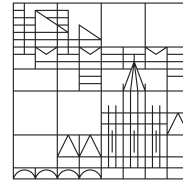


Universität  
Konstanz



## **Modulhandbuch / Module Directory**

### **Bachelor-Studiengang Informatik / Bachelor study programme Computer Science (PO 2015)**

Fachbereich Informatik und Informationswissenschaft  
Mathematisch-Naturwissenschaftliche Sektion  
Universität Konstanz

Department of Computer and Information Science  
Faculty of Sciences  
University of Konstanz

Stand: Oktober 2017  
last changed: October 2017

# Inhaltsverzeichnis

<b>1</b>	<b>Fachbereich Informatik und Informationswissenschaft</b>	<b>1</b>
1.1	Studienangebot des Fachbereichs . . . . .	1
1.2	Abgrenzung Informatik und Information Engineering . . . . .	2
1.3	Entwicklung und Akkreditierung der Studiengänge . . . . .	3
<b>2</b>	<b>Bachelorstudiengang Informatik</b>	<b>4</b>
2.1	Qualifikationsziele des Bachelorstudiengangs Informatik . . . . .	4
2.2	Der Aufbau des Bachelorstudiengangs Informatik . . . . .	5
2.3	Musterstudienplan Bachelor Informatik . . . . .	6
<b>3</b>	<b>Department of Computer and Information Science</b>	<b>7</b>
3.1	Study programmes . . . . .	7
3.2	Delimiting Computer Science and Information Engineering . . . . .	8
3.3	Development and accreditation of the study programmes . . . . .	9
<b>4</b>	<b>Bachelor Study Programme Computer Science</b>	<b>10</b>
4.1	Qualification targets of the Bachelor study programme Computer Science . . . . .	10
4.2	Structure of the Bachelor study programme Computer Science . . . . .	11
4.3	Sample timetable Bachelor study programme Computer Science . . . . .	12
<b>5</b>	<b>Modulverzeichnis / Module directory</b>	<b>13</b>
5.1	Studiumsvorbereitung / Preparatory Course . . . . .	14
5.1.1	Brückenkurs Mathematik / Preparatory Course Mathematics . . . . .	14
5.2	Grundstudium - 1. Semester / Basic studies - 1. term . . . . .	15
5.2.1	Konzepte der Informatik / Principles of Computer Science 1 . . . . .	15
5.2.2	Programmierkurs 1 (Imperative Sprache) / Programming Course 1 (Imperative Programming) . . . . .	17
5.2.3	Rechnersysteme und -netze / Computer Systems . . . . .	18
5.2.4	Analysis I / Analysis I . . . . .	19
5.2.5	Schlüsselqualifikation für Informatiker / Key Competences for Undergraduate Students of Information Engineering and Computer Science . . . . .	20
5.3	Grundstudium - 2. Semester / Basic studies - 2. term . . . . .	21
5.3.1	Konzepte der Programmierung / Programming Concepts . . . . .	21
5.3.2	Programmierkurs 2 (Deklarative Sprache) / Programming Course 2 (Declarative Programming) . . . . .	22
5.3.3	Datenbanksysteme / Database Systems . . . . .	23
5.3.4	Mathematik: Diskrete Strukturen / Mathematics: Discrete Structures . . . . .	24
5.4	Grundstudium - 3. Semester / Basic studies - 3. term . . . . .	25
5.4.1	Betriebssysteme und systemnahe Programmierung / Operating Systems and Systems Programming . . . . .	25
5.4.2	Programmierkurs 3 (Systemnahe Programmierung) / Programming Course 3 (System Programming) . . . . .	26
5.4.3	Software Engineering / Software Engineering . . . . .	27
5.4.4	Algorithmen und Datenstrukturen / Algorithms and Data Structures . . . . .	28

