Module im Bereich: Grundlagen Informatik

- Computer Graphics - an Introduction
- Database Concepts
- Introduction to Computer Science for Engineers
- Introduction to Simulation
- Logik 2
- Programmierparadigmen

Computer Graphics - an Introduction

Winter

(in German: Computer Graphics - an Introduction)

Module-ID: FIN-INF-110472

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
ICG	6	Winter	starting at 1.	1 Semester	english	Master

Link:	LSF
Limited accessability:	false
Responsibility:	Holger Theisel
Lecturer:	Holger Theisel
Classes:	• Lecture *Computer Graphics - an Introduction* • Exercise class *Computer Graphics - an Introduction*
Applicability in curriculum:	- M.Sc. DE: Grundlagen Informatik - M.Sc. VC: Visual Computing

Intended learning outcomes:
The students

- understand the basic concepts of computer graphics
- can apply their knowledge to create their own simple rendering program
- work in groups on exercises and programming tasks
- learn both the theoretical basics and applications of computer graphics
- are able to acquire in-depth knowledge of computer graphics after finishing the course

Content:

- Introduction, history, application areas of Computer graphics
- Modeling and acquisition of graphical data
- Transformations
- Clipping
- Rasterization and antialiasing
- Lighting
- Texturing
- Visibility
- Ray tracing
- Modern concepts of computer graphics at a glance

Workload:

- 56h contact time (lecture + exercise class)
- 94h self study (lecture material, solution of exercises and assignments)

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
• Successful completion of the exercises • Completing a programming task	Exam: Written exam 120 min. Exam certificate: Passing the written exam 120 min.	• Lectures (2 SWS) • Exercises (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	

Media:	Literature:
	• J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes: Computer Graphics – Principles and Practice (second Edition). AddisonWesley Publishing Company, Inc., 1996 • J.Encarnacao, W. D. Salomon: Computer Graphics Geometric Modeling, Springer, 1999A. • Watt: 3D Computer Graphics. Addison-Wesley Publishing Company, Inc., 2000

Comments:
(nur anrechenbar wenn nicht vorher die deutschsprachige Bachelor-LV Computergraphik I belegt

wurde)

Database Concepts

Summer

(in German: Database Concepts)

Module-ID: FIN-INF-120529

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
DB1	6	Summer	starting at 1.	1 Semester	english	Master

Link:	LSF
Limited accessability:	
Responsibility:	Gunter Saake
Lecturer:	Gunter Saake
Classes:	• Lecture *Database Concepts* • Exercise class *Database Concepts*
Applicability in curriculum:	- M.Sc. DE: Grundlagen Informatik

Intended learning outcomes:
The students ...

- understand basic concepts of database systems
- are able to model relational databases
- are familiar with relational database languages
- can develop database applications
- model relational databases based on complex and changing practical usage scenarios

Content:

- Properties of Database Systems

- Architectures
- Conceptual Design of Relational Databases
- Relational Database Model
- Mapping of ER-Schemas to Relations
- Database Languages (Relational Algebra, SQL)
- Formal Design Criteria and Normalization
- Database Application Programming
- Further Database Concepts, e.g., Views, Triggers, Access Rights

Workload:
56 contact hours + 124 h self study

Pre-examination requirements:	**Type of examination:**	**Teaching method / lecture hours per week (SWS):**
Successful participation in the exercise	Written exam (120 min)	• 2 SWS Lecture • 2 SWS Exercise

Prerequisites according to examination regulations:	**Recommended prerequisites:**
Cannot be attended together with “Datenbanken 1”	

Media:	**Literature:**
	• Datenbanken - Konzepte und Sprachen. Gunter Saake, Kai-Uwe Sattler, Andreas Heuer, March 2013, ISBN 3-8266-9453-8, MitpVerlag; Edition 5 • Fundamentals of Database Systems. Ramez Elmasri, Shamkant B. Navathe, April 2010, ISBN 0-136-08620-9, Addison Wesley; Edition 6

Comments:

Introduction to Computer Science for Engineers

Winter

(in German: Introduction to Computer Science for Engineers)

Module-ID: FIN-INF-120440

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
ICSE	6	Winter	starting at 1.	1 Semester	english	Bachelor

Link:	LSF
Limited accessability:	false
Responsibility:	Dr. Christian Braune
Lecturer:	Dr. Christian Braune
Classes:	• Lecture *Introduction to Computer Science for Engineers* • Exercise class *Introduction to Computer Science for Engineers*
Applicability in curriculum:	- M.Sc. DE: Grundlagen Informatik - M.Sc. DE: Informatik Grundlagen (SPO 2027)

