Module Catalogue for Computer Science (M.Sc.)
of the Faculty of Minden Campus

Please note: The German version of this document is the legally binding version. The English translation provided here is for information purposes only.

Business Engineering and IT Project Management								Abbr. BPM
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.0	150 h	5	1st/2nd sem.	Annual	Summer	1 sem.	Compulsory	M.Sc.
1	Course type		Contact hours	Self-study	Forms of teaching (forms of learning)		Planned group size	Language
	Sem. lessons		4 SCH/60 h	90 h	To be announced in course		35	German and English
2	Learning outcomes/competences The students analyse innovations in information and communication technology in their capacity as drivers of restructuring and reorganisation of complex enterprises. They identify, describe and define new production processes, sales and acquisition channels, administrative structures or business processes as a result of new IT infrastructures. They categorise operational changes in the spectrum from pure automation to new process models to real paradigm shifts. They plan exemplary implementations of such new technology and corporate structures with the participation of IT and business management. They apply procedures, methods and tools such as enterprise architecture, service management, project management and (intercultural) communication skills. In particular, students should develop a holistic business engineering approach to the company							
3	Contents The following and other topics are chosen as the main focus of the programme, based on the current scientific and social discussion and current research projects: • Advanced aspects of project management • Methods and instruments of IT project management • Project management frameworks and process models e.g. PMI, PRINCE2 • Enterprise Architecture Methodology • Enterprise architecture process models e.g. TOGAF 9.1 • IT service management and frameworks e.g. ITIL business process modelling and management • (Intercultural) communication in change management • Social implications of corporate transformations							
4	Participation requirements None							
5	Form of assessment Oral examination or term paper or project paper or scientific poster or course publication manuscript or research funding proposal or internship, excursion or daily log or portfolio or learning diary or course examination							
6	Condition for the award of credit points Certificate of successful participation Module examination pass							
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)							
8	Module coordinator Prof. Dr. Dominic Becking							
9	Other information Literature: • o.A., ITIL Lifecycle Suite 2011, The Stationery Office Ltd, London 2011 • o.A., Managing Successful Projects with PRINCE 2, The Stationery Office Ltd, London 2009 • PMI (Hg.), A Guide to the Project Management Body of Knowledge, Newton Square, USA, 2008 • Tiemeyer, E., Handbuch IT-Projektmanagement: Vorgehensmodelle, • Managementinstrumente, Good Practices, München, 2010							

	• The Open Group (Hg.), TOGAF Version 9.1, Reading, UK, 2011 • Keller, W., IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur optimalen IT- Unterstützung, Heidelberg, 2012 • Brocke, J.v., Rosemann, M. (Hgg.), Handbook on Business Process Management, Springer, 2011

Formal Models in Computer Science								Abbr. FOM
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.0	150 h	5	1st/2nd sem.	Annual	Winter	1 sem.	Compulsory	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	45 h	To be announced in course	60	English
	Exercise	2 SCH/30 h	45 h		30	English

2 Learning outcomes/competences

Students who have completed this module know different formal methods of modelling sequential and distributed systems. They can evaluate these systems using examples and analyse and model them using a suitable method. They know about the applicability and limitations of these methods and of modelling in general.
Depending on the structure and characteristics of a system, they can independently carry out modelling with formal languages, automata, graphs, Petri nets. They work with the mathematical terminology of propositional and predicate logic.
They can also formulate exemplary requirements for software or hardware models in the specification languages LTL and CTL and carry out simple model checking. Furthermore, the students know the basic methodical procedure of software verification according to the Hoare calculus and are able to apply it to simple examples.

3 Contents

Basic concepts of modelling
Modelling with formal languages and automata
Graph-theoretical models
Modelling with Petri nets
Propositional logic
Predicate logic
Examples from Model Theory
Introduction to Model Checking
Linear Temporal Logic (LTL)
Computation Tree Logic (CTL)
Programme verification according to the Hoare calculus

4 Participation requirements

None

5 Form of assessment

Written examination

6 Condition for the award of credit points

Passed exam

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Dipl.-Inf. BC George

9 Other information

-

Research Project								Abbr. FUE
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.1	150 h	5	1st/2nd sem.	Annual	Winter	1 sem.	Compulsory	M.Sc.
1	Course type		Contact hours	Self-study	Forms of teaching (forms of learning)		Planned group size	Language
	Practical / Seminar		4 SCH/60 h	90 h	Project work		15	German
2	Learning outcomes/competences Students are encouraged to work independently and to solve problems in a scientific question. They independently develop a project result in a scientific procedure, implement it and document it. Students are put in a position to work on a concrete research project.							
3	Contents The contents are derived from concrete research questions posed by the lecturers in the computer science department. Thus, topics also arise from ongoing, possibly interdisciplinary research projects. - Defining and structuring complex problems - Interface definition - Project monitoring and implementation of a project with cooperating groups, if applicable - Documentation of results by means of scientific publication and lecture							
4	Participation requirements None							
5	Form of assessment Project submission							
6	Condition for the award of credit points Module examination pass							
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)							
8	Module coordinator Prof. Dr. Dr.-Ing. Matthias König							
9	Other information							

Complexity of Computation								Abbr. MKO
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.3	150 h	5	1st/2nd sem.	Annual	Winter	1 sem.	Compulsory	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	30 h	To be announced in course	35	German
	Exercise	2 SCH/30 h	60 h		20	German

2 Learning outcomes/competences

Expertise:
Students know and understand the importance of complexity theory for modern computer science. They know and understand the basic concepts and basic techniques of complexity theory, the most important complexity classes and their hierarchies, as well as their relation to algorithmic problems.

Methodological competence:
Students are able to apply essential methods of complexity theory to solve concrete problems concerning complexity classes and corresponding hierarchies. They recognise the principle limits of possible solutions with computer assistance. They can classify concrete algorithmic problems in terms of their complexity and thus filter out suitable algorithmic techniques.

Social competence: Due to the group work, the students are able to develop solutions in the group and to solve tasks cooperatively.

3 Contents

This module makes a selection among the results of complexity theory so that the importance of complexity theory for a modern computer science comes into focus.
The following contents will be learned:

- Complexity classes, P vs. NP
- Reductions and completeness
- Space complexity
- Hierarchy records
- Relativisation and oracle
- Turing machines
- Circuit complexity
- Polynomial time hierarchy
- Probabilistic complexity classes
- Interactive proof systems

4 Participation requirements

Formal: -
Content: Knowledge of algorithms and data structures, basics of theoretical computer science

5 Form of assessment

Oral examination or written exam

6 Condition for the award of credit points

Passing the module examination.

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Christoph Thiel

9 Other information

Literature:

- Balcazar / Diaz / Gabarro, Structural Complexity I and II, Springer, 2011.
- C. H. Papadimitriou. Computational Complexity. Addison-Wesley. Reading. 1995.
- U. Schöning, Theoretische Informatik - kurz gefasst, Spektrum Akademischer Verlag; 2008
- I. Wegener, Komplexitätstheorie: Grenzen der Effizienz von Algorithmen, Springer;

	Auflage: 2003 • Current scientific articles

Special Topics in Software Engineering								Abbr. SGS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.1	300 h	10	1st/2nd sem.	Annual	Summer	1 sem.	Compulsory	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	4 SCH/60 h	60 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h	150 h		15	German

2 Learning outcomes/competences

- Students learn about and understand abstract modelling of software systems.
- They can use cloud computing procedures with confidence.
- They understand modern architectural patterns and apply them.
- Students learn about agile process models and apply them practically through methodical and structured software development in a team.

3 Contents

Selected current topics from the field of software engineering and treatment of research-related and practice-relevant issues. Investigation of software engineering methods and technologies in the following areas:

Efficient requirements engineering
tracing, Volere
Model-driven software development
MDSD, MOF, CIM, PIM, PSM, model, meta-model, DSL
Aspect-oriented software development
Crosscutting Concerns, Join Points, PointCuts, Advice, Aspect
Event Driven Architecture
JMS, ESB, CEP, DSMS
Interactive Web Applications
MVC, MVP, MVVM
Semantic web applications
Open Data, Linked Data, RDF/S, OWL, SPARQL, RIF RuleML
Agile methods and processes
TDD, Kanban, BDD, Scrum, CI, CD
Cloud Computing
Service models (XaaS), deployment models (private, public, hybrid)

4 Participation requirements

None

5 Form of assessment

Performance test

6 Condition for the award of credit points

Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Jörg Brunsmann

9 Other information

Elective Module List 1: Compiler Construction								Abbr. CB
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q level
1.10	150 h	5	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	45 h	To be announced in course	60	German
	Practical	2 SCH/30 h	45 h		15	German

2 Learning outcomes/competences

Participants gain in-depth basic knowledge of compiler construction. They are familiar with the structure of compilers and the phases of translation and can apply this knowledge to create their own languages and compilers. Special attention is given to the creation and application of domain-specific languages (DSL). Students are able to understand and classify current trends and fundamental research in the field of compiler construction. The procedures covered in the practical course are applied to creating a (small) compiler.

3 Contents

Selection of topics for lecture:

- Overview of programming paradigms and concepts
- Formal languages, syntax and grammar
- Lexical analysis: Scanner, transition tables, symbol tables
- Syntactic analysis: Parser, top-down vs. bottom-up (LL(k) vs.LR(k)), Abstract Syntax Tree
- Static analysis, type systems
- Code transformation, intermediate code and code generation
- Parser/compiler generators (e.g. ANTLR, Flex, Bison)
- Domain-specific languages (internal, external), implementation with Xtext/Xtend

4 Participation requirements

None

5 Form of assessment

Oral examination or written exam

6 Condition for the award of credit points

Certificate of successful participation and passed module examination

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr.-Ing. Carsten Gips

9 Other information

Aho, Lam, Sethi, Ullman: "Compilers: Principles, Techniques, and Tools", Addison Wesley, 2013
Torczon, Cooper: "Engineering a Compiler", Academic Press, 2011 Grune et al.: "Modern Compiler Design", Springer, 2012
Parr, T.: "Language Implementation Pattern", Pragmatic Programmers, 2010 Peter Sestoft: "Programming Language Concepts", Springer, 2012
Clausing, A.: "Programmiersprachen", Spektrum Akademischer Verlag, 2011 Güting, Erwig.: "Übersetzerbau", Springer, 1999
Voelter, M.: "DSL Engineering: Designing, Implementing and Using Domain-Specific Languages", CreateSpace Independent Publishing Platform, 2013
Ghosh, D.: "DSLs in Action", Manning, 2011
Bettini, L.: "Implementing Domain-Specific Languages with Xtext and Xtend", PACKT Publishing, 2013

Elective Module List 1: Computational Geometry								Abbr. CG
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.11	150 h	5	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	45 h	To be announced in course	60	English
	Practical / Seminar	2 SCH/30 h	45 h		15	German

2	Learning outcomes/competences Students can apply methods from the field of computational geometry. Students will be able to understand and classify basic research in the field of computer graphics
3	Contents Algorithms, data structures and problems from the field of computational geometry are dealt with, e.g. the Art Gallery Problem or Post Office Problem. The methods and algorithms covered are implemented and intensively examined in the practical course.
4	Participation requirements Formal: - Content: Basic knowledge of computer graphics (e.g. participation in the lecture of the module Computer Graphics in the bachelor's degree programme)
5	Form of assessment Oral exam or written exam
6	Condition for the award of credit points Certificate of successful participation Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr. Kerstin Müller
9	Other information Bender M., Brill, M.: Computergrafik, Hanser Verlag, http://www.vislab.de Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition. Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics Principles and Practice, Addison-Wesley de Berg, M., Cheong, O., van Kreveld, M., Overmars, M Computational Geometry - Algorithms and Applications, Springer

Elective Module List 1: Advanced Computer Graphics								Abbr. GDV
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.12	150 h	5	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	45 h	To be announced in course	60	English
	Practical / Seminar	2 SCH/30 h	45 h		15	German

2	Learning outcomes/competences Students will be able to apply methods from parts of the areas of Computer Aided Geometric Design and visualisation. Students will be able to understand and classify basic research in the field of computer graphics.
3	Contents In computer graphics, images are created from data. This includes the sub-areas of geometric modelling and image synthesis as well as visualisation. In the lecture and practical course Computer Graphics, one of the following topics is dealt with in depth: Computer Aided Geometric Design, image synthesis and visualisation. Further current procedures and methods of computer graphics are learned as far as possible within the framework of accompanying industrial cooperations and research projects. The methods and algorithms covered are implemented and intensively examined in the practical course.
4	Participation requirements Formal: - Content: Basic knowledge of computer graphics (e.g. participation in the “Computer Graphics” lecture from the bachelor’s degree programme)
5	Form of assessment Oral exam or written exam
6	Condition for the award of credit points Certificate of successful participation Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr. Kerstin Müller
9	Other information Bender M., Brill, M.: Computergrafik, Hanser Verlag, http://www.vislab.de Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition. Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics Principles and Practice, Addison-Wesley de Berg, M., Cheong, O., van Kreveld, M., Overmars, M Computational Geometry - Algorithms and Applications, Springer

Elective Module List 1: Managing Innovation and Technology Transfer								Abbr. IMT
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.13	150 h	5	1st/2nd sem.	Bi-annual	Winter/ summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	45 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h	45 h		15	German

2	Learning outcomes/competences Students understand the basic mechanisms of processes underlying innovations. They can classify proprietary knowledge and evaluate the possibility of securing this knowledge. They can put their own ideas into business plans.
3	Contents - Knowledge and knowledge management - Creativity techniques - Continuous and disruptive innovation - Intellectual property - Business planning - Strategy and market entry - Entrepreneurship
4	Participation requirements None
5	Form of assessment Term paper
6	Condition for the award of credit points Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr. Martin Hoffmann, (covered by Prof. Dr. Matthias König)
9	Other information

Elective Module List 1: Machine Learning Methods								Abbr. MML
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.14	150 h	5	1st/2nd sem.	Bi-annual	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	45 h	To be announced in course	60	German
	Practical course	2 SCH/30 h	45 h		15	German

2 Learning outcomes/competences

The aim is to teach the advantages and disadvantages of various machine learning methods and algorithms. The students are able to independently derive and apply a solution approach for a given problem from the methods learned.

Through the practical applications and the basic knowledge imparted in the lecture, the students are prepared for special areas of the subject field.

3 Contents

Selection of topics:

- Classification, for example with Naive Bayes, kNN, Roccio, SVM, Markov Random Fields
- Clustering methods, for example with k-Means, EM
- Reinforcement Learning
- Active learning
- Frameworks (TensorFlow, Microsoft Azure,)
- Deep Learning, Convolutional Neural Networks
- Sequential Data, HMM, LDS, Particle Filter

4 Participation requirements

Formal: -
Content: Linear algebra, basic knowledge of artificial intelligence, programming skills

5 Form of assessment

Oral examination or written exam

6 Condition for the award of credit points

Certificate of successful participation and passed module examination

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr.-Ing. Carsten Gips, (covered by Prof. Dr. Dr.-Ing. Matthias König)

9 Other information

Literature will be announced in the course

Elective Module List 1: Mobile Applications								Abbr. MAN
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.15	150 h	5	1st/2nd sem.	Bi-annual	Winter/ summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	45 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h	45 h		15	German

2 Learning outcomes/competences

Participants gain advanced competences in the field of software development for mobile devices. They are familiar with common hardware platforms and mobile operating systems and can select and use suitable frameworks for software development in a problem-oriented manner.
Practical examples are used to practise the procedures covered.

3 Contents

Fundamentals of software development for mobile systems (status quo, research subjects, trends)
Overview of current hardware and operating systems
Current software frameworks for mobile applications
Design of mobile systems (including description of mobile architectures)
Implementation of mobile systems
Cross Platform Development
Testing mobile software systems
Resource management in mobile systems
Security and privacy
Energy aspects
Data storage
Sensor technology and context (including geo-localisation)

4 Participation requirements

None

5 Form of assessment

Oral examination

6 Condition for the award of credit points

Certificate of successful participation
Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Martin Hoffmann, (covered by Prof. Dr. Matthias König)

9 Other information

Modern Database Systems								Abbr. MDB
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.16	150 h	5	1st/2nd sem.	Bi-annual	Summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons Practical / Seminar	2 SCH/30 h 2 SCH/30 h	45 h 45 h	To be announced in course	35 15	German and English

2 Learning outcomes/competences

Driven by extreme requirements in the area of Big Data, there has been a stormy development in the area of database systems beyond the classic RDBMS. Students acquire an overview of the theory, architectures, implementation techniques, languages and applications of new database systems. They make reasoned decisions for the application of new DBMSs. They can install and administer new DBMSs, analysing and implementing application software requirements. They can analyse the performance of databases and increase it through suitable technical measures. They fill DBMSs with large amounts of sample data and address them from applications they have implemented themselves.

3 Contents

The following and other topics are chosen as the main focus of the programme, based on the current scientific and social discussion and current research projects:

- Semi-structured and unstructured database content
- Databases for modern applications
- Databases for extreme applications
- NoSQL DBMS of various flavours
- DBMSs with heterogeneous architecture principles
- APIs of modern DBMSs
- Languages and development environments for modern DBMSs
- Parameterisation and optimisation of modern DBMSs
- Measuring and assessing the performance of modern and classic DBMSs
- DaaS

4 Participation requirements

Content: In-depth knowledge of database systems (e.g. attendance of a course in databases and a specialisation course e.g. implementation techniques of databases)

5 Form of assessment

Oral examination or term paper or project work or scientific poster or course publication manuscript or research funding proposal or internship, field trip or daily log or portfolio or learning diary or course examination

6 Condition for the award of credit points

Certificate of successful participation
Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Dominic Becking

9 Other information

Literature:

- Journals and proceedings on the topic
- Harrison, G., Next Generation Databases: NoSQL, NewSQL, and Big Data, NYC, 2015 Friedland, A. et.al., N NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Hanser, München 2011
- Redmond, E; Wilson, J.R.: Seven Databases in Seven Weeks A Guide to Modern Databases and the NoSQL Movement. Pragmatic Programmers, o.O. 2.Aufl., 2018
- Tiwari, S.: Professional NoSQL. Wiley, Indianapolis, 2011

Elective Module List 1: Operations Research								Abbr. OR
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.17	150 h	5	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	90 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h			15	German

2 Learning outcomes/competences

Students will be able to formulate operations research problems and solve them using the methods they have learned. They are also involved in the implementation of the procedures to ensure that they can confidently implement and develop them in a programming language.

3 Contents

For example:

- Fundamentals of Operations Research
- Network Planning Technique
- Linear programming
- Queueing theory
- Graph theory
- Combinatorial and dynamic optimisation simulation
- Applications, e.g.
    - Revenue management
    - Logistics and Supply Chain Management
    - Forecasting method

4 Participation requirements

Content: Good knowledge of algebra, analysis, numerics, stochastics, optimisation

5 Form of assessment

Written examination or oral examination

6 Condition for the award of credit points

Certificate of successful participation
Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Dr.-Ing. Matthias König

9 Other information

Literature will be announced in the course.

