



UNIVERSITÄT
DES
SAARLANDES

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

MODULHANDBUCH

Eingebettete Systeme BSc

9. Juni 2021

Liste der Modulbereiche und Module

1	Pflichtbereich	3
1.1	Einführung in Eingebettete Systeme	4
1.2	Embedded Systems	5
1.3	Grundlagen der Elektrotechnik 1	7
1.4	Höhere Mathematik für Ingenieure I	8
1.5	Höhere Mathematik für Ingenieure II	9
1.6	Höhere Mathematik für Ingenieure III	10
1.7	Perspektiven der Informatik	11
1.8	Programmierung 1	12
1.9	Programmierung 2	14
1.10	Proseminar	16
1.11	Softwarepraktikum	18
2	Wahlpflichtbereich Informatik	20
2.1	Digital Transmission & Signal Processing	21
2.2	Grundzüge der Theoretischen Informatik	23
2.3	Grundzüge von Algorithmen und Datenstrukturen	25
2.4	Nebenläufige Programmierung	26
2.5	Systemarchitektur	29
3	Wahlpflichtbereich Engineering	31
3.1	Elektronische Systeme	32
3.2	Grundlagen der Automatisierungstechnik	33
3.3	Grundlagen der Signalverarbeitung	34
3.4	Mikroelektronik 1	35
3.5	Schaltungstechnik	36
3.6	Systemtheorie und Regelungstechnik 1	37
4	Stammvorlesungen	38
4.1	Audio/Visual Communication and Networks	39

4.2	Aufbau- und Verbindungstechnik 1	41
4.3	Automation Systems	42
4.4	Compiler Construction	43
4.5	Computational Electromagnetics 1	44
4.6	Computational Electromagnetics 2	45
4.7	Computer Architecture	46
4.8	Data Networks	48
4.9	Digital Signal Processing	50
4.10	Digital Transmission & Signal Processing	51
4.11	Distributed Systems	53
4.12	Einführung in die elektromagnetische Feldsimulation	54
4.13	Elektrische Antriebe	55
4.14	Elektronik / Bauelemente	56
4.15	High Frequency Engineering	57
4.16	High Speed Electronics	58
4.17	Image Processing and Computer Vision	59
4.18	Messtechnik und Sensorik	61
4.19	Mikroelektronik 2	62
4.20	Mikroelektronik 3	63
4.21	Mikroelektronik 4	64
4.22	Mikromechanische Bauelemente	65
4.23	Mikrotechnologie	66
4.24	Multimedia Transport	67
4.25	Operating Systems	69
4.26	Pattern and Speech Recognition	71
4.27	Security	72
4.28	Software Engineering	73
4.29	Statistical Natural Language Processing	75
4.30	Systemtheorie und Regelungstechnik 1	76
4.31	Systemtheorie und Regelungstechnik 2	77
4.32	Theoretische Elektrotechnik 2	78
4.33	Verification	79
5	Bachelor-Seminar und -Arbeit	80
5.1	Bachelor-Arbeit	81
5.2	Bachelor-Seminar	82

Modulbereich 1

Pflichtbereich

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Bernd Finkbeiner, Ph.D

Dozent/inn/en Prof. Bernd Finkbeiner, Ph.D
Prof. Dr. Martina Maggio

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

- Written exam at the end of the course.
- Demonstration of the implemented system.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

The course is accompanied by a laboratory project, in which a non-trivial embedded system has to be realized.

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students should learn methods for the design, the implementation, and the validation of safety-critical embedded systems.

Inhalt

Embedded Computer Systems are components of a technical system, e.g. an air plane, a car, a household machine, a production facility. They control some part of this system, often called the plant, e.g. the airbag controller in a car controls one or several airbags. Controlling means obtaining sensor values and computing values of actuator signals and sending them.

Most software taught in programming courses is transformational, i.e. it is started on some input, computes the corresponding output and terminates. Embedded software is reactive, i.e. it is continuously active waiting for signals from the plant and issuing signals to the plant.

Many embedded systems control safety-critical systems, i.e. malfunctioning of the system will in general cause severe damage. In addition, many have to satisfy real-time requirements, i.e. their reactions to input have to be produced within fixed deadlines.

According to recent statistics, more than 99% of all processors are embedded. Processors in the ubiquitous PC are a negligible minority. Embedded systems have a great economical impact as most innovations in domains like avionics, automotive are connected to advances in computer control. On the other hand, failures in the design of such systems may have disastrous consequences for the functioning of the overall system. Therefore, formal specification techniques and automatic synthesis of software are used more than in other domains.

The course will cover most aspects of the design and implementation of embedded systems, e.g. specification mechanisms, embedded hardware, operating systems, scheduling, validation methods.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Grundlagen der Elektrotechnik 1

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Studiendekan bzw. Studienbeauftragter der Fakultät NT

Dozent/inn/en Dozent/inn/en der Mathematik

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen benotete schriftliche Abschlussprüfung; Die Zulassung zur Prüfung erfordert die erfolgreiche Bearbeitung der Übungsaufgaben (Bekanntgabe der genauen Regeln zu Beginn der Lehrveranstaltung)

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzzeit Vorlesung und Übung
+ 120 h Vor- und Nachbereitung, Übungsbearbeitung
+ 60h Klausurvorbereitung
= 270 h (= 9 ECTS)

Modulnote Abschlussprüfungsnote

Sprache

Lernziele / Kompetenzen

Beherrschung der grundlegenden Begriffe, Methoden und Techniken der Analysis und linearen Algebra sowie die Fähigkeit, diese in ersten Anwendungen umzusetzen (auch mithilfe von Computern).

Inhalt

- Aussagen, Mengen und Funktionen
- Zahlbereiche: **N**, **Z**, **Q**, **R**, vollständige Induktion
- Kombinatorik, Gruppen, Körper
- Reelle Funktionen, Polynominterpolation
- Folgen, Reihen, Maschinenzahlen
- Funktionenfolgen, Potenzreihen, Exponentialfunktion
- Der **Rn**: Vektorraum, Geometrie und Topologie
- Die komplexen Zahlen

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, aktive Teilnahme an den Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Höhere Mathematik für Ingenieure II

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Höhere Mathematik für Ingenieure III

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Wintersemester	1 Semester	2	2

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Nachweis über das inhaltliche Verständnis von mindestens drei Vorträgen, z.B. durch schriftliche Ausarbeitung oder Test.

Lehrveranstaltungen / SWS 2 SWS Vorlesung

Arbeitsaufwand 30 h Präsenzstudium
+ 30 h Eigenstudium
= 60 h (= 2 ECTS)

Modulnote Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde (unbenotet).

Sprache Deutsch / English

Lernziele / Kompetenzen

Frühzeitige Motivierung und Überblick über die zentralen wissenschaftlichen Fragestellungen der Informatik, sowie über die Kompetenzen der Saarbrücker Informatik.

Inhalt

Vorträge durch wöchentlich wechselnde Dozent/inn/en bieten einen Querschnitt durch die Forschungsthemen der Saarbrücker Informatik. Die Themen spannen einen attraktiven Bogen von aktuellster Forschung zu anspruchsvollen Problemen der industriellen Praxis.

Literaturhinweise

Material wird passend zu den zu den einzelnen Vorträgen bereitgestellt.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Perspectives in Computer Science*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka
 Prof. Dr.-Ing. Holger Hermanns
 Prof. Bernd Finkbeiner, Ph.D

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

- zwei Klausuren (Mitte und Ende der Vorlesungszeit)
- Die Note wird aus den Klausuren gemittelt und kann durch Leistungen in den Übungen verbessert werden.
- Eine Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
 + 2 SWS Übung
 = 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
 + 180 h Eigenstudium
 = 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch und Englisch

Lernziele / Kompetenzen

- höherstufige, getypte funktionale Programmierung anwenden können
- Verständnis rekursiver Datenstrukturen und Algorithmen, Zusammenhänge mit Mengenlehre
- Korrektheit beweisen und Laufzeit abschätzen
- Typabstraktion und Modularisierung verstehen
- Struktur von Programmiersprachen verstehen
- einfache Programmiersprachen formal beschreiben können
- einfache Programmiersprachen implementieren können
- anwendungsnahe Rechenmodelle mit maschinennahen Rechenmodellen realisieren können
- Praktische Programmiererfahrung, Routine im Umgang mit Interpretern und Übersetzern

Inhalt

- Funktionale Programmierung
- Algorithmen und Datenstrukturen (Listen, Bäume, Graphen; Korrektheitsbeweise; asymptotische Laufzeit)
- Typabstraktion und Module
- Programmieren mit Ausnahmen
- Datenstrukturen mit Zustand
- Struktur von Programmiersprachen (konkrete und abstrakte Syntax, statische und dynamische Syntax)
- Realisierung von Programmiersprachen (Interpreter, virtuelle Maschinen, Übersetzer)

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Programming 1*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Sommersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Sebastian Hack

Dozent/inn/en Prof. Dr. Sebastian Hack
Prof. Dr. Jörg Hoffmann

Zulassungsvoraussetzungen *Programmierung 1* und *Mathematik für Informatiker 1* und Mathematikveranstaltungen im Studiensemester oder vergleichbare Kenntnisse aus sonstigen Mathematikveranstaltungen (empfohlen)

Leistungskontrollen / Prüfungen Prüfungsleistungen werden in zwei Teilen erbracht, die zu gleichen Teilen in die Endnote eingehen. Um die Gesamtveranstaltung zu bestehen, muss jeder Teil einzeln bestanden werden.

Im **Praktikumsteil** müssen die Studierenden eine Reihe von Programmieraufgaben selbstständig implementieren. Diese Programmieraufgaben ermöglichen das Einüben der Sprachkonzepte und führen außerdem komplexere Algorithmen und Datenstrukturen ein. Automatische Tests prüfen die Qualität der Implementierungen. Die Note des Praktikumsteils wird maßgeblich durch die Testergebnisse bestimmt.

Im **Vorlesungsteil** müssen die Studierenden Klausuren absolvieren und Übungsaufgaben bearbeiten. Die Aufgaben vertiefen dabei den Stoff der Vorlesung. Die Zulassung zu der Klausur hängt von der erfolgreichen Bearbeitung der Übungsaufgaben ab.

Im Praktikumsteil kann eine Nachaufgabe angeboten werden

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
+ 180 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

Sprache Deutsch und Englisch

Lernziele / Kompetenzen

Die Studierenden lernen die Grundprinzipien der imperativen /objektorientierten Programmierung kennen. Dabei wird primär Java als Programmiersprache verwendet.

In dieser Vorlesung lernen sie:

- wie Rechner Programme ausführen
- Die Grundlagen imperativer und objektorientierter Sprachen
- kleinere, wohlstrukturierte Programme in C zu schreiben
- mittelgroße objektorientierte Systeme in Java zu implementieren und zu testen
- sich in wenigen Tagen eine neue imperative/objektorientierte Sprache anzueignen, um sich in ein bestehendes Projekt einzuarbeiten

Inhalt

- Imperatives Programmieren
- Objekte und Klassen
- Klassendefinitionen
- Objektinteraktion
- Objektsammlungen
- Objekte nutzen und testen
- Vererbung
- Dynamische Bindung
- Fehlerbehandlung
- Klassendesign und Modularität
- Systemnahe Programmierung

sowie spezifische Vorlesungen für die Programmieraufgaben.

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Programming 2*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Semester	1 Semester	2	5

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen Grundlegende Kenntnisse im jeweiligen Teilbereich des Studienganges.

Leistungskontrollen / Prüfungen

- Thematischer Vortrag mit anschließender Diskussion
- Aktive Teilnahme an der Diskussion
- Gegebenenfalls kurze schriftliche Ausarbeitung und/oder Projekt

Lehrveranstaltungen / SWS 2 SWS Proseminar

Arbeitsaufwand 30 h Präsenzstudium
+ 120 h Eigenstudium
= 150 h (= 5 ECTS)

Modulnote Wird aus den Leistungen im Vortrag und der schriftlichen Ausarbeitung und/oder dem Seminarprojekt ermittelt. Die genauen Modalitäten werden von dem/der jeweiligen Dozenten/in bekannt gegeben.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Die Studierenden haben am Ende der Veranstaltung ein grundlegendes Verständnis aktueller oder fundamentaler Aspekte eines spezifischen Teilbereiches der Informatik erlangt.

Sie haben insbesondere grundlegende Kompetenz im eigenständigen wissenschaftlichen Recherchieren, Einordnen, Zusammenfassen, Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.

Im Vergleich zum Seminar liegt der Fokus beim Proseminar auf der Aneignung der grundlegenden wissenschaftlichen Arbeitsweisen.

Inhalt

Unter Anleitung werden folgende Punkte praktisch geübt:

- Lesen und Verstehen wissenschaftlicher Arbeiten
- Diskutieren der Arbeiten in der Gruppe
- Analysieren, Zusammenfassen und Wiedergeben des spezifischen Themas
- Präsentationstechnik

Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.

Der typische Ablauf eines Proseminars ist üblicherweise wie folgt:

- Vorbereitende Gespräche zur Themenauswahl
- Regelmäßige Treffen mit Diskussion ausgewählter Beiträge
- ggf. Bearbeitung eines themenbegleitenden Projekts
- Vortrag und ggf. Ausarbeitung zu einem der Beiträge

Literaturhinweise

Material wird dem Thema entsprechend ausgewählt.

Weitere Informationen

Die jeweils zur Verfügung stehenden Proseminare werden vor Beginn des Semesters angekündigt und unterscheiden sich je nach Studiengang.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		Vorlesungsfr. Zeit nach dem SS	7 Wochen	BLOCK	9

Modulverantwortliche/r Prof. Dr. Sven Apel

Dozent/inn/en Prof. Dr. Sven Apel

Zulassungsvoraussetzungen Die Teilnahme am Softwarepraktikum setzt umfangreiche Programmierkenntnisse voraus, wie sie in den Vorlesungen *Programmierung 1* und *Programmierung 2* vermittelt werden.

Für die Teilnahme am Softwarepraktikum werden eigene Laptops benötigt, die selbst mitgebracht werden müssen.

Leistungskontrollen / Prüfungen Das Ziel des Softwarepraktikums ist es, in einer Gruppe von Studierenden ein nicht-triviales Softwaresystem zu erstellen. Dazu müssen eine Reihe von Dokumenten (Entwurf, Quellcode, Tests, etc.) erstellt und abgegeben werden. Bewertet wird die Korrektheit und Qualität der Dokumente sowie die fristgerechte Abgabe.

Das Softwarepraktikum gliedert sich in eine Übungsphase, eine Gruppenphase und eine Einzelphase. In der Übungsphase werden täglich Minitests zu den aktuellen Vorlesungsinhalten durchgeführt und bewertet.

In der Gruppenphase wird ein substantielles Softwaresystem im Team geplant, entworfen, implementiert und getestet. Um zur Gruppenphase zugelassen zu werden, müssen die Studierenden die Minitests der Übungsphase bestehen.

In der Einzelphase wird ein kleineres Softwaresystem oder eine Erweiterung eines bestehenden Systems (z.B. aus der Gruppenphase) von den Studierenden jeweils allein entwickelt. Voraussetzung für die Einzelphase ist die erfolgreiche Absolvierung der Gruppenphase.

Die Softwaresysteme der Gruppen- und Einzelphase, sowie die zugehörigen Dokumente (Entwurf, Quellcode, Tests, etc.), werden auf Basis der Prinzipien und Qualitätsstandards der Vorlesung bewertet. Genauere Prüfungsmodalitäten werden zu Beginn des Softwarepraktikums in der Vorlesung bekannt gegeben.

Lehrveranstaltungen / SWS täglich Projektarbeit mit Betreuung
teilweise Vorlesung

Arbeitsaufwand 35 h Vorlesung
+ 235 h Projektarbeit
= 270 h (= 9 ECTS)

Modulnote unbenotet

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden erwerben die Fähigkeit, im Team zu arbeiten und Probleme des Software Engineerings zu lösen.

Die Studierenden wissen, welche Probleme beim Durchführen eines Softwareprojekts auftreten können, und wie diese gelöst werden können.

Sie können eine komplexe Aufgabenstellung eigenständig in ein Softwareprodukt umsetzen, das den Anforderungen des Kunden entspricht. Hierfür wählen sie einen passenden Entwicklungsprozess, der Risiken früh erkannt und minimiert, und wenden diesen an.

Sie sind vertraut mit Grundzügen des Softwareentwurfs wie schwache Kopplung, hohe Kohäsion, Geheimnisprinzip sowie Entwurfs- und Architekturmustern und sind in der Lage, einen Entwurf anhand dieser Kriterien zu erstellen, zu beurteilen und zu verbessern.

Sie beherrschen Techniken der Qualitätssicherung wie Testen und Debugging und wenden diese an.

Inhalt

- Softwareentwurf
- Softwaretesten
- Teamarbeit
- Debugging

Literaturhinweise

- Software Engineering. I. Sommerville, Addison-Wesley, 2004.
- Software Engineering: A Practitioner's Approach. R. Pressman, McGraw Hill Text, 2001.
- Using UML: Software Engineering with Objects and Components. P. Stevens, et al., Addison-Wesley, 1999.
- UML Distilled. M. Fowler, et al., Addison-Wesley, 2000.
- Objects, Components and Frameworks with UML, D. D'Souza, et al., Addison-Wesley, 1999.
- Designing Object-Oriented Software. R. Wirfs-Brock, et al., Prentice Hall, 1990.
- Design Patterns. Elements of Reusable Object-Oriented Software. E. Gamma, et al., Addison-Wesley, 1995.
- Head First Design Patterns. E. Freeman, et al. O'Reilly, 2004.
- Software Architecture: Perspectives on an Emerging Discipline. M. Shaw, et al., Prentice-Hall, 1996.
- Refactoring: Improving the Design of Existing Code. M. Fowler, et al., Addison-Wesley, 1999.
- Software Testing and Analysis: Process, Principles and Techniques. M. Pezze, Wiley. 2007.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Software Engineering Lab*.

Modulbereich 2

Wahlpflichtbereich Informatik

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en Prof. Dr.-Ing. Thorsten Herfet

Zulassungsvoraussetzungen The lecture requires a solid foundation of mathematics (differential and integral calculus) and probability theory. The course will, however, refresh those areas indispensably necessary for telecommunications and potential intensification courses and by this open this potential field of intensification to everyone of you.

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorials
 Passing the final exam in the 2nd week after the end of courses.
 Eligibility: Weekly exercises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.

Lehrveranstaltungen / SWS 4 h lectures
 + 2 h tutorial
 = 6 h (weekly)

Arbeitsaufwand 90 h of classes
 + 180 h private study
 = 270 h (= 9 ECTS)

Modulnote Final exam mark

Sprache English

Lernziele / Kompetenzen

Digital Signal Transmission and Signal Processing refreshes the foundation laid in "Signals and Systems" [Modulkennung]. Including, however, the respective basics so that the various facets of the introductory study period (Bachelor in Computer Science, Vordiplom Computer- und Kommunikationstechnik, Elektrotechnik or Mechatronik) and the potential main study period (Master in Computer Science, Diplom-Ingenieur Computer- und Kommunikationstechnik or Mechatronik) will be paid respect to.

Inhalt

As the basic principle, the course will give an introduction into the various building blocks that modern telecommunication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords, but we will also deal with dedicated pieces like A/D- and D/A-converters and quantizers in a little bit more depth.

The course will refresh the basic transformations (Fourier, Laplace) that give access to system analysis in the frequency domain, it will introduce derived transformations (z, Hilbert) for the analysis of discrete systems and modulation schemes and it will briefly introduce algebra on finite fields to systematically deal with error correction schemes that play an important role in modern communication systems.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Weitere Informationen

This module was formerly also known as *Telecommunications I*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Raimund Seidel

Dozent/inn/en Prof. Dr. Raimund Seidel
 Prof. Dr. Bernd Finkbeiner
 Prof. Dr. Kurt Mehlhorn
 Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen *Programmierung 1 und 2* und *Mathematik für Informatiker 1 und 2* oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.

Leistungskontrollen / Prüfungen Erfolgreiche Bearbeitung der Übungsaufgaben berechtigt zur Klausurteilnahme.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
 + 2 SWS Übung
 = 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
 + 180 h Eigenstudium
 = 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden kennen verschiedene Rechenmodelle und ihre relativen Stärken und Mächtigkeiten.

Sie können für ausgewählte Probleme zeigen, ob diese in bestimmten Rechenmodellen lösbar sind oder nicht.

Sie verstehen den formalen Begriff der Berechenbarkeit wie auch der Nicht-Berechenbarkeit.

Sie können Probleme aufeinander reduzieren.

Sie sind vertraut mit den Grundzügen der Ressourcenbeschränkung (Zeit, Platz) für Berechnungen und der sich daraus ergebenden Komplexitätstheorie.

Inhalt

Die Sprachen der Chomsky Hierarchie und ihre verschiedenen Definitionen über Grammatiken und Automaten; Abschlusseigenschaften; Klassifikation von bestimmten Sprachen („Pumping lemmas“);

Determinismus und Nicht-Determinismus;

Turing Maschinen und äquivalente Modelle von allgemeiner Berechenbarkeit (z.B. μ -rekursive Funktionen, Random Access Machines) Reduzierbarkeit, Entscheidbarkeit, Nicht-Entscheidbarkeit;

Die Komplexitätsmaße Zeit und Platz; die Komplexitätsklassen P und NP;

Grundzüge der Theorie der NP-Vollständigkeit

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Introduction to Theoretical Computer Science*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Wintersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Raimund Seidel

Dozent/inn/en Prof. Dr. Raimund Seidel
 Prof. Dr. Kurt Mehlhorn
 Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen *Programmierung 1 und 2, und Mathematik für Informatiker 1 und 2* oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.

Leistungskontrollen / Prüfungen Erfolgreiche Bearbeitung der Übungsblätter berechtigt zur Klausurteilnahme.

Lehrveranstaltungen / SWS 2 SWS Vorlesung
 + 2 SWS Übung
 = 4 SWS

Arbeitsaufwand 60 h Präsenzstudium
 + 120 h Eigenstudium
 = 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden lernen die wichtigsten Methoden des Entwurfs von Algorithmen und Datenstrukturen kennen: Teile-und-Herrsche, Dynamische Programmierung, inkrementelle Konstruktion, „Greedy“, Dezimierung, Hierarchisierung, Randomisierung. Sie lernen Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell zu analysieren und auf Basis dieser Analysen zu vergleichen. Sie lernen verschiedene Arten der Analyse (schlechtester Fall, amortisiert, erwartet) einzusetzen.

Die Studierenden lernen wichtige effiziente Datenstrukturen und Algorithmen kennen. Sie sollen die Fähigkeit erwerben, vorhandene Methoden durch theoretische Analysen und Abwägungen für ihre Verwendbarkeit in tatsächlich auftretenden Szenarien zu prüfen. Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen und Datenstrukturen unter dem Aspekt von Performanzgarantien zu entwickeln oder anzupassen

Inhalt

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Fundamentals of Data Structures and Algorithms*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Sommersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D
Prof. Dr. Verena Wolf

Zulassungsvoraussetzungen *Programmierung 1 und 2, Softwarepraktikum, und Grundzüge der Theoretischen Informatik* (empfohlen)

Leistungskontrollen / Prüfungen Zwei Klausuren (Mitte und Ende der Vorlesungszeit), praktisches Projekt.
Nachklausuren finden innerhalb der letzten Wochen vor dem Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS **Element T – Theorie (2 SWS):**
8 Vorlesungen: 6 Wochen
4 Übungen: 6 Wochen
Element A – Anwendung (2 SWS):
9 Vorlesungen: 6 Wochen
4 Übungen: 6 Wochen
Element P – Praxis (Eigenstudium):
Semesterbegleitend 8 schriftliche Reflektionen (Prüfungsvorleistungen),
anschließend Projektarbeit über ca. 2 Wochen
= 4 SWS

Arbeitsaufwand **Element T:**
24 h Präsenz, 36 h Selbststudium
Element A:
26 h Präsenz, 34 h Selbststudium
Element P:
60 h Selbststudium
50 h Präsenzstudium
+ 130 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren (im Anschluss an die Elemente T und A), sowie den Prüfungsvorleistungen (Element P) ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben. Alle Modulelemente sind innerhalb eines Prüfungszeitraumes erfolgreich zu absolvieren.

Sprache Englisch

Lernziele / Kompetenzen

Die Teilnehmer lernen die Nebenläufigkeit von Berechnungen als ein weitreichendes, grundlegendes Prinzip in der Theorie und Anwendung der modernen Informatik kennen. Durch die Untersuchung und Verwendung unterschiedlicher formaler Modelle gewinnen die Teilnehmer ein vertieftes Verständnis von Nebenläufigkeit. Dabei lernen die Teilnehmer wichtige formale Konzepte der Informatik korrekt anzuwenden. Das im ersten Teil der Veranstaltung erworbene theoretische Wissen wird in der zweiten Hälfte in der (Programmier-)Praxis angewendet. Dabei lernen die Teilnehmer Verwendung der Programmierparadigmen „Shared Memory“ und „Message Passing“ zuerst gemeinsam in der Programmiersprache `pseuCo`, bevor sie dann diese Fähigkeiten auf Java und teilweise Go übertragen. Außerdem lernen die Teilnehmer verschiedene Phänomene

des nebenläufigen Programmierens in den formalen Modellen zu beschreiben und mit deren Hilfe konkrete Lösungen für die Praxis abzuleiten. Des Weiteren untersuchen die Teilnehmer in der Praxis existierende Konzepte auf ihre Verlässlichkeit hin. Ein spezifischer Aspekt dieser beruflichen Praxis ist das taktisch adäquate Reagieren auf Problemstellungen der Nebenläufigkeit unter engen Zeitvorgaben.

Inhalt

Nebenläufigkeit als Konzept

- Potentieller Parallelismus
- Tatsächlicher Parallelismus
- Konzeptioneller Parallelismus

Nebenläufigkeit in der Praxis

- Objektorientierung
- Betriebssysteme
- Multi-core Prozessoren, Coprozessoren
- Programmierte Parallelität
- Verteilte Systeme (Client-Server, Peer-to-Peer, Datenbanken, Internet)

Die Schwierigkeit von Nebenläufigkeit

- Ressourcenkonflikte
- Fairness
- Gegenseitiger Ausschluss
- Verklemmung (Deadlock)
- gegenseitige Blockaden (Livelock)
- Verhungern (Starvation)

Grundlagen der Nebenläufigkeit

- Sequentielle vs. Nebenläufige Prozesse
- Zustände, Ereignisse und Transitionen
- Transitionssysteme
- Beobachtbares Verhalten
- Determinismus vs. Nicht-Determinismus
- Algebren und Operatoren

CCS: Der Kalkül kommunizierender Prozesse

- Konstruktion von Prozessen: Sequenz, Auswahl, Rekursion
- Nebenläufigkeit und Interaktion
- Strukturelle operationelle Semantik
- Gleichheit von Beobachtungen
- Implementierungsrelationen
- CCS mit Datentransfer

Programmieren von Nebenläufigkeit

- pseuCo
- Message Passing in pseuCo und Go
- Shared Memory in pseuCo und Java
- Monitore und Semaphoren
- Shared Objects und Threads in Java
- Shared Objects und Threads als Transitionssysteme

Programmier- und Analyseunterstützung

- Erkennung von Verklemmungen
- Zusicherung von Sicherheit und Lebendigkeit
- Model-Basiertes Design von Nebenläufigkeit
- Software Architekturen für Nebenläufigkeit

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Concurrent Programming*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Sommersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Jan Reineke

Dozent/inn/en Prof. Dr. Jan Reineke

Zulassungsvoraussetzungen *Programmierung 1, Programmierung 2* (im selben Semester) und *Mathematik für Informatiker 1* oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.

Leistungskontrollen / Prüfungen Prüfungsleistungen werden in zwei Teilen erbracht, die beide in die Endnote eingehen. Um die Gesamtveranstaltung zu bestehen, muss jeder Teil einzeln bestanden werden.

Im *Projektteil* müssen die Studierenden eine Reihe von Projekten selbstständig bearbeiten. Diese Projekte vertiefen das praktische Verständnis des Vorlesungsstoffes in den Bereichen Rechnerarchitektur und Betriebssysteme.

Im *Vorlesungsteil* müssen die Studierenden Klausuren absolvieren und Übungsaufgaben und/oder Minitests bearbeiten. Die erfolgreiche Bearbeitung der Übungsblätter beziehungsweise der Minitests ist Voraussetzung zur Teilnahme an der Klausur.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
+ 180 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen, Minitests und praktischen Projekten ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden sollen die Funktionsweise und die wichtigsten Eigenschaften moderner Rechnerarchitekturen und Betriebssystemen kennenlernen.

Außerdem sollen die Studierenden die der Implementierung solcher Systeme zugrundeliegenden Entwurfsprinzipien verstehen.

Inhalt

1. Rechnerarchitektur
 - a. Boolesche Algebra und Schaltkreise
 - b. Zahlendarstellungen und arithmetische Schaltkreise
 - c. Befehlssatzarchitekturen
 - d. Mikroarchitekturen, insbesondere der Entwurf eines einfachen Reduced Instruction Set Computers, sowie Techniken zur Leistungsoptimierung.
2. Betriebssysteme
 - a. Virtualisierungsmechanismen
 - b. Planungsalgorithmen
 - c. Dateisysteme

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *System Architecture*.

Modulbereich 3

Wahlpflichtbereich Engineering

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Grundlagen der Automatisierungstechnik

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Grundlagen der Signalverarbeitung

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Systemtheorie und Regelungstechnik 1

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Modulbereich 4

Stammvorlesungen

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en Prof. Dr.-Ing. Thorsten Herfet

Zulassungsvoraussetzungen Solid foundation of mathematics (differential and integral calculus) and probability theory. The course will build on the mathematical concepts and tools taught in TC I while trying to enable everyone to follow and to fill gaps by an accelerated study of the accompanying literature. *Signals and Systems* as well as *Digital Transmission and Signal Processing (TC I)* are strongly recommended but not required.

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorials Passing the final exam
Oral exam directly succeeding the course. Eligibility: Weekly excersises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Final Exam Mark

Sprache English

Lernziele / Kompetenzen

AVCN will deepen the students' knowledge on modern communications systems and will focus on wireless systems.

Since from a telecommunications perspective the combination of audio/visual data – meaning inherently high data rate and putting high requirements on the realtime capabilities of the underlying network – and wireless transmission – that is unreliable and highly dynamic with respect to the channel characteristics and its capacity – is the most demanding application domain.

Inhalt

As the basic principle the course will study and introduce the building blocks of wireless communication systems. Multiple access schemes like TDMA, FDMA, CDMA and SDMA are introduced, antennas and propagation incl. link budget calculations are dealt with and more advanced channel models like MIMO are investigated. Modulation and error correction technologies presented in Telecommunications I will be expanded by e.g. turbo coding and receiver architectures like RAKE and BLAST will be introduced. A noticeable portion of the lecture will present existing and future wireless networks and their extensions for audio/visual data. Examples include 802.11n and the terrestrial DVB system (DVB-T2).

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Weitere Informationen

This module was formerly also known as *Telecommunications II*.

Aufbau- und Verbindungstechnik 1

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Sebastian Hack

Dozent/inn/en Prof. Dr. Sebastian Hack

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Written exam at the end of the course, theoretical exercises, and compiler-laboratory project.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
 + 2 h tutorial
 = 6 h (weekly)

Arbeitsaufwand 90 h of classes
 + 180 h private study
 = 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students learn, how a source program is lexically, syntactically, and semantically analyzed, and how they are translated into semantically equivalent machine programs. They learn how to increase the efficiency by semantics-preserving transformations. They understand the automata-theoretic foundations of these tasks and learn, how to use the corresponding tools.

Inhalt

Lexical, syntactic, semantic analysis of source programs, code generation for abstract and real machines, efficiency-improving program transformations, foundations of program analysis.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Computational Electromagnetics 1

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Computational Electromagnetics 2

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. W.-J. Paul

Dozent/inn/en Prof. Dr. W.-J. Paul

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen **Studying:** Students should listen to the lectures, read the lecture notes afterwards and understand them. They should solve the exercises alone or in groups. Students must present and explain their solutions during the tutorials.

Exams: Students who have solved 50 % of all exercises are allowed to participate in an oral exam at the end of the semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

After attending this lecture students know how to design pipelined processors with interrupt mechanisms, caches and MMUs. Given a benchmark they know how to analyse, whether a change makes the processor more or less cost effective.

Inhalt

General comment: constructions are usually presented together with correctness proofs

- Complexity of Architectures
 - Hardware cost and cycle time
 - Compilers and benchmarks
- Circuits
 - Elementary computer arithmetic
 - Fast adders
 - Fast multipliers
- Sequential processor design
 - DLX instruction set
 - Processor design
- Pipelining
 - Elementary pipelining
 - Forwarding
 - Hardware-Interlock
- Interrupt mechanisms
 - Extension of the instruction set
 - Interrupt service routines

- hardware construction
- Caches
 - Specification including consistency between instruction and data cache
 - Cache policies
 - Bus protocol
 - Hardware construction (k-way set associative cache, LRU replacement, realisation of bus protocols by automat)
- Operating System Support
 - Virtual and Physical machines
 - Address translation
 - Memory management unit (MMU) construction
 - Virtual memory simulation

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Dr. Anja Feldmann

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Qualification for final exam through mini quizzes during classes
- Possibility to get bonus points through excellent homework
- Final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

After taking the course students have

- a thorough knowledge regarding the basic principles of communication networks,
- the fundamentals of protocols and concepts of protocol,
- Insights into fundamental motivations of different pragmatics of current network solutions,
- Introduction to practical aspects of data networks focusing on internet protocol hierarchies

Inhalt

Introduction and overview

Cross section:

- Stochastic Processes, Markov models,
- Fundamentals of data network performance assessment
- Principles of reliable data transfer
- Protokols and their elementary parts
- Graphs and Graphalgorithms (maximal flow, spanning tree)
- Application layer:
- Services and protocols
- FTP, Telnet
- Electronic Mail (Basics and Principles, SMTP, POP3, ..)
- World Wide Web (History, HTTP, HTML)

- Transport Layer:
 - Services and protocols
 - Addressing
 - Connections and ports
 - Flow control
 - QoS
 - Transport Protocols (UDP, TCP, SCTP, Ports)
- Network layer:
 - Services and protocols
 - Routing algorithms
 - Congestion Control
 - Addressing
 - Internet protocol (IP)
- Data link layer:
 - Services and protocols
 - Medium access protocols: Aloha, CSMA (-CD/CA), Token passing
 - Error correcting codes
 - Flow control
 - Applications: LAN, Ethernet, Token Architectures, WLAN, ATM
- Physical layer
 - Peer-to-Peer and Ad-hoc Networking Principles

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en Prof. Dr.-Ing. Thorsten Herfet

Zulassungsvoraussetzungen The lecture requires a solid foundation of mathematics (differential and integral calculus) and probability theory. The course will, however, refresh those areas indispensably necessary for telecommunications and potential intensification courses and by this open this potential field of intensification to everyone of you.

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorials
 Passing the final exam in the 2nd week after the end of courses.
 Eligibility: Weekly exercises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.

Lehrveranstaltungen / SWS 4 h lectures
 + 2 h tutorial
 = 6 h (weekly)

Arbeitsaufwand 90 h of classes
 + 180 h private study
 = 270 h (= 9 ECTS)

Modulnote Final exam mark

Sprache English

Lernziele / Kompetenzen

Digital Signal Transmission and Signal Processing refreshes the foundation laid in "Signals and Systems" [Modulkennung]. Including, however, the respective basics so that the various facets of the introductory study period (Bachelor in Computer Science, Vordiplom Computer- und Kommunikationstechnik, Elektrotechnik or Mechatronik) and the potential main study period (Master in Computer Science, Diplom-Ingenieur Computer- und Kommunikationstechnik or Mechatronik) will be paid respect to.

Inhalt

As the basic principle, the course will give an introduction into the various building blocks that modern telecommunication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords, but we will also deal with dedicated pieces like A/D- and D/A-converters and quantizers in a little bit more depth.

The course will refresh the basic transformations (Fourier, Laplace) that give access to system analysis in the frequency domain, it will introduce derived transformations (z, Hilbert) for the analysis of discrete systems and modulation schemes and it will briefly introduce algebra on finite fields to systematically deal with error correction schemes that play an important role in modern communication systems.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Weitere Informationen

This module was formerly also known as *Telecommunications I*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Peter Druschel, Ph.D.

Dozent/inn/en Prof. Peter Druschel, Ph.D.
Allen Clement, Ph.D

Zulassungsvoraussetzungen *Operating Systems or Concurrent Programming*

Leistungskontrollen / Prüfungen

- Regular attendance at classes and tutorials.
- Successful completion of a course project in teams of 2 students. (Project assignments due approximately every 2 weeks.)
- Passing grade on 2 out of 3 written exams: midterm, final exam, and a re-exam that takes place during the last two weeks before the start of lectures in the following semester.
- Final course grade: 50% project, 50% best 2 out of 3 exams.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Introduction to the principles, design, and implementation of distributed systems.

Inhalt

- Communication: Remote procedure call, distributed objects, event notification, Inhalt dissemination, group communication, epidemic protocols.
- Distributed storage systems: Caching, logging, recovery, leases.
- Naming. Scalable name resolution.
- Synchronization: Clock synchronization, logical clocks, vector clocks, distributed snapshots.
- Fault tolerance: Replication protocols, consistency models, consistency versus availability trade-offs, state machine replication, consensus, Paxos, PBFT.
- Peer-to-peer systems: consistent hashing, self-organization, incentives, distributed hash tables, Inhalt distribution networks.
- Data centers. Architecture and infrastructure, distributed programming, energy efficiency.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Einführung in die elektromagnetische Feldsimulation

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Elektrische Antriebe

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert

Zulassungsvoraussetzungen Undergraduate mathematics (e.g. Mathematik für Informatiker I-III) and elementary programming knowledge in C

Leistungskontrollen / Prüfungen

- For the homework assignments one can obtain up to 24 points per week. Actively participating in the classroom assignments gives 12 more points per week, regardless of the correctness of the solutions. To qualify for both exams one needs 2/3 of all possible points.
- Passing the final exam or the re-exam.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from the performance in the exam or the re-exam. The better grade counts.

Sprache English

Lernziele / Kompetenzen

Broad introduction to mathematical methods in image processing and computer vision. The lecture qualifies students for a bachelor thesis in this field. Together with the completion of advanced or specialised lectures (9 credits at least) it is the basis for a master thesis in this field.

Inhalt

Inhalt

1. Basics
 - 1.1 Image Types and Discretisation
 - 1.2 Degradations in Digital Images
2. Colour Perception and Colour Spaces
3. Image Transformations
 - 3.1 Continuous Fourier Transform
 - 3.2 Discrete Fourier Transform
 - 3.3 Image Pyramids
 - 3.4 Wavelet Transform
4. Image Compression
5. Image Interpolation
6. Image Enhancement
 - 6.1 Point Operations

- 6.2 Linear Filtering and Feature Detection
- 6.3 Morphology and Median Filters
- 6.3 Wavelet Shrinkage, Bilateral Filters, NL Means
- 6.5 Diffusion Filtering
- 6.6 Variational Methods
- 6.7 Deconvolution Methods
- 7. Texture Analysis
- 8. Segmentation
 - 8.1 Classical Methods
 - 8.2 Variational Methods
- 9. Image Sequence Analysis
 - 9.1 Local Methods
 - 9.2 Variational Methods
- 10. 3-D Reconstruction
 - 10.1 Camera Geometry
 - 10.2 Stereo
 - 10.3 Shape-from-Shading
- 11. Object Recognition
 - 11.1 Hough Transform
 - 11.2 Invariants
 - 11.3 Eigenspace Methods

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Mikromechanische Bauelemente

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		winter semester	1 semester	4	9

Modulverantwortliche/r Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en Prof. Dr.-Ing. Thorsten Herfet

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorial.
Paper as well as programming exercises for exam qualification
Final exam
A re-exam takes place before the start of lectures in the following semester

Lehrveranstaltungen / SWS 3 h lectures
+ 1 h tutorial
= 4 h (weekly)

Arbeitsaufwand 60 h of classes
+ 90 h private study
+ 120 h programming exercise
= 270 h (= 9 ECTS)

Modulnote Graded absolute 1.0-n.b. and relative A-F

Sprache English

Lernziele / Kompetenzen

The course deals with Media Transport over the Internet. After the course students know how data- and mediatransport is solved in today's Internet and have a good understanding of so called erasure channels.

Besides the pure transport protocol design the course complements the fundamentals laid in TCI and TCII by introducing state-of-the-art error codes (Van-der-Monde-Codes, Fountain Codes) and by engineering tasks like the design of a Digital PLL.

Inhalt

The course introduces media transmission over packet channels, specifically the Internet. After establishing a Quality of Service framework built on ITU requirements the course models erasure channels without and with memory. Key characteristics like the channel capacity and the minimum redundancy information are derived.

The second part of the course introduces current media transport protocol suites (TCP, UDP, RTP, RTSP) and middleware (ISMA, DLNA, UPnP, DVB-IPI).

In the second half of the course audiovisual coders used in the Internet are introduced (H.264, AAC), state-of-the-art forward error coding schemes (Van-der-Monde-Codes, Fountain Codes) are explained and essential elements like a Digital Phase-locked Loop are developed.

Literaturhinweise

The course will come with a self contained manuscript. The most essential monographs used for and referenced within the manuscript are available in the Computer Science Library of Saarland University.

Weitere Informationen

This module was formerly also known as *Future Media Internet*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Peter Druschel, Ph.D.

Dozent/inn/en Prof. Peter Druschel, Ph.D.
Björn Brandenburg, Ph.D

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen Regular attendance at classes and tutorials
Successful completion of a course project in teams of 2 students
Passing 2 written exams (midterm and final exam)
A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Introduction to the principles, design, and implementation of operating systems

Inhalt

Process management:

- Threads and processes, synchronization
- Multiprogramming, CPU Scheduling
- Deadlock

Memory management:

- Dynamic storage allocation
- Sharing main memory
- Virtual memory

I/O management:

- File storage management
- Naming
- Concurrency, Robustness, Performance

Virtual machines

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Pattern and Speech Recognition

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Michael Backes

Dozent/inn/en Prof. Dr. Michael Backes
Prof. Dr. Cas Cremers

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam is normally provided (as written or oral examination).

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

Description, assessment, development and application of security mechanisms, techniques and tools.

Inhalt

- Basic Cryptography,
- Specification and verification of security protocols,
- Security policies: access control, information flow analysis,
- Network security,
- Media security,
- Security engineering

Literaturhinweise

Will be announced on the course website

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Sven Apel

Dozent/inn/en Prof. Dr. Sven Apel

Zulassungsvoraussetzungen

- Knowledge of programming concepts (as taught in the lectures *Programmierung 1* and *Programmierung 2*)
- Basic knowledge of software processes, design, and testing (as taught and applied in the lecture *Softwarepraktikum*)

Leistungskontrollen / Prüfungen Beside the lecture and weekly practical exercises, there will be a number of assignments in the form of mini-projects for each student to work on (every two to three weeks). The assignments will be assessed based on the principles covered in the lecture. Passing all assignments is a prerequisite for taking the final written exam. The final grade is determined only by the written exam. Further examination details will be announced by the lecturer at the beginning of the course. In short:

- Passing all assignments (prerequisite for the written exam)
- Passing the written exam

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h exercises
= 6 h (weekly)

Arbeitsaufwand 90 h of classes and exercises
+ 180 h private study and assignments
= 270 h (= 9 ECTS)

Modulnote The grade is determined by the written exam. Passing all assignments is a prerequisite for taking the written exam. The assignments do not contribute to the final grade. Further examination details will be announced by the lecturer at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

- The students know and apply modern software development techniques.
- They are aware of key factors contributing to the complexity of real-world software systems, in particular, software variability, configurability, feature interaction, crosscutting concerns, and how to address them.
- They know how to apply established design and implementation techniques to master software complexity.
- They are aware of advanced design and implementation techniques, including collaboration-based design, mixins/traits, aspects, pointcuts, advice.
- They are aware of advanced quality assurance techniques that take the complexity of real-world software systems into account: variability-aware analysis, sampling, feature-interaction detection, predictive performance modeling, etc.
- They appreciate the role of non-functional properties and know how to predict and optimize software systems regarding these properties.
- They are able to use formal methods to reason about key techniques and properties covered in the lecture.