5.4.5	Lineare Algebra I / Linear Algebra I . . . . .	29
5.5	Grundstudium - 4. Semester / Basic studies - 4. term . . . . .	30
5.5.1	Software-Projekt / Software Project . . . . .	30
5.5.2	Theoretische Grundlagen der Informatik / Theoretical Computer Science . . . . .	31
5.5.3	Computergrafik und interaktive Systeme / Computer Graphics and Interactive Systems . . . . .	32
5.5.4	Statistik I / Statistics I . . . . .	34
5.5.5	Numerik I / Numeric I . . . . .	35
5.6	Vertiefungsstudium / Advanced studies . . . . .	36
5.6.1	Analyse und Visualisierung von Information / Analysis & Visualization . . . . .	36
5.6.2	Allgemeine Seminarbeschreibung / General description of seminars . . . . .	37
5.6.3	Allgemeine Bachelor-Projektbeschreibung / General description of the Bachelor's project . . . . .	39
5.6.4	Information Visualization II . . . . .	41
5.6.5	Applied Visual Analytics . . . . .	43
5.6.6	Geographic Information Systems (GIS) . . . . .	45
5.6.7	Multimedia Database Systems . . . . .	46
5.6.8	Database System Architecture and Implementation . . . . .	47
5.6.9	Data Stream Management Systems . . . . .	49
5.6.10	Data Warehousing und OLAP . . . . .	51
5.6.11	XML Technologies . . . . .	52
5.6.12	Transactional Information Systems . . . . .	54
5.6.13	Usability Engineering: Design . . . . .	55
5.6.14	Usability Engineering: Evaluation . . . . .	57
5.6.15	New Directions in Human-Computer Interaction: Blended Interaction . . . . .	59
5.6.16	Visual Information Seeking Systems . . . . .	61
5.6.17	Virtual and Augmented Reality . . . . .	63
5.6.18	Illustrative Computer Graphics . . . . .	64
5.6.19	Modelling in Computer Graphics . . . . .	65
5.6.20	Image Analysis and Computer Vision . . . . .	66
5.6.21	Introduction to Mathematical Image Analysis . . . . .	67
5.6.22	Digital Signal Processing . . . . .	68
5.6.23	Numerical Continuation Methods . . . . .	69
5.6.24	Eye Tracking: Theory and Practice . . . . .	70
5.6.25	Design and Analysis of Algorithms . . . . .	71
5.6.26	Network Analysis . . . . .	72
5.6.27	Graph Drawing . . . . .	73
5.6.28	Big Data and Scripting . . . . .	74
5.6.29	Introduction to Data Analysis . . . . .	75
5.6.30	Data Mining 2: Frequent Pattern Mining . . . . .	76
5.6.31	Data Mining 2: Artificial Neural Networks and Deep Learning . . . . .	77
5.6.32	Web Information Retrieval . . . . .	78
5.6.33	Computational Methods for Document Analysis . . . . .	80
5.6.34	Algorithms in Bioinformatics . . . . .	81
5.6.35	Computational Systems Biology: Modelling and Simulation . . . . .	82
5.6.36	Computational Systems Biology: Representation and Visualisations . . . . .	83
5.6.37	Petri Nets . . . . .	84
5.6.38	Functional Safety for Embedded Systems . . . . .	85
5.6.39	Directed Studies: Testing of Software . . . . .	86
5.6.40	Model Checking of Software and Systems . . . . .	87
5.6.41	Advanced Model Checking . . . . .	88
5.6.42	High-Performance Systems and Networks . . . . .	89
5.6.43	Open Source Software Development . . . . .	90

# 1 Fachbereich Informatik und Informationswissenschaft

Mit seinen zwölf Professuren, drei Juniorprofessuren und circa 80 Wissenschaftlerinnen und Wissenschaftlern deckt der Fachbereich Informatik und Informationswissenschaft der Universität Konstanz das für einen Informatikstudiengang benötigte Fachspektrum ab und ist im Hinblick auf seinen Forschungsschwerpunkt **Exploration und Visualisierung großer Datenmengen** einzigartig und exzellent besetzt.

- Prof. Dr. Michael Berthold (Bioinformatik und Information Mining), zur Zeit vertreten durch PD Dr. Christian Borgelt
- Prof. Dr. Ulrik Brandes (Algorithmik)
- Prof. Dr. Oliver Deussen (Computergrafik und Medieninformatik)
- Prof. Dr. Bastian Goldlücke (Computer Vision und Image Analysis)
- Prof. Dr. Daniel Keim (Datenanalyse und Visualisierung)
- Apl. Prof. Dr. Sven Kosub (Theoretische Informatik)
- Prof. Dr. Stefan Leue (Software and Systems Engineering)
- Prof. Dr. Harald Reiterer (Mensch-Computer Interaktion)
- Prof. Dr. Dietmar Saupe (Multimedia Signalverarbeitung)
- Prof. Dr. Marc Scholl (Datenbanken und Informationssysteme)
- Prof. Dr. Falk Schreiber (Computational Life Sciences)
- Prof. Dr. Marcel Waldvogel (Verteilte Systeme)
- Juniorprofessor Dr. Bela Gipp (Informationswissenschaft)
- Juniorprofessor Dr. Michael Grossniklaus (Datenbanken und Informationssysteme)
- Juniorprofessorin Dr. Tatjana Petrov (Modellierung komplexer selbstorganisierender Systeme)

Der Fachbereich gilt international als Zentrum der Spitzenforschung im Bereich der Datenanalyse und -visualisierung. Dazu trägt auch der im gleichen Themenumfeld angesiedelte transregionale Sonderforschungsbereich „Quantitative Methods for Visual Computing“ bei, welcher von der Deutschen Forschungsgemeinschaft (DFG) gefördert wird.

## 1.1 Studienangebot des Fachbereichs

Der Fachbereich Informatik und Informationswissenschaft bietet folgende Studiengänge im Hauptfach an:

- Bachelor of Science Informatik
- Bachelor of Science Information Engineering
- Master of Science Computer and Information Science
- Bachelor of Education Informatik
- Master of Education Informatik (ab WS 2017/2018)
- Staatsexamenstudiengang Informatik (kein Studienstart mehr möglich)

Zusätzlich kann Informatik als Nebenfach oder im Wahlpflichtbereich vieler Studiengänge belegt werden.

Das Bachelorstudium Information Engineering ist auf drei Jahre angelegt, das Bachelorstudium Informatik auf vier Jahre. Beide Studiengänge vermitteln grundlegende Kenntnisse und Fähigkeiten, die zu typischen Berufen in der Informationsgesellschaft befähigen.

Direkt im Anschluss, nach einigen Jahren Berufserfahrung oder auch nach einem anderen Studium können vertiefte Kenntnisse im Rahmen des Masterstudiums erworben werden. Die Dauer des Masterstudiums hängt dabei vom Bachelorabschluss ab. Bei dreijährigen Bachelorabschlüssen dauert das Masterstudium zwei Jahre, bei vierjährigen äquivalenten Bachelorstudiengängen dauert das Masterstudium ein Jahr.

Das Angebot beider Bachelorstudiengänge ist grundsätzlich deutschsprachig, für Vertiefungsveranstaltungen und die einschlägige Literatur sind in aller Regel aber auch Englischkenntnisse erforderlich. Der Masterstudiengang Computer and Information Science ist englischsprachig.

Die Erläuterungen in diesem Modulhandbuch sind eine Beschreibung des Studiengangs. Die **verbindlichen** Details sind in der gültigen **Prüfungsordnung** festgelegt.

Jedes Studienjahr ist in zwei Hälften geteilt, das Winter- und das Sommersemester. Abgesehen von den Abschlussarbeiten sind alle Prüfungen studienbegleitend, was bedeutet, dass die Inhalte der Veranstaltungen jeweils bis zum Ende des Semesters, in dem das Modul angeboten wird, geprüft werden. Prüfungen finden in Form von Hausarbeiten, mündlichen Prüfungen oder Klausuren statt.