Elective module List 1: Optimisation								Abbr. OPT
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.18	150 h	5	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	90 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h			15	German

2	Learning outcomes/competences Students recognise when a problem is amenable to an optimisation procedure and can formulate it as an appropriate optimisation problem. They are familiar with the essential procedures for solving these problems and are able to assess, apply and implement.
3	Contents Basics and algorithms for - Linear optimisation - Discrete optimisation - Non-linear optimisation (with/without constraints) - Multicriteria optimisation o as well as applications of optimisation. From the perspective of computer science, the focus is on algorithms or numerical solution methods.
4	Participation requirements Content: Good knowledge of linear algebra, analysis, numerics
5	Form of assessment Written examination or oral examination
6	Condition for the award of credit points Certificate of successful participation Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr. Dr.-Ing. Matthias König
9	Other information Literature will be announced in the course.

Elective Module List 1: Reliable and Secure Software Systems								Abbr. ZSS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.19	150 h	5	1st/2nd sem.	Annual	Winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	30 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h	60 h		15	German

2 Learning outcomes/competences

Expertise:
Students know typical standards for developing secure and reliable software systems and for evaluating such systems. In particular, they know current methods for the specification, design and testing of reliable and secure software systems.
Methodological competence:
The students are able to select and practically apply the appropriate basic techniques for the specification, design, testing and verification of reliable and secure software systems. They can assess the quality of software systems in terms of security and reliability according to various standards, and identify and implement opportunities for improvement.
Social competence: Due to the group work, the students are capable of finding solutions collaboratively and solving tasks cooperatively.

3 Contents

- Fundamentals and models for reliability and safety;
- Description techniques for reliable and secure software systems
- Standards for the evaluation and development of reliable and secure software systems
- Vulnerability- and risk-based process models for the development of reliable and secure software systems
- Standards for testing and debugging software systems
- Case studies in research and industry

4 Participation requirements

Formal: -
Content: Knowledge of software engineering

5 Form of assessment

Oral examination or written exam

6 Condition for the award of credit points

Passing the module examination.

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Christoph Thiel

9 Other information

Literature:

- N. Storey, Safety Critical Computer Systems, Addison Wesley, 1996
- C. P. Pfleeger, Security in Computing, 4th ed., Prentice Hall 2007
- J. Humble, D. Farley, Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation , Addison-Wesley 2010
- Current professional articles

Elective Module List 1: Blockchain Systems								Abbr. BCS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
1.20	150 h	5	1st/2nd sem.	Bi-annual	Winter/ summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	45 h	To be announced in course	35	German
	Practical / Seminar	2 SCH/30 h	45 h		15	German

2 Learning outcomes/competences

- The students know the requirements as well as the advantages and limitations of blockchain-based, decentrally organised networks compared to traditional distributed databases.
- Students learn competences for the technical integration of distributed ledger technologies, taking into account the critical system parameters (decentralisation, scalability, bandwidth, cycle time) and are enabled to specifically select suitable implementations in potential application scenarios.
- The students know established or experimental alternatives to the blockchain-based approach and can name relevant technical aspects.
- Students also have the ability to critically examine public blockchain protocols from an economic and socially relevant perspective.

3 Contents

- Distributed Ledger Technologies.
- Systematic overview and use cases (e.g. Bitcoin, Ethereum, Hyperledger).
- Design, parameterisation and realisation of a blockchain system.
- Cryptographic foundations of pseudonymous and anonymous public transactions.
- Consensus building in decentrally organised networks (e.g. PoW, PoS, RPCA).
- Social aspects of a digital (token) economy.
- Blockchain middleware and Dapp development (e.g. Esprezzo, Solidity).
- Approaches to scaling (e.g. Lightning, Drivechain, Plasma).
- Current state of research on established and alternative approaches (e.g. DAGs).

4 Participation requirements

None

5 Form of assessment

Project/seminar paper, oral exam or written exam.

6 Condition for the award of credit points

Practical course with test and passed module examination

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Jörg Brunsmann

9 Other information

Elective Module List 2: Computer-Aided Geometric Design								Abbr. CAG
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.10	300 h	10	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	210 h	To be announced in course	60	
	Practical / Seminar	4 SCH/60 h			15	German

2 Learning outcomes/competences

After the course, the students have an overview of suitable procedures and methods in the field of Computer-Aided Geometric Design and can implement the selected algorithms. Students are able to plan and realise a small research project and are able to understand and classify research work in the field covered.

3 Contents

This course covers current procedures and techniques from the field of Computer Aided Geometric Design. The following topics are examples of this: Bezier and B-spline techniques, subdivision surfaces, and current research in the field of CAGD.
The theoretical foundations are developed in the lecture. In the practical course, data from current industrial and research projects are suitably visualised. Accompanying industrial cooperations and research projects are envisaged. The computer graphics specific applications are to be worked on in a team. The internship part can be used as preparation for the master thesis in the field of computer graphics.

4 Participation requirements

Formal: -
Content: Fundamentals of computer graphics
(e.g. participation in a lecture on computer graphics)

5 Form of assessment

Project work

6 Condition for the award of credit points

Successful completion of the project work

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Kerstin Müller

9 Other information

- Gerald Farin: Curves and Surfaces for CAGD: A Practical Guide, Morgan Kaufmann
- Bender M., Brill, M.: Computergrafik, 2. Auflage, Hanser Verlag, 2005. http://www.vislab.de
- Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition.
- Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics Principles and Practice, Addison-Wesley

Elective Module List 2: Data Science								Abbr. DS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.11	300 h	10	1st/2nd sem.	Bi-annual according to demand	Winter/ summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	210 h	To be announced in course	60	German and English
	Practical	4 SCH/60 h			15	German and English

2 Learning outcomes/competences

Driven by extreme demands in the field of Big Data, there is a trend towards consolidating methods and findings from different disciplines of computer science into a unified Data Science. The students develop a scientific approach to the fields of data science by means of a complex project from the field of analysis and processing of large data sets.
The project set by the lecturer serves to develop and apply scientific methods of computer science, in particular the inventive method of scientific knowledge acquisition, mathematical and statistical methods for data analysis and empirical research, especially experimental methods for performance measurement. Students gain practical experience in project work in scientific research, the design and implementation of research prototypes, the scientific classification of results and presentation and depiction of the results in typical formats for scientific publication.