Inhalt

- Domain analysis, feature modeling
- Automated reasoning about software configuration using SAT solvers

- Runtime parameters, design patterns, frameworks
- Version control, build systems, preprocessors
- Collaboration-based design
- Aspects, pointcuts, advice
- Expression problem, preplanning problem, code scattering & tangling, tyranny of the dominant decomposition, inheritance vs. delegation vs. mixin composition
- Feature interaction problem (structural, control- & data-flow, behavioral, non-functional feature interactions)
- Variability-aware analysis and variational program representation (with applications to type checking and static program analysis)
- Sampling (random, coverage)
- Machine learning for software performance prediction and optimization

Literaturhinweise

- Feature-Oriented Software Product Lines: Concepts and Implementation. S. Apel, et al., Springer, 2013.
- Generative Programming: Methods, Tools, and Applications: Methods, Techniques and Applications. K. Czarnecki, et al., Addison-Wesley, 2000.
- Mastering Software Variability with FeatureIDE. J. Meinicke, et al., Springer, 2017.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		every summer semester	1 semester	4	6

Modulverantwortliche/r Prof. Dr. Dietrich Klakow

Dozent/inn/en Prof. Dr. Dietrich Klakow

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen Written Exam

Lehrveranstaltungen / SWS 2 h lectures
+ 2 h tutorial
= 4 h (weekly)

Arbeitsaufwand 60 h of classes
+ 120 h problem solving and private study
= 180 h (= 6 ECTS)

Modulnote Final Exam Mark

Sprache English

Lernziele / Kompetenzen

Acquire core competencies in the mathematical basics of language processing and practice the implementation of essential methods.

Inhalt

- language processing: basic terms
- mathematical foundations
- word sense disambiguation
- part-of-speech tagging
- named-entity recognition
- information retrieval
- text classification

Literaturhinweise

Chris Manning and Hinrich Schütze
Foundations of Statistical Natural Language Processing

Systemtheorie und Regelungstechnik 1

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Theoretische Elektrotechnik 2

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students become familiar with the standard methods in computer-aided verification. They understand the theoretical foundations and are able to assess the advantages and disadvantages of different methods for a specific verification project. The students gain first experience with manual correctness proofs and with the use of verification tools.

Inhalt

- models of computation and specification languages: temporal logics, automata over infinite objects, process algebra
- deductive verification: proof systems (e.g., Floyd, Hoare, Manna/Pnueli), relative completeness, compositionality
- model checking: complexity of model checking algorithms, symbolic model checking, abstraction case studies

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Modulbereich 5

Bachelor-Seminar und -Arbeit

Bachelor-Arbeit

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Semester	3 Monate	-	12

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Professoren der Fachrichtung

Zulassungsvoraussetzungen Erfolgreicher Abschluss des *Bachelor-Seminars*

Leistungskontrollen / Prüfungen Schriftliche Ausarbeitung. Sie beschreibt sowohl das Ergebnis der Arbeit als auch den Weg, der zu dem Ergebnis führte. Der eigene Anteil an den Ergebnissen muss klar erkennbar sein. Außerdem Präsentation der Bachelorarbeit in einem Kolloquium, in dem auch die Eigenständigkeit der Leistung des Studierenden überprüft wird.

Lehrveranstaltungen / SWS keine

Arbeitsaufwand 30 h Betreuung durch den Lehrstuhl
+ 330 h Eigenstudium
= 360 h (= 12 ECTS)

Modulnote Beurteilung der Bachelorarbeit durch die Gutachter.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Die Bachelor-Arbeit ist eine Projektarbeit, die unter Anleitung ausgeführt wird. Sie soll der Kandidaten/die Kandidatin in der Lage versetzen, innerhalb einer vorgegebenen Frist ein Problem aus dem Gebiet der Informatik selbständig zu lösen und die Ergebnisse in wissenschaftlich angemessener Form zu dokumentieren.

Inhalt

Bearbeitung einer aktuellen Problemstellung aus der Informatik unter Anleitung. Adäquate Dokumentation der Ergebnisse in Form einer wissenschaftlichen Abschlussarbeit.

Das Thema wird in enger Absprache mit dem anleitenden Dozenten definiert.

Literaturhinweise

Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem anleitenden Dozenten.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		jedes Semester	variabel	2	9

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen Erwerb von mindestens 120 CP

Leistungskontrollen / Prüfungen

- Schriftliche Ausarbeitung der Aufgabenstellung der Bachelorarbeit und der relevanten wissenschaftlichen Literatur
- Vortrag über die geplante Aufgabenstellung mit anschließender Diskussion
- Aktive Teilnahme an der Diskussion

Lehrveranstaltungen / SWS 2 SWS Seminar

Arbeitsaufwand 30 h Präsenzstudium (Seminar)
+ 30 h Betreuung durch den Lehrstuhl
+ 210 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus den Leistungen im Vortrag und der schriftlichen Ausarbeitung ermittelt. Die genauen Modalitäten werden von dem/der jeweiligen Dozenten/in bekannt gegeben.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Im Bachelorseminar erwirbt der Studierende unter Anleitung die Fähigkeit zum wissenschaftlichen Arbeiten im Kontext eines angemessenen Themengebietes.

Am Ende des Bachelorseminars sind die Grundlagen für eine erfolgreiche Anfertigung der Bachelorarbeit gelegt, und wesentliche Lösungsansätze bereits eruiert.

Das Bachelorseminar bereitet somit die Themenstellung und Ausführung der Bachelorarbeit vor.

Es vermittelt darüber hinaus praktische Fähigkeiten des wissenschaftlichen Diskurses. Diese Fähigkeiten werden durch die aktive Teilnahme an einem Lesekreis vermittelt, in welchem die Auseinandersetzung mit wissenschaftlich anspruchsvollen Themen geübt wird.

Inhalt

Einarbeitung in ein wissenschaftliches Themengebiet innerhalb der Informatik.

Anfertigung einer schriftlichen Ausarbeitung der Aufgabenstellung der Bachelorarbeit und der relevanten wissenschaftlichen Literatur.

Fachvortrag über das Themengebiet und die geplante Aufgabenstellung der Bachelorarbeit.

Das Thema wird in enger Absprache mit dem anleitenden Dozenten definiert.

Literaturhinweise

Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem Dozenten