Der zu erwartende Aufwand für eine Veranstaltung wird gemäß ECTS (European Credit Transfer System) in Credits gemessen, wobei ein Credit für 30 Stunden Arbeitsaufwand steht. In den Studiengängen sind pro Semester 30 ECTS-Credits zu erwerben.

## 1.2 Abgrenzung Informatik und Information Engineering

In den folgenden Abschnitten werden die Begriffe Informatik und Information Engineering näher erläutert und beide Begriffe voneinander abgegrenzt, um die Unterschiede der Studienfächer aufzuzeigen.

Die klassische **Informatik** hat das Ziel, ein grundlagen- und anwendungsorientiertes Studium informationsverarbeitender Prozesse und Maschinen in ganzer Breite zu sein. Die Grundlagen der Informatik liegen in den Bereichen Mathematik und Ingenieurwissenschaft. Unser Studiengang vermittelt hierbei im Gegensatz zu Fachhochschulen und anderen Ausbildungsstätten insbesondere auch theoretische Konzepte und Methoden, die nicht aktuellen Trends unterliegen. Für die Informatikerinnen und Informatiker im Berufsleben bedeutet dies eine weitgehende Unabhängigkeit von Branchentrends. Der Anteil an Mathematik - von Algebra über Analysis bis hin zu Logik - und Theorie ist daher in diesem Fach höher. Da dieser Studiengang acht Semester dauert, gibt das Curriculum ausreichend Raum für eine Praxisphase (ein Semester Auslandsaufenthalt oder ein Praxissemester in der Wirtschaft, gerne auch im Ausland).

**Information Engineering** umfasst für uns alle Aspekte der Gewinnung, Aufbereitung, Bereitstellung, Extraktion, Analyse und Vermittlung von Information. Wir verstehen es als eine angewandte Informatik, die sich mit allen Aufgaben, die entlang der Prozesskette „Daten - Information - Wissen“ anfallen, beschäftigt. In ihrem Zentrum stehen daher Konzeption und Konstruktion informationsverarbeitender Systeme sowie entsprechende Methoden und Anwendungen. Beispiele für relevante Themen sind effiziente Codierungsverfahren zur Übermittlung von Multimedia-Daten, Anfrageauswertung in XML-Datenbanken, Data Mining in der Medikamentenherstellung oder die Visualisierung von Wetterdaten. Weiter gehören dazu auch die Nutzerfreundlichkeit von Computerprogrammen und Web-Designs sowie ethische Aspekte der Verfügbarkeit von Information. Das Fach vereint daher grundlegende Themen der praktischen und angewandten Informatik mit nutzungsorientierten Herangehensweisen der Informationswissenschaft.

Die Ausbildung ruht auf den drei Säulen Informatik, Umgang mit Information sowie Mathematik mit einem Schwerpunkt im Bereich der Datenexploration und Visualisierung. Die Lehre ist nah an der Forschung des Fachbereichs - getreu dem Leitsatz der Universität Konstanz: „**Lehre aus**

**Forschung“**. Der Fachbereich ist im Hinblick auf seinen Forschungsschwerpunkt **„Visualisierung und Exploration großer Datenräume“** in Deutschland einzigartig breit ausgelegt und von hohem internationalen Ansehen.

Das Programm wird ergänzt durch einen von den Studierenden zu wählenden fachfremden Anteil, der darauf vorbereiten soll, dass die Arbeit unsere Absolventinnen und Absolventen in aller Regel den Umgang mit Informationsbedürfnissen aus anderen Anwendungsfeldern beinhaltet. Der durch Wirtschaft, öffentlichen Dienst und Gesellschaft entstehende Bedarf an der Beherrschung immer größer werdender Informationsmengen eröffnet hierbei ein weites Spektrum zur Umsetzung der erworbenen Kompetenzen. Typische Berufsbilder sind neben vielen informatiknahen Tätigkeiten vor allem im Umfeld der Informationsdienstleistungen zu finden.

### **1.3 Entwicklung und Akkreditierung der Studiengänge**

Information Engineering wird seit dem Wintersemester 1999/2000 als erster konsekutiver Bachelor- und Masterstudiengang an der Universität Konstanz angeboten. Im Zuge der Akkreditierung durch die Fachagentur ASIIN im Jahre 2005 (die erste an der Universität Konstanz) wurde das Studienprogramm im Wintersemester 2006/2007, im Wintersemester 2010/11 und im Wintersemester 2014/15 überarbeitet und weiter verbessert.

Seit Wintersemester 2010/11 wird zusätzlich zum Bachelorstudiengang Information Engineering der vierjährige Bachelorstudiengang Informatik angeboten. Der Masterstudiengang wurde zum Sommersemester 2015 von Information Engineering auf Computer and Information Science umbenannt. Die aktuelle Akkreditierung der Studiengänge erfolgte am 18.02.2015 im Rahmen der Systemakkreditierung und ist gültig bis zum 28.02.2019.

## 2 Bachelorstudiengang Informatik

### 2.1 Qualifikationsziele des Bachelorstudiengangs Informatik

Der Bachelorstudiengang Informatik vermittelt ein breites Spektrum an Fachwissen und die notwendigen Grundlagen für den Einstieg in die berufliche Praxis. Ziel des Informatik-Bachelorstudiengangs ist die Vermittlung der grundlegenden Methoden der angewandten Informatik mit einer soliden theoretischen und mathematischen Kompetenz.

Die Informatik nimmt in einer beständig zunehmenden Zahl von Branchen eine Schlüsselrolle ein. Die hohe Relevanz des Gebietes kann daran abgelesen werden, dass führende Unternehmen aller Branchen den Informatikbereich als einen strategischen Schwerpunkt identifiziert haben. Das Portfolio des Bachelorstudiengangs Informatik greift den Expertisebedarf potentieller Arbeitgeber direkt auf.

Wesentliche Zielgruppe sind Schulabgängerinnen und Schulabgänger, die einen grundständigen Informatik-Studiengang aufnehmen wollen und von Vertiefungsmöglichkeiten im Forschungsschwerpunkt des Fachbereichs Informatik und Informationswissenschaft, einer curricular verankerten externen Praxisphase im Ausland oder in der Wirtschaft profitieren wollen. Ferner soll ein Direkteinstieg in ein zukunftssträchtiges Arbeitsmarktsegment oder ein weiterführendes Studium bzw. eine Promotion möglich sein.

Diese Studienanfängerinnen und -anfänger unterscheiden sich von denen, die das Bachelorstudium Information Engineering aufnehmen, durch größeres Interesse an mathematischen und systemnahen Grundlagen.

Der Studiengang ist

1. berufsqualifizierend (statt -befähigend),
2. vierjährig mit einsemestriger externer Praxisphase,
3. mit Fast Track-Option zur anschließenden Aufnahme einer Promotion,
4. mit Mentorenprogramm für eine leichtere Studierbarkeit und
5. mit hoher Durchlässigkeit von und zu dem Bachelorstudiengang Information Engineering und dem Teilstudiengang Bachelor of Education Informatik konzipiert.

**Kompetenzen.** Die Absolventinnen und Absolventen sind in der Lage, Methoden der Informatik und Informationswissenschaft anzuwenden und in Projekten umzusetzen. Neben den fachspezifischen Kompetenzen (formale, algorithmische und mathematische Kompetenzen, Analyse-, Design-, Realisierungs- und Projektmanagement-Kompetenzen sowie technologische Kompetenzen) werden auch fachübergreifende Kompetenzen und überfachliche Schlüsselqualifikationen im Bachelorstudium vermittelt. Der Forschungsschwerpunkt des Fachbereichs spiegelt sich auch in der Konzeption des Studiengangs wider. So werden unabhängig von der individuellen Schwerpunktsetzung Kompetenzen im Bereich der Methoden und Systeme zur Visualisierung, Analyse, Exploration und Verarbeitung großer Informationsmengen vermittelt. Die verpflichtende externe Praxisphase (Praktikum oder Auslandsstudienaufenthalt) fördert zusätzlich die Berufsqualifizierung sowie Sozial-, Selbst- und Sprachkompetenzen.

Das wissenschaftliche Ausbildungsprofil der Absolventinnen und Absolventen ist daher nicht nur für die akademische Forschung, sondern insbesondere auch für die Nachfrage der Wirtschaft konzipiert. Dieser Bedarf besteht unter anderem in der Forschung und Entwicklung der produzierenden Industrie und des Informationsmarktes.

## 2.2 Der Aufbau des Bachelorstudiengangs Informatik

Das Bachelorstudium Informatik ist auf eine Regelstudienzeit von acht Semestern ausgelegt und in ein Basisstudium und ein projektorientiertes Vertiefungsstudium gegliedert. Es setzt auf eine breite Basis in den Bereichen Mathematik und Informatikgrundlagen. Zusätzlich ermöglichen die acht Semester die Einführung eines Praxis- oder Auslandssemesters, für das in der Prüfungsordnung 30 ECTS vorgesehen sind. Im Vertiefungsstudium werden zusätzlich 30 ECTS in den Forschungsgebieten des Fachbereichs Informatik und Informationswissenschaft absolviert.

Den Zugang zum Studium regelt die Zulassungs- und Immatrikulationsordnung der Universität Konstanz. Da über die darin festgelegten Bedingungen hinaus keine Beschränkungen bestehen, reicht die Allgemeine Hochschulreife. Internationale Studierende müssen zusätzlich ausreichende Sprachkenntnisse nachweisen.

Das Basisstudium (Umfang 120 ECTS) umfasst vier Semester und beinhaltet insbesondere Module aus den folgenden Gebieten:

- Modul Informatik 1 (Konzepte der Informatik und Programmierkurs 1)\*,
- Modul Informatik 2 (Konzepte der Programmierung und Programmierkurs 2)\*,
- Modul Informatik 3 (Betriebssysteme und Systemnahe Programmierung und Programmierkurs 3),
- Rechnersysteme und -netze\*,
- Datenbanksysteme\*,
- Software Engineering,
- Softwareprojekt,
- Algorithmen und Datenstrukturen,
- Theoretische Grundlagen der Informatik,
- Computergrafik und interaktive Systeme,
- Analysis,
- Lineare Algebra,
- Diskrete Strukturen,
- Statistik oder Numerik,
- Schlüsselqualifikationen.

Von den mit \* markierten Modulen müssen mindestens zwei von vier für das Bestehen der **Orientierungsprüfung** bis zum Ende des zweiten Semesters erfolgreich absolviert werden.

Das Vertiefungsstudium umfasst ebenfalls vier Semester (Umfang 120 ECTS). Hier wird das Pflichtmodul Analyse und Visualisierung von Information belegt sowie Lehrveranstaltungen aus dem Vertiefungsangebot des Fachbereichs sowie ergänzend auch fachfremde Veranstaltungen, Schlüsselqualifikationen und/oder Sprachkurse. Während des Studiums ist im 6. oder 7. Semester eine externe Praxisphase abzuleisten, wahlweise in Form eines sechsmonatigen Praktikums oder eines Auslandssemesters. Das Bachelor-Projekt und -Seminar bereiten auf die dreimonatige Bachelor-Arbeit (15 ECTS) im achten Semester vor. Die Abschlussnote setzt sich (zu unterschiedlichen Anteilen) aus den studienbegleitend erworbenen Noten und denen der Bachelor-Arbeit und des Kolloquiums über die Bachelor-Arbeit zusammen.

Erfolgreichen Absolventinnen und Absolventen wird der akademische Grad eines **Bachelor of Science (B.Sc.) im Fach Informatik (Computer Science)** verliehen. Die Absolventinnen und Absolventen sind für die breite Vielfalt an Aufgaben in der globalen Informationsgesellschaft befähigt. Der vierjährige Abschluss ist berufsqualifizierend und bringt durch das Praxissemester bereits erste Berufserfahrungