of this course students should be familiar with important language constructs of Haskell and their semantics (e.g., expressions, local declarations, higher-order function abstraction, recursion, lazy and eager evaluation, referential transparency, algebraic data types, monads); be able to use these language concepts to solve algorithmic problems; be familiar with the lambda calculus as an operational semantics behind functional programming; understand the difference between imperative and declarative programming styles; have an appreciation of the close relationship between programming language types and specification and the role of type checking as a static program analysis method; be familiar with polymorphic Hindley-Milner style type systems.	
Remark: The main language of instruction in this course is English. However, the lectures and/or tutorials may be delivered in German if all participating students are fluent in German.	
prerequisites for the module: none	
Recommended prior knowledge: Elementary concepts in logic and discrete mathematics for computer scientists; Basic programming skills; English language skills at Level B2 (UniCert II) or above.	**Admission requirements:** none

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Introduction to Functional Programming **Mode of Delivery:** Lectures **Lecturers:** Prof. Ph.D. Michael Mendler **Language:** English/German **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: Through prepared class presentations and direct interactions with the students the lecturer introduces the topics of the course in detail, poses exercises and suggests literature for self-study.	
Literature: • Pierce, B. C.: Types and Programming Languages, MIT Press, 2002	

• Thompson, S.: Haskell – The Craft of Functional Programming, Addison-Wesley 1999.	
2. Introduction to Functional Programming **Mode of Delivery:** Practicals **Lecturers:** Prof. Ph.D. Michael Mendler **Language:** English/German **Frequency:** every winter semester **Contents:** The tutorials deepen the students' understanding of the theoretical concepts and constructions covered in the lectures through practical exercises. Participants are given the opportunity to discuss their solutions to homework question sheets and sample solutions are presented by the tutors or lecturer for selected exercises. The tutorials also provide exam preparation.	**2,00 Weekly Contact Hours**
Examination Written examination / Duration of Examination: 90 minutes **Description:** 90 min written examination. The exam takes place during the regular exam period after the end of the semester.	

Module GdI-Proj-M Master's Project Theoretical Foundations of Computing *Masterprojekt Grundlagen der Informatik*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Ph.D. Michael Mendler	
Contents: The project will be conducted either individually or in small student teams depending on the topic which will fall into one of the current active research areas of the informatics theory group (GDI). The results of the project are documented in written form in a work report and orally presented in a research talk. The project typically consist of theoretical research based on the literature and some software implementation.	
Learning outcomes: By conducting supervised research the project implementation work, the students will be able to gain an understanding of further central issues in the theory of computing, beyond the contents covered in regular modules. They will also be able to deepen their knowledge of the practical application of theoretical concepts discussed in theory modules they have previously attended and develop important research skills.	
prerequisites for the module: none	
Recommended prior knowledge: Students are expected to possess general skills and knowledge in the planning, organisation and execution of software projects, such as acquired in a previous software engineering lab module. Typically, students have previously also attended courses on research methods. In addition, for projects in the theoretical foundations of computer science we strongly recommend: a good command of English, elementary formal logic, basic knowledge in the theory of machines and languages, computer architecture, operating systems, non-procedural programming. Module Functional Programming of Reactive Systems (GdI-FPRS-M) - recommended Module Introduction to Functional Programming (GdI-IFP-M) - recommended	**Admission requirements:** none

Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Master's Project Theoretical Foundations of Computing **Mode of Delivery:** **Lecturers:** Prof. Ph.D. Michael Mendler **Language:** English/German **Frequency:** every semester	**4,00 Weekly Contact Hours**
Learning outcome: To be announced at the beginning of the semester.	
Contents:	

Project planning meetings, tutorials on the project topics, final presentation and poster **Literature:** Relevant literature will be announced at the beginning of the semester.	
Examination Coursework Assignment and Colloquium / Duration of Examination: 20 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regelmäßige Teilnahme an der Lehrveranstaltung **Description:** Preparation of the final written project report and poster presentation with colloquium.	

Module HCI-MCI-M Human-Computer Interaction *Mensch-Computer-Interaktion*	6 ECTS / 180 h
(since WS21/22) Person responsible for module: Prof. Dr. Tom Gross	
Contents: Advanced theoretical, methodological, and practical foundation of Human-Computer Interaction	
Learning outcomes: The aim of this module is to teach advanced knowledge and skills in the area of human-computer interaction as well as a broad theoretical and practical methodological expertise concerned with the design, conception, and evaluation of ubiquitous systems. Students of this course learn the relevant literature and systems in breadth and depth and are later able to critical review new literature and systems.	
Remark: http://www.uni-bamberg.de/hci/leistungen/studium The workload for this module is roughly structured as following: • Attendance of the lectures and assignments: 45 hours • Credits of the lecture (incl. research and study of additional sources): ca. 30 hours • Credits of the assignments (incl. research and study of additional sources, but without optional homework assignment): ca. 30 hours • Solving the optional homework assignments: overall ca. 45 hours • Exam preparation: ca. 30 hours (based on the above mentioned preparation and revision of the subject material) The default language of instruction in this course is German, but can be changed to English on demand. All course materials (incl. exams) are available in English.	
prerequisites for the module: none	
Recommended prior knowledge: Module Algorithms and data structures (MI-AuD-B) Module Introduction to Algorithms, Programming and Software (DSG-EiAPS-B)	**Admission requirements:** Passing the written exam

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Human - Computer Interaction **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Tom Gross **Language:** German/English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: After an introduction into the subject the following topics are covered in this lecture: • Mobile human-computer interaction	

- Adaptivity and adaptibility
- Information visualisation
- Tangible user interaction
- Usability engineering
- Usability and economics

Literature:
The course is based on a compilation of different sources; as additional sources and as a reference are recommended:

- Jacko, J.A. and Sears, A., (Eds.). Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. Lawrence Erlbaum, Hillsdale, NJ, 2002.
- Hammond, J., Gross, T. and Wesson, J., (Eds.). Usability: Gaining a Competitive Edge. Kluwer Academic Publishers, Dordrecht, 2002.

Examination
Oral examination
Description:
The oral exam takes 30 minutes and is worth a total of 90 points. Depending on the number of attendees the form of the exam can be changed to a written exam with 90 minutes and a total of 90 points. The final form of the exam is announced in the first lecture at the beginning of the term.

During the semester students can do assignments, which are optional. They are 12 points in total. The type of optional homework assignments as well as the deadlines are announced in detail at the beginning of the term. If the oral exam is passed (as a rule 50% of the points have to be reached) the points from the assignments are a bonus and added to the points from the oral exam. In any case, a top grade of 1,0 is also reachable without solving the assignments.

Module Units

Human-Computer Interaction — **2,00 Weekly Contact Hours**
Mode of Delivery: Practicals
Lecturers: Scientific Staff Mensch-Computer-Interaktion
Language: German/English
Frequency: every winter semester

Contents:
Practical assignments based on the subjects of the lecture.

Literature:
Cf. lecture

Examination
Written examination / Duration of Examination: 90 minutes
Description:
In Abhängigkeit der Teilnehmerzahl wird die Modulprüfung entweder in Form einer Klausur oder in Form einer mündlichen Prüfung durchgeführt.

Die Festlegung erfolgt zu Semesterbeginn und wird im ersten Lehrveranstaltungstermin bekannt gegeben. In der Klausur über 90 Min. können 90 Punkte erzielt werden. Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die Prüfung bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der Prüfung auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.	

Module HCI-Prop-M Propaedeutic: Human-Computer-Interaction *Propädeutikum Mensch-Computer-Interaktion*	3 ECTS / 90 h
(since WS24/25) Person responsible for module: Prof. Dr. Tom Gross	
Contents: Scientific foundation of the research field of Human-Computer Interaction	
Learning outcomes: The aim of this module is a general introduction to and teaching of fundamental paradigms and scientific methods of the organisation, the written documentation, oral presentation of research activities in Human-Computer Interaction. The primary focus is on domain-specific documentation and presentation of designs, prototypes, and user studies.	
Remark: http://www.uni-bamberg.de/hci/leistungen/studium The workload for this module is roughly structured as following: • Participation in the course meetings (theoretical foundation; practical case studies): ca. 30 hours • Working on the case studies: ca. 30 hours • Preparation of presentation: ca. 15 hours • Writing of term paper: ca. 15 hours The default language of instruction in this course is German, but can be changed to English on demand. All course materials (incl. exams) are available in English.	
prerequisites for the module: none	

Recommended prior knowledge: none		**Admission requirements:** none
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Propaedeutic: Human-Computer-Interaction **Mode of Delivery:** **Lecturers:** Prof. Dr. Tom Gross, Scientific Staff Mensch-Computer-Interaktion **Language:** German/English **Frequency:** every winter semester	**3,00 Weekly Contact Hours**
Contents: This seminar is concerned with the documentation and presentation of current concepts, technologies, and tools and user studies of human-computer interaction.	
Literature: The course is based on a compilation of different sources; as additional sources and as a reference are recommended:	

Jacko, Julie A., ed. Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. (3rd ed.). Lawrence Erlbaum, Hillsdale, NJ, 2012.	
Examination Internship report / Duration of Examination: 30 minutes Duration of Coursework: 4 months **Description:** Written term paper and presentation on the chosen topic by the participant, incl. discussion	

Module HCI-Sem-HCC-M Master-Seminar Human-Centred Computing *Masterseminar Human-Centred Computing*	3 ECTS / 90 h
(since WS24/25) Person responsible for module: Prof. Dr. Tom Gross	
Contents: Advanced active scientific work on own current concepts, technologies and tools of Human-Computer Interaction	
Learning outcomes: The aim of this course is the acquisition of abilities that allow the independent research and presentation of topics in the field of human-computer interaction on basis of the existing literature. The focus lies on the development of skills that allow to critically and systematically review literature in order to develop and present an own perspective.	
Remark: http://www.uni-bamberg.de/hci/leistungen/studium The workload for this module is roughly structured as following: • Participation in the seminars (introduction to the topics, discussions, presentations): ca. 20 hours • Literature review and getting familiar with the topic: ca. 25 hours • Preparation of presentation: ca. 15 hours • Writing of term paper: ca. 30 hours The default language of instruction is German and can be changed to English based on students' needs. All course materials (incl. exams) are available in English	
prerequisites for the module: none	
Recommended prior knowledge: Module Human-Computer Interaction (HCI-MCI-M)	**Admission requirements:** Passing the exam

Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Human-Centred Computing **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Tom Gross, Scientific Staff Mensch-Computer-Interaktion **Language:** German/English **Frequency:** every summer semester	**2,00 Weekly Contact Hours**
Contents: This seminar is concerned with novel research methods in the fields of human-computer interaction, computer-supported cooperative work, and ubiquitous computing.	
Literature: To be announced at the beginning of the course	
Examination	

Internship report / Duration of Examination: 30 minutes Duration of Coursework: 4 months **Description:** Written term paper and presentation on the chosen topic by the participant, incl. discussion	

Module HCI-Sem-M Master-Seminar Human-Computer Interaction *Masterseminar Mensch-Computer-Interaktion*	3 ECTS / 90 h
(since WS24/25) Person responsible for module: Prof. Dr. Tom Gross	
Contents: Advanced active scientific work on current concepts, technologies and tools of Human-Computer Interaction	
Learning outcomes: The aim of this course is the acquisition of abilities that allow the independent research and presentation of topics in the field of human-computer interaction on basis of the existing literature. The focus lies on the development of skills that allow to critically and systematically review literature in order to develop and present an own perspective.	
Remark: http://www.uni-bamberg.de/hci/leistungen/studium The workload for this module is roughly structured as following: • Participation in the seminars (introduction to the topics, discussions, presentations): ca. 20 hours • Literature review and getting familiar with the topic: ca. 25 hours • Preparation of presentation: ca. 15 hours • Writing of term paper: ca. 30 hours The default language of instruction in this course is German, but can be changed to English on demand. All course materials (incl. exams) are available in English.	
prerequisites for the module: none	
Recommended prior knowledge: Module Human-Computer Interaction (HCI-MCI-M)	**Admission requirements:** Passing the exam

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Human-Computer Interaction **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Tom Gross, Scientific Staff Mensch-Computer-Interaktion **Language:** German/English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: This seminar is concerned with topics on current concepts, technologies, and tools of human-computer interaction.	
Literature: To be announced at the beginning of the course	
Examination Internship report / Duration of Examination: 30 minutes	

Duration of Coursework: 4 months **Description:** Written term paper and presentation on the chosen topic by the participant, incl. discussion	

Module HCI-US-B Ubiquitous Systems *Ubiquitäre Systeme*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Tom Gross	
Contents: Theoretical, methodological, and practical foundation of Ubiquitous Computing	
Learning outcomes: The aim of this module is to teach advanced knowledge and skills in the aerea of ubiquitous systems as well as abroad theoretical and practical methodological expertise concerned with the design, conception and evaluation of ubiquitous systems. Students of this course learn the relevant literature and systems in breadth and depth and should be able to critical review new litarature and systems	
Remark: htp://www.uni-bamberg.de/hci/leistungen/studium The workload for this module is roughly structured as following: • Attendance of the lectures and assignments: 45 hours • Credits of the lecture (incl.research and study of additional sources): ca. 30 Hours • Credits of the assignments ((incl.research and study of additional sources, excluding optional homework assignment): ca. 30 hours • Solving the optional homework assignments: overall ca. 45 hours • Exam preparation: ca. 30 hours (based on the above mentioned preparation and revision of the subject material) The default language of instruction in this course is German, but can be changed to English on demand. All course materials (incl. exams) are available in English.	
prerequisites for the module: none	
Recommended prior knowledge: Module Algorithms and data structures (MI-AuD-B) Module Introduction to Algorithms, Programming and Software (DSG-EiAPS-B)	**Admission requirements:** Passing the written exam

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Ubiquitous Systems **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Tom Gross **Language:** German/English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: This lecture gives an introduction to the subject of Ubiquitous Computing—that is, the paradigm of invisible computing, with computers embedded into everyday objects that act as client and server and communicate with each other—and includes the following conceptual, technical and methodological topics:	

- Basic concepts
- Base technology and infrastructures
- Ubiquitous systems and prototypes
- Context awareness
- User interaction
- Ubiquitous systems in a broad context and related topics

Literature:
The course is based on a compilation of different sources; as additional sources and as a reference are recommended:

- Krumm, J. (Ed.). Ubiquitous Computing Fundamentals. Taylor & Francis Group, Boca Raton, FL, 2010.

Examination
Oral examination
Description:
The oral exam takes 30 minutes and is worth a total of 90 points. Depending on the number of attendees the form of the exam can be changed to a written exam with 90 minutes and a total of 90 points. The final form of the exam is announced in the first lecture at the beginning of the term.

During the semester students can do assignments, which are optional. They are 12 points in total. The type of optional homework assignments as well as the deadlines are announced in detail at the beginning of the term. If the oral exam is passed (as a rule 50% of the points have to be reached) the points from the assignments are a bonus and added to the points from the oral exam. In any case, a top grade of 1,0 is also reachable without solving the assignments.

Module Units

Ubiquitous Systems — **2,00 Weekly Contact Hours**
Mode of Delivery: Practicals
Lecturers: Scientific Staff Mensch-Computer-Interaktion
Language: German/English
Frequency: every winter semester

Contents:
Practical assignments based on the subjects of the lecture including the programming of small prototypes

Literature:
Cf. lecture

Examination
Written examination / Duration of Examination: 90 minutes
Description:
In Abhängigkeit der Teilnehmerzahl wird die Modulprüfung entweder in Form einer Klausur oder in Form einer mündlichen Prüfung durchgeführt.

Die Festlegung erfolgt zu Semesterbeginn und wird im ersten Lehrveranstaltungstermin bekannt gegeben. In der Klausur über 90 min. können 90 Punkte erzielt werden. Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die Prüfung bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der Prüfung auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.	

Module ISPL-MDP-M Managing Digital Platforms *Managing Digital Platforms*	6 ECTS / 180 h

(since WS23/24)
Person responsible for module: Prof. Dr. Thomas Kude

Contents:
Digital platforms are ubiquitous in industries and in society and both researchers and practitioners have recognized their disruptive potential. Large technology companies, such as Apple, Alibaba, Amazon, or SAP, rely on a platform business model and the emergence of the thriving platform economy has contributed to the meteoric rise of some platform owners to top the lists of the most valuable companies in the world. The central actors in the context of digital platforms include the platform owner that provides the platform itself along with interfaces and other resources, outside third-party actors that provide complementary products and services, as well as the users of the platform. For example, in the context of mobile app ecosystems, complementors can leverage platform functionality of iOS or Android to create apps and use Apple's App Store or the Google Play Store to offer them to iPhone or Android users.

In this course, we develop a comprehensive understanding of the management of digital platforms through an in-depth exploration of the roles and mechanisms of digital platforms and the surrounding ecosystems. After laying the foundations of digital platform management, we will dive into advanced questions of platform design and management, e.g., related to platform launch, to governing third-party contributions, or to key success factors for the various actors in digital platform ecosystems. The course relies on both theoretical insights and practical cases across industries and companies.

Learning outcomes:
After the course, participants will be able to...

- Recognize the growing importance of digital platforms
- Analyze the underlying mechanisms and the roles of different actors in digital platform ecosystems
- Make decisions regarding the governance of different types of platforms
- Develop strategies and business models for complementor organizations that benefit from and depend on digital platforms

Remark:
The required workload of 180h is approximately subdivided into:

- 56h for participation in lecture and exercise
- 124h for preparation and post-processing of sessions as well as exam preparation

prerequisites for the module:
none

Recommended prior knowledge: Good command of the English language		**Admission requirements:** none
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Managing Digital Platforms **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Thomas Kude **Language:** English	**2,00 Weekly Contact Hours**

Frequency: every summer semester **Contents:** In the lecture, we will work on central topics of managing platform ecosystems, including, but not limited to: • Foundations of digital platforms • Launching and monetizing digital platforms • Digital platform governance • The role of complementors in digital platforms **Literature:** The specific literature that we will use in the course will be communicated or distributed in class or through the learning platform (VC). Students may have to purchase cases.	
2. Managing Digital Platforms **Mode of Delivery:** Practicals **Lecturers:** Prof. Dr. Thomas Kude **Language:** English **Frequency:** every summer semester **Contents:** In the exercise, we will deepen and practice the content of the lecture through examples and case discussions, some of which will be done in groups. **Literature:** See lecture	**2,00 Weekly Contact Hours**

Examination Written examination / Duration of Examination: 90 minutes **Description:** The exam questions will include the content from lecture, exercises, and assignments. Students can reach 90 points in the exam. Students may obtain additional points to improve their grade though the voluntary participation in group or individual assignments These points can be included in the exam points if a student would pass the exam without the additional points. The respective assignments, the available time, and the points that can be reached in each assignment will be communicated if and once such voluntary assignments are offered. The best grade (1,0) can be reached without participating in the voluntary assignments.	

Module KTR-Mobi-M Mobile Communication *Mobilkommunikation*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(since SS20) Person responsible for module: Prof. Dr. Udo Krieger	

Contents:

The course presents the fundamentals of mobile communication. We sketch the underlying standards, system architectures and their realizations as well as current research and development trends. Due to the complexity of the field the course can only present some basic important aspects of those mobile communication systems that exhibit the strongest growth in the markets and affect all busieness areas of the information societies at most. The course will focus on the technical system and design perspectives regarding the service architectures and local or wide area mobile communication networks.

The following topics are discussed in detail:
• technical foundation of wireless transmission
• media access control protocols
• resource management protocols in mobile communication networks (including resource assignment strategies at the radio layer, handoff management, error control protocols, scheduling etc.)
• mobility support at the network layer by mobile IP
• transport protocols and their enhancements
• wirelss LANs and their development (IEEE802.11 standards, WiMAX etc.)
• wireless wide area networks based on TDMA technology (GSM basics and protocols, GPRS)
• data communication in wireless wide area networks (UMTS, HSPA, LTE, LTE-A etc.)
• service architectures for mobile networks (including Android programming and WebRTC architectures)

Learning outcomes:

The students are encouraged to independent scientific work. They learn the fundamentals of mobile communication and are trained to analyze the applied protocols and communication algorithms in a systematic manner. Students are instructed to investigate the sketched mobile communication protocols by measurements using Wireshark and other tools, to evaluate their performance, and to develop new protocol elements. The processing of design, programming, and performance assessment tasks by teams of students and the effective arrangement of workgroups is part of the training.

Remark:

The module can be selected by exchange students and master students speaking only English.

prerequisites for the module:

none

Recommended prior knowledge:	**Admission requirements:**
• substantial knowledge of the foundations of data communication similar to module KTR-Datkomm-B • good knowledge of programming in JAVA (or C++) • knowledge of algorithms and data structures similar to module MI-AuD-B Module Algorithms and Data Structures (AI-AuD-B) - recommended Module Advanced Java Programming (DSG-AJP-B) - recommended	governed by examination regulations (StuFPO)

Module Data communication (KTR-Datkomm-B) - recommended		
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units

Mobile Communication Course — **4,00 Weekly Contact Hours**

Mode of Delivery: Lectures and Practicals
Lecturers: Prof. Dr. Udo Krieger
Language: English/German
Frequency: every winter semester

Learning outcome:

The students are encouraged to independent scientific work. They learn the fundamentals of mobile communication and are trained to analyze the applied protocols and communication algorithms in a systematic manner. Students are instructed to investigate the sketched mobile communication protocols by measurements using Wireshark and other tools, to evaluate their performance, and to develop new protocol elements. The processing of design, programming, and performance assessment tasks by teams of students and the effective arrangement of workgroups is part of the training.

Contents:

The course presents the fundamentals of mobile communication. We sketch the underlying standards, system architectures and their realizations as well as current research and development trends. Due to the complexity of the field the course can only present some basic important aspects of those mobile communication systems that exhibit the strongest growth in the markets and affect all busieness areas of the information societies at most. The course will focus on the technical system and design perspectives regarding the service architectures and local or wide area mobile communication networks.

The following topics are discussed in detail:
• technical foundation of wireless transmission
• media access control protocols
• resource management protocols in mobile communication networks (including resource assignment strategies at the radio layer, handoff management, error control protocols, scheduling etc.)
• mobility support at the network layer by mobile IP
• transport protocols and their enhancements
• wirelss LANs and their development (IEEE802.11 standards, WiMAX etc.)
• wireless wide area networks based on TDMA technology (GSM basics and protocols, GPRS)
• data communication in wireless wide area networks (UMTS, HSPA, LTE, LTE-A etc.)
• service architectures for mobile networks (including Android programming and WebRTC architectures)

The content of the lectures is illustrated by exercises and laboratories covering important aspects of the protocol stacks in mobile networks. The independent

processing of tasks, the qualified presentation and critical discussion of the outcomes by teams of students is part of the course. It improves the technical understanding and provides means to work as project leader in industry on those topics.

The course can be supplemented by the module Foundations of Internet Communcation (KTR-GIK-M) with its instructive tasks executed in the router laboratory, by master seminars and projects or a master's thesis on related topics in next generation networks.

The language of the course wil be announced during the first lecture.

Literature:

- Schiller, J.: Mobile Communications. Pearson-Education, Munich, 2004.
- Walke, B.: Mobile Radio Networks, Wiley, 2002.
- Pahlavan, K., Krishnamurthy, P.: Principles of Wireless Networks, A Unified Approach. Prentice Hall, 2002.
- Pahlavan, K., Krishnamurthy, P.: Networking Fundamentals: Wide, Local and Personal Area Communications, Wiley, 2009.
- Holma, H., Toskala, A.: LTE for UMTS, Evolution to LTE-Advanced, 2. ed, Wiley, 2011.

Examination

Oral examination / Duration of Examination: 30 minutes

Description:

30 minutes oral examination covering all topics of the lectures and practicals.

The language of the examination wil be announced during the first lecture.

Module MOBI-ADM-M Advanced Data Management *Advanced Data Management*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(since SS21) Person responsible for module: Prof. Dr. Daniela Nicklas	
Contents: With the rapid growth of the internet and more and more observable processes, many data sets became so large that they cannot be processed with traditional database methods any more. This modul covers advanced data management and integration techniques (also known under the term “big data”) that are useful when dealing with very large data sets.	
Learning outcomes: The students will understand the challenges of big data, and will be able to apply some of the new techniques to deal with it.	
Remark: The main language of instruction in this course is English. However, the lectures and/or tutorials may be delivered in German if all participating students are fluent in German. The written reports/seminar essay and the presentation may be delivered in English or in German.	
prerequisites for the module: none	
Recommended prior knowledge: Foundations of relational databases, relational algebra and SQL; e.g. from Modul SEDA-DMS-B: Data management systems	**Admission requirements:** none

Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Lectures Advanced Data Management **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Daniela Nicklas **Language:** English **Frequency:** every summer semester	**2,00 Weekly Contact Hours**
Contents: The lecture will cover various algorithms for clustering, association rule mining, or page ranking and their scalable processing using map and reduce methods, data integration, data cleansing and entity recognition. The exercises will be built upon the Hadoop framework. The language of the course will be announced in the first lecture.	
Literature: L. Wiese, Advanced Data Management, For SQL, NoSQL, Cloud and Distributed Databases. Berlin, Boston: De Gruyter, 2015	
2. Practicals Advanced Data Management **Mode of Delivery:** Practicals **Lecturers:** Prof. Dr. Daniela Nicklas	**2,00 Weekly Contact Hours**

Language: English
Frequency: every summer semester

Contents:
see Lectures

The language of the course will be announced in the first lecture.

Examination
Written examination / Duration of Examination: 75 minutes
Description:
Central written exam. The examination language is English.

The exam questions will be in English. The questions can be answered in English or German. The content that is relevant for the exam consists of the content presented in the lecture and in the practical assignments.

The exam consists of 7 tasks of which only 6 will be graded. The exam time includes a reading time of 15 minutes to select the tasks to be completed within the scope of the choices.

Participants who submit solutions for practical assignments can achieve bonus points. Details regarding the number of assignments, the number of bonus points per assignment, the conversion factor from bonus points to exam points (e.g., 10:1) and the type of assignments will be announced in the first practical assignment session.

If the points achieved in the exam are sufficient to pass the exam on its own (generally, this is the case when at least 50% of the points have been obtained), the converted bonus points will be added to the points achieved in the exam.

The grade 1.0 can be achieved without the bonus points.

Module MOBI-DSC-M Data Streams and Complex Event Processing *Data Streams and Complex Event Processing*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(since WS20/21) Person responsible for module: Prof. Dr. Daniela Nicklas	
Contents: The management of data streams and foundations of event processing: Applications, systems, query languages, continuous query processing, and security in distributed data stream management systems. The modul covers the following topics: Architectures of data stream management systems; Query languages; Data stream processing; Complex event processing; Security in data stream management systems; Application of data stream management systems	
Learning outcomes: Understand the challenges of data stream management and complex event processing Recognize and link basic building blocks of data stream management tasks in different frameworks and systems Develop and program queries on data streams and event streams in different query languages to process data streams and detect event patterns Understand basic implementation techniques for data stream operators Understand the main security challenges and solutions in data stream management systems	
prerequisites for the module: none	
Recommended prior knowledge: Foundations of relational databases, relational algebra and SQL; e.g. from Modul MOBI-DBS-B: Database Systems	**Admission requirements:** none

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Data Streams and Complex Event Processing **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Daniela Nicklas **Language:** English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Learning outcome: Understand the challenges of data stream management and complex event processing Recognize and link basic building blocks of data stream management tasks in different frameworks and systems Develop and program queries on data streams and event streams in different query languages to process data streams and detect event patterns Understand basic implementation techniques for data stream operators	

Understand the main security challenges and solutions in data stream management systems **Contents:** The management of data streams and foundations of event processing: Applications, systems, query languages, continuous query processing, and security in distributed data stream management systems. The modul covers the following topics: Architectures of data stream management systems; Query languages; Data stream processing; Complex event processing; Security in data stream management systems; Application of data stream management systems	
Examination Oral examination / Duration of Examination: 15 minutes **Description:** oral or written exam (will be announced in class at the beginning of the semester). The examination language is English.	

Module Units	
Data Streams and Complex Event Processing **Mode of Delivery:** Practicals **Language:** English **Frequency:** every winter semester **Contents:** see lecture	**2,00 Weekly Contact Hours**

Examination Written examination / Duration of Examination: 60 minutes **Description:** oral or written exam (will be announced in class at the beginning of the semester). The examination language is English.	

Module MOBI-Proj-M Master Project Mobile Software Systems *Master Project Mobile Software Systems*	6 ECTS / 180 h
(since SS24) Person responsible for module: Prof. Dr. Daniela Nicklas	
Contents: Applications of in mobile software systems, which are taken from current research activities in mobile, context-aware systems and data stream management, are carried out in part individually and in part in small teams of students, from conception, via theoretical and/or practical realization, to evaluation. In particular, the project concerns the development of sound concepts pertaining to the task to be addressed under the given project constraints. This requires studying the current research literature and relevant approaches on the project's topic. An example of a project task would be the conceptual development, the prototypic implementation, and the case-study-driven evaluation of a small sensor-based, mobile system, which would require knowledge from the modul MOBI-DSC Data streams and event processing. The tasks in the project will be tailored to Master level.	
Learning outcomes: Students will deepen their knowledge regarding the conceptual problems that arise when carrying out theoretical and/or practical research on software projects, and regarding approaches to possible solutions. Since this will be done by means of the intensive conduct of a research topic in Mobile Software Systems, students will gain important experience in carrying out research-oriented projects, from project planning, to the abstract and concrete design, to the realization, to the documentation of results in a scientific project report.	
Remark: The main language of instruction in this course is English. However, the lectures and/or tutorials may be delivered in German if all participating students are fluent in German. The written reports/seminar essay and the presentation may be delivered in English or in German.	
prerequisites for the module: none	

Recommended prior knowledge: Foundations of relational databases, relational algebra and SQL; e.g. from Modul SEDA-DMS-B: Data management systems		**Admission requirements:** none
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Master project Mobile Software Systems **Mode of Delivery:** **Lecturers:** Prof. Dr. Daniela Nicklas **Language:** English/German **Frequency:** every winter semester	**4,00 Weekly Contact Hours**
Contents: The language of the course will be announced in the first lecture.	

Examination Coursework Assignment and Colloquium **prerequisites for module examination:** Regelmäßige Teilnahme an der Lehrveranstaltung **Description:** The language of the exam will be announced in the first lecture.	

Module MOBI-SEM-M Master-Seminar Mobile Software Systems *Master-Seminar Mobile Software Systems*	3 ECTS / 90 h
(since WS17/18) Person responsible for module: Prof. Dr. Daniela Nicklas	
Contents: Sensors continuously supply data that often cannot be understood by machines in its raw form. The topics in this seminar deal with different processes of how to obtain better information from continuous (sensor) data streams.	
Learning outcomes: gaining professional competence regarding the critical and systematic analysis of scientific literature; learning techniques to structure complex facts in the field of software systems science in systematic manner; evaluation of competing approaches; learning techniques to present scientific topics in professional manner and to write scientific papers.	
Remark: The module covers independent study and presentation of a topic on the chosen subject area, using scientific methods. Details on the topic and literature will be will be announced by the lecturer offering this module a the beginning of the seminar. The seminar thesis and the presentation may be delivered in English or in German	
prerequisites for the module: none	
Recommended prior knowledge: Scientific research and writing, e.g. from the module "IAIWAI-B Wissenschaftliches Arbeiten" or "SSS-SRW-M Scientific Research on Writing for Master´s Students".	**Admission requirements:** none
Frequency: every winter semester **Recommended semester:** **Minimal Duration of the Module:** 1 Semester	

Module Units	
Mobile Software Systems **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Daniela Nicklas **Language:** English **Frequency:** every winter semester **Contents:** The language of the course will be announced in the first course.	**2,00 Weekly Contact Hours**
Examination Coursework Assignment with presentation **Description:** The language of the exam will be announced in the first course.	

Module PSI-AdvaSP-M Advanced Security and Privacy *Advanced Security and Privacy*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium

(since SS24)
Person responsible for module: Prof. Dr. Dominik Herrmann

Contents:
Information security and privacy are relevant in almost all information systems today. Many real-world use cases have complex security and privacy requirements involving multiple parties. Often there are multiple stakeholders with different, sometimes even contradictory interests. For instance, some use cases call for a solution that allows a service provider to process sensitive data without learning its content. In other cases it is not the content but some meta information such as location and usage intensity that has to be protected. And then there are scenarios where seemingly harmless pieces of data can be used to disclose or infer very personal pieces of information about an individual.

This module covers advanced techniques for information security and privacy that can be used to satisfy the complex requirements of practical systems. It builds upon the basic concepts in information security that are introduced in the module "Introduction to Security and Privacy" (PSI-IntroSP-B).

Learning outcomes:
This module is designed to bring students towards the research boundaries in the field of security and privacy technologies by covering a selection of contemporary topics in depth. The focus of the module is on technical safeguards that can be used by system designers and users to enforce properties such as confidentiality and integrity. Moreover, sophisticated attacks on security and privacy are explained.

Successful students will be able to explain attack strategies and defenses discussed in recent research papers. They will also be able to analyze whether a particular attack or defense is relevant in a specific scenario. Finally, they will be able to implement selected attacks and defenses with a programming language of their choice.

Remark:
This module is taught in English. It consists of a lecture and tutorials. During the course of the tutorials there will be theoretical and practical assignments (task sheets). Assignments and exam questions can be answered in English or German.

Lecture and tutorials are partially taught in form of a paper reading class. Participants are expected to read the provided literature in advance and participate in the discussions.

Workload breakdown:

- Lecture: 22.5 hours (2 hours per week)
- Tutorials: 22.5 hours (2 hours per week)
- Preparation and studying during the semester: 30 hours
- Assignments: 67.5 hours
- Preparation for the exam (including the exam itself): 37.5 hours

prerequisites for the module:
none

Recommended prior knowledge:	**Admission requirements:**
Participants should be familiar with basic concepts in information security and privacy, which can be acquired, for instance, by taking the module "Introduction to Security and Privacy" (PSI-IntroSP-B).	none

This includes basic knowledge about the commonly used security terminology, common types of malware and attacks, buffer overflows and related attacks, cryptography, network security, web security, and concepts of privacy. Moreover, participants should have practical experience with at least one scripting or programming language such as Python or Java. Module Introduction to Security and Privacy (PSI-IntroSP-B) - recommended		
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Advanced Security and Privacy **Mode of Delivery:** Lectures **Language:** English/German **Frequency:** every summer semester **Learning outcome:** cf. module description **Contents:** Selected topics: • Authentication techniques • Privacy on the web (e.g., online tracking) • Privacy enhancing technologies (e.g., Tor) • Security and privacy aspects of e-mail • Usability aspects in security and privacy • Ethical aspects information security • Advanced techniques in software security (e.g., symbolic execution) • Advanced cryptographic building blocks • Other current topics in privacy and security Some parts of the lecture are aligned with current events and recently published research. The selected topics are therefore subject to change. **Literature:** Selected books: • R. Anderson: Security Engineering • A. Shostack: Threat Modelling • J.-P. Aumasson: Serious Cryptography • W. Stallings: Computer Security: Principles and Practice • B. Schneier et al.: Cryptography Engineering • J. Erickson: Hacking: The Art of Exploitation • J. Katz & Y. Lindell: Introduction to Modern Cryptography • L. Cranor & S. Garfinkel: Security and Usability	**2,00 Weekly Contact Hours**
2. Tutorials for Advanced Security and Privacy **Mode of Delivery:** Practicals **Language:** English/German	**2,00 Weekly Contact Hours**

Frequency: every summer semester	

Examination Written examination / Duration of Examination: 110 minutes **Description:** The exam time includes a reading time of 20 minutes. The content that is relevant for the exam consists of the content presented in the lecture and tutorials (including the assignments) as well as the content of the discussed papers. The maximum number of points that can be achieved in the exam is 100. Participants that solve all assignments correctly can collect up to 10 bonus points. Details regarding the number of assignments, the number of points per assignment, and the type of assignments will be announced in the first lecture. If the points achieved in the exam are sufficient to pass the exam on its own (generally, this is the case when at least 50 points have been obtained), the bonus points will be added to the points achieved in the exam. The grade 1.0 can be achieved without the bonus points.	

Module PSI-DiffPriv-M Introduction to Differential Privacy *Introduction to Differential Privacy*	6 ECTS / 180 h
(since WS23/24) Person responsible for module: Prof. Dr. Dominik Herrmann further responsible : Graf, Christian Alexander	

Contents:

The protection of personal data is an organizational as well as a technical challenge. Privacy-by-design is a reasonable requirement that is anything but easy to implement. This is especially true if a system deals with data that is meant to be published.What is more, a mathematically meaningful definition of privacy has only been available for less than a decade.

The lecture addresses different concepts and approaches for de-identification and attacks on privacy of published datasets. Its focus is on bringing you an in-depth understanding of differential privacy. Theoretical foundations, concepts and examples of state-of-the-art algorithms are introduced and explored in greater depth by means of practical exercises.

Contents:

1. Fundamental concepts of Data Privacy (8h)

- Outline of topic and its impact on society and economy
- A short history of data privacy
- Privacy by design and privacy frameworks
- Attacker models and attack patterns
- Different approaches to define privacy and their downsides
- Motivation and conceptual idea of Differential Privacy

2. Mathematical Foundations (20h)

- a review of important concepts from analysis, stochastic and statistics
- properties of important distributions, e.g. Gauss-, Exponential- and Laplace-distribution
- some useful theorems

3. An overview over common methods used in statistical disclosure control (10h)

- common methods used for de-identification and approaches to define privacy in depth
- common methods used for disclosure risk estimation and determination of data utility

4. Algorithmic foundations of Differential Privacy (16h)

- generalized data base models
- randomized algorithms
- mathematical definition and properties of differential privacy
- measuring privacy-loss and utility
- post processing immunity of dp-methods
- alternative dp definitions

5. Different approaches to achieve Differential Privacy (10h)

For instance:

- DIP (distribution invariant differential privacy)

- GAN-approaches
- Existing Software frameworks for de-identification

Learning outcomes:

- understand and apply de-identification approaches and attacks on privacy
- understand and apply fundamental stochastic and statistical methods used in statistical disclosure control
- understand the mathematical concepts of differential privacy following Dwork et. al.
- apply examples for dp-algorithms in example scenarios
- know different approaches towards differential privacy

prerequisites for the module:
none

Recommended prior knowledge:		**Admission requirements:**
none		none
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** Semester

Module Units

Lecture and Tutorial — **4,00 Weekly Contact Hours**

Mode of Delivery: Lectures and Practicals
Language: English
Frequency: every winter semester

Contents:
see module description

Literature:
Provisional recommended literature:

- Claire McKay Bowen: Protecting Your Privacy in a Data-Driven World
- Dwork, Roth: The Algorithmic Foundations of Differential Privacy, Foundations and Trends in
- Theoretical Computer Science Vol. 9, Nos. 3–4 (2013)
- SPECIAL ISSUE: A New Generation of Statisticians Tackles Data Privacy, CHANCE Magazine, 33:4, 2020.

Literature on probability theory and statistics:

- Ludwig Fahrmeier, Rita Künstler, Iris Pigeot, Gerhard Tutz, Statistik - Der Weg zur Datenanalyse, 8. Auflage, Springer, 2016.
- William Feller, An Introduction to Probability Theory and Its Applications Vol.I, 3.Auflage, John Wiley & Sons, 1968.
- David J.C. MacKay: Information Theory, Inference, and Learning Algorithms., Cambridge University Press, 2003.

Examination
/ Duration of Examination: 90 minutes

Module PSI-ProjectSP-M Project Security and Privacy *Project Security and Privacy*	6 ECTS / 180 h

(since SS24)
Person responsible for module: Prof. Dr. Dominik Herrmann

Contents:
In this project participants work independently on problems related to current research activities of the Privacy and Security in Information Systems Group. Instructors will provide guidance and supervision.

Learning outcomes:
Successful students will be able to independently work on research problems in security and privacy. They will also be able to implement tools and/or analyze data in order to answer a research question. Finally, they will be able to present their work in a talk and document their approach and results in a written report.

Remark:
This project is taught in English unless all participants are fluent in German. The workload of this project is equivalent to 270 hours.

Workload breakdown:

- 60 hrs: Getting familiar with the problem and preliminaries: reading related work, and understanding potentially existing source code
- 20 hrs: Preparing the talk (including time for attendance of other talks)
- 110 hrs: Implementing tools and/or analyzing data
- 80 hrs: Writing final report with approach and methods

prerequisites for the module:
none

Recommended prior knowledge:	**Admission requirements:**
Participants should have advanced knowledge and practical skills in information security and privacy, which can be acquired, for instance, in the module PSI-IntroSP-B and a security-related seminar or project. Depending on the actual topic participants may be expected to be familiar with commonly used security terminology, common types of malware and attacks, buffer overflows and related attacks, cryptography, network security, web security, and concepts of privacy. Moreover, participants should have practical experience with at least one scripting or programming language such as Python or Java. Alternatively, participants should have strong skills in empirical data collection and data analytics (statistics and/or machine learning). Experience with Linux environments, web technologies, and network protocols is recommended.	none

Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Project Security and Privacy **Mode of Delivery:** **Language:** English/German	**6,00 Weekly Contact Hours**

Frequency: every semester

Learning outcome:
cf. module description

Contents:
Potential topics include

- empirical studies, either manually (surveying security properties of systems) or automatically (e.g., web crawls),
- creating scanning tools and platforms where results can be published in a meaningful way (e.g., PrivacyScore.org),
- analyzing data sets for aspects of security and privacy, and
- implementing cryptographic or anonymization techniques in a secure fashion, e.g., for encrypted storage in cloud services.

Literature:
Literature will be announced at the beginning of the project.

Examination
Coursework Assignment and Colloquium / Duration of Examination: 30 minutes
Duration of Coursework: 3 months
prerequisites for module examination:
Regular attendance at project meetings.
Description:
The module examination consists of two parts: Firstly, the participants submit a written report (in English) that includes the source code, datasets, and analysis scripts. Secondly, the participants give a talk in which they defend their work (in English; in German if all participants are fluent in German) by presenting related work, their approach, and results. The maximum number of points that can be achieved in the module examination is 100.

Optionally, participants can submit intermediary results (in English) to collect up to 20 bonus points. If the module examination is passed on its own (generally, this is the case when at least 50 points are obtained), the bonus points will be added to the points achieved in the module examination. The grade 1.0 can be achieved without the bonus points. Details regarding the number of optional submissions during the semester, their type, the points per submission, and the respective deadlines will be announced in the first session of the project.

Module PSI-Sem-M Seminar Research Topics in Security and Privacy *Seminar Research Topics in Security and Privacy*	3 ECTS / 90 h
(since WS24/25) Person responsible for module: Prof. Dr. Dominik Herrmann	
Contents: This seminar provides in-depth coverage of advanced topics in one of the fields of information security and privacy. Participants learn to review, analyze, and discuss scientific sources (books and essays). While participants are expected to perform the actual research independently and mostly on their own, the instructors provide extensive support throughout the seminar. The instructors will provide guidance on scientific methods, e.g., how to approach a topic, how to find relevant literature, how to read a paper efficiently, how to write a seminar report, and how to give a good talk. Participants will be asked to deliver manageable chunks of work throughout the semester (such as summarizing literature in a survey, writing a draft of the term paper, etc.). The actual topics are subject to change. A list of available topics is made available before the first session via the seminar's VC course.	
Learning outcomes: The participants learn to find, read, and summarize scientific texts. They also learn to assess statements and to discuss them critically. Finally, they learn to write scientific texts and to present their results in a talk.	
Remark: The default language in this seminar is English, unless all participants are fluent in German.	
prerequisites for the module: none	

Recommended prior knowledge: Participants should have basic knowledge in software engineering, foundations of computing, operating systems, and networks. Knowledge in information security and privacy (obtained, e.g., in PSI-IntroSP-B and by having completed a seminar or thesis in the field of information security) is strongly recommended.		**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Seminar Research Topics in Security and Privacy **Mode of Delivery:** Seminar **Language:** English/German **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: cf. module description	
Literature: • Alley: The Craft of Scientific Writing	

• Anderson: Security Engineering • Pfleeger et al.: Security in Computing • Stallings & Brown: Computer Security: Principles and Practice • Strunk & White: The Elements of Style Other relevant literature is presented in the first session.	

Examination Internship report / Duration of Examination: 30 minutes Duration of Coursework: 2 months **prerequisites for module examination:** Continuous attendance in the seminar sessions is mandatory, cf. §9 (10) APO. **Description:** The module examination consists of two parts, a term paper (in English) and a talk (in English; in German if all participants are fluent in German). The maximum number of points that can be achieved in the module examination is 100. Details regarding the number of points that can be achieved in the talk and in the report will be announced in the first session of the project. Optionally, participants can participate in writing and presentation labs, where they can submit intermediary results (in English) or give mock presentations (in English). Participants can thereby earn 20 bonus points. If the module examination is passed on its own (generally, this is the case when at least 50 points are obtained), the bonus points will be added to the points achieved in the module examination. The grade 1.0 can be achieved without the bonus points.	

Module SNA-OSN-M Project Online Social Networks *Projekt zu Online Social Networks*	6 ECTS / 180 h
(since SS23) Person responsible for module: Prof. Dr. Oliver Posegga	
Contents: This module is an introduction to the analysis of online social networks. The aim is twofold: to provide students with the tools necessary to undertake research into online networks, and to give an overview of the type of questions these data can answer.	
Learning outcomes: At the conclusion of the course, students should know not only how to calculate basic network metrics on pre-existing data sets, but also how to capture an online social network efficiently with the intent of answering a specific research question. Further goals: • Learn how the radical innovation process in small teams works • Learn how to collaborate in multidisciplinary intercultural virtual teams • Learn how to find trendsetter and trends on the Internet and social media • Learn how to predict trends using SNA und statistical forecasting techniques	
Remark: The main language of instruction in this course is English. The written reports/seminar essay and the presentation have to be delivered in English.	
prerequisites for the module: none	
Recommended prior knowledge: We recommend attending at least one of the following courses: • Social Network Analysis (SNA-ASN-M) • Theories of Social Networks (SNA-NET-M)	**Admission requirements:** keine

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Online Social Networks **Mode of Delivery:** Practicals **Lecturers:** Prof. Dr. Oliver Posegga **Language:** English/German **Frequency:** every winter semester	**4,00 Weekly Contact Hours**
Contents: The course will define online networks, examine how they differ from offline social networks, and consider theoretical and methodological issues associated with their analysis. The sessions will explore different strategies to retrieve and analyze online network data, and present different empirical scenarios to which those tools have been applied.	
Literature:	

• Gloor, P. A. Swarm Creativity, Competitive Advantage Through Collaborative Innovation Networks. Oxford University Press, 2006 Further literature will be announced in the lecture.	

Examination Coursework Assignment and Colloquium / Duration of Examination: 30 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regelmäßige Teilnahme an der Lehrveranstaltung **Description:** Die Gewichtung der Prüfungsleistungen Hausarbeit und Kolloquium wird zu Beginn der Lehrveranstaltung von der Dozentin bzw. dem Dozenten bekannt gegeben.	

Module SSS-PraktIntKon-M Internship in an International Context *Praktikum im internationalen Kontext*	12 ECTS / 360 h
(since WS24/25) Person responsible for module: Prof. Ph.D. Michael Mendler	
Contents: As an internship in an international context, a subject-specific internship geared to the professional field of Software Systems Science must be proven, which must be completed in an international context, preferably abroad. The internship can be completed in a foreign or internationally operating domestic company (or research institution) in private or public hands. An internship placement must be chosen in such a way that it meets the training objectives of § 39 Para. 1.	
Learning outcomes: • Gain work experience in an international context, for international students specifically in the German labour market • Transfer and application of the (theoretical) knowledge learned at the university in the industrial practice • Reflection on one's own strengths and weaknesses by taking responsibility for small projects, to boost confidence in one’s abilities, to improve social skills • To learn to communicate constructively in a team, to create technical solutions in a partially specified context, under time and resource constraints • Networking with potential employers	
Remark: Proof of the internship must be provided in the form of an internship certificate from the organizational unit where the internship was completed and a written internship report. The internship certificate and the internship report must be submitted together to the module manager.	
prerequisites for the module: none	
Recommended prior knowledge: It is strongly recommended that you only start an internship once you have earned at least 30 ECTS credits in module groups A1 and A2.	**Admission requirements:** none

Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Examination Praktikumsbericht, unbenotet **Description:** The report must contain at least 4 pages of continuous text, not including the cover page.	

Module SSS-Thesis-M Master's Thesis in Software Systems Science *Master Thesis in Software Systems Science*	30 ECTS / 900 h
(since SS23) Person responsible for module: Prof. Ph.D. Michael Mendler further responsible : Professors of Computer Science	
Contents: The module for the master's thesis comprises 30 ECTS credit points and is assessed through a written exam in the form of a master's thesis document and an oral exam conducted as a colloquium. The topic of the master's thesis must be taken from one of the research areas specified in Appendix 2a of the study an examination regulations. Topics outside of these areas may also be admitted on request but must be individually approved by the examination board. For such an exception it must be plausibly justified that the chosen topic is related to the curriculum of the master's degree programme in International Software Systems Science.	
Learning outcomes: Through the successful completion of the master's thesis the examinee • demonstrates that they are able to conduct independent research; • produce technical solutions to a research problem of substantial size, • arising and identified from the current state of the art and • critically evaluate the contributions made. on the basis of the specific knowledge acquired during their degree studies.	
prerequisites for the module: The master's thesis cannot be registered and thus confirmed by the examination board until at least 60 ECTS credit points have been successfully completed towards the degree.	
Recommended prior knowledge: It is assumed that candidates are familiar with academic research and have the necessary skills for independent literature research and technical writing such as acquired through a bachelor thesis.	**Admission requirements:** none

Frequency: every semester	**Recommended semester:** 4.	**Minimal Duration of the Module:** 1 Semester

Examination Coursework Assignment / Duration of Coursework: 6 months **Description:** The marks obtained from the written work is weighted 67% of the total grade for the master's thesis module.	

Examination Colloquium **Description:** The examination includes a presentation (Kolloquium) of a duration between 20 and 60 minutes. The purpose of the presentation is for the student to defend their	

main results of the thesis. The thesis will be weighted with 67%, the presentation with 33%. The presentation will take place before or after the grading of the thesis, according to the student's preference.	

Module SWT-ASV-M Applied Software Verification *Applied Software Verification*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Gerald Lüttgen	
Contents: This module focuses on the increasingly important field of automated software verification, which aims at increasing the quality of today's complex computer systems. Students will be introduced to modern automated software verification and, in particular, to software model checking, and will be familiarised with a variety of important formal verification concepts, techniques and algorithms, as well as with state-of-the-art verification tools.	
Learning outcomes: On completion of this module, students will be able to thoroughly analyse software using modern software verification tools and understand the state-of-the-art techniques and algorithms that drive cutting-edge development environments offered by major software companies.	
Remark: The main language of instruction is English. The lectures and practicals may be delivered in German if all participating students are fluent in German. The total workload of 180 hrs. is split approximately as follows: • 30 hrs. attending lectures (Vorlesungen) • 30 hrs. attending practicals (Übungen) • 60 hrs. preparing and reviewing the lectures and practicals, including researching literature, studying material from additional sources and applying software tools • 30 hrs. working on the assignment (Hausarbeit) • 30 hrs. preparing for the colloquium (Kolloquium)	
prerequisites for the module: none	

Recommended prior knowledge: Basic knowledge in algorithms and data structures, mathematical logic and theoretical computer science.		**Admission requirements:** none
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Applied Software Verification **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Gerald Lüttgen **Language:** English **Frequency:** every summer semester	**2,00 Weekly Contact Hours**
Contents: The lectures (Vorlesungen) will address the following topics in automated software verification: (i) state machines, linear-time properties and algorithms for state space exploration; (ii) LTL model checking; (iii) SAT solving and bounded model checking; (iv) decision procedures and SMT solving; (v) software	

model checking; (vi) predicate abstraction. In addition, state-of-the-art software verification tools will be introduced. **Literature:** • Baier, C., Katoen, J.-P. Principles of Model Checking. MIT Press, 2008. • Clarke, E., Grumberg, O., Kroening, D., Peled, D. and Veith, H. Model Checking. 3rd. ed. MIT Press, 2018. • Huth, M. and Ryan, M. Logic in Computer Science. 2nd ed. Cambridge University Press, 2004. • Kroening, D. and Strichman, O. Decision Procedures: An Algorithmic Point of View. Springer, 2008. • Loeckx, J. and Sieber, K. The Foundations of Program Verification. 2nd ed. Wiley, 1987.	
2. Applied Software Verification **Mode of Delivery:** Practicals **Lecturers:** Scientific Staff Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen **Language:** English **Frequency:** every summer semester **Contents:** Students will practice the various theoretical and practical concepts taught in the lectures (Vorlesungen) by applying them to solve verification problems using several modern model-checking tools, and also by engaging in pen-and-paper exercises. Emphasis will be put on presenting and discussing the solutions to the exercises by and among the students, within the timetabled practicals (Übungen). **Literature:** - see the corresponding lectures -	**2,00 Weekly Contact Hours**

Examination Coursework Assignment and Colloquium / Duration of Examination: 20 minutes Duration of Coursework: 3 weeks **Description:** Assignment (Hausarbeit) consisting of questions that practice, review and deepen the knowledge transferred in the lectures and practicals (Vorlesungen und Übungen). The assignment is set in English language, while answers may be provided in either English or German. Colloquium (Kolloquium) consisting of questions testing the knowledge transferred in the lectures and practicals (Vorlesungen und Übungen), on the basis of the submitted solutions to the assignment (Hausarbeit). The colloquium can be held electively in English or German language.	

Module SWT-PR1-M Masters Project in Software Engineering and Programming Languages *Masterprojekt Softwaretechnik und Programmiersprachen*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Gerald Lüttgen	

Contents:

Topics in Software Engineering and Programming Languages are carried out individually or in teams of students, from conception, via theoretical and/or practical realization, to evaluation. In particular, the project concerns the development of sound concepts pertaining to the task to be addressed under the given project constraints. This requires studying academic literature and relevant technologies and approaches on the project's topic.

An example of a project task would be the conceptual development, the prototypic implementation, and the case-study-driven evaluation of tools for software verification, which requires the prior attendance of the module "Applied Software Verification" (SWT-ASV-M), or equivalent knowledge.

Learning outcomes:

Students will deepen their knowledge regarding the conceptual problems that arise when carrying out scientific projects related to Software Systems Science, and regarding approaches to possible solutions. Students will also gain important experience in carrying out such projects, from project planning, to the abstract and concrete design, to the realization, to the documentation of results in a scientific project report.

Remark:

The main language of instruction is English. The module may be delivered in German if all participating students are fluent in German. A regular participation in the project meetings is necessary.

The total workload of 180 hrs. is split approximately as follows:

- 10 hrs. participating in introductions to and tutorials on methods, software tools, and giving presentations on the project status
- 20 hrs. completing the exercises for bonus points
- 115 hrs. researching and familiarization with the project topic and conducting the project work
- 35 hrs. compilating a project report (written assignment/schriftliche Hausarbeit) and preparation of the Colloquium (Kolloquium).

prerequisites for the module:

none

Recommended prior knowledge: Basic knowledge in software engineering and programming languages, knowledge in the subject matter of the project topic.		**Admission requirements:** none
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Masters Project in Software Engineering and Programming Languages **Mode of Delivery:**	**4,00 Weekly Contact Hours**

Lecturers: Prof. Dr. Gerald Lüttgen, Scientific Staff Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
Language: English/German
Frequency: every semester

Learning outcome:
To be announced at the beginning of the project.

Contents:
Conduct of the project, accompanied by tutorials and regular project meetings.

Literature:
To be announced at the beginning of the project.

Examination
Coursework Assignment and Colloquium, schriftliche Hausarbeit mit Kolloquium /
Duration of Examination: 20 minutes
Duration of Coursework: 12 weeks
prerequisites for module examination:
Regelmäßige Teilnahme an den zugehörigen Lehrveranstaltungen
Description:
Production of a written report on the software project carried out (written assignment/schriftliche Hausarbeit). The student may choose whether to write/ compose the project report in English or German.

Discussion of this project report and of the developed artefacts in the context of the wider project topic (Colloquium/Kolloquium). The examination language is either English or German and may be chosen by the student at the colloquium.

Module SYSNAP-OSE-M Operating Systems Engineering *Operating Systems Engineering*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Michael Engel	

Contents:

Operating systems and related system software such as hypervisors form the basis of today's computer systems. The design and implementation of the core parts of system software can have significant impact not only on the performance of a computer system, but also on other aspects such a safety, security, and energy efficiency. Thus, the design and implementation of operating systems is a highly relevant topic for students working in all areas of computer science, from small embedded systems to large virtualized Cloud infrastructures.

This module concentrates on the central part ("kernel") of an operating system, i.e. the part of the system running in a privileged processor mode that interacts directly with hardware. Based on seminal publications, students will investigate different architectures of kernels, such as monolithic, micro- and exokernels, hypervisors and also unikernels. Mechanisms and policies of operating systems will be analyzed with respect to their functional as well as non-functional properties. The analysis of mechanisms dependent on a specific processor architecture will be explained using the modern and open RISC-V processor architecture.

A central part of this module will consist of code reading and the development of pieces of code for a small operating system. Different aspects of operating system functionality will be demonstrated through existing code. Constraints of, extension possibilities for, as well as alternative approaches to implement a given functionality will be discussed; this discussion will then form the basis for the implementation of a given feature in the practical exercises. An example for this is the discussion of file systems; here, features of a given traditional inode-based file system will be discussed and analyzed and alternative implementations, such as log-structured file systems, will be investigated and implemented in a basic form.

Learning outcomes:

The module is designed to enable students to not only understand the internals of operating systems, but also learn about different aspects of their implementation and the interaction between hardware and software. Starting from a thorough analysis of the internals of modern operating systems, this module will continue to present and discuss novel and non-traditional approaches to operating systems in the second half of the semester.

Successful students will be able to understand design and implementation aspects of system software as well as to comprehend and critically analyze proposed new approaches from the literature. They will also be able to understand the structure of and extend a given operating system code base with new functionality and test as well as evaluate functional and non-functional properties of the implementation. By writing system-level code running directly on hardware (or a hardware emulator), students will also be able to gain a better understanding of the operation of hardware and its interaction with software.

prerequisites for the module:

none

Recommended prior knowledge:	**Admission requirements:**
Participants should be familiar with basic concepts of operating systems and computer architecture, e.g. as acquired by	-

taking the module "Grundlagen der Rechnerarchitektur und Betriebssysteme" (Inf-GRABS-B). In addition, knowledge of C programming, debugging using gdb, using the Unix command line, and software construction tools (e.g. make) are useful.		
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units

1. Vorlesung Operating Systems Engineering — **2,00 Weekly Contact Hours**

Mode of Delivery: Lectures
Lecturers: Prof. Dr. Michael Engel
Language: German/English
Frequency: every summer semester

Learning outcome:
cf. module description

Contents:
cf. module description

Literature:

- Russ Cox, Frans Kaashoek and Robert Morris, "xv6: a simple, Unix-like teaching operating system", MIT PDOS group 2020, https://pdos.csail.mit.edu/6.S081/2020/xv6/book-riscv-rev1.pdf
- Zhao Jiong, "A Heavily Commented Linux Source code", http://www.oldlinux.org/download/ECLK-5.0-WithCover.pdf
- Marshall Kirk McKusick et al., "The Design and Implementation of the 4.4 BSD Operating System", Addison-Wesley 1996, ISBN-13: 978-0132317924
- Uresh Vahalia, "Unix: the New Frontiers", Pearson 1996, ISBN-13: 978-0131019089
- John Lions, "Commentary on the 6th Edition Unix System", 1977, https://warsus.github.io/lions-/
- David Patterson and Andrew Waterman, "The RISC-V Reader: An Open Architecture Atlas", Strawberry Canyon 2017, ISBN-13: 978-0999249116$
- Andrew Waterman, Krste Asanovic and John Hauser (eds.), "The RISC-V Instruction Set Manual Volume II: Privileged Architecture", Document Version 20211203, https://github.com/riscv/riscv-isa-manual/releases/download/Priv-v1.12/riscv-privileged-20211203.pdf

In addition, selected papers will be provided.

2. Übung Operating Systems Engineering — **2,00 Weekly Contact Hours**

Mode of Delivery:
Lecturers: Prof. Dr. Michael Engel
Language: German/English
Frequency: every summer semester

Learning outcome:
cf. module description

Contents:

cf. module description	

Examination Coursework Assignment and Colloquium / Duration of Examination: 30 minutes Duration of Coursework: 3 months **Description:** Oral examination concerning the topics discussed in the lecture, exercises and assignment. Students may choose English or German as the language for the oral examination. Examinations will take place at the end of the summer term or at the begin of the winter term (students may choose one of them). Students are assumed to work on a programming assignment ('schriftliche Hausarbeit') during the semester that is introduced at the beginning of the semester and uses the most important technologies discussed during the semester. **Note:** Without working on the programming assignment over the term students may run into problems during their oral examination (Kolloquium) as we discuss questions concerning topics from the lectures as well as from the assignment; questions about the assignment are based on the assignment solution programmed by the students.	

Module SYSNAP-Project-M Project Systems Programming *Projekt Systemnahe Programmierung*	6 ECTS / 180 h
(since SS24) Person responsible for module: Prof. Dr. Michael Engel	

Contents:

Students work (in groups) on a small yet realistic project to develop a standalone piece of system software that is not solvable in acceptable time by a single student. Hence, besides

- basic literature research to find approaches to solve the problem(s) at hand and to get used to the state-of-the-art technology required,
- analyzing, designing, architecting, programming and testing the practical solution,

skills such as planning, delegating and organizing work in groups are practiced.
Note: The topics of this master project are - compared to bachelor projects - more advanced and lead to advanced skills in the development of operating systems, machine-level and assembler programming as well as debugging.

Learning outcomes:

Students learn how to

- work independently and in groups on selected problems using the knowledge and skills provided by other modules,
- work with state-of-the-art tools and refer to recent scientific literature to look for problem solutions,
- architect and implement an operating system kernel interacting with emulators and real hardware,
- read, understand and apply data sheets as well as processor and peripheral user manuals
- document and present their work in an understandable manner to others,
- interact with others to discuss pros and cons of different solution approaches,
- organize work in groups, esp., how to delegate work, to fix interfaces and work under time constraints.

prerequisites for the module:
none

Recommended prior knowledge: Modules SYSNAP-OSE and/or SYSNAP-Virt		**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units

Projekt Systemnahe Programmierung **Mode of Delivery:** **Lecturers:** Prof. Dr. Michael Engel **Language:** German/English **Frequency:** every semester	**4,00 Weekly Contact Hours**
Learning outcome: see module description	
Contents: see module description	

Literature: Based on the concrete project topics literature will be provided at the start of the semester.	
Examination Coursework Assignment and Colloquium / Duration of Examination: 30 minutes Duration of Coursework: 3 months **prerequisites for module examination:** As this is a project in groups and the topic of the examination is the project work of each student, each student has to declare which part of the project and report is due to his own work. **Description:** A project report written in the style of a scientific publication is required. Master students are also expected to write reviews of their fellow students' papers in a round of peer review. In addition, delivery of the developed software based on the project work indicating which are the on achievements during the project. Oral examination concerning the technologies used in the project as well as the work of the group a student belongs to with an emphasis on her or his own work.	

Module SYSNAP-SEM-M Seminar System Software *Seminar System Software*		3 ECTS / 90 h
(since SS24) Person responsible for module: Prof. Dr. Michael Engel		
Contents: Current topics in system software, including operating systems, hypervisors, just-in-time compilation and hardware-software interfacing. Topics cover the full spectrum of research topics in these fields, from the analysis, design, implementation and evaluation of current system software, to the discussion and evaluation of novel research proposals.		
Learning outcomes: Students will compile and acquire current topics in operating systems by independently carrying out and documenting a literature survey, and by preparing and delivering a coherent, comprehensible presentation to their peers. Students will also be able to scientifically discuss topics in system software with their peers.		
prerequisites for the module: none		
Recommended prior knowledge: Basic knowledge in system software, machine-level programming and computer architecture and in the subject matter of the seminar. Additionally, basic knowledge of scientific methods is expected.		**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Seminar **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Michael Engel **Language:** German/English **Frequency:** every semester	**2,00 Weekly Contact Hours**
Learning outcome: cf. module description	
Contents: cf. module description	
Literature: Recent papers on system software related to the respective focus of the seminar, announced at the start of the semester.	
Examination Internship report / Duration of Examination: 30 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regular participation in the group meetings **Description:** Review of a written elaboration on the most important aspects of the topic, including a correct list of references.	

Participation in peer reviewing the other participants; free holding of a a presentation based on presentation documents including discussion of the contents with the seminar participants.	

Module SYSNAP-Virt-M Virtualization *Virtualisierung*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Michael Engel	
Contents: Virtualization is the basis of a significant part of the Internet infrastructure today. It is used in different contexts such as system-level virtualization for co-hosting virtual machines in Cloud infrastructures or just-in-time translation of JavaScript code in web applications. This module discusses virtualization technologies on all layers of the hardware/software stack, from system-level virtualization to virtual machines for high-level languages. Based on publications and real-world code examples, students will investigate different architectures of virtual machines. The design and implementation of virtualization technologies will be analyzed through the investigation of real-world open-source code examples for common hardware, such as x86, ARM and RISC-V.	
Learning outcomes: The module is designed to enable students to understand the different approaches to virtualization and learn details about their design and implementation. Students will learn to analyze the advantages and disadvantages of virtualization on different layers of a computer system and will gain experience in isolation and security properties of virtualized systems. Successful students will be able to understand design and implementation aspects of different virtualization approaches as well as to comprehend and critically analyze proposed new approaches from the literature. They will also be able to understand the structure of and extend a given virtualization system code base with new functionality and test as well as evaluate functional and non-functional properties of the implementation.	
prerequisites for the module: none	
Recommended prior knowledge: Participants should be familiar with basic concepts of operating systems and computer architecture, e.g. as acquired by taking the module "Grundlagen der Rechnerarchitektur und Betriebssysteme" (Inf-GRABS-B). In addition, knowledge of C programming, debugging using gdb, using the Unix command line, and software construction tools (e.g. make) are useful.	**Admission requirements:** -
Frequency: every winter semester / **Recommended semester:** / **Minimal Duration of the Module:** 1 Semester	

Module Units	
1. Vorlesung Virtualisierung **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Michael Engel **Language:** German/English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Learning outcome: c.f. module description	
Contents:	

c.f. module description

Literature:

- Jim Smith and Ravi Nair,
  Virtual Machines: Versatile Platforms for Systems and Processes
  Morgan Kaufmann, 1st edition 2005, ISBN-13: 978-1558609105
- Steven Hand, Andrew Warfield, Keir Fraser, Evangelos Kotsovinos, Dan Magenheimer
  Are Virtual Machine Monitors Microkernels Done Right?
  Proceedings of HotOS'05, 2005
- Gernot Heiser, Volkmar Uhlig and Joshua LeVasseur,
  Are virtual-machine monitors microkernels done right?,
  ACM SIGOPS Oper. Syst. Rev., vol. 40, number 1, 2006
- Barham, Paul, et al.,
  Xen and the art of virtualization,
  ACM SIGOPS operating systems review 37.5 (2003): 164-177
- Heiser, Gernot, and Kevin Elphinstone.
  L4 microkernels: The lessons from 20 years of research and deployment,
  ACM Transactions on Computer Systems (TOCS) 34.1 (2016): 1-29
- Engler, Dawson R., M. Frans Kaashoek, and James O'Toole Jr.,
  Exokernel: An operating system architecture for application-level resource management,
  ACM SIGOPS Operating Systems Review 29.5 (1995): 251-266
- Aycock, John,
  A brief history of just-in-time,
  ACM Computing Surveys (CSUR) 35.2 (2003): 97-113

Additional selected papers will be provided as required.

2. Übung Virtualisierung — **2,00 Weekly Contact Hours**

Mode of Delivery:
Lecturers: Prof. Dr. Michael Engel
Language: German/English
Frequency: every winter semester

Learning outcome:
c.f. module description

Contents:
c.f. module description

Examination
Coursework Assignment and Colloquium / Duration of Examination: 30 minutes
Duration of Coursework: 3 months

Description:
Oral examination concerning the topics discussed in the lecture, exercises and assignment. Students may choose English or German as the language for the oral examination. Examinations will take place at the end of the winter term or at the begin of the summer term (students may choose one of them).

Students are assumed to work on a programming assignment ('schriftliche Hausarbeit') during the semester that is introduced at the beginning of the semester and uses the most important technologies discussed during the semester.	

Module VIS-IVVA-M Advanced Information Visualization and Visual Analytics *Advanced Information Visualization and Visual Analytics*	6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Fabian Beck	
Contents: The course discusses methods for interactive information visualization and systems for explorative visual analysis. Visualizations blend with algorithmic solutions and get adopted to domain-specific needs. Giving a research-oriented perspective, the design and evaluation of such methods is the focus of the course, as well as their practical and interdisciplinary application in various fields.	
Learning outcomes: The students recognize the possibilities and limitations of data visualization and are able to apply visualization methods to concrete application examples. They understand the foundations of visual perception and cognition as well as their implications for the visual representation of data. They have a sound overview of possibilities for the visual representation of abstract data and are able to adapt visualization techniques to new problems and justify design decisions. On a conceptual level, they are able to integrate visualization techniques with interaction techniques and algorithmic solutions and design visual analytics solutions. They can evaluate visualization techniques in quantitative and qualitative user studies.	
Remark: The workload for this module typically is as follows: • Lecture and exercise sessions: 45h • Preparation and review of the lecture: 30h • Work on exercises and assignments: 75h • Preparation for the exam: 30h	
prerequisites for the module: none	
Recommended prior knowledge: Basic knowledge in information visualization (e.g., as provided through VIS-GIV-B) is recommended; knowledge in programming, algorithms and data structures, human-computer-interaction, and machine learning and data science can be beneficial.	**Admission requirements:** none

Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Advanced Information Visualization and Visual Analytics **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Fabian Beck **Language:** English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: See module description	

Literature: Further material and reading will be announced in the course.	
2. Advanced Information Visualization and Visual Analytics **Mode of Delivery:** Practicals **Lecturers:** N.N. **Language:** English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**
Contents: In the exercise sessions, lecture contents are expanded upon and their application is practiced.	

Examination Written examination / Duration of Examination: 90 minutes **Description:** By voluntarily handing in graded assignments (semesterbegleitende Studienleistungen) during the semester, points can be collected to improve the grade, which can be credited to the exam, provided that the exam is also passed without points from assignments. At the beginning of the course, it will be announced whether graded assignments are offered. If offered, the number, type, scope and processing time of the assignments as well as the number of achievable points per assignment and in the module examination will also be announced at this time. A grade of 1.0 can also be achieved without points from the assignments.	

Module VIS-Sem-M Master Seminar Information Visualization *Masterseminar Informationsvisualisierung*	3 ECTS / 90 h
(since SS22) Person responsible for module: Prof. Dr. Fabian Beck	
Contents: The seminar investigates current trends in a subarea of visualization research. Based on an extensive literature review, different visualization approaches will be compared and evaluated. All participants work on individually assigned topics that contribute different facets to an overarching seminar topic.	
Learning outcomes: Students learn to independently research and find the latest research results regarding a given research topic in applied computer science. They discuss and evaluate state-of-the-art research results and develop a deep understanding of the individual topic, its potential use and application as well as limitations. They practice methods of scientific communication in oral and written form.	
Remark: The workload for this module typically is as follows: • Sessions: 20h • Literature search and reading: 25h • Preparation of presentation: 15h • Report writing: 30h	
prerequisites for the module: none	
Recommended prior knowledge: None required, but basic knowledge in visualization, human-computer-interaction, or machine learning and data science can be beneficial.	**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Masterseminar Informationsvisualisierung **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Fabian Beck, N.N. **Language:** English/German **Frequency:** every semester	**2,00 Weekly Contact Hours**
Contents: See module description	
Literature: Further material and reading will be announced in the course.	

Examination	
Coursework Assignment with presentation / Duration of Examination: 30 minutes Duration of Coursework: 4 months	

prerequisites for module examination: Regular participation in the course **Description:** The language of the course and exam will be announced in the first session of the course.	

Module xAI-DL-M Deep Learning *Deep Learning*		6 ECTS / 180 h
(since WS24/25) Person responsible for module: Prof. Dr. Christian Ledig		
Contents: Deep Learning is a form of machine learning that learns hierarchical concepts and representations directly from data. Enabled by continuously growing dataset sizes, compute power and rapidly evolving open-source frameworks Deep Learning based AI systems continue to set the state of the art in many applications and industries. The course will provide an introduction to the most relevant techniques in the field of Deep Learning and a broad range of its applications.		
Learning outcomes: In this course students will learn/recap some fundamentals from mathematics and machine learning that are critical for the introduction of the concept of Deep Learning. Participants will learn about various foundational technical aspects including optimization and regularization strategies, cost functions and important network architectures such as Convolutional Networks. Students will further get an insight into more advanced concepts such as sequence modelling and generative modelling. Participants will further learn about representative architectures of important algorithm categories, e.g., classification, detection, segmentation, some of their concrete use cases and how to evaluate them. The lecture is accompanied by exercises and assignments that will help participants develop practical, hands-on experience. In those exercises students will learn how to implement and evaluate Deep Learning algorithms using Python and its respective commonly used libraries.		
Remark: The lecture is conducted in English. The workload of this module is expected to be roughly as follows: • Lecture: 22.5h (equals the 2 SWS) • Preparation of lectures and analysis of further sources: 30h (over the 15 weeks term) • Exercise classes accompanying lecture: 22.5h (equals the 2 SWS) • Work on the actual assignments: 75h (over the 15 weeks term) • Preparation for exam: 30h		
prerequisites for the module: none		
Recommended prior knowledge: Strongly recommended: Good working knowledge of programming (in particular Python), Mathematics for Machine Learning [xAI-MML] Further recommended (or similar): Bachelorproject Erklärbares Maschinelles Lernen [xAI-Proj-B], Lernende Systeme / Machine Learning [KogSys-ML-B], Einführung in die Künstliche Intelligenz / Introduction to AI [KogSys-KI-B], Algorithmen und Datenstrukturen [AI-AuD-B]		**Admission requirements:** none
Frequency: every winter semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Deep Learning **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Christian Ledig **Language:** English/German **Frequency:** every winter semester **Learning outcome:** c.f. module description **Contents:** The lecture will be held in English. The following is a selection of topics that will be addressed in the course • Relevant concepts in linear algebra, probability and information theory • Deep feedforward networks • Convolutional Neural Networks • Regularization, Batch Normalization • Optimization (Backpropagation, Stochastic Gradient Decent) and Cost Functions • Classification (binary, multiclass, multilabel) • Object Detection & Segmentation • Generative Modelling • Attention mechanisms & Transformer Networks • Evaluation of ML approaches **Literature:** • Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning, MIT Press, 2016 • Zhang, Lipton, et al.: Dive into Deep Learning (https://d2l.ai/) Further literature will be announced at the beginning of the course.	**2,00 Weekly Contact Hours**
2. Deep Learning **Mode of Delivery:** Practicals **Lecturers:** N.N. **Language:** English/German **Frequency:** every winter semester **Learning outcome:** see module description **Contents:** Further exploration of concepts discussed in the lecture, often accompanied by assigments and programming exercises implemented in Python and the corresponding machine/deep learning libraries. **Literature:** see lecture description	**2,00 Weekly Contact Hours**

Examination	
Written examination / Duration of Examination: 90 minutes **Description:**	

The content that is relevant for the exam consists of the content presented in the lecture and exercises/tutorials (including the assignments) as well as additional content of the discussed literature, which will be highlighted. Participants can collect bonus points by working on and solving the assignments discussed during the exercises/tutorials. Details regarding the number of assignments, the number of points per assignment, and the type of assignments will be announced in the lecture. If the points achieved in the exam are sufficient to pass the exam on its own, the bonus points (at most 20% of the maximum achievable points in the exam) will be added to the points achieved in the exam. The grade 1.0 can be achieved without the bonus points.	

Module xAI-Sem-M1 Master Seminar Explainable Machine Learning *Masterseminar Erklärbares Maschinelles Lernen*	3 ECTS / 90 h
(since SS22) Person responsible for module: Prof. Dr. Christian Ledig	
Contents: Machine Learning holds great promise to transform a variety of industries including healthcare. However, there are key challenges when translating AI technology reliably into practice. In this seminar students will learn about a selected subarea of machine learning often in the context of a particular application. The seminar will enable students to apply knowledge from corresponding lectures and exercises and independently explore a particular research-oriented topic based on published literature. The seminar focuses on a wide spectrum of aspects not limited to pure technical questions.	
Learning outcomes: Students will learn about the potential as well as current challenges when translating AI systems into practice. Participants will learn to independently research their specific topic by deep diving into and structuring published literature. Within the seminar students learn to present and communicate state-of-the-art research results in both oral (presentation) and written form (technical report). Seminar participants will further learn about and critically discuss scientific questions with their peers. In comparison to the Bachelor Seminar this Master Seminar is more ambitious in terms of complexity of selected topics as well as expectations with respect to delivered reports and presentations.	
Remark: This seminar is generally conducted in English. The workload of this module is expected to be roughly as follows: • Attendance of seminar / presentation: 20h • Literature review and familiarization with topic: 25h • Preparation of presentation: 15h • Written report: 30h	
prerequisites for the module: none	
Recommended prior knowledge: Recommended completion of module "Lernende System / Machine Learning" or "Einführung in die KI / Introduction into AI" or „Deep Learning“	**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Master Seminar Explainable Machine Learning **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Christian Ledig **Language:** English/German **Frequency:** every semester	**2,00 Weekly Contact Hours**
Contents: see module description	

Literature: Will be announced at the beginning of the course.	

Examination Coursework Assignment with presentation / Duration of Examination: 30 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regular attendance of seminar and other presentations. **Description:** The seminar will be held in English including the report and presentations.	

Module Handbook Summary

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A1 Software System Science		**36 - 54**		
	In module group A1, at least 6 ECTS credits must be completed in each focus area. Each module can only be included once in the module group.				
	Focus Area: S1: Distributed and Mobile Systems		**6 - 36**		
DSG-DSAM-M	Distributed Systems Architectures and Middleware	every winter semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
DT-DBCPU-M	Database Systems for modern CPU	every summer semester(1)	6	4 Lectures and Practicals	Written examination 90 minutes
GdI-FPRS-M	Functional Programming of Reactive Systems	every summer semester	6	2 Lectures 2 Practicals	Written examination 90 minutes Oral examination 30 minutes
KTR-Mobi-M	Mobile Communication	every winter semester	6	4 Lectures and Practicals	Oral examination 30 minutes
MOBI-DSC-M	Data Streams and Complex Event Processing	every winter semester(1)	6	2 Lectures 2 Practicals	Oral examination 15 minutes Written examination 60 minutes
PSI-AdvaSP-M	Advanced Security and Privacy	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 110 minutes
PSI-DiffPriv-M	Introduction to Differential Privacy	every winter semester(1)	6	4 Lectures and Practicals	90 minutes
SWT-ASV-M	Applied Software Verification	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 weeks 20 minutes
	Focus Area: S2: Software Analysis and Verification		**6 - 30**		

AISE-Auto	Automation of First- and Higher-Order Logic	every summer semester(1)	6	6 Lectures and Practicals	Oral examination 30 minutes
AISE-UL	Universal Logic & Universal Reasoning	every winter semester(1)	6	2 Lectures and Practicals 2 Practicals	Written examination (AISE-UL: Universal Logic & Universal Reasoning (Universelle Logik & Universelles Schließen))
GdI-FPRS-M	Functional Programming of Reactive Systems	every summer semester	6	2 Lectures 2 Practicals	Written examination 90 minutes Oral examination 30 minutes
GdI-IFP-M	Introduction to Functional Programming	every winter semester	6	2 Lectures 2 Practicals	Written examination 90 minutes
SWT-ASV-M	Applied Software Verification	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 weeks 20 minutes
	Focus Area: S3: Service-oriented Architectures		**6 - 24**		
AlgoK-Algo	Algorithms	alle 4 Semester(1)	6	4 Lectures and Practicals	Sonstiges
MOBI-ADM-M	Advanced Data Management	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 75 minutes
SYSNAP-OSE-M	Operating Systems Engineering	every summer semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months 30 minutes
SYSNAP-Virt-M	Virtualization	every winter semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months

					30 minutes
	Focus Area: S4: Communication Systems and Protocols		**6 - 36**		
DT-DBCPU-M	Database Systems for modern CPU	every summer semester(1)	6	4 Lectures and Practicals	Written examination 90 minutes
KTR-Mobi-M	Mobile Communication	every winter semester	6	4 Lectures and Practicals	Oral examination 30 minutes
MOBI-DSC-M	Data Streams and Complex Event Processing	every winter semester(1)	6	2 Lectures 2 Practicals	Oral examination 15 minutes Written examination 60 minutes
PSI-AdvaSP-M	Advanced Security and Privacy	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 110 minutes
PSI-DiffPriv-M	Introduction to Differential Privacy	every winter semester(1)	6	4 Lectures and Practicals	90 minutes
SWT-ASV-M	Applied Software Verification	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 weeks 20 minutes
SYSNAP-OSE-M	Operating Systems Engineering	every summer semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months 30 minutes
SYSNAP-Virt-M	Virtualization	every winter semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months 30 minutes

Module Handbook Summary

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A2 Domain-specific Software System Science		**0 - 18**		
EESYS-ADAML-M	Applied Data Analytics and Machine Learning in R	every winter semester	6	2 Lectures 2 Practicals	Written examination 90 minutes
EESYS-ES-M	Energy Efficient Systems	every summer semester	6	2 Lectures 2 Practicals	Written examination 90 minutes
HCI-MCI-M	Human-Computer Interaction	every winter semester	6	2 Lectures 2 Practicals	Oral examination Written examination 90 minutes
HCI-US-B	Ubiquitous Systems	every winter semester	6	2 Lectures 2 Practicals	Oral examination Written examination 90 minutes
ISPL-MDP-M	Managing Digital Platforms	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 90 minutes
SNA-OSN-M	Project Online Social Networks	every winter semester	6	4 Practicals	Coursework Assignment and Colloquium 4 months 30 minutes
VIS-IVVA-M	Advanced Information Visualization and Visual Analytics	every winter semester(1)	6	2 Lectures 2 Practicals	Written examination 90 minutes
xAI-DL-M	Deep Learning	every winter semester(1)	6	2 Lectures 2 Practicals	Written examination 90 minutes

Module Handbook Summary

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A3 Seminar and Project		**9**		
	Elective Area: Project		**6**		
DT-Proj-M	Project: Data Engineering	every semester(1)	6	4	Colloquium, Coursework Assignment 3 months 30 minutes
GdI-Proj-M	Master's Project Theoretical Foundations of Computing	every semester	6	4	Coursework Assignment and Colloquium 4 months 20 minutes
MOBI-Proj-M	Master Project Mobile Software Systems	every winter semester(1)	6	4	Coursework Assignment and Colloquium
PSI-ProjectSP-M	Project Security and Privacy	every semester(1)	6	6	Coursework Assignment and Colloquium 3 months 30 minutes
SWT-PR1-M	Masters Project in Software Engineering and Programming Languages	every winter semester	6	4	Coursework Assignment and Colloquium (schriftliche Hausarbeit mit Kolloquium) 12 weeks 20 minutes
SYSNAP-Project-M	Project Systems Programming	every semester(1)	6	4	Coursework Assignment and Colloquium 3 months 30 minutes
	Elective Area: Seminar		**3**		
AlgoK-Sem-M	Master Seminar Algorithms and Complexity Theory	winter and summer	3	2 Seminar	Internship report 4 months 30 minutes

		semester, on demand(1)			
DT-DB42-M	Database Systems - The question to or the better answer than 42?	winter and summer semester, on demand(1)	3	2 Seminar	Internship report 14 days 30 minutes
HCI-Prop-M	Propaedeutic: Human-Computer-Interaction	every winter semester(1)	3	3	Internship report 4 months 30 minutes
HCI-Sem-HCC-M	Master-Seminar Human-Centred Computing	every summer semester	3	2 Seminar	Internship report 4 months 30 minutes
HCI-Sem-M	Master-Seminar Human-Computer Interaction	every winter semester	3	2 Seminar	Internship report 4 months 30 minutes
MOBI-SEM-M	Master-Seminar Mobile Software Systems	every winter semester(1)	3	2 Seminar	Coursework Assignment with presentation
PSI-Sem-M	Seminar Research Topics in Security and Privacy	every semester(1)	3	2 Seminar	Internship report 2 months 30 minutes
SYSNAP-SEM-M	Seminar System Software	every semester(1)	3	2 Seminar	Internship report 4 months 30 minutes
VIS-Sem-M	Master Seminar Information Visualization	every semester(1)	3	2 Seminar	Coursework Assignment with presentation 4 months 30 minutes
xAI-Sem-M1	Master Seminar Explainable Machine Learning	every semester(1)	3	2 Seminar	Coursework Assignment with presentation 4 months 30 minutes

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A4 Master's Thesis		**30**		
SSS-Thesis-M	Master's Thesis in Software Systems Science	every semester	30		Coursework Assignment 6 months Colloquium

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A5 International Experience		**27**		
	Modules amounting to 27 ECTS credits must be completed in module group A5.				
	Elective Area: Guided study abroad		**0 - 27**		
	Modules amounting to 0 to 27 ECTS credits can be included in the compulsory elective area a, which are completed as part of a guided study abroad program at a foreign university, provided that they differ significantly from the modules to be completed in accordance with these regulations and can be assigned to module groups A1, A2 or A3.				
	Elective Area: Internship in an International Context		**0 - 12**		
	For the internship a foreign or internationally acting domestic company (or research institute) may be selected. It has to offer a specific internship related to relevant topics of software systems science. The documentation of the internship requires the delivery of the following items to the degree programme representative: • written report of 4 pages at least, reporting on the tasks and achievements, and • a certificate issued by the hosting institution or the organizational unit that has realized the internship.				
SSS-PraktIntKon-M	Internship in an International Context	every semester(1)	12		Writen Report on Practical Training
	Elective Area: Foreign Languages		**0 - 15**		
	In the elective area 5c, Foreign languages, modules comprising up to 15 ECTS credits can be taken from the range offered by the University's Language Centre. Excluded are modules in English and modules in the language in which the university entrance qualification was obtained. Details, in particular the modules available for selection and the module examinations, are described in the *Modulhandbuch des Sprachenzentrums der Otto-Friedrich-Universität Bamberg:* https://www.uni-bamberg.de/abt-studium/aufgaben/modulhandbuecher/sprachenzentrum/modulhandbuch-sprachenzentrum/				
	Elective Area: Further Modules from A1 and/or A2		**0 - 27**		
	In this area, modules from A1 or A2, which were not previously completed, can be taken. It is not possible to complete additional seminars or prejects (modules from A3).				