Intended learning outcomes:
Students who finish this course

- know the basic principles of programming in an imperative, object-oriented programming language
- understand that design of an algorithm can impact its runtime

- know some of the most commonly used data structures and what their respective advantages and disadvantages are

Content:
Introduction to:

- imperative programming paradigm
- basic concepts of object-oriented programming
- programming in a commonly used programming language (e.g. Python)
- generic programming
- fundamental data structures:
  - trees (binary trees, search-trees and self-balancing search trees)
  - hash tables
  - graphs
- abstract data types: lists, stacks, queues
- main algorithms for fundamental tasks such as sorting and searching
- methods to observe algorithm complexity or performance (Big-O notation).
- fundamental types of algorithm design paradigms: Divide-and-Conquer, Greedy, Backtracking and Searching, and Dynamic Programming

Workload:
180 h (70 h contact hours + 110 h complementary reading and realization of the exercises/assignments)

Pre-examination requirements:	**Type of examination:**	**Teaching method / lecture hours per week (SWS):**
Prerequisites for admission: successful completion of assignments (voting & assessment). Details will be explained during the first lecture.	Written examination, 120 min	Lecture (2 SWS) Exercise (2 SWS)

Prerequisites according to examination regulations:	**Recommended prerequisites:**
keine	

Media:	**Literature:**
	Computer Science - An Interdisciplinary Approach, R. Sedgewick and K. Wayne, Addison-Wesley, 2016, ISBN 0-13-407642-7 Algorithms, 4th Edition, R. Sedgewick and K. Wayne, Addison-Wesley, 2011, ISBN 0-321-57351-X Data Structures and Algorithm in Java, 6th Edition, M.T. Goodrich and R. Tamassia and M.H. Goldwasser, Wiley, 2014, ISBN 1-118-77133-4

Comments:

Introduction to Simulation

Winter

(in German: Introduction to Simulation)

Module-ID: FIN-INF-120345

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
ItS	6	Winter	starting at 1.	1 Semester	english	Master

Link:	LSF
Limited accessability:	
Responsibility:	Graham Horton
Lecturer:	Graham Horton
Classes:	• Lecture *Introduction to Simulation* • Exercise class *Introduction to Simulation*
Applicability in curriculum:	- M.Sc. DKE: Fundamentals of Data Science - M.Sc. DE: Grundlagen Informatik - M.Sc. DE: Informatik Grundlagen (SPO 2027)

Intended learning outcomes:
Students who complete the course ...

- can analyse discrete, continuous and hybrid systems using a professional simulation tool
- know how to obtain an accurate result efficiently when integrating initial value problems numerically
- can carry out input and output analysis to obtain statistically significant results for stochastic, discrete-event models
- can select an appropriate modelling paradigm for a given situation
- can perform a simulation study and interpret its results

Content:
Introduction to the fundamentals and steps in a simulation study:

- discrete-event simulation
- random variables and random number generation
- statistical data analysis
- ordinary differential equations and numerical integration
- stochastic Petri nets
- the AnyLogic simulation system
- discrete-time Markov chains
- agent-based simulation

Workload:
56 contact hours + 124 h self study

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	2h written exam	• 2 SWS lecture • 2 SWS exercise class

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	Basic engineering mathematics

Media:	Literature:
	Banks, Carson, Nelson, Nicol: Discrete-Event System Simulation

Comments:

Logik 2

Winter

(engl. Logic 2)

Modulnummer: FIN-INF-120540

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
Logik2	10	Winter	1.	1 Semester	deutsch	Master

Link zum LSF:	LSF
Zugang beschränkt:	true
Verantwortung:	Fabian Neuhaus
Dozent:in:	Fabian Neuhaus
Lehrveranstaltungen:	Vorlesung Logik 2 Übung Logik 2
Verwendbarkeit:	- M.Sc. INF: Informatik - M.Sc. INF: Informatik Wahlpflicht (SPO 2027) - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. WIF: Informatik Wahlpflicht (SPO 2027) - M.Sc. DKE: Fundamentals of Data Science - M.Sc. DE: Grundlagen Informatik - M.Sc. DE: Informatik Grundlagen (SPO 2027) - M.Sc. VC: Computer Science - M.Sc. VC: Computer Science (SPO 2027)

Angestrebte Lernergebnisse:

- Anwendungen von ausgewählten Erweiterungen der klassischen Logik zur Wissensrepräsentation
- Anwendungen von ausgewählten nicht-klassischen Logiken zur Wissensrepräsentation
- Anwendung von Beweiskalkülen für Logiken
- Verstehen von fortgeschrittenen semantischen Methoden für Logiken
- Verstehen von metalogischen Beweisen und die Fähigkeit einfache metalogische Beweise selbst zu führen

Inhalt:
Die Veranstaltung baut auf der Pflichtveranstaltung Logik auf, in der die klassische Aussagen- und Prädikatenlogik eingeführt wird. In Logik 2 werden einerseits weiterführende Themen der klassischen Logik behandelt (u.a. Korrektheits-, Vollständigkeits- und Kompaktheitsbeweise). Darüber hinaus werden Erweiterungen der klassischen Logik sowie nichtklassische Logiken behandelt, z.B. Modallogik, Temporallogik, epistemische Logik, Konditionallogik, mehrwertige Logiken und Logik höherer Stufe.

Arbeitsaufwand:

- Präsenzzeiten: 14 X 4h = 56h
- selbständige Arbeit = 244h
  (Vorbereiten der Vorlesung,
  Lösen Übungsaufgaben,
  Vorbereitung Präsentation)

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:
• 2 Drittel der Übungsaufgaben votiert • mindestens 2 Präsentationen	Mündliche Prüfung	Übung 2 SWS Vorlesung 2 SWS

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
keine	Erfolgreicher Abschluss des Modul Logik bzw. fundiertes Wissen im Bereich der Themen des Moduls"Logik.

Medienformen:	Literatur:

Hinweise:
Anmeldung via E-Learning-Plattform

Programmierparadigmen

Sommer

(engl. Programming Paradigms)

Modulnummer: FIN-INF-110121

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
PGP	5	Sommer	ab 2.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	false
Verantwortung:	Dr. Christian Braune
Dozent:in:	Dr. Christian Braune
Lehrveranstaltungen:	• Vorlesung Programmierparadigmen • Übung Programmierparadigmen
Verwendbarkeit:	- B.Sc. INF: Informatik II - Pflicht - B.Sc. INF: Aufbaumodule (SPO 2027) - B.Sc. CV: Informatik - Wahlpflicht - B.Sc. CV: Computervisualistik Wahlpflicht (SPO 2027) - B.Sc. INGINF: Informatik - Wahlpflicht - B.Sc. INGINF: Informatik Wahlpflicht (SPO 2027) - B.Sc. WIF: Gestalten und Anwenden - Wahlpflicht - B.Sc. WIF: Informatik Pflicht (SPO 2027) - B.Sc. INF (bilingual): Informatik II - Pflicht - B.Sc. INF (bilingual): Aufbaumodule (SPO 2027) - M.Sc. DE: Grundlagen Informatik - M.Sc. DE: Informatik Grundlagen (SPO 2027)

Angestrebte Lernergebnisse:
Studierende, die diesen Kurs erfolgreich abschließen, ...

- kennen die gängigsten Paradigmen zum Algorithmenentwurf,
- sind in der Lage, diese anzuwenden,
- können beurteilen, in welchen Situationen welches Paradigma am geeignetsten ist

- und verstehen welche Vor- und Nachteile sich aus der Verwendung der jeweiligen Paradigmen ergeben.

Inhalt:

- Konzepte der wesentlichen Paradigmen
  - Prozedurale Programmierung
  - Objektorientierte Programmierung
  - Funktionale Programmierung
  - Logische Programmierung
  - Parallele Programmierung
  - Quantencomputing
- Technische Umsetzung der Paradigmen in Programmiersprachen
- Anwendung der Paradigmen in Programmiersprachen wie z.B. C, Java, Scala, Haskell, Prolog, Python
- Entscheidungskriterien für Paradigmen

Arbeitsaufwand:
56 h Präsenzzeit 94 h selbstständige Arbeit

Prüfungsvorleistungen:	**Studien-/Prüfungsleistungen:**	**Lehrform / SWS:**
Prüfungsvorleistung erforderlich	Prüfung: mündliche Prüfung (wenn es hinreichend viele Teilnehmer gibt: schriftliche Klausur, 120 Minuten)	• Vorlesung (2 SWS) • Übung (2 SWS)

Voraussetzungen nach Prüfungsordnung:	**Empfohlene Voraussetzungen:**
keine	*Einführung in die Informatik, M1* Der parallele Besuch der Veranstaltungen *Algorithmen und Datenstrukturen* und *M5* ist von Vorteil

Medienformen:	**Literatur:**

Hinweise: