Otto-Friedrich Universität Bamberg

Module Handbook

Master's Degree Programme
International Software Systems Science

Faculty of Information Systems
and Applied Computer Sciences

According to the valid version of the study and examination regulations of 06.03.2015 for the Master's degree programme International Software Systems Science at the Otto Friedrich University of Bamberg. Valid from summer semester 2024 on for students who started their studies before the summer semester 2021.

Valid: 17.05.2024

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 the previously valid module handbook (cf. no. 2b) 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*.

Notice on the validity of newer versions of examination regulations:

The provisions in § 36 apply in accordance with the Examination Regulations of 18.01.2021 (valid for students who started their studies from the summer semester 2021 onwards): A specialization field is indicated in the final examination certificate at the request of the student. A corresponding form for the application is published by the examination board.

List of deviations in the module program MSc. International Software System Science compared to the Examination Regulations of 06.03.2015

- The examination **duration** specifications in the column "Prüfungen" have been deleted from the study and examination regulations. The examination duration of a module is specified in the module handbook.
- The module **SWT-PCC-M** (elective module for students starting before WS19/20, compulsory module for students starting from WS19/20 on) will no longer be offered as of SS21. It was first replaced by the new module SWT-CPS-M until summer semester 2023 and from winter semester 2023/24 on by the module **SWT-SWQ-M**, which was an elective module in module group A1 before.
- The elective module **GdI-FP-M** will no longer be offered as of SS21.
- The module KogSys-ML-M is now offered as KogSys-ML-B and therefore is no longer eligible in the Master’s program International Software System Science.
- The elective module **DSG-SRDS-M** will no longer be offered as of WS22/23.
- The elective module **SME-STE-M** will no longer be offered as of WS23/24.

Equivalence List M.Sc. International Software Systems Science
Examination Regulations of 06.03.2015

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)**
MOBI-DSC	Data Streams and Complex Event Processing	SS 18	MOBI-DSC-M	Data Streams and Complex Event Processing	WS 1819
GdI-AFP-M	Advanced Functional Programming	WS 2021	GdI-FPRS-M	Functional Programming of Reactive Systems	SS 21
EESYS-DAE-M	Data Analytics in der Energieinformatik	SS 21	EESYS-ADAML-M	Applied Data Analytics and Machine Learning in R	WS 2122
KogSys-ML-M	Lernende Systeme (Machine Learning)	SS22	KogSys-ML-B	Einführung in Maschinelles Lernen	WS22/23

Date of the equivalence and deviations list: 12.01.2024

Modules

AISE-Auto: Automation of First- and Higher-Order Logic....10
AISE-UL: Universal Logic & Universal Reasoning....12
AlgoK-Algo: Algorithms....15
AlgoK-Sem-M: Master Seminar Algorithms and Complexity Theory....17
DSG-DSAM-M: Distributed Systems Architectures and Middleware....18
DSG-DistrSys-M: Distributed Systems....20
DSG-SOA-M: Service-Oriented Architecture and Web Services....23
DSG-Sem-M: Master Seminar in Distributed Systems....26
DT-CPP-M: Advanced Systems Programming in C++ (Master)....28
DT-DB42-M: Database Systems - The question to or the better answer than 42?....30
DT-DBCPU-M: Database Systems for modern CPU....31
EESYS-ADAML-M: Applied Data Analytics and Machine Learning in R....33
EESYS-ES-M: Energy Efficient Systems....36
GdI-CSNL-M: Computational Semantics of Natural Language....39
GdI-Sem-M: Master's Seminar Theoretical Computer Science....41
HCI-MCI-M: Human-Computer Interaction....43
HCI-Prop-M: Propaedeutic: Human-Computer-Interaction....46
HCI-Sem-HCC-M: Master-Seminar Human-Centred Computing....48
HCI-Sem-M: Master-Seminar Human-Computer Interaction....50
HCI-US-B: Ubiquitous Systems....52
KTR-GIK-M: Foundations of Internet Communication....55
KTR-MAKV-M: Modeling and Analysis of Communication Networks and Distributed Systems....58
KTR-MMK-M: Multimedia Communication in High Speed Networks....61
KTR-Mobi-M: Mobile Communication....64
KTR-SSSProj-M: KTR Master Project Software Systems Science....67
KTR-Sem-M: Master Seminar Communication Systems and Computer Networks....70
MOBI-ADM-M: Advanced Data Management....72
MOBI-DSC-M: Data Streams and Complex Event Processing....74
MOBI-PRS-M: Master Project Mobile Software Systems (SoSySc)....76

MOBI-SEM-M: Master-Seminar Mobile Software Systems........ 78
PSI-AdvaSP-M: Advanced Security and Privacy........ 79
PSI-ProjectCAD-M: Project Complex Attacks and Defenses........ 82
PSI-ProjectSP-M: Project Security and Privacy........ 85
PSI-Sem-M: Seminar Research Topics in Security and Privacy........ 87
SNA-OSN-M: Project Online Social Networks........ 89
SSS-PraktIntKon-M: Internship in an International Context........ 91
SSS-Thesis-M: Master's Thesis in Software Systems Science........ 92
SWT-ASV-M: Applied Software Verification........ 94
SWT-PCC-M: Principles of Compiler Construction........ 96
SWT-PR2-M: SWT Masters Project in Software Systems Science........ 98
SWT-SEM-M: Seminar in Software Engineering and Programming Languages (Master)........ 100
SWT-SWQ-M: Software Quality........ 102
SYSNAP-OSE-M: Operating Systems Engineering........ 104
SYSNAP-PMAP-M: Processor Microarchitecture and Performance........ 107
SYSNAP-Project-M: Project Systems Programming........ 109
SYSNAP-SEM-M: Seminar System Software........ 111
SYSNAP-Virt-M: Virtualization........ 113
VIS-IVVA-M: Advanced Information Visualization and Visual Analytics........ 116
VIS-Sem-M: Master Seminar Information Visualization........ 118
xAI-DL-M: Deep Learning........ 120
xAI-MML-M: Mathematics for Machine Learning........ 123
xAI-Sem-M1: Master Seminar Explainable Machine Learning........ 126

Index by areas of study

1) Software Systems Science for students starting before WS 19/20 (Modulgruppe) ECTS: 30 - 48

In module groups A1 and A2, modules totalling 48 ECTS points are to be completed in accordance with the minimum and maximum limits applicable to the module groups.

Please note that the module SWT-PCC-B is no longer offered as of the winter semester 2020/21.

AISE-Auto: Automation of First- and Higher-Order Logic (6 ECTS, every summer semester).................. 10

AISE-UL: Universal Logic & Universal Reasoning (6 ECTS, every winter semester)................................12

AlgoK-Algo: Algorithms (6 ECTS, alle 4 Semester)... 15

DSG-DSAM-M: Distributed Systems Architectures and Middleware (6 ECTS, every winter semester).....18

DSG-DistrSys-M: Distributed Systems (6 ECTS, every summer semester)...20

DSG-SOA-M: Service-Oriented Architecture and Web Services (6 ECTS, every summer semester)....... 23

DT-CPP-M: Advanced Systems Programming in C++ (Master) (6 ECTS, every winter semester)........... 28

DT-DBCPU-M: Database Systems for modern CPU (6 ECTS, every summer semester).........................31

KTR-GIK-M: Foundations of Internet Communication (6 ECTS, every summer semester).......................55

KTR-MAKV-M: Modeling and Analysis of Communication Networks and Distributed Systems (6 ECTS, every summer semester)... 58

KTR-MMK-M: Multimedia Communication in High Speed Networks (6 ECTS, every summer semester)..61

KTR-Mobi-M: Mobile Communication (6 ECTS, every winter semester)...64

MOBI-ADM-M: Advanced Data Management (6 ECTS, every summer semester).................................. 72

MOBI-DSC-M: Data Streams and Complex Event Processing (6 ECTS, every winter semester).............74

PSI-AdvaSP-M: Advanced Security and Privacy (6 ECTS, every summer semester).............................. 79

SWT-ASV-M: Applied Software Verification (6 ECTS, every summer semester).....................................94

SWT-SWQ-M: Software Quality (6 ECTS, every winter semester)..102

SYSNAP-OSE-M: Operating Systems Engineering (6 ECTS, every summer semester)........................ 104

SYSNAP-PMAP-M: Processor Microarchitecture and Performance (6 ECTS, every summer semester)..107

SYSNAP-Virt-M: Virtualization (6 ECTS, every winter semester)...113

2) A1 Software Systems Science for students starting from WS 1920 onwards (Modulgruppe) ECTS: 30 - 48

In module groups A1 and A2, modules totalling 48 ECTS credits must be completed in accordance with the minimum and maximum limits applicable to the module groups.

Please note that the module SWT-PCC-B is no longer offered as of the winter semester 2020/21. Instead, the module SWT-SWQ-B can be taken and recognised in the compulsory area.

a) compulsory part (Teilmodulgruppe) ECTS: 24

DSG-DSAM-M: Distributed Systems Architectures and Middleware (6 ECTS, every winter semester).....18

KTR-GIK-M: Foundations of Internet Communication (6 ECTS, every summer semester).......................55

MOBI-DSC-M: Data Streams and Complex Event Processing (6 ECTS, every winter semester).............74

SWT-PCC-M: Principles of Compiler Construction (6 ECTS, every summer semester).......................... 96

SWT-SWQ-M: Software Quality (6 ECTS, every winter semester)..102

b) elective modules (Teilmodulgruppe) ECTS: 0 - 24

AISE-Auto: Automation of First- and Higher-Order Logic (6 ECTS, every summer semester)..................10

AISE-UL: Universal Logic & Universal Reasoning (6 ECTS, every winter semester)...............................12

AlgoK-Algo: Algorithms (6 ECTS, alle 4 Semester)... 15

DSG-DistrSys-M: Distributed Systems (6 ECTS, every summer semester)..20

DSG-SOA-M: Service-Oriented Architecture and Web Services (6 ECTS, every summer semester).......23

DT-CPP-M: Advanced Systems Programming in C++ (Master) (6 ECTS, every winter semester)........... 28

DT-DBCPU-M: Database Systems for modern CPU (6 ECTS, every summer semester).........................31

KTR-MAKV-M: Modeling and Analysis of Communication Networks and Distributed Systems (6 ECTS, every summer semester).. 58

KTR-MMK-M: Multimedia Communication in High Speed Networks (6 ECTS, every summer semester)..61

KTR-Mobi-M: Mobile Communication (6 ECTS, every winter semester)..64

MOBI-ADM-M: Advanced Data Management (6 ECTS, every summer semester).................................. 72

PSI-AdvaSP-M: Advanced Security and Privacy (6 ECTS, every summer semester)............................. 79

SWT-ASV-M: Applied Software Verification (6 ECTS, every summer semester).....................................94

SYSNAP-OSE-M: Operating Systems Engineering (6 ECTS, every summer semester)........................ 104

SYSNAP-PMAP-M: Processor Microarchitecture and Performance (6 ECTS, every summer semester)...107

SYSNAP-Virt-M: Virtualization (6 ECTS, every winter semester)...113

3) A2 Domain-specific Software Systems Science (Modulgruppe) ECTS: 0 - 18

In module groups A1 and A2, modules totalling 48 ECTS points are to be completed in accordance with the minimum and maximum limits applicable to the module groups.

EESYS-ADAML-M: Applied Data Analytics and Machine Learning in R (6 ECTS, every winter semester)....33

EESYS-ES-M: Energy Efficient Systems (6 ECTS, every summer semester).... 36

GdI-CSNL-M: Computational Semantics of Natural Language (6 ECTS, every summer semester)....39

HCI-MCI-M: Human-Computer Interaction (6 ECTS, every winter semester).... 43

HCI-US-B: Ubiquitous Systems (6 ECTS, every winter semester)....52

SNA-OSN-M: Project Online Social Networks (6 ECTS, every winter semester)....89

VIS-IVVA-M: Advanced Information Visualization and Visual Analytics (6 ECTS, every winter semester)....116

xAI-DL-M: Deep Learning (6 ECTS, every winter semester).... 120

xAI-MML-M: Mathematics for Machine Learning (6 ECTS, every summer semester)....123

4) A3 Seminar and Project (Modulgruppe) ECTS: 12

a) Elective Unit A3WP1: Seminar (Teilmodulgruppe) ECTS: 3

AlgoK-Sem-M: Master Seminar Algorithms and Complexity Theory (3 ECTS, winter and summer semester, on demand)....17

DSG-Sem-M: Master Seminar in Distributed Systems (3 ECTS, every semester)....26

DT-DB42-M: Database Systems - The question to or the better answer than 42? (3 ECTS, winter and summer semester, on demand)....30

GdI-Sem-M: Master's Seminar Theoretical Computer Science (3 ECTS, winter or summer semester, on demand)....41

HCI-Prop-M: Propaedeutic: Human-Computer-Interaction (3 ECTS, every winter semester).... 46

HCI-Sem-HCC-M: Master-Seminar Human-Centred Computing (3 ECTS, every summer semester)....48

HCI-Sem-M: Master-Seminar Human-Computer Interaction (3 ECTS, every winter semester).... 50

KTR-Sem-M: Master Seminar Communication Systems and Computer Networks (3 ECTS, winter or summer semester, on demand)....70

MOBI-SEM-M: Master-Seminar Mobile Software Systems (3 ECTS, every winter semester)....78

PSI-Sem-M: Seminar Research Topics in Security and Privacy (3 ECTS, every winter semester)....87

SWT-SEM-M: Seminar in Software Engineering and Programming Languages (Master) (3 ECTS, every semester)....100

SYSNAP-SEM-M: Seminar System Software (3 ECTS, every semester)....111

VIS-Sem-M: Master Seminar Information Visualization (3 ECTS, every semester).... 118

xAI-Sem-M1: Master Seminar Explainable Machine Learning (3 ECTS, every semester)...................... 126

b) Project (Teilmodulgruppe) ECTS: 9

MOBI-PRS-M: Master Project Mobile Software Systems (SoSySc) (9 ECTS, every summer semester)...76

SWT-PR2-M: SWT Masters Project in Software Systems Science (9 ECTS, every semester).................98

KTR-SSSProj-M: KTR Master Project Software Systems Science (9 ECTS, every semester)................. 67

PSI-ProjectCAD-M: Project Complex Attacks and Defenses (9 ECTS, every semester).......................... 82

PSI-ProjectSP-M: Project Security and Privacy (6 ECTS, every semester)...85

SYSNAP-Project-M: Project Systems Programming (6 ECTS, every semester)....................................109

5) A4: Masters Thesis (Modulgruppe) ECTS: 30

SSS-Thesis-M: Master's Thesis in Software Systems Science (30 ECTS, every semester).....................92

6) A5 International Experience (Modulgruppe) ECTS: 30

According to the examination regulations (StuFPO) Appendix 1, students have four options regarding the Module Group A5, *International Experience*, which may also be combined:

(1) to study modules of software systems science at a university abroad for at least one semester or

(2) to accomplish a traineeship in an international context, preferentially abroad, that covers topics of the occupational field of software systems science with a volume of at least 360 working hours (12 ECTS credits).

(3) to accomplish *further* modules of module groups A1 and A2 (Examination Regulations, App. 1)

(4) to accomplish up to 18 ECTS credits in modules of foreign languages (neither English nor native language).

a) Guided graduate study abroad (Teilmodulgruppe) ECTS: 0 - 30

Regarding the study of software systems science modules at a university abroad, courses with a workload equivalent to 30 ECTS credits can be accomplished.
The courses that are selected at a foreign university have to be approved by learning agreements. For own planning security reasons, learning agreements have to be signed by those Professors at University of Bamberg responsible for the chosen subject, as well as the head of the Examination Board, before the graduate study abroad is initiated.

b) Internship in an International context (Teilmodulgruppe) ECTS: 0 - 12

Regarding the elective area 5b, *Internship in an international context*, with an equivalent workload of 12 ECTS credits, 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 (12 ECTS, every semester).......................... 91

c) Foreign languages (Teilmodulgruppe) ECTS: 0 - 18

In the elective area 5c, *Foreign languages*, modules comprising up to 18 ECTS credits can be taken from the range offered by the University's Language Centre. Excluded are modules of the English language and modules of the language in which the university entrance qualification was obtained.
Details, in particular the modules available for selection and the respective Module examinations are described (in German) in the *Modulhandbuch des Sprachenzentrums der Otto-Friedrich-Universität Bamberg.*

d) further modules from module groups A1 and/ or A2 (Module Group) ECTS: 0 - 30

Additional, not previously completed modules from A1 or A2 module groups’ required elective options in accordance with the Examination Regulations, Appendix 1.

Module AISE-Auto Automation of First- and Higher-Order Logic *Automation of First- and Higher-Order Logic*	6 ECTS / 180 h

(since SS23)
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:
verpflichtende Nachweise de

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:** ECTS-Bedingungen de

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

Module Units

Automation of First- and Higher-Order Logic **Language:** German **Frequency:** every summer semester **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.	**6,00 Weekly Contact Hours**

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) **Mode of Delivery:** Practicals **Lecturers:** Prof. Dr. Christoph Benzmüller **Language:** English **Frequency:** every winter semester	**2,00 Weekly Contact Hours**

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 DSG-DistrSys-M Distributed Systems

Distributed Systems

6 ECTS / 180 h
45 h Präsenzzeit
135 h Selbststudium

(since SS20)
Person responsible for module: Prof. Dr. Guido Wirtz

Contents:
Nowadays infrastructure and business relies more or less on distributed systems of various flavors. Most of our civilization would not work any more if all distributed systems would fail. So, that should be a good reason for anyone planning to work in the context of IT to learn at least about the characteristics and basic issues of such systems. The course introduces to the different flavors of and issues with distributed systems, discusses the most basic problems arising with this kind of systems and presents solutions and techniques that are essential to make distributed systems work. Additionally, the course also teaches how to build simple distributed systems using Java-based technologies like process interaction, synchronization, remote message invocation and web service infrastructure. Students are required to work (in groups) on assignments in order to combine the theoretical concepts with practical experience and ... Yes, we program!

Learning outcomes:
Students know about the characteristics and different flavors of distributed systems and understand the essential differences compared to monolithic, centralized systems as well as their consequences when designing and building distributed systems. Students are able to apply the basic algorithmic techniques and programming paradigms in order to build simple distributed systems themselves. Students have gained basic experience with practically building and running distributed systems.

Remark:
The language of instruction in this course is English.

The overall workload of 180h for this module consists of:

- weekly classes: 22.5h
- tutorials: 22.5h
- Work on assignment: 75h
- Literature study 30h
- preparation for and time of the final exam: 30h

This course is intended for 2nd/3rd year bachelor students as well as master students which have not enrolled in a similar course during their bachelor studies. In case of questions don't hesitate to contact the person responsible for this module.

prerequisites for the module:
none

Recommended prior knowledge:
Knowledge of the basics of computer science in general, esp. operating systems, as well as practical experience in Java programming, as the subjects taught in DSG-EiAPS-B and DSG-EiRBS-B. Preferable also knowledge about multithreading and synchronization like, e.g., the subject-matters of DSG-PKS-B.

Module Introduction to Parallel and Distributed Programming (DSG-PKS-B) - recommended

Admission requirements:
none

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

Module Units

1. Lecture Distributed Systems — **2,00 Weekly Contact Hours**

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

Learning outcome:
c.f. module description

Contents:
c.f. module description

Literature:

- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems - Concepts and Design. Pearson Education UK, 2011 (5. edition);
- Andrew Tanenbaum, Marten van Steen: Distributed Systems - Principles and Paradigms, 2017 (3rd edition)
- Additional research literature will be provided during the term for selected readings and discussions

2. Tutorial Distributed Systems — **2,00 Weekly Contact Hours**

Mode of Delivery: Practicals
Lecturers: Scientific Staff Praktische Informatik
Language: German
Frequency: every summer semester

Learning outcome:
c.f. 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.

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 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 DSG-SOA-M Service-Oriented Architecture and Web Services *Service-Oriented Architecture and Web Services*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(since SS20) Person responsible for module: Prof. Dr. Guido Wirtz	

Contents:

Building enterprise-scale IT systems requires sound concepts for integrating software. **Service-oriented architectures (SOAs)** have been the number one answer to this integration challenge for years. Indeed, service orientation is and will be a cornerstone in modularizing large IT landscapes and alignment with business needs is the driving factor for service engineering. A SOA composes an IT system from services in a loosely-coupled manner. Each service implements a business task and therefore have a clear value attribution. When business needs change, the loose coupling of services allows for quick adjustment of the SOA. In recent years, Microservices have been put forward as a new paradigm for organizing software-intensive systems as a set of small services that communicate using lightweight communication technologies and are *independently deployable by fully automated deployment machinery*. Conceptually, Microservices and SOA share a lot, but the Microservices paradigm puts a lot more emphasis on automation in development and therefore is a better fit for modern development practices.

When moving beyond company boundaries and opening up the solution space is necessary, **software ecosystems (SECOs)**come into play. Software ecosystems integrate software contributions from independent organizational entities and enable software products and solutions that a single company cannot realize alone. Prominent representatives of software ecosystems are Android and the Playstore or iOS and the AppStore. But the paradigm of software ecosystems goes far beyond mobile platforms and also covers application areas in the cloud domain or the embedded domain.

Skilled software architects therefore reconcile the business views and technical views for the benefit of the enterprise and therefore need both, advanced knowledge in business process and workflow management as well as a rock-solid understanding of service engineering and distributed computing.

This course will introduce you to the world of architectures for large-scale software by giving a brief overview on distributed systems and software architecture in general. Then SOAs as an architectural paradigm and Web Services (WSDL + REST) as SOA implementation technology will be treated in detail. SOA will be contrasted to Microservices and the development aspects that Microservices focuses on will be discussed. Software ecosystems then will be introduced as a paradigm for organizing software systems and container technology (Linux Containers (LXC) and Docker) as a frequent implementation means for software ecosystems will be introduced. In particular, we will investigate what building industry-grade ecosystems based on container technology means in practice.

- Conceptual Foundations of SOA
- SOA Characterisitics
- Microservices
- WSDL and Basic Web Services
- REST-ful Services
- Software Ecosystems
- Container technology

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 service development and SOA tools.

Also, you will get a grasp of current services research and you will get the chance to discuss selected publications with your lecturers.

Learning outcomes:
Students know about the different aspects of service-oriented architectures and their practical use. Students

- Understand the characteristics of SOAs, Microservices and SECOs and its implications on IT systems.
- Know relevant technologies and standards in the field and being able to combine some of these to develop basic Web Services and service compositions
- Being able to compare WSDL Web Services to REST Web Services
- Being able to use container technology for integrating software
- Being able to judge IT architectures from a SOA/Microservices/SECO perspective.
- Being able to understand and discuss scientific work in the area

Remark:
The main language of instruction in this course is English.

The overall workload of 180h for this module consists of:

- weekly classes: 22.5h
- tutorials: 22.5h
- Work on assignment: 75h
- Literature study 30h
- preparation for and time of final exam: 30h

prerequisites for the module:
Basic knowledge in software engineering and in distributed systems as introduced, e.g., in the modules DSG-IDistrSys-B or DSG-DistrSys-M.

Recommended prior knowledge: Basic knowledge in software engineering and distributed systems. Module Introduction to Distributed Systems (DSG-IDistrSys) - recommended	**Admission requirements:** none

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

Module Units	
1. Lectures Service-Oriented Architecture and Web Services **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Guido Wirtz, Scientific Staff Praktische Informatik **Language:** English **Frequency:** every summer semester **Learning outcome:** c.f. overall module description **Contents:** c.f. overall module description **Literature:**	**2,00 Weekly Contact Hours**

SOA is still a fast emerging field - most recent version of standards and up-to-date literature will be provided at the beginning of each course.	
2. Practicals Service-Oriented Architecture and Web Services **Mode of Delivery:** Practicals **Lecturers:** Scientific Staff Praktische Informatik **Language:** English/German **Frequency:** every summer 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. S**tudents 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.	

<table>
<tr><td colspan="2">Module DSG-Sem-M Master Seminar in Distributed Systems
<i>Masterseminar zu Verteilten Systemen</i></td><td>3 ECTS / 90 h</td></tr>
<tr><td colspan="3">(since SS24)
Person responsible for module: Prof. Dr. Guido Wirtz</td></tr>
<tr><td colspan="3">Contents:
This module is intended to offer an in-depth study of specific topics in distributed systems that go well beyond the topics discussed in DSG-DistrSys-M, DSG-SOA-M or DSG-DSM-M. We try to close the gap between 'standard' lecture topics often dealing with the (required) basics and the state-of-the-art related to a specific research question regarding distributed systems in general, SOC and SOA, server-side middleware, process languages, as well as questions w.r.t. standard conformance, interoperability and correctness based on 'ground-breaking' as well as up-to-date research papers from international journals and/or conferences.</td></tr>
<tr><td colspan="3">Learning outcomes:
Students will learn how to read and work on research papers, how to present their essence as an outline talk to colleguages (students) and how to guide discussion sessions based on scientific talks. Students will be able to classify and compare results from papers in the context of a specific research question. Moreover, students will become proficient in the developments of the specialized research area that is the topic of the particular course.</td></tr>
<tr><td colspan="3">Remark:
The seminar will regularly be taught in English.</td></tr>
<tr><td colspan="3">prerequisites for the module:
none</td></tr>
<tr><td colspan="2">Recommended prior knowledge:
Basic knowledge about distributed systems as offered, e.g., by the course
<i>DSG-IDistrSys-B</i> oder <i>DSG-DistrSys-M</i> or similar knowledge.
Dependend on the topic of the specific seminar, additional knowledge as discussed in DSG-SOA-M or DSG-DSAM-M may be helpful (ask if in doubt before enrolling in the course)</td><td>Admission requirements:
none</td></tr>
<tr><td>Frequency: every semester</td><td>Recommended semester:
2.</td><td>Minimal Duration of the Module:
1 Semester</td></tr>
</table>

<table>
<tr><td colspan="2">Module Units</td></tr>
<tr><td>Master Seminar in Distributed Systems
Mode of Delivery: Seminar
Lecturers: Prof. Dr. Guido Wirtz
Language: English/German
Frequency: every semester
Learning outcome:
see module description
Contents:
see module description</td><td>2,00 Weekly Contact Hours</td></tr>
</table>

Literature: depends on specific topics of each seminar and will be given in the introductionary meeting	
Examination Internship report / Duration of Examination: 30 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regelmäßige Teilnahme an der Lehrveranstaltung **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 DT-CPP-M Advanced Systems Programming in C++ (Master)

Fortgeschrittene Systemprogrammierung in C++ (Master)

6 ECTS / 180 h

(since WS23/24 to SS24)
Person responsible for module: Prof. Dr. Maximilian Schüle

Contents:
In diesem Modul wird die fortgeschrittene Systemprogrammierung in C++ gelehrt. Dabei lernen die Teilnehmer nicht nur ihr Wissen in kleinen Programmierhausaufgaben anzuwenden sondern auch das gelernte Wissen in einer übergreifenden Projektarbeit zu kombinieren.

Learning outcomes:
Anwendung komplexer C++-Systemprogrammierung in eigenständiger Projektarbeit

prerequisites for the module:
none

Recommended prior knowledge:	**Admission requirements:**
none	none

Frequency: every winter semester	**Recommended semester:** from 3.	**Minimal Duration of the Module:** 1 Semester

Module Units

Fortgeschrittene Systemprogrammierung in C++ (Master)
Mode of Delivery: Lectures and Practicals
Lecturers: Prof. Dr. Maximilian Schüle
Language: English
Frequency: every winter semester

6,00 Weekly Contact Hours

Learning outcome:
Anwendung komplexer C++-Systemprogrammierung in eigenständiger Projektarbeit

Contents:
In diesem Modul wird die fortgeschrittene Systemprogrammierung in C++ gelehrt. Dabei lernen die Teilnehmer nicht nur ihr Wissen in kleinen Programmierhausaufgaben anzuwenden sondern auch das gelernte Wissen in einer übergreifenden Projektarbeit zu kombinieren.

Literature:
Primary

- C++ Reference Documentation
- Lippman, 2013. C++ Primer (5th edition).
- Stroustrup, 2013. The C++ Programming Language (4th edition).
- Meyers, 2015. Effective Modern C++. 42 Specific Ways to Improve Your Use of C++11 and C++14.

Supplementary

- Aho, Lam, Sethi & Ullman, 2007. Compilers. Principles, Techniques & Tools (2nd edition).

- Tanenbaum, 2006. Structured Computer Organization (5th edition).	
Examination Colloquium, Coursework Assignment / Duration of Examination: 30 minutes Duration of Coursework: 4 months	

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 SS23) Person responsible for module: Prof. Dr. Maximilian Schüle	

Contents:

In this seminar, we study the challenges of modern database systems. We discuss the topic along with very recent publications about database systems for machine learning and knowledge discovery

Learning outcomes:

Selbständig Publikationen verfassen

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:** Semester

Module Units

Datenbanksysteme - Die Frage zu oder die bessere Antwort auf 42? **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Maximilian Schüle **Language:** German **Frequency:** winter and summer semester, on demand **Learning outcome:** Selbständig Publikationen verfassen **Contents:** In this seminar, we study the challenges of modern database systems. We discuss the topic along with very recent publications about database systems for machine learning and knowledge discovery	**2,00 Weekly Contact Hours**

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

<table>
<tr><td colspan="2">Module DT-DBCPU-M Database Systems for modern CPU
<i>Datenbanksysteme für moderne CPU</i></td><td>6 ECTS / 180 h</td></tr>
<tr><td colspan="3">(since WS23/24 to SS24)
Person responsible for module: Prof. Dr. Maximilian Schüle</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td colspan="3">Learning outcomes:
Konzepte von Datenbanksystemen verstehen und Datenbanksysteme implementieren können inkl. für moderne Hardware</td></tr>
<tr><td colspan="3">prerequisites for the module:
none</td></tr>
<tr><td colspan="2">Recommended prior knowledge:
MOBI-DBS-B</td><td>Admission requirements:
none</td></tr>
<tr><td>Frequency: every summer semester</td><td>Recommended semester:</td><td>Minimal Duration of the Module:
1 Semester Semester</td></tr>
</table>

<table>
<tr><td colspan="2">Module Units</td></tr>
<tr><td>Datenbanksysteme für moderne CPU
Mode of Delivery: Lectures and Practicals
Lecturers: Prof. Dr. Maximilian Schüle
Language: English
Frequency: every summer semester
Learning outcome:
Konzepte von Datenbanksystemen verstehen und Datenbanksysteme implementieren können inkl. für moderne 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. <i>Database Systems: The Complete Book</i>
• D. E. Knuth.The Art of Computer Programming Volume III</td><td>6,00 Weekly Contact Hours</td></tr>
</table>

- 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 Oral examination alone / Duration of Examination: 20 minutes	

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: 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.	**Admission requirements:** 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 **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.	**2,00 Weekly Contact Hours**
2. Practicals Data Analytics in Energy Informatics **Mode of Delivery:** Practicals **Language:** German/English **Frequency:** every winter semester **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	**2,00 Weekly Contact Hours**

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-CSNL-M Computational Semantics of Natural Language *Computational Semantics of Natural Language*	6 ECTS / 180 h

(since WS23/24)
Person responsible for module: Prof. Ph.D. Michael Mendler
further responsible : Luke Burke

Contents:

The formal study of natural language syntax and semantics has developed as a very lively sub-field of linguistics in the past 50 years, with the typed lambda calculus in particular providing a way of giving compositional analyses of meanings in natural language. Recently, monads and continuations have been employed as tools in natural language syntax and semantics. The aim of this module is to introduce the use of monads and continuations in natural language semantics and to discuss different approaches to the formal representation of quantifier scope ambiguities in natural language. The basics of natural language semantics (typed lambda calculus) will be briefly introduced, before discussing a continuation-based approach to quantification in natural language, which will be contrasted with other approaches. Monads representing focus, intensionality and non-determinism in natural language will be discussed. We will look at how analyses of the meaning of sentences can be represented in Haskell.

Importantly, the course may differ slightly from other courses in that assessment will not concentrate on technical exercises; rather, we require careful reading and dissection of relevant literature on the topic, since the primary mode of assessment will be via seminar presentations and essays, and you will be assessed on your understanding of, and your independent analysis of, relevant literature discussed in lectures. Independent reading of this literature will in fact be essential.

This course may also be of interest to students in philosophy and linguistics.

Learning outcomes:

At the end of this course students should be familiar with different approaches to the formal representation of quantifier scope ambiguities in natural language; be familiar with how monads and continuations have been used in natural language semantics; be familiar with the use of Haskell to formalise analyses in natural language semantics; be able to produce and manipulate terms of the typed lambda calculus to represent how meanings combine; have an understanding of how both logics and trees have been used to represent natural language syntax; be acquainted with logics such as Montague’s “Intensional Logic” and Gallin’s Ty2.

Remark:

The workload for this module consists of:

- participation in lectures and tutorial sessions: 45hrs
- individual preparation and reading: 105hrs
- exam preparation and oral exam: 30hrs

prerequisites for the module:

none

Recommended prior knowledge: Willingness to read relevant literature, critically discuss and analyse it and write about it. Basic logic (GdI-MfI-1: Mathematik fur Informatik or an equivalent level of understanding). Some knowledge of modal logic more basic than that required for (GdI-MTL: Modal and	**Admission requirements:** English language skills at Level B2 (UniCert II) or above.

Temporal Logic). Knowledge of the typed lambda calculus (abstraction and application) and elementary Haskell (Gdl-IFP: Introduction to Functional Programming) would be very useful, though not essential.		
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** Semester

Module Units	
Computational Semantics of Natural Language **Language:** English **Frequency:** every summer semester **Contents:** Through prepared class presentations, essay writing, and direct interactions with the students the lecturer introduces the topics of the course in detail. The seminars deepen the students' understanding of the theoretical concepts and constructions covered in the lectures through presentations, which involve comparing alternative analyses of linguistic phenomena. **Literature:** • van Eijck, J. And Unger, Christina, “Computational Semantics with Functional Programming”, Cambridge University Press 2010 • Barker, C. and Shan, C.-C., “Continuations and natural language”, Volume 53. Oxford studies in Theoretical Linguistics, Oxford University Press, 2014 • Carpenter, Bob, “Type-Logical Semantics”, MIT Press (1997) • Keenan, Edward, and Stabler, Edward, “Mathematical structures in Language”, CSLI publications, Stanford, 2016 • Gallin, Daniel, “Intensional and Higher-Order Modal logic. North Holland, 1975.	**4,00 Weekly Contact Hours**

Examination Portfolio / Duration of Examination: 80 minutes **Description:** The components of the portfolio will be announced at the beginning of each semester.	

Module GdI-Sem-M Master's Seminar Theoretical Computer Science *Masterseminar Grundlagen der Informatik*	3 ECTS / 90 h
(since WS17/18) Person responsible for module: Prof. Ph.D. Michael Mendler	

Contents:
The GdI seminar will be held on a semesterly basis on varying topics in the area of theoretical foundations of computer science.

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 in writing and orally. Promotion of the scientific curiosity and the formation of a self-confident research attitude towards Computer Science Engineering.

Remark:
The written seminar essay and the presentation may be delivered in English or in German.

prerequisites for the module:
none

Recommended prior knowledge: Discrete Mathematics, elementary Logic and Algebra. Introduction to Theoretical Computer Sciences, Functional Programming; Distributed Systems; English language skills at level B2 (UniCert II) or above.		**Admission requirements:**
Frequency: winter or summer semester, on demand	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units

Master's Seminar Theoretical Computer Science **Mode of Delivery:** Seminar **Lecturers:** Michael Mendler, N.N. **Language:** English/German **Frequency:** winter or summer semester, on demand **Contents:** The GdI seminar will be held on a semesterly basis on varying topics in the area of theoretical foundations of computer science. **Literature:** Pertinent literature will be selected and announced during the first classes at the beginning of the semester.	**2,00 Weekly Contact Hours**
Examination Coursework Assignment with presentation / Duration of Examination: 30 minutes Duration of Coursework: 4 months **prerequisites for module examination:** Regelmäßige Teilnahme an der Lehrveranstaltung **Description:**	

The examination language will be announced in the first course.	

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 **Contents:** After an introduction into the subject the following topics are covered in this lecture: • Mobile human-computer interaction	**2,00 Weekly Contact Hours**

- 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
Mode of Delivery: Practicals
Lecturers: Scientific Staff Mensch-Computer-Interaktion
Language: German/English
Frequency: every winter semester

2,00 Weekly Contact Hours

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 WS17/18 to SS24)
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	**3,00 Weekly Contact Hours**

Mode of Delivery:
Lecturers: Prof. Dr. Tom Gross, Scientific Staff Mensch-Computer-Interaktion
Language: German/English
Frequency: every winter semester

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 Coursework Assignment with presentation / 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 WS17/18 to SS24)
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 **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	**2,00 Weekly Contact Hours**
Examination	

Coursework Assignment with presentation / 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 WS17/18 to SS24)
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 Coursework Assignment with presentation / 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 WS21/22 to SS24)
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 **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:	**2,00 Weekly Contact Hours**

- 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
Mode of Delivery: Practicals
Lecturers: Scientific Staff Mensch-Computer-Interaktion
Language: German/English
Frequency: every winter semester

2,00 Weekly Contact Hours

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 KTR-GIK-M Foundations of Internet Communication

Grundbausteine der Internet-Kommunikation

6 ECTS / 180 h
45 h Präsenzzeit
135 h Selbststudium

(since SS20)
Person responsible for module: Prof. Dr. Udo Krieger

Contents:

The course provides an introduction to the theoretical foundations of important technical issues related to the fundamentals of Internet communication, the data link layer, routing and transport protocols in IP networks, as well as advanced topics such as real-time communication and security in IP networks. The implementation of the learnt concepts in terms of predetermined configuration tasks in the communication laboratory by small teams of students constitutes the tutorial part of the course. For this purpose, guidelines, technical instructions, and tools will be provided.

The implementation tasks include the configuration and testing of computer networks in the laboratory setting. Operating system and required software components like Wireshark, Atheris and Vyatta software router will be provided. The basic handling of the hardware and software itself will be perfomed by the students as part of their individual intellectual efforts within te couse.

Learning outcomes:

The important skill to provide a qualified assessment of current communicaton technologies and corresponding practical knowledge can only be acquired by team-oriented processes subject to time constraints and the clear specification of technical and administrative objectives. In the course Foundations of Internet Communication and its tutorials in the router laboratory students will learn to work independently with a high level of responsibility as self-confident member of a successful team.

It is the objective of the course that the students acquire practical knowledge on modern data communication in Internet and learn how communication concepts can be developed, implemented and judged with th highest level of expertise.

The course is open to bachelor students in their transition phase to the master prgram. It attempts to prepare for the job in communication industry related fields. Master students in the first semester and exchange students from abroad are invited to join the course.

Remark:

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

The workload is composed of the following items:

- participation in lectures, tutorials in the laboratory, laboratory meetings: 45 hours
- preparation, execution, post-processing of lectures and tutorials in the laboratory: 100 hours
- preparation of the examination: 35 hours

prerequisites for the module:

none

Recommended prior knowledge:

- data communication similar to module KTR-Datkomm-B
- fundamental knowledge on programming in JAVA (or C++)
- working knowledge on LINUX is recommended, but not assumed

Module Algorithms and Data Structures (AI-AuD-B) - recommended

Admission requirements:

governed by examination regulations (StuFPO)

Module Introduction to Algorithms, Programming and Software (DSG-EiAPS-B) - recommended Module Data communication (KTR-Datkomm-B) - recommended	

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

Module Units

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

4,00 Weekly Contact Hours

Learning outcome:

The important skill to provide a qualified assessment of current communicaton technologies and corresponding practical knowledge can only be acquired by team-oriented processes subject to time constraints and the clear specification of technical and administrative objectives. In the course Foundations of Internet Communication and its tutorials in the router laboratory students will learn to work independently with a high level of responsibility as self-confident member of a successful team.

It is the objective of the course that the students acquire practical knowledge on modern data communication in Internet and learn how communication concepts can be developed, implemented and judged with th highest level of expertise.

The course is open to bachelor students in their transition phase to the master prgram. It attempts to prepare for the job in communication industry related fields. Master students in the first semester and exchange students from abroad are invited to join the course.

Contents:

The course provides an introduction to the theoretical foundations of important technical issues related to the fundamentals of Internet communication, the data link layer, routing and transport protocols in IP networks, as well as advanced topics such as real-time communication and security in IP networks. The implementation of the learnt concepts in terms of predetermined configuration tasks in the communication laboratory by small teams of students constitutes the tutorial part of the course. For this purpose, guidelines, technical instructions, and tools will be provided.

The implementation tasks include the configuration and testing of computer networks in the laboratory setting. Operating system and required software components like Wireshark, Atheris and Vyatta software router will be provided. The basic handling of the hardware and software itself will be perfomed by the students as part of their individual intellectual efforts within the couse.

The organization of the laboratories is following the framework of industry. It comprises definition, preparation, implementation and presentation phases. An incremental processing is performed like in industrial projects. It means

- a segmentation into specific work packages,

- its division into tasks and subtasks including milestones
- the presentation of intermediate results
- a final report with presentation

Further laboratories related to current research issues in "Future Generation Internet" will be integrated into the course on demand. Details are discussed in the first lecture.

An actual list of studied topics and related references are presented in the first lecture.

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

Literature:
Foundations:
- J. Liebeherr, M. Elzarki: Mastering Networks, An Internet Lab Manual, Pearson Education, Boston, 2004.

Further references related to specific workpackages:
- Kurose, J., Ross, K.W.: Computer Networking – a Top-Down Approach, Addison-Wesley, 2013 .
- Tanenbaum, A. S.: Computer Networks, Pearson Education, 2010.
- Leon-Garcia, A., Widjaja, I.: Communication Networks, McGraw-Hill, Boston, 2nd ed. 2004.
- Flaig, G., u.a.: Internet-Telefonie, Open source Press, München, 2006.

An up-to-date list is provided by the course.

Examination
Coursework Assignment and Colloquium / Duration of Examination: 30 minutes
Duration of Coursework: 4 months
Description:
The evaluation of the course will take place after completion of all lectures within the examination cycle. It is based on following items:
- assessment of the chapters composed by the candidate in the final course report about all workpackages written by a team of students
- presentation and explanation of specific tasks and outcomes of laboratories by an individual colloquium lasting 30 minutes

The evaluation rules of these components will be announced during the first lecture. The overall individual grading has to reach the level "satisfactory/ ausreichend (4.0)" to pass the examination of the module.

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

Module KTR-MAKV-M Modeling and Analysis of Communication Networks and Distributed Systems *Modellierung und Analyse von Kommunikationsnetzen und Verteilten Systemen*	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium

(since WS17/18)
Person responsible for module: Prof. Dr. Udo Krieger

Contents:

The course deals with the analysis and performance evaluation of complex distributed systems such as telecommunication systems, computer networks and complex networks as well as cloud computing systems. The latter are transformed to abstract system-theoretical models and their associated parameters. The models are used to analyze the system behavior, and to predict relevant performance metrics such as utilization, throughput, waiting and response times of request, person or data flows in distributed systems or social networks. Such predictions have great importance regarding economic or technical design and decision processes in future generation networks and their distributed service architectures.

The course presents the modeling of distributed systems and discusses associated description methods such as relevant load and machine models. The system-theorectical analysis of these models and the included resource assignment and management strategies are sketched based on simple analytic methods like Markov chains, algebraic and numerical solution methods for queueing models.

Learning outcomes:

It is the objective of the course to teach students the fundamentals of measurement, analysis, and performance evaluation methods in modern computer and communication networks, and distributed systems. Students will learn how they can apply the underlying system-theoretical monitoring, modeling, and analysis techniques to a given technical context. The application of the sketched models and methods is illustrated by exercises covering views of distributed systems with a realistic characteristic. Students are encouraged to apply a given methodology to new technical contexts and scientific tasks.

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:**
• solid knowledge of calculus (like Mathematik I) and linear algebra (like Mathematik für Informatik 2) • basic knowledge of probability theory and statistics • programming experience in JAVA (or C++)	governed by examination regulations (StuFPO)

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

Module Units	
Modeling and Analysis of Communication Networks and Distributed Systems **Mode of Delivery:** Lectures and Practicals **Lecturers:** Prof. Dr. Udo Krieger	**4,00 Weekly Contact Hours**

Language: English/German
Frequency: every summer semester

Learning outcome:

It is the objective of the course to teach students the fundamentals of measurement, analysis, and performance evaluation methods in modern computer and communication networks, and distributed systems. Students will learn how they can apply the underlying system-theoretical monitoring, modeling, and analysis techniques to a given technical context. The application of the sketched models and methods is illustrated by exercises covering views of distributed systems with a realistic characteristic. Students are encouraged to apply a given methodology to new technical contexts and scientific tasks.

Contents:

The course deals with the analysis and performance evaluation of complex distributed systems such as telecommunication systems, computer networks and complex networks. The latter are transformed to abstract system-theoretical models and their associated parameters. The models are used to analyze the system behavior, and to predict relevant performance metrics such as utilization, throughput, waiting and response times of request, person or data flows in distributed systems or social networks. Such predictions have great importance regarding economic or technical design and decision processes in future generation networks and their distributed service architectures.

The course presents the modeling of distributed systems and discusses associated description methods such as relevant load and machine models. The system-theorectical analysis of these models and the included resource assignment and management strategies are sketched based on simple analytic methods like Markov chains, algebraic and numerical solution methods for queueing models, and simulative analysis schemes.

The content of the lectures is illustrated by exercises and laboratories covering important performance aspects in high-speed networks and distributed systems. Knowledge and skills to perform an efficient system analysis, system monitoring, and performance evaluation will be trained in this manner. 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 language of the course wil be announced during the first lecture.

Literature:

- G. Bolch, S. Greiner, H. de Meer, K. S. Trivedi: Queueing Networks and Markov Chains. Wiley, 2nd ed., 2006.
- R. Nelson: Probability, Stochastic Processes, and Queueing Theory. Springer, 1995.

A list of further references is presented in the first lecture.

Examination
Oral examination / Duration of Examination: 30 minutes
Description:

30 minutes oral examination related to the technical topics of all lectures and practicals. The language of the examination wil be announced during the first lecture.	

Module KTR-MMK-M Multimedia Communication in High Speed Networks

Multimedia-Kommunikation in Hochgeschwindigkeitsnetzen

6 ECTS / 180 h
45 h Präsenzzeit
135 h Selbststudium

(since WS17/18)
Person responsible for module: Prof. Dr. Udo Krieger

Contents:

Based on the foundations of data communication, this advanced course of the master program presents the design of high-speed networks (HSN) and the advanced protocol elements of the signaling and user plane that are required to implement new real-time and multimedia services. It includes the digital switching technologies and protocol stacks of HSNs, the quality-of-service architectures, as well as the traffic management protocols of these next generation IP networks. The extension of the TCP/IP protocol stack to realize communication relations among mobile or stationary end systems that are supported by quality-of-service guarantees and associated improved switching concepts are discussed in detail by lectures of the course.

These lectures focus on effective access technologies and new transport and QoS-architectures in the core network like Diffserv, MPLS and GMPLS. Further, enhancement of IPv4 by IPv6 switching and the extension of TCP by moern multipath concepts such as MPTCP and SCTP are presented. Advanced QoS-management concepts, effective resource and traffic management schemes like buffer management by RED, RIO or schedluing by WFQ, are discuessed, too. Furthermore, we present new architectures for next generation networks (NGNs) such as software-defined networks and information-centric networks.

Modern multimedia service architectures with interactive applications for third to fourth generation Internet like Web applications based on HTTP 2.0, WebRTC, peer-to-peer VoIP and media streaming applications are sketched.

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 thesis on related topics in next generation networks.

Learning outcomes:

The students will be enabled to work independently according to the highest scientific standards on design and analysis tasks associated with high-speed network protocols. They will learn about the fundamentals of multimedia communication in high-speed networks and the systematic analysis of the applied communication algorithms by means of an interactive tutorial concept. They will assess the implementations of existing network protocols and to evaluate their performance by means of a measurement analysis with Wireshark and other tools. The processing of the design, assessment, measurement, and implementation tasks will be performed by teams of students. Thus, learning effective teamwork is part of the course.

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:**
	governed by examination regulations (StuFPO)

• successful examination in data communication similar to module KTR-Datkomm-B and substantial knowledge of related technical concepts • knowledge in progamming with JAVA (or C++) Module Advanced Java Programming (DSG-AJP-B) - recommended Module Data communication (KTR-Datkomm-B) - recommended		
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Multimedia Communication in High Speed Networks **Mode of Delivery:** Lectures and Practicals **Lecturers:** Prof. Dr. Udo Krieger **Language:** English/German **Frequency:** every summer semester	**4,00 Weekly Contact Hours**
Learning outcome: The students will be enabled to work independently according to the highest scientific standards on design and analysis tasks associated with high-speed network protocols. They will learn about the fundamentals of multimedia communication in high-speed networks and the systematic analysis of the applied communication algorithms by means of an interactive tutorial concept. They will assess the implementations of existing network protocols and to evaluate their performance by means of a measurement analysis with Wireshark and other tools. The processing of the design, assessment, measurement, and implementation tasks will be performed by teams of students. Thus, learning effective teamwork is part of the course.	
Contents: Based on the foundations of data communication, this advanced course of the masters programme presents the design of high-speed networks (HSN) and the advanced protocol elements of the signaling and user plane that are required to implement new real-time and multimedia services. It includes the digital switching technologies and protocol stacks of HSNs, the quality-of-service architectures, as well as the traffic management protocols of these next generation IP networks. The extension of the TCP/IP protocol stack to realize communication relations among mobile or stationary end systems that are supported by quality-of-service guarantees and associated improved switching concepts are discussed in detail by lectures of the course. These lectures focus on effective access technologies and new transport and QoS-architectures in the core network like Diffserv, MPLS and GMPLS. Further, the enhancement of IPv4 by IPv6 switching and the extension of TCP by modern multipath concepts such as MPTCP and SCTP are presented. Advanced QoS-management concepts, effective resource and traffic management schemes like buffer management by RED, RIO or schedluing by weighted fair queueing (WFQ), are discussed, too. Furthermore, we present new architectures for next generation networks (NGNs) such as software-defined networks and information-centric networks.	

Modern multimedia service architectures with interactive applications for third to fourth generation Internet like Web applications based on HTTP 2.0, WebRTC, peer-to-peer VoIP and media streaming applications are sketched.

The content of the lectures is illustrated by exercises and laboratories covering important aspects of the protocol stacks in high-speed 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:

- Kurose, J., Ross, K.W.: Computernetzwerke – ein Top-Down-Ansatz mit Schwerpunkt Internet, Pearson Studium, München, 2013.
- Kurose, J.F., Ross, K.W.: Computer Networking, A Top-Down Approach Featuring the Internet, Pearson Addison-Wesley, 7th ed., 2017.
- Leon-Garcia, A., Widjaja, I.: Communication Networks, McGraw-Hill, Boston, 2nd ed. 2004.
- Comer, D.: Computernetzwerke und Internets, Pearson Studium, München, 2001.

Weitere Literatur wird in der Vorlesung benannt.

Examination

Oral examination / Duration of Examination: 30 minutes

Description:

30 minutes oral examination related to the technical topics of all lectures and practicals.

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

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:
- 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

Admission requirements:
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 **Mode of Delivery:** Lectures and Practicals **Lecturers:** Prof. Dr. Udo Krieger **Language:** English/German **Frequency:** every winter semester	**4,00 Weekly Contact Hours**

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 KTR-SSSProj-M KTR Master Project Software Systems Science

KTR Masterprojekt Software Systems Science

9 ECTS / 270 h
70 h Präsenzzeit
200 h Selbststudium

(since SS24)
Person responsible for module: Prof. Dr. Udo Krieger

Contents:
Important skills regarding the planning, development and implementation of new commmunication technologies, their advanced services, and the related protocols in next generation networks can only be learnt by team oriented development projects subject to stringent time and resource contraints, and clear development objectives, similar to an industrial project environment. After a short training phase and based on an autonomous working mode, students will learn by a teamwork project to solve advanced communication tasks and to implement new communication services associated with current research issues of the professorship. Actual topics will be announced on the web page of the module.

Learning outcomes:
The students are encouraged to independent scientific work. They learn how to plan, develop and implement new advanced multimedia services and communication protocols in next generation networks. They are trained to efficiently implement the applied protocols and to analyze the performance of the communication algorithms in a systematic manner. Students are instructed to investigate their developed protocol code elements by measurements and other tools, to evaluate their performance, and to develop improved protocol units. The processing of design, programming, and performance assessment tasks by teams of students and the effective arrangement of the groupwork is part of the training.

The project follows scientific standards and deals with research issues of the professorship. The overall objective is to develop skills and knowledge required for a successful career in industry or research in the field of communication engineering.

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

prerequisites for the module:
A bachelor degree in computer science, computer engineering or mathematics is required. Students must be enroled in the masters degree programme "M.Sc. International Software Systems Science".

Recommended prior knowledge:

- good knowledge in mathematics and statistics, similar to module Mathematik für Informatiker 2
- good programming skills in JAVA (or C++)
- good knowledge in data communication, similar to module KTR-Datkomm-B
- solid methodological know-how in planning and execution of software projects, similar to the module "Software Engineering Lab" (SWT-SWL-B)

Module Introduction to Parallel and Distributed Programming (DSG-PKS-B) - recommended
Module Data communication (KTR-Datkomm-B) - recommended

Admission requirements:
governed by examination regulations (StuFPO)

Module Mathematics for Computer Science 2 (Linear Algebra) (KTR-MfI-2) - recommended Module Software Engineering Lab (SWT-SWL-B) - recommended	

Frequency: every semester	**Recommended semester:** 2.	**Minimal Duration of the Module:** 1 Semester

Module Units	
KTR Master Project Software Systems Science **Mode of Delivery:** **Lecturers:** Prof. Dr. Udo Krieger **Language:** English/German **Frequency:** every semester **Learning outcome:** The details are sketched previously. **Contents:** Important skills regarding the planning, development and implementation of new commmunication technologies, their advanced services, and the related protocols in next generation networks can only be learnt by team oriented development projects subject to stringent time and resource contraints, and clear development objectives, similar to an industrial project environment. After a short training phase and based on an autonomous working mode, students will learn by a teamwork project to solve advanced communication tasks and to implement new communication services associated with current research issues of the professorship. The organization of the project is following the framework of industry. It comprises definition, preparation, implementation and presentation phases. An incremental processing is performed like in industrial projects. It means • a segmentation into specific work packages, • its division into tasks and subtasks including milestones • the presentation of intermediate results • a final report with presentation and an individual colloquium to defend the outcome. Research and development tasks are related to current research issues in "Future Generation Internet" and will be integrated into the module. An actual list of studied topics and related references are presented in the first lecture. The language of the course wil be announced during the first lecture. **Literature:** A reference list will be provided in the first meeting of the project.	**6,00 Weekly Contact Hours**
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:**	

The course duration is one semester. The assessment of the module covers the results of the project report, written either as groupwork or on an individual basis by the student, the project presentation, and the final colloquium arranged on an individual basis. The language of the course and its examination is announced during the first lecture.	

Module KTR-Sem-M Master Seminar Communication Systems and Computer Networks

Hauptseminar zu Kommunikationssystemen und Rechnernetzen

3 ECTS / 90 h

(since SS20)
Person responsible for module: Prof. Dr. Udo Krieger

Contents:
The seminar will discuss hot topics in the fields of stationary and mobile communication networks, new Internet services as well as fog and cloud computing architectures. The development of powerful transport and edge computing platforms for future generation software-defined networks supporting quality-of-service and mobility requirements will constitute a technical focus of the seminar.

Learning outcomes:
A major competence objective is given by the ability to evaluate the scientific literature in a critical manner and to apply new scientific results while solving a technical problem at hand. We shall improve the ability to adopt effectively the new technical methodologies stemming from the fields of software-defined communication networks, the theory of distributed systems, and the foundations of computer science.

Remark:
The workload comprises the following components:

- personal presence phases including topic dissemination and discussions witht the lecturers: 20 hours
- preparation of the technical topic and writing of the report: 54 hours
- preparation of the oral presentation: 16 hours

prerequisites for the module:

- knowledge on topics of the module Foundations of Intenet Communication (KTR-GIK-M)

Module Foundations of Internet Communication (KTR-GIK-M) - Pflicht

Recommended prior knowledge:

- basic knowledge on the principles of data communication
- additional knowledge according to the technical specification of the offered seminar

Module Data communication (KTR-Datkomm-B) - recommended

Admission requirements:
none

Frequency: winter or summer semester, on demand

Recommended semester:

Minimal Duration of the Module:
1 Semester

Module Units

Seminar KTR-Master
Mode of Delivery: Advanced seminar
Lecturers: Prof. Dr. Udo Krieger
Language: English/German
Frequency: winter and summer semester, on demand

2,00 Weekly Contact Hours

Learning outcome:
The students will prepare the writing of a master's thesis and their industrial or scientific employment. A major competence objective is given by the ability to evaluate the scientific literature in a critical manner and to apply new scientific results while solving a technical problem at hand.

Contents:
The seminar will discuss hot topics in the fields of stationary and mobile communication networks, new Internet services as well as fog and cloud computing architectures. The development of powerful transport and edge computing platforms for future generation software-defined networks supporting quality-of-service and mobility requirements will constitute the technical focus of the seminar.

The seminar offers a student the perspectives on the system-theoretical foundations of actual technical topics arising in the rapidly evolving areas of modern communication and fog/cloud computing systems. It is the objective of study to independently adopt the new technical methodologies stemming from the fields of software-defined communication networks, the theory of distributed systems, and the foundations of computer science.

Passing the examination of the seminar is, in general, a prerequisite to successfully write a master's thesis at the Professorship of Computer Science or in cooperation with industrial peers.

The used language of the module will be announced during the first session of the seminar.

Literature:
The relevant reference list will be announced during the first session.

Examination
Coursework Assignment with presentation / Duration of Examination: 40 minutes
Duration of Coursework: 4 months
prerequisites for module examination:
Regelmäßige Teilnahme an der Lehrveranstaltung
Description:
The final grade evaluates the written report (- this phase lasts at most 4 months -) and the oral presentation as equally weighted components. Both the report and oral presentation have to achieved at least the grade 4.0 to pass the examination.

The language of the examination will be announced during the first session of the seminar.

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 **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	**2,00 Weekly Contact Hours**
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-PRS-M Master Project Mobile Software Systems (SoSySc) *Master Project Mobile Software Systems (SoSySc)*	9 ECTS / 270 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-M 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 and 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: Basic programming skills (Java or Python); scientific writing skills, e.g., obtained from the course SSS-SRW-M or from a scientific Bachelor thesis; basic knowledge in Mobile Computing as offered, e.g., by the course MOBI-MSS-B. Dependent on the topic of the specific project, additional knowledge as discussed in the courses MOBI-DSC-M or MOBI-ADM-M can be required.	**Admission requirements:** none

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

Module Units

Master Project Mobile Software Systems (SoSySc) **Mode of Delivery:** **Lecturers:** Prof. Dr. Daniela Nicklas	**6,00 Weekly Contact Hours**

Language: English/German **Frequency:** every summer semester	
Contents: Conduct of the project, accompanied by regular meetings between students and lecturer. 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:** Als Prüfungsleistung ist eine Hausarbeit sowie ein Kolloquium zu erbringen. Die Bearbeitungsfrist der Hausarbeit und die Prüfungsdauer des Kolloquiums werden zu Beginn einer jeden Lehrveranstaltung von der Projektleiterin bzw. dem Projektleiter bekannt gegeben. Production of a written report on the software project carried out (Assignment/ Hausarbeit). Discussion of this project report and of the developed artefacts in the context of the wider project topic (Colloquium/Kolloquium). The term of the project report and of the colloquium will be announced at the beginning of each course by the project leader.	

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

2,00 Weekly Contact Hours

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. 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-ProjectCAD-M Project Complex Attacks and Defenses

Project Complex Attacks and Defenses

9 ECTS / 270 h

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

Contents:

Breaking into information systems is exciting, but impractical due to ethical and legal concerns. However, offensive competences and adversarial thinking are essential to build secure systems. In this project students will get the opportunity to acquire practical security skills in a dedicated training environment.

The goal of this project is to build and extend the "Insekta" platform. This web-based tool provides a frontend for virtual machines that can be used to study selected topics in security and privacy on one's own and at one's own pace.

This project is offered together with PSI-ProjectPAD, which focuses on conceptually simpler attacks and defenses.

The participants of the project familiarize themselves with security weaknesses in information systems and apply this knowledge to develop vulnerable services which others can use for training. To this end, participants form groups, read about attacks and defenses in textbooks and research papers, and discuss various options to implement them. Instructors will provide extensive and on-demand support to enable the participants to implement a vulnerable service that can be exploited to learn about a particular vulnerability.

Besides implementing vulnerable services, the participants prepare training materials, which consist of questions and tasks to test one's knowledge as well as step-by-step instructions. These training materials may also contain interactive elements for an improved learning experience.

The project also takes into account attacks on privacy, e.g., re-identifying individuals in anonymized datasets and communication networks, tracking users on the Internet, inferring sensitive attributes from seemingly harmless data traces, as well as mitigations, e.g., depersonalization strategies and differential privacy mechanisms. Here, practical activities consist in the preparation of datasets and scripts for analysis.

Learning outcomes:

Successful students will be able to describe attacks and defenses from textbooks and research papers in easily understandable form. They will also be able to carry out selected attacks in practice and implement defenses with a programming language of their choice.

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:

- 20 hrs: Getting familiar with the platform
- 50 hrs: Reading papers and researching security vulnerabilities
- 20 hrs: Preparing the talk (including time for attendance of other talks)
- 90 hrs: Implementing the vulnerable service and defenses
- 90 hrs: Writing training material and documentation

Note that there is another project (PSI-ProjectPAD) with a workload equivalent to 180 hours.

prerequisites for the module: none		
Recommended prior knowledge: This project is primarily intended for students in master programs. Students in bachelor programs can participate, if they are qualified. 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). 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. Experience with Linux environments, web technologies, and network protocols is recommended.		**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
Project Complex Attacks and Defenses **Mode of Delivery:** **Language:** English/German **Frequency:** every semester	**6,00 Weekly Contact Hours**

Learning outcome:
cf. module description

Contents:
Potential topics include:

- web security (injection flaws and other issues mentioned in the OWASP Top 10)
- network security (such as DNS cache poisoning and rebinding attacks)
- security issues in C programs (buffer overflows, etc.)
- cryptography (low-level attacks on ciphers, high-level attacks on protocols, e.g., TLS)
- business logic failures
- misconfigurations
- attacks on availability (denial of service)
- attacks on privacy (such as inference, tracking, re-identification, fingerprinting)
- privacy defenses (such as k-anonymity, related concepts, differential privacy)

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 of the vulnerable service and the training material. 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 theoretical and practical aspects of their vulnerable service as well as relevant mitigations. 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-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 SS24)
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, reviewing the work of others, writing a draft of the term paper, reviewing the draft of other students, etc.). They will receive feedback by their peers and by the instructors.

The actual topics are subject to change. A list of available topics is made available before the first session via UnivIS or VC.

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.

Students who participate in the optional peer review process will also learn techniques to give useful feedback to others as well as how to accept feedback for one's own work.

Remark:
The default language in this seminar is English, unless all participants are fluent in German.

prerequisites for the module:
none

Recommended prior knowledge:	**Admission requirements:**
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.	none

Frequency: every winter 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 **Contents:**	**2,00 Weekly Contact Hours**

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: 3 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 submit intermediary results (in English) such as surveys, written reviews for the work of other participants, and a draft of the term paper. 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 WS19/20) 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: none		**Admission requirements:** none
Frequency: every semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Examination Praktikumsbericht, unbenotet **Description:** at least 4 pages	

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 WS19/20)
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:	**Admission requirements:**
Basic knowledge in algorithms and data structures, mathematical logic and theoretical computer science. Knowledge of the module "Foundations of Software Analysis" (SWT-FSA-B) - or equivalent - is desirable.	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 **Contents:** The lectures (Vorlesungen) will address the following topics in automated software verification: (i) state machines, assertions and algorithms for state	**2,00 Weekly Contact Hours**

space exploration; (ii) temporal logics for specifying program properties; (iii) model checking using binary decision diagrams; (iv) SAT-based bounded model checking; (v) software model checking based on decision procedures; (vi) abstraction-based software model checking. In addition, several 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

2,00 Weekly Contact Hours

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 -

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-PCC-M Principles of Compiler Construction *Principles of Compiler Construction*	6 ECTS / 180 h

(since WS20/21)
Person responsible for module: Prof. Dr. Gerald Lüttgen

Contents:
The module teaches the theoretical and practical principles of compiler construction, from lexical analysis and parsing, to semantic analysis, to code generation and optimisation.

Learning outcomes:
On completion of this module, students will be familiar with all phases of a modern compiler – from lexical analysis and parsing, to semantic analysis and finally code generation and code optimisation – and will have a deep understanding of the workings of compilers. As a result, students will be able to use compilers more effectively and learn better debugging practices. Students will also be able to start building compilers on their own.

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. reviewing the lectures, including researching and studying material from additional sources
- 30 hrs. attending practicals (Übungen)
- 30 hrs. preparing and reviewing the practicals, including researching and studying material from additional sources
- 60 hrs. working on the assignment (Hausarbeit) and preparing for the colloquium (Kolloquium)

prerequisites for the module:
none

Recommended prior knowledge:	**Admission requirements:**
Basic knowledge in programming languages, in the theoretical foundations of Computer Science (especially in language theory and automata theory) and in algorithms and data structures.	none

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

Module Units

1. Principles of Compiler Construction **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Gerald Lüttgen **Language:** English/German **Frequency:** every summer semester	**2,00 Weekly Contact Hours**

Contents:
Students will be familiarised with a variety of theoretical and practical concepts, techniques and algorithms employed in compiler construction, which reach from language theory, to automata theory, and to data flow analysis. The lectures will

focus on the following aspects of compiler construction: lexical analysis, parsing, abstract syntax, semantic analysis, code generation and code optimisation.

Literature:

- Louden, K. C. Compiler Construction: Principles and Practice. Course Technology, 1997.
- Aho, A. V., Lam, M. S., Sethi, R. and Ullman, J. D. Compilers: Principles, Techniques, and Tools, 2nd ed. Pearson, 2007.
- Fischer, C. N., Cytron, R. K. and LeBlanc Jr., R. J. Crafting a Compiler. Pearson, 2010.
- Muchnick, S. S. Advanced Compiler Design and Implementation, Morgan Kaufmann, 1997.

2. Principles of Compiler Construction

2,00 Weekly Contact Hours

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

Contents:
Students will practice the theoretical concepts taught in the lectures by applying them to a variety of exercises, so that they can appreciate the diverse range of foundations that make modern programming languages possible. The exercises will largely be pen-and-paper exercises but may also involve some work using computers. Emphasis will be put on presenting and discussing the solutions to the exercises by and among the students, within the timetabled practicals (Übungen). Students can gain further practical experience in compiler construction by attending one of the modules "Masterprojekt Softwaretechnik und Programmiersprachen" (SWT-PR1-M) or "Masters Project in Software Systems Science" (SWT-PR2-M).

Literature:
- see the corresponding lectures -

Examination
Coursework Assignment and Colloquium / Duration of Examination: 20 minutes
Duration of Coursework: 3 weeks

Description:
Assignment (Hausarbeit) consisting of questions practicing, reviewing and deepening the knowledge transferred in the lectures (Vorlesungen) and practicals (Übungen). The examination 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 (Vorlesungen) and practicals (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-PR2-M SWT Masters Project in Software Systems Science *SWT Masters Project in Software Systems Science*	9 ECTS / 270 h

(since WS19/20)
Person responsible for module: Prof. Dr. Gerald Lüttgen

Contents:

Attention: The module SWT-PR2-M cannot take place in winter semester 2019/20!

Topics in Software Systems Science 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. Another example would be designing and implementing a compiler of a small programming language in either an imperative, object-oriented or functional language, which requires the prior attendance of the module "Principles of Compiler Construction" (SWT-PCC-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 270 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
- 195 hrs. researching and familiarization with the project topic and conducting the project work
- 45 hrs. compiling a project report (Assignment/Hausarbeit) and preparation of the Colloquium (Kolloquium).

The project report can be written/composed in either English or German.

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

Module Units	
SWT Masters Project in Software Systems Science **Mode of Delivery:** Practicals **Lecturers:** Prof. Dr. Gerald Lüttgen **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.	**6,00 Weekly Contact Hours**

Examination Coursework Assignment and Colloquium / Duration of Examination: 30 minutes Duration of Coursework: 12 weeks **prerequisites for module examination:** Regular participation in the practicals. **Description:** Production of a written report on the software project carried out (Assignment/ Hausarbeit). Discussion of this project report and of the developed artefacts in the context of the wider project topic (Colloquium/Kolloquium).	

Module SWT-SEM-M Seminar in Software Engineering and Programming Languages (Master) *Seminar Software Engineering and Programming Languages (Master)*	3 ECTS / 90 h

(since WS17/18)
Person responsible for module: Prof. Dr. Gerald Lüttgen

Contents:
Current topics in software engineering and programming languages. This may comprise the full spectrum of research topics in these fields, from the analysis, comparison and evaluation of current software technologies and tools, to the discussion and evaluation of novel research proposals.

Learning outcomes:
Students will compile and acquire current topics in software engineering and programming languages 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 software engineering and programming languages with their peers.

Remark:
The main language of instruction is English. The seminar may be delivered in German if all participating students are fluent in German. Regular participation in the presentations is required.

The total workload of 90 hrs. is split approximately as follows:

- 20 hrs. consultations and presentations (Referate), including discussions
- 25 hrs. literature research and familiarization and evaluation of literature
- 45 hrs. working on the assignment (Hausarbeit) and preparation for the presentation (Referat)

prerequisites for the module:
none

Recommended prior knowledge: Basic knowledge in software engineering, in programming languages 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	
Software Engineering and Programming Languages (Master) **Mode of Delivery:** Seminar **Lecturers:** Prof. Dr. Gerald Lüttgen, Scientific Staff Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen **Language:** English/German **Frequency:** every semester **Contents:** Various current topics in software engineering and programming languages, which complement and/or extend the technical and methodological aspects of the degree programme's modules related to these fields. **Literature:** Will be allocated according to the topics to be discussed.	**2,00 Weekly Contact Hours**

Examination Coursework Assignment with presentation / Duration of Examination: 40 minutes Duration of Coursework: 8 weeks **prerequisites for module examination:** Regular participation in the seminar. **Description:** Assignment (Hausarbeit) consisting of a written report on the topic assigned to the student. Presentation (Referat) on the topic assigned to the student, including a discussion.	

Module SWT-SWQ-M Software Quality

Software Quality

6 ECTS / 180 h

(since WS21/22)
Person responsible for module: Prof. Dr. Gerald Lüttgen

Contents:
Software quality is fundamental for a software product's reliable, safe and secure operation, for its maintainability and reusability, and for user and customer satisfaction. Engineering high-quality software products and managing their development involves the application of advanced techniques, methods and tools for software quality assurance. This module focuses, in particular, on model-based testing, software inspection, software measurement, and static analysis, which are indispensable in today's agile software engineering practice.

Learning outcomes:
On completion of this module, students will be familiar with important concepts and techniques of software quality and their role in modern software engineering. In particular, students will be able to apply state-of-the-art methods and tools for achieving and monitoring software quality, and devise strategies for software quality assurance in different product and organizational contexts.

Remark:
The language of instruction is English.

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

- 30 hrs. attending lectures (Vorlesungen)
- 30 hrs. attending practicals (Übungen)
- 90 hrs. preparing and reviewing lectures and practicals, researching literature and studying material from additional sources
- 30 hrs. preparing for the written exam (Klausur)

prerequisites for the module:
none

Recommended prior knowledge: Basic knowledge in Software Engineering, such as gained, e.g., in the module "Foundations of Software Engineering" (SWT-FSE-B). In particular, good knowledge of the Unified Modeling Language (UML) is expected.	**Admission requirements:** none

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

Module Units

1. Software Quality **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Gerald Lüttgen, Alexander Kraas **Language:** English **Frequency:** every winter semester **Learning outcome:** – see the module's learning outcomes/competences (Lernziele/Kompetenzen) listed above –	**2,00 Weekly Contact Hours**

Contents:

The following topics will be covered in this module:

- Software quality within agile software engineering
- Fundamental testing concepts and techniques
- Automated, model-based testing
- Inspections and reviews
- Software measurement
- Static analysis
- Software quality management

Literature:

- Goericke, S. (editor). The Future of Software Quality Assurance. Springer, 2020.
- Kramer, A. and Legeard, B. Model-Based Testing Essentials. Wiley, 2016.
- Meyers, G. J. et al. The Art of Software Testing, 3rd ed. Wiley, 2012.
- O’Reagan, G. Concise Guide to Software Testing. Springer, 2019.
- O’Reagan, G. Introduction to Software Quality. Springer, 2014.
- Utting, M. and Legeard, B. Practical Model-Based Testing – A Tools Approach. Morgan Kaufmann, 2007.
- Walkinshow, N. Software Quality Assurance. Springer, 2017.

2. Software Quality

2,00 Weekly Contact Hours

Mode of Delivery: Practicals

Lecturers: Scientific Staff Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen

Language: English

Frequency: every winter semester

Learning outcome:

– see the module's learning outcomes/competences (Lernziele/Kompetenzen) listed above –

Literature:

– see the corresponding lectures –

Examination

Written examination / Duration of Examination: 90 minutes

Description:

Written exam (Klausur) consisting of questions that relate to the contents of the lectures (Vorlesungen) and practicals (Übungen) of this module.

The exam is passed if at least 50% of the available points are reached.

Module SYSNAP-OSE-M Operating Systems Engineering *Operating Systems Engineering*	6 ECTS / 180 h
(since SS22) 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 "Einführung in Rechner- und Betriebssysteme" (PSI-EiRBS-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 **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Michael Engel **Language:** German/English **Frequency:** every summer semester	**2,00 Weekly Contact Hours**
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 **Mode of Delivery:** **Lecturers:** Prof. Dr. Michael Engel **Language:** German/English **Frequency:** every summer semester	**2,00 Weekly Contact Hours**
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-PMAP-M Processor Microarchitecture and Performance *Processor Microarchitecture and Performance*	6 ECTS / 180 h
Person responsible for module: Prof. Dr. Michael Engel further responsible : Werner Haas	

Contents:

Modern computer systems include high-performance processors which enable computationally demanding applications such as video and audio processing, handling of big data amounts or deep neural networks. Exploiting this performance potential for modern applications, however, is difficult, since increased performance levels could only be achieved by introducing additional complexity into the architecture of computer systems – for example, multiprocessor and multicore systems, multi-level memory hierarchies, and memory models with relaxed consistency.

This course gives an insight into architectural details of modern processor architecture and their impact on non-functional properties. Whereas performance is the central topic of the course, additional non-functional properties such as energy consumption and security will be discussed. In addition to gaining theoretical insight into modern features of processor and system architecture, the course also discusses the interaction of software and hardware and how to optimize software for given architectural features.

Learning outcomes:

The module is designed to enable students to not only understand the internals of modern microprocessors and computer systems, but also learn about the non-functional properties involved and how the interaction between hardware and software relates to these. Starting with an overview of contemporary processors, this module will present and discuss different performance-improving aspects of processor architectures and their impact on software.

Successful students will develop an understanding of modern processor architectures and the related systems as well as the resulting non-functional properties. They can comprehend and critically analyze existing and proposed new approaches from the literature. By writing code and analyzing the impact of different architectural features on the software, students will be able to gain a better understanding of the operation of hardware and its interaction with software and be able to optimize software for a given architecture and memory hierarchy.

prerequisites for the module:

verpflichtende Nachweise de

Recommended prior knowledge: Fundamentals of computer architecture and operating systems, e.g. module PSI-EiRBS-B Operating Systems Engineering (SYSNAP-OSE-M) and/or Virtualization (SYSNAP-Virt-M)		**Admission requirements:** none
Frequency: every summer semester	**Recommended semester:**	**Minimal Duration of the Module:** 1 Semester

Module Units	
1. Lecture Processor Microarchitecture and Performance **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Michael Engel	**2,00 Weekly Contact Hours**

Language: English/German **Frequency:** every summer semester **Learning outcome:** cf. module description **Contents:** 1 Intro/Recap: stored program arch, ISA, abstraction, iron law of performance 2 Simple pipelining: pipeline hazards, superscalar processing, exception handling 3 Caches: direct mapped, set/fully associative, memory hierarchy 4 Virtual memory: segmentation, paging, TLB, aliases/synonyms, VP/PP caches 5/6 Out of order execution – register renaming, Tomasulo algorithm – memory disambiguation, load/store queues 7/8 Branch prediction – branch history – branch targets 9 Symmetric multiprocessing: sequential consistency, cache coherence protocols 10 Virtualisation: processor modes, sensitive instructions, multi-level translation 11 Side channels: cache state, timing sources, resource contention 12 Transient execution attacks: Meltdown, Spectre, Retpoline **Literature:** John L. Hennessy, David A. Patterson Computer Architecture: A Quantitative Approach Morgan Kaufmann, 6th Edition 2017 ISBN-13: 978-0128119051 John Paul Shen, Mikko H. Lipasti Modern Processor Design: Fundamentals of Superscalar Processors Waveland Pr Inc, Reprint Edition 2013 ISBN-13: 978-1478607830	
2. Exercises Processor Microarchitecture and Performance **Mode of Delivery:** **Lecturers:** Prof. Dr. Michael Engel **Language:** English/German **Frequency:** every summer semester **Learning outcome:** cf. module description **Contents:** cf. module description **Literature:** cf. module description	**2,00 Weekly Contact Hours**
Examination Portfolio / Duration of Coursework: 3 months	

<table>
<tr><td colspan="2">Module SYSNAP-Project-M Project Systems Programming
<i>Projekt Systemnahe Programmierung</i></td><td>6 ECTS / 180 h</td></tr>
<tr><td colspan="3">(since SS24)
Person responsible for module: Prof. Dr. Michael Engel</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td colspan="3">prerequisites for the module:
none</td></tr>
<tr><td colspan="2">Recommended prior knowledge:
Modules SYSNAP-OSE and/or SYSNAP-Virt</td><td>Admission requirements:
none</td></tr>
<tr><td>Frequency: every semester</td><td>Recommended semester:</td><td>Minimal Duration of the Module:
1 Semester</td></tr>
</table>

<table>
<tr><td colspan="2">Module Units</td></tr>
<tr><td>Projekt Systemnahe Programmierung
Mode of Delivery:
Lecturers: Prof. Dr. Michael Engel
Language: German/English
Frequency: every semester
Learning outcome:
see module description
Contents:
see module description</td><td>4,00 Weekly Contact Hours</td></tr>
</table>

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.	

<table>
<tr><td colspan="2">Module SYSNAP-SEM-M Seminar System Software
<i>Seminar System Software</i></td><td>3 ECTS / 90 h</td></tr>
<tr><td colspan="3">(since SS24)
Person responsible for module: Prof. Dr. Michael Engel</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td colspan="3">prerequisites for the module:
none</td></tr>
<tr><td colspan="2">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.</td><td>Admission requirements:
none</td></tr>
<tr><td>Frequency: every semester</td><td>Recommended semester:</td><td>Minimal Duration of the Module:
1 Semester</td></tr>
</table>

<table>
<tr><td colspan="2">Module Units</td></tr>
<tr><td>Seminar
Mode of Delivery: Seminar
Lecturers: Prof. Dr. Michael Engel
Language: German/English
Frequency: every semester
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.</td><td>2,00 Weekly Contact Hours</td></tr>
<tr><td>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.</td><td></td></tr>
</table>

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 WS22/23)
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:	**Admission requirements:**
Participants should be familiar with basic concepts of operating systems and computer architecture, e.g. as acquired by taking the module "Einführung in Rechner- und Betriebssysteme" (PSI-EiRBS-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 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 WS23/24 to SS24) 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 and programming; knowledge in 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 **Contents:** See module description **Literature:**	**2,00 Weekly Contact Hours**

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 **Contents:** In the exercise sessions, lecture contents are expanded upon and their application is practiced.	**2,00 Weekly Contact Hours**

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 **Contents:** See module description **Literature:** Further material and reading will be announced in the course.	**2,00 Weekly Contact Hours**

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 WS23/24)
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-M] Further recommended: Bachelorproject Erklärbares Maschinelles Lernen [xAI-Proj-B], Lernende Systeme / Machine Learning [KogSys-ML-B], Einführung in die Künstliche Intelligenz / Introduction to AI [AI-KI-B], Mathematik für Informatik 2 (Lineare Algebra) [KTR-MfI-2], 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-MML-M Mathematics for Machine Learning *Mathematics for Machine Learning*	6 ECTS / 180 h

(since SS23)
Person responsible for module: Prof. Dr. Christian Ledig

Contents:

The course aims to establish a common mathematical foundation for the further study of advanced machine learning techniques. The content is selected specifically to be most relevant for students interested in machine learning problems and covers a broad range of concepts from, e.g., linear algebra, vector calculus, probability theory, statistics, and optimization.

Learning outcomes:

In this course students will learn fundamental mathematical concepts that are important prerequisites for the deeper understanding of the field of machine learning. The overarching goal of this course is to build a mathematical foundation by selectively covering the most essential mathematical concepts form a broad range of mathematical disciplines. Dependent on previous background, students will get the chance to learn critical ML-relevant mathematics for the first time or consolidate concepts that have been partially covered in their previous curriculum.

The lecture is accompanied by exercises and assignments that will help participants develop both theoretical and practical experience. In those exercises students will get the opportunity to learn how to apply and prove theoretical concepts as well as implement some concrete algorithms in 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: No specific prior knowledge is required, but the following will be helpful. • Working knowledge of programming (e.g., in Python). • Completion of mathematical courses addressing concepts of linear algebra (e.g., KTR-MfI-2), calculus (e.g., WiMa-B-002), or statistics (e.g., Stat-B).	**Admission requirements:** none

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

Module Units	
1. Mathematics for Machine Learning **Mode of Delivery:** Lectures **Lecturers:** Prof. Dr. Christian Ledig	**2,00 Weekly Contact Hours**

Language: English/German
Frequency: every summer 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

- Linear Algebra (e.g., vector spaces, span, basis, rank)
- Analytic Geometry (e.g., norms, inner product, projections)
- Matrix decompositions (e.g., Eigenvectors, SVD)
- Vector calculus (e.g., derivatives, Taylor series)
- Information Theory (e.g., entropy, KL divergence)
- Probability theory and distributions
- Statistics (e.g., estimators, tests)
- Optimization (e.g., gradient based)
- Machine Learning Problems (e.g., Density estimation, Dimensionality Reduction)

Literature:
- Marc. Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong: Mathematics for Machine Learning, Cambridge University Press, 2020

Further literature will be announced at the beginning of the course.

2. Mathematics for Machine Learning
Mode of Delivery: Practicals
Lecturers: N.N.
Language: English/German
Frequency: every summer semester

2,00 Weekly Contact Hours

Learning outcome:
see module description

Contents:
Further exploration of concepts discussed in the lecture by specific assigments and some programming exercises implemented predominantly in Python.

Literature:
see lecture description

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 **Contents:** see module description	**2,00 Weekly Contact Hours**

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.	

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	Software Systems Science for students starting before WS 19/20		**30 - 48**		
	In module groups A1 and A2, modules totalling 48 ECTS points are to be completed in accordance with the minimum and maximum limits applicable to the module groups. Please note that the module SWT-PCC-B is no longer offered as of the winter semester 2020/21.				
AISE-Auto	Automation of First- and Higher-Order Logic	every summer semester(1)	6	6	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))
AlgoK-Algo	Algorithms	alle 4 Semester(1)	6	4 Lectures and Practicals	Sonstiges
DSG-DSAM-M	Distributed Systems Architectures and Middleware	every winter semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
DSG-DistrSys-M	Distributed Systems	every summer semester(2020)	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
DSG-SOA-M	Service-Oriented Architecture and Web Services	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
DT-CPP-M	Advanced Systems Programming in C++ (Master)	every winter semester(1)	6	6 Lectures and Practicals	Colloquium, Coursework Assignment 4 months

					30 minutes
DT-DBCPU-M	Database Systems for modern CPU	every summer semester(1)	6	6 Lectures and Practicals	Oral examination alone 20 minutes
KTR-GIK-M	Foundations of Internet Communication	every summer semester(on demand also WS)	6	4 Lectures and Practicals	Coursework Assignment and Colloquium 4 months 30 minutes
KTR-MAKV-M	Modeling and Analysis of Communication Networks and Distributed Systems	every summer semester	6	4 Lectures and Practicals	Oral examination 30 minutes
KTR-MMK-M	Multimedia Communication in High Speed Networks	every summer semester	6	4 Lectures and Practicals	Oral examination 30 minutes
KTR-Mobi-M	Mobile Communication	every winter semester	6	4 Lectures and Practicals	Oral examination 30 minutes
MOBI-ADM-M	Advanced Data Management	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 75 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
SWT-ASV-M	Applied Software Verification	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 weeks 20 minutes

SWT-SWQ-M	Software Quality	every winter semester(1)	6	2 Lectures 2 Practicals	Written examination 90 minutes
SYSNAP-OSE-M	Operating Systems Engineering	every summer semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months 30 minutes
SYSNAP-PMAP-M	Processor Microarchitecture and Performance	every summer semester(1)	6	2 Lectures 2	Portfolio 3 months
SYSNAP-Virt-M	Virtualization	every winter semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months 30 minutes

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A1 Software Systems Science for students starting from WS 1920 onwards		**30 - 48**		
	In module groups A1 and A2, modules totalling 48 ECTS credits must be completed in accordance with the minimum and maximum limits applicable to the module groups. Please note that the module SWT-PCC-B is no longer offered as of the winter semester 2020/21. Instead, the module SWT-SWQ-B can be taken and recognised in the compulsory area.				
	Teilmodulgruppe: compulsory part		**24**		
DSG-DSAM-M	Distributed Systems Architectures and Middleware	every winter semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
KTR-GIK-M	Foundations of Internet Communication	every summer semester(on demand also WS)	6	4 Lectures and Practicals	Coursework Assignment and Colloquium 4 months 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
SWT-PCC-M	Principles of Compiler Construction	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 weeks 20 minutes
SWT-SWQ-M	Software Quality	every winter semester(1)	6	2 Lectures 2 Practicals	Written examination 90 minutes
	Teilmodulgruppe: elective modules		**0 - 24**		
AISE-Auto	Automation of First- and Higher-Order Logic		6	6	Oral examination 30 minutes

		every summer semester(1)			
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))
AlgoK-Algo	Algorithms	alle 4 Semester(1)	6	4 Lectures and Practicals	Sonstiges
DSG-DistrSys-M	Distributed Systems	every summer semester(2020)	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
DSG-SOA-M	Service-Oriented Architecture and Web Services	every summer semester	6	2 Lectures 2 Practicals	Coursework Assignment and Colloquium 3 months 15 minutes
DT-CPP-M	Advanced Systems Programming in C++ (Master)	every winter semester(1)	6	6 Lectures and Practicals	Colloquium, Coursework Assignment 4 months 30 minutes
DT-DBCPU-M	Database Systems for modern CPU	every summer semester(1)	6	6 Lectures and Practicals	Oral examination alone 20 minutes
KTR-MAKV-M	Modeling and Analysis of Communication Networks and Distributed Systems	every summer semester	6	4 Lectures and Practicals	Oral examination 30 minutes
KTR-MMK-M	Multimedia Communication in High Speed Networks	every summer semester	6	4 Lectures and Practicals	Oral examination 30 minutes
KTR-Mobi-M	Mobile Communication		6	4 Lectures and Practicals	Oral examination

		every winter semester			30 minutes
MOBI-ADM-M	Advanced Data Management	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 75 minutes
PSI-AdvaSP-M	Advanced Security and Privacy	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 110 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-PMAP-M	Processor Microarchitecture and Performance	every summer semester(1)	6	2 Lectures 2	Portfolio 3 months
SYSNAP-Virt-M	Virtualization	every winter semester(1)	6	2 Lectures 2	Coursework Assignment and Colloquium 3 months 30 minutes

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A2 Domain-specific Software Systems Science		**0 - 18**		
	In module groups A1 and A2, modules totalling 48 ECTS points are to be completed in accordance with the minimum and maximum limits applicable to the module groups.				
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
GdI-CSNL-M	Computational Semantics of Natural Language	every summer semester(1)	6	4	Portfolio 80 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	Written examination 90 minutes Oral examination
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
xAI-MML-M	Mathematics for Machine Learning	every summer semester(1)	6	2 Lectures 2 Practicals	Written examination 90 minutes

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A3 Seminar and Project		**12**		
	Teilmodulgruppe: Elective Unit A3WP1: Seminar		**3**		
AlgoK-Sem-M	Master Seminar Algorithms and Complexity Theory	winter and summer semester, on demand(1)	3	2 Seminar	Internship report 4 months 30 minutes
DSG-Sem-M	Master Seminar in Distributed Systems	every semester	3	2 Seminar	Internship report 4 months 30 minutes
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
GdI-Sem-M	Master's Seminar Theoretical Computer Science	winter or summer semester, on demand	3	2 Seminar	Coursework Assignment with presentation 4 months 30 minutes
HCI-Prop-M	Propaedeutic: Human-Computer-Interaction	every winter semester(1)	3	3	Coursework Assignment with presentation 4 months 30 minutes
HCI-Sem-HCC-M	Master-Seminar Human-Centred Computing	every summer semester	3	2 Seminar	Coursework Assignment with presentation 4 months 30 minutes
HCI-Sem-M	Master-Seminar Human-Computer Interaction	every winter semester	3	2 Seminar	Coursework Assignment with presentation 4 months 30 minutes

KTR-Sem-M	Master Seminar Communication Systems and Computer Networks	winter or summer semester, on demand(Regelturnus: WS)	3	2 Advanced seminar	Coursework Assignment with presentation 4 months 40 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 winter semester(1)	3	2 Seminar	Internship report 3 months 30 minutes
SWT-SEM-M	Seminar in Software Engineering and Programming Languages (Master)	every semester	3	2 Seminar	Coursework Assignment with presentation 8 weeks 40 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
	Teilmodulgruppe: Project		**9**		
MOBI-PRS-M	Master Project Mobile Software Systems (SoSySc)	every summer semester(1)	9	6	Coursework Assignment and Colloquium

SWT-PR2-M	SWT Masters Project in Software Systems Science	every semester	9	6 Practicals	Coursework Assignment and Colloquium 12 weeks 30 minutes
KTR-SSSProj-M	KTR Master Project Software Systems Science	every semester(Beginn WS)	9	6	Coursework Assignment and Colloquium 4 months 30 minutes
PSI-ProjectCAD-M	Project Complex Attacks and Defenses	every semester(1)	9	6	Coursework Assignment and Colloquium 3 months 30 minutes
PSI-ProjectSP-M	Project Security and Privacy	every semester(1)	6	6	Coursework Assignment and Colloquium 3 months 30 minutes
SYSNAP-Project-M	Project Systems Programming	every semester(1)	6	4	Coursework Assignment and Colloquium 3 months 30 minutes

Module Handbook Summary

ID	Module	Semester	ECTS	Weekly Contact Hours	Examination
	A4: Masters 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		**30**		

According to the examination regulations (StuFPO) Appendix 1, students have four options regarding the Module Group A5, *International Experience*, which may also be combined:

(1) to study modules of software systems science at a university abroad for at least one semester or

(2) to accomplish a traineeship in an international context, preferentially abroad, that covers topics of the occupational field of software systems science with a volume of at least 360 working hours (12 ECTS credits).

(3) to accomplish *further* modules of module groups A1 and A2 (Examination Regulations, App. 1)

(4) to accomplish up to 18 ECTS credits in modules of foreign languages (neither English nor native language).

Teilmodulgruppe: Guided graduate study abroad 0 - 30

Regarding the study of software systems science modules at a university abroad, courses with a workload equivalent to 30 ECTS credits can be accomplished.

The courses that are selected at a foreign university have to be approved by learning agreements. For own planning security reasons, learning agreements have to be signed by those Professors at University of Bamberg responsible for the chosen subject, as well as the head of the Examination Board, before the graduate study abroad is initiated.

Teilmodulgruppe: Internship in an International context 0 - 12

Regarding the elective area 5b, *Internship in an international context*, with an equivalent workload of 12 ECTS credits, 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

Teilmodulgruppe: Foreign languages 0 - 18

In the elective area 5c, *Foreign languages*, modules comprising up to 18 ECTS credits can be taken from the range offered by the University's Language Centre. Excluded are modules of the English language and modules of the language in which the university entrance qualification was obtained.

Details, in particular the modules available for selection and the respective Module examinations are described (in German) in the *Modulhandbuch des Sprachenzentrums der Otto-Friedrich-Universität Bamberg.*

Module Group: further modules from module groups A1 and/ or A2 0 - 30

Additional, not previously completed modules from A1 or A2 module groups' required elective options in accordance with the Examination Regulations, Appendix 1.