3 Contents

Data science uses methods and findings from different areas of computer science. The application of such methods from a critical-scientific point of view is an essential content of the course.
Selection of possible topics:

- Cloud computing
- Modern database architectures
- Data analysis
- Data visualisation
- Information Retrieval
- Knowledge Discovery
- Data Mining
- Bad Data and Data Cleansing
- Multivariate Statistics
- Time Series Analysis

In consultation with the lecturer, students select project topics from the field of Big Data/Data Science in groups and work on them as a project group over a semester. They research the state of the art and science, formulate a research question and develop the required skillset. They use current project management methods and tools. They implement selected parts of the modelling into functioning software. They contextualise the results within the current scientific discussion and create a presentation of the results using a typical format for a scientific publication.

4 Participation requirements

Formal: -
Content: In-depth knowledge of database systems (e.g. attendance of a course in databases and of a specialisation course (e.g. implementation techniques of databases), basic knowledge of scientific work (e.g. attendance of a specialised seminar)

5 Form of assessment

	Oral examination or term paper or project work or scientific poster or course publication manuscript or research funding proposal or internship, excursion or daily log or portfolio or learning diary or course examination
6	Condition for the award of credit points Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr. Dominic Becking
9	Other information References: Journals and proceedings on the topic Friedland, A. et.al., N NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Hanser, München 2011 O Neill, C.; Schutt, R.: Doing Data Science. O Reilly, Cambridge USA, 2013 McCallum, Q.E.: Bad Data Handbook. O Reilly, Cambridge USA, 2012 McKinney, W.: Python for Data Analysis. O Reilly, Cambridge USA, 2013 Witten, I.H. et.al.: Data Mining. Elsevier, Burlington USA, 2011

Elective Module List 2: Modelling and Designing Software Systems								Abbr. DMS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.12	300 h	10	1st/2nd sem.	Annual according to demand	Winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	210 h	To be announced in course	60	German and English
	Practical	4 SCH/60 h			15	German and English

2 Learning outcomes/competences

The aim of the course is to develop the students' problem-solving skills so that they can take on technical leadership tasks in large software development projects. Particular attention is paid to methodological competence and here especially to methods of independent scientific work. In the project set by the lecturer, the students acquire competences in the development and application of scientific methods of computer science, in particular the inventive method of gaining scientific knowledge.
In consultation with the lecturer, students select project topics in groups and work on these over a semester as a project group. They research the state of the art and science, formulate a research question and requirements and create a complete professional and technical modelling and documentation. They use current project management methods and tools. Students develop designs and implement research prototypes They implement selected parts of the modelling into functioning software. They contextualise the results within the current scientific discussion and create a presentation of the results using a typical format for a scientific publication.

3 Contents

Large software systems are part of the enterprise architecture of every medium and large company. Scientific approaches to the design and modelling of such systems are widely discussed and continuously changing. The application of current methods from a critical-scientific point of view is an essential content of the course. Choice of topics:

- Current languages and models of software design
- Requirements engineering and modelling
- Architecture models and architecture patterns
- Methods and procedures of IT enterprise architecture
- Design of complex information systems
- Structures for data and information
- Mobile data and data mobilisation
- Replication, distribution
- SaaS and DaaS
- Cloud computing

4 Participation requirements

Formal: -
Content: Basic knowledge of databases (e.g. attendance of a course on databases), basic knowledge of software engineering (e.g. attendance of a course on software engineering), basic knowledge of scientific work (e.g. attendance of a specialised seminar)

5 Form of assessment

Oral examination, or term paper, or project paper, or scientific poster, or course publication manuscript, or research funding proposal, or practical, excursion or daily log, or portfolio, or learning diary, or course examination

6 Condition for the award of credit points

Certificate of successful participation
Module examination pass

7 Application of the module (in the following study programmes)

	Computer Science (M.Sc.)
8	Module coordinator
	Prof. Dr. Dominic Becking
9	Other information
	References: • Journal: Software and Systems Modeling, Springer, Heidelberg • Starke, G; Effective Software Architectures, Hanser, Munich, 2011 Authors' • Collective, TOGAF Version 9.1, The Open Group, London, 2011

Elective Module List 2: Distributed Organic Computing Systems								Abbr. DOC
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.13	300 h	10	1st/2nd sem.	Bi-annual	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	45 h	To be announced in course	35	German
	Practical / Seminar	4 SCH/60 h	165 h		15	German

2	Learning outcomes/competences Students deepen their knowledge in the field of distributed systems and acquire sound knowledge of concepts and applications of cloud computing and peer-to-peer systems, ad-hoc and sensor networks and infrastructure-less communication networks, among others. In particular, nature-analogue algorithms and the functioning of self-organising distributed systems are considered. The students develop their own approaches for future system architectures (e.g. from the research area of organic computing). They will gain practical experience in project work (self-study), in scientific research, in the design and implementation of system architectures, the subsequent evaluation, presentation and presentation of the results (from a technical and scientific point of view).
3	Contents Preliminary scientific work in the field of cloud computing and peer-to-peer systems provides approaches for organising complex abstracted IT infrastructures that can be dynamically adapted to changing conditions of use and provide services at different levels, e.g. computing capacity, data storage, network capacities and software services. Ad-hoc and sensor networks are examined as special cases. To this end, the requirements and goals of such systems are first defined and discussed. In addition, concepts from the field of system architectures and approaches from the field of self-organising algorithms are presented and evaluated.
4	Participation requirements Formally: -, Content: Advanced knowledge in the area of distributed systems (e.g. attendance of a course in computer networks, communication networks or similar)
5	Form of assessment Oral examination
6	Condition for the award of credit points Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr.-Ing. Martin Hoffmann
9	Other information References: Journals and proceedings on the topic C. Müller-Schloer, S. Tomforde (ed.): Organic Computing Technical Systems for Survival in the Real World, Springer ISBN 978-3-319-68476-5, 2017 Rolf P. Würtz (ed.): Organic Computing (Understanding Complex Systems), Springer Verlag Berlin, 2008, 356 p., hardcover ISBN 978-3540776567 Peer-to-Peer Systems and Applications; Ralf Steinmetz und Klaus Wehrle (Hrsg.); 629 Seiten, Springer, ISBN 3-540-29192-X, 2005. Peer-to-Peer-Netzwerke; Peter Mahlmann und Christian Schindelhauer; 293 Seiten, Springer, ISBN 978-3-540-33991-5, 2007.

Internet of Things and Smart Systems								Abbr. IOT
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.14	300 h	10	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	210 h	To be announced in course	35	German
	Practical	4 SCH/60 h			15	German

2 Learning outcomes/competences

Students understand the development of smart systems and can apply them. For this purpose, technologies of the Internet of Things (or Industry 4.0, cyber-physical systems) are taught and analysed. The aim is to promote independence and practical problem-solving skills as well as the ability to work scientifically on one's own. The project provided by the lecturer serves to deepen knowledge in the field of programming Internet of Things or Smart Systems and constitutes practical experience in project planning and realisation of a research prototype.

3 Contents

The seminar-based teaching covers procedures and technologies for the implementation of Internet-of-Things applications, which are tailored to the project content. The students carry out an analysis of the procedures and their implementation in the project context.

Examples of topics covered in the area of the Internet of Things:

- Protocols
- Architectures
- Algorithms
- Frameworks and APIs
- Interaction with mobile applications and cloud computing
- Applications
  - Smart Home, Smart Cities, Smart Health, Smart Energy...
  - Condition monitoring, predictive maintenance

Project process:

Largely independent processing of a more complex task within the framework of a research and development project. The lecturer defines the objective and conducts a regular discourse on the progress of the project. He/she also sets milestones and the respective form each project will take in cooperation with students.

4 Participation requirements

Formal: -
Content: Programming skills, knowledge of embedded systems

5 Form of assessment

Successful completion of the project work

6 Condition for the award of credit points

Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Dr.-Ing. Matthias König

9 Other information

Literature will be announced in the course

Elective Module List 2: Computer Music								Abbr. MI2
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.15	300 h	10	1st/2nd sem.	Annual according to demand	Winter/ summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	210 h	To be announced in course	60	German and English
	Practical	4 SCH/60 h			15	German and English

2 Learning outcomes/competences

With a turnover of about 2 billion euros in Germany alone, the music industry is one of the most important target industries for the field of media informatics. Music informatics deals with all computer-based techniques and the development of applications for the composition, production, distribution, billing/licensing and enjoyment of music and other audio products. In addition, special aspects of music management, the music business and the technical support of creative processes of music creators are the subject of the field. The students develop a scientific approach to this important sub-field of media informatics by means of a complex project from the field of music informatics. The project set by the lecturer serves to develop and apply scientific methods of computer science, in particular the inventive method of scientific knowledge acquisition, physical and mathematical methods for sound generation, composition, mastering, etc. The students thereby incorporate knowledge about music as a universal cultural phenomenon into their considerations and familiarise themselves with scientific literature from anthropology, psychology and cultural studies. Students gain practical experience in project work in scientific research, the design and implementation of research prototypes, the scientific classification of the results and the presentation and presentation of the results in typical formats of scientific publication. In consultation with the lecturer, the students select project topics and work on these over a semester as a project group. They research the state of the art and science, formulate a research question and develop the required skillset. They use current project management methods and tools. They implement selected parts of the modelling into functioning software. They contextualise the results within the current scientific discussion and create a presentation of the results using a typical format for a scientific publication.

3 Contents

Music informatics uses methods and findings from various fields of computer science, physics, mathematics and cultural studies. The application of such methods from a critical-scientific point of view is an essential content of the course.
Choice of topics:

- Mathematical foundations of music
- Physical foundations of music
- Analogue and digital sound generators
- Audio digitisation and audio formats
- MIDI
- Virtual instruments and VST
- Digital sound processing and modification
- Special audio programming languages
- Audio libraries for all-purpose programming languages, esp. C/C++
- Agogic and the human factor
- Music as a universal human phenomenon
- Psychoacoustics and musical enjoyment
- DAW programming

4	Participation requirements None
5	Form of assessment Oral examination or term paper or project paper or scientific poster or course publication manuscript or research funding proposal or internship, excursion or daily log or portfolio or learning diary or course examination
6	Condition for the award of credit points Certificate Module examination pass
7	Application of the module (in the following study programmes) Computer Science (M.Sc.)
8	Module coordinator Prof. Dr. Dominic Becking
9	Other information References: • Current journals and proceedings on the topic. • Steppat, M.: Audioprogrammierung. Hanser, München, 2014. • Boulanger, R., Lazzarini, V. (Hgg.): The Audio Programming Book. MIT Press, Cambridge USA, 2011. • Mazzola, G.: Elemente der Musikinformatik. Birkhäuser, Basel, 2006. • Loy, G.: Musimathics the mathematical foundations of music, Vol. 1 u. 2. MIT Press, Cambridge USA, 2007. • Gouveia, D.: Getting Started with C++ Audio Programming for Game Development. Packt Publishing, Birmingham, 2013. • Brown, A. R.: Making Music with Java. o.O., 2005

Elective Module List 2: Natural Language Processing and Information Retrieval								Abbr. NLP
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.16	300 h	10	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Seminar lessons	2 SCH/30 h	45 h	To be announced in course	35	German
	Practical	4 SCH/60 h	165 h		15	German

2 Learning outcomes/competences

The aim is to promote independence and practical problem-solving skills as well as the ability to work scientifically on one's own. The lecture builds on the knowledge acquired in the Machine Learning Methods module and transfers the methods learned to the field of Natural Language Processing. In particular, text mining methods such as sentiment analysis and recommender systems are discussed. Students will be able to understand and classify current research in the field covered. The project provided by the lecturer will deepen the topics covered and additionally serves as practical experience in project planning and realisation of a research prototype.

3 Contents

Selection of topics:

- Natural language processing, language models, speech recognition, syntactic analysis, semantics
- Question answering and summarisation systems
- Spelling correction methods, recommender systems
- Sentiment analysis (pre-processing, feature extraction, feature selection, text classification)
- Information retrieval (crawler, tolerant retrieval, vector space model)

Project process:

Largely independent work on a more complex task within the framework of a research and development project in a team, which can also be worked on in cooperation with research and development departments of companies. As a rule, the project groups consist of 3-5 students who come together freely and choose a project leader from among themselves and develop according to procedural models agreed with the lecturer. The lecturer defines the objective and conducts a regular discourse on the progress of the project. He/she also agrees upon milestones and preferred forms of communication and cooperation in discussion with the team.

4 Participation requirements

None

5 Form of assessment

Project work

6 Condition for the award of credit points

Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr.-Ing. Carsten Gips

9 Other information

Elective Module List 2: Robotics, Vision and Control								Abbr. RVC
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.17	300 h	10	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	210 h	To be announced in course	35	German
	Practical	4 SCH/60 h			15	German

2 Learning outcomes/competences

The aim is to promote independence and practical problem-solving skills as well as the ability to work scientifically on one's own. Students will be able to understand, apply and evaluate current research in the field covered.
The project provided by the lecturer serves to deepen the knowledge and application of the learned contents in the field of robotics and computer vision, as well as a practical experience in project planning and realisation of a research prototype.

3 Contents

The seminar-based teaching deals with methods of robotics and computer vision, which are coordinated with the project content. The students carry out an analysis of the procedures and their implementation in the project context.
A current problem from the subfields of robotics and computer vision is chosen for the project. Examples of these sub-areas are:

- Video and motion analysis
- Stereo image analysis, Structure from Motion
- Image segmentation, object recognition, scene understanding
- Machine Learning / Deep Robot Learning
- SLAM and vSLAM
- Path Planning and Navigation
- Robot manipulations, kinematics and control
- Visual Servoing
- Human-Robot Interaction

Project process:
Largely independent processing of a more complex task within the framework of a research and development project, which can also be processed in cooperation with R&D departments of companies. The lecturer defines the objective and conducts a regular discourse on the progress of the project. He/she also sets milestones and the respective form each project will take in cooperation with students.

4 Participation requirements

Formal: -
Content: Machine learning methods, image recognition and pattern processing, programming skills

5 Form of assessment

Successful completion of the project work

6 Condition for the award of credit points

Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Dr.-Ing. Matthias König

9 Other information

Literature will be announced in the course

Elective Module List 2: System Security								Abbr. SYS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.18	300 h	10	1st/2nd sem.	Annual	Summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons	2 SCH/30 h	30 h	To be announced in course	35	German
	Practical / Seminar	4 SCH/60 h	180 h		15	German

2 Learning outcomes/competences

Expertise:
Students know the methods and techniques of typical system security standards, especially for the collection and formal description of security requirements, security policies and abstract security models. Students understand the application scenarios, possibilities and limitations of the different standards. Furthermore, students know security measures and mechanisms for implementing the models, such as access and information flow control models.

Methodological competence:
Students are able to describe the security development lifecycle, check compliance with corresponding specifications, verify the correctness of security policies and models with regard to the various standards and, conversely, independently develop suitable policies and formal models for colloquially formulated security requirements and implement these with suitable security measures and mechanisms.

Social competence: Due to the group work, the students are able to develop solutions in the group and to solve tasks cooperatively.

3 Contents

- Security Development Lifecycle: The path from safety requirements via formal safety models to the safe system
- Standards for the analysis of safety requirements
- Formulation of security policies
- Standards and formalisms for security models
- Security architectures
- Security measures and mechanisms
- SELinux, Kerberus, other current security control solutions

4 Participation requirements

Formal: -
Content: Knowledge of algorithms and data structures, basics of theoretical computer science

5 Form of assessment

Successful completion of the project work

6 Condition for the award of credit points

Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Christoph Thiel

9 Other information

Literature:

- Claudia Eckert: IT-Sicherheit, 6. Auflage, Oldenbourg, 2009 Ross Anderson:
- Security Engineering, Wiley & Sons, 2008
- Jack Koziol et.al.: The Shellcoder's Handbook, Wiley & Sons, 2007
- Michael Howard et.al.: The Security Development Lifecycle, Microsoft Press, 2009

	• Patrick Horster, Systemsicherheit, Vieweg, 2015 • Current professional articles

Elective Module List 2: Data Mining								Abbr. DM
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.19	300 h	10	1st/2nd sem.	Bi-annual	Winter/ summer	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Sem. lessons Practical / Seminar	2 SCH/30 h 4 SCH/60 h	45 h 165 h	Lecture Team Project	15	German

2 Learning outcomes/competences

The aim is to promote independence and practical problem-solving skills as well as the ability to work scientifically on one's own in a computer science core subject, and to increase awareness of the major challenges in the context of digitalisation in industry and society. Students acquire core competences in data mining and the application of machine learning methods. They will learn technologies and algorithms for all processes in the application of data mining to data from the lecturers' current research projects.
Knowledge in the field of AI is deepened and practical experience is gained in project planning and realisation of a research prototype in the form of software.

3 Contents

Interdisciplinary approaches are typical for research-relevant applications of computer science. In this subject, fundamental elements on the respective project-relevant research topics, which are to be taken from current topics from the research focus Interdisciplinary Research for Sustainable, Renewable and Safe Energy Concepts and the Solar Computing Lab of the university. These include the basics of sensor technology, e.g. based on RaspberryPi, yield analysis and error diagnostics in photovoltaics, renewable energies.
Special methods of environmental informatics are to be taught by the lecturer in the course depending on the project content and field of application, analysed by the students and used in the project-specific implementations.
Selection of topics for knowledge transfer:

- The data mining process with
  - Record sensor data
  - Filter, clean and consolidate data
  - Data analysis for data reduction
  - Application of machine learning methods (neural networks, deep learning, cluster algorithms)
  - Formulating scientific questions and conducting experiments
- Programming with libraries for statistics and machine learning (e.g. Python, NumPy, Pandas, SciPy, Jupyter, IPython)

The focus is largely on the independent processing of a complex task within the framework of a research and development project in a team, which can also be processed in cooperation with research and development departments of companies. As a rule, the project groups consist of 2–4 students who are to come together freely, choose a project leader from among themselves and develop according to procedural models agreed with the lecturers. The lecturer defines the interdisciplinary objective and conducts a regular discourse on the progress of the project. Milestones and forms of communication and cooperation are agreed with the team.

4 Participation requirements

Formal: -
Content: Programming in Java, Python or JavaScript, basics of artificial intelligence, databases

5 Form of assessment

Successful completion of the project work

6 Condition for the award of credit points

Module examination pass

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8	Module coordinator Prof. Dr.-Ing. Grit Behrens
9	Other information Volker Runkler Data Mining Modelle und Algorithmen intelligenter Datenanalyse , Springer Vieweg 2015 ,ISBN 978-3-8348-2171-3 Ian H. Witten Data Mining: Practical Machine Learning Tools and Techniques, ELSEWVIER 2017, ISBN 978-0128042915 Thomas Haslwanter An Introduction to Statistics with Python , Springer Nature 2016, ISBN 978-3-319-28316-6 Miroslav Kubat An Introdruction to Machine Learning , Springer Nature 2017, ISBN 978-3-319-63912-3

Elective Module List 2: Visualisation								Abbr. VIS
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
2.20	300 h	10	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Lecture	2 SCH/30 h	45 h	To be announced in course	60	
	Practical / Seminar	4 SCH/60 h	165 h		15	German

2 Learning outcomes/competences

After the course, the students have an overview of suitable procedures and methods for visual problem solving of current problems in computer graphics and can implement the selected algorithms. Students are able to plan and realise a small research project and are able to understand and classify research work in the field covered.

3 Contents

Computer graphics is the field of data processing concerned with the capture, storage, processing and output of graphic representations. Within the framework of this course, students will acquire current visualisation techniques for presenting and solving interdisciplinary problems. The theoretical foundations are developed in the lecture. In the practical course, data from current industrial and research projects are suitably visualised. Accompanying industrial cooperations and research projects are envisaged. The questions focus on the following areas of computer graphics:

- Virtual reality applications.
- Computer Games and Serious Games
- Animation and Graphic Simulation
- Human Computer Interaction and Information Visualisation

The computer graphics specific applications are to be worked on in a team. The internship section can be used as preparation for the master thesis in the field of computer graphics.

4 Participation requirements

Formal: -

Content: Fundamentals of computer graphics

(e.g. participation in the lecture “Computer Graphics” in the bachelor’s degree course)

5 Form of assessment

Project work

6 Condition for the award of credit points

Successful completion of the project work

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Kerstin Müller

9 Other information

- Bender M., Brill, M.: Computergrafik, 2. Auflage, Hanser Verlag, 2005. http://www.vislab.de
- Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition.
- Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics Principles and Practice, Addison-Wesley

Elective Module List 2: Advanced Deep Learning for Computer Vision								Abbr. ADL
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q level
2.21	300 h	10	1st/2nd sem.	Bi-annual according to demand	Summer/ winter	1 sem.	Compulsory elective	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (learning methods)	Planned group size	Language
	Lecture	2 SCH/30 h	45 h	To be announced in course.	35	German
	Practical / Seminar	4 SCH/60 h	165 h		15	German

2 Learning outcomes/competences

The students are familiar with current applications for the use of deep artificial neural networks in the field of computer vision. They are able to understand and evaluate current research in the field covered. Students are able to implement selected procedures, using appropriate software libraries for this purpose.

Working in project teams, they are able to independently discuss issues in the covered area in groups, develop solutions and implement them in practice. In addition, students learn to work on tasks in a cooperative manner and to carry them out within a specified period of time.

3 Contents

The lecture covers both basic and advanced deep learning methods and architectures. For many applications in the field of computer vision, these represent the current state of the art. In terms of content, it focuses on applications in the field of computer vision. Examples of possible content include:

- Revision of the fundamentals of neural networks and CNNs
- Bias effects
- Object recognition
- Image segmentation
- Object tracking
- Generative Adversarial Neural Networks (GANs) and applications
- Deep learning frameworks

Project process:

In consultation with the lecturer, the students select topics and work on these in groups over the course of the semester. They research the state of the art in technology and science, implement a chosen solution approach as working software, evaluate and document their results and regularly present their findings.

4 Participation requirements

Formally: -

Content: Linear algebra, basic knowledge of machine learning, programming skills

5 Form of assessment

Written examination or oral examination or term paper or project work or performance examination or scientific poster or short publication manuscript or research funding application or practical project, excursion or daily protocol or portfolio or learning diary or OSPE or (according to Section 14 (4) RPO) a combination of different forms of assessment.

Examinations may be held digitally.

6 Condition for the award of credit points

Certificate of successful participation and passed module examination

7 Application of the module (in the following study programmes):

Computer Science (M.Sc.)

8 Module coordinator

Prof. Dr. Jan Rexilius

9 Other information

Literature will be announced in the course.

Master Thesis								Abbr. MA
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
3.0	720 h	24	3rd sem.	Any semester	Summer/ winter	1 sem.	Compulsory	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	0.5 SCH individual lecturer-based faculty tutoring	20 h	700 h	Master thesis	1	German or English

2 Learning outcomes/competences

Students acquire the skills to work independently on a complex, practice-related computer science topic using scientific methods, i.e. to analyse the problem, identify possible solutions, classify them in the state of the art in science/technology, implement them and finally evaluate them, as well as to write a sophisticated scientific paper on the topic.

3 Contents

The master thesis is intended to prove that students are able to solve a complex, extensive and technically demanding problem using scientific methods within a limited period of time and to document the theoretical and practical knowledge acquired in the process in a comprehensible manner at a high level.

1. Define the concrete details of a task
2. Preparation of a timetable
3. Evaluation and listing of the techniques and methods to be used
4. Creation of a software concept
5. Implementation and documentation of the software solution
6. Overall view, test and evaluation of the solution
7. Presentation of the solution in the form of the master thesis.

In contrast to the bachelor thesis, this involves working on a more demanding and possibly more extensive topic at a higher scientific level over a longer period of time.

4 Participation requirements

None

5 Form of assessment

Master thesis assessed by two examiners

6 Condition for the award of credit points

Master thesis passed

7 Application of the module (in the following study programmes)

Computer Science (M.Sc.)

8 Module coordinator

All lecturers in the field of computer science who have passed at least the relevant master examination at a university or a comparable examination or who have acquired a comparable qualification and who have exercised relevant independent teaching activities in the master programme.

9 Other information

Colloquium								Abbr. KOL
No.	Workload	Credit points	Study semester	Frequency	Sem.	Duration	Type	Q-level
3.1	180 h	6	3rd sem.	Any semester	Summer/ winter	1 sem.	Compulsory	M.Sc.

1	Course type	Contact hours	Self-study	Forms of teaching (forms of learning)	Planned group size	Language
	Colloquium	According to need	180 h	Lecture and thesis defence	1	German or English

2	Learning outcomes/competences
	Students are able to present the results of the master thesis, its subject-specific foundations, its interdisciplinary connections and its extra-subject references orally to present, independently justify and assess their significance for practice.
3	Contents
	Colloquium lecture
4	Participation requirements
	Formal: Admission is granted to those who have proven that they meet the requirements for admission to the master thesis as specified in the MPO, have passed all but two examinations during the course of study and have received a grade of at least 4.0 for the master thesis. Content: -
5	Form of assessment
	Oral examination
6	Condition for the award of credit points
	Module examination pass
7	Application of the module (in the following study programmes)
	Computer Science (M.Sc.)
8	Module coordinator
	All lecturers in the field of computer science who have passed at least the relevant master examination at a university or a comparable examination or who have acquired a comparable qualification and who have exercised relevant independent teaching activities in the master programme.
9	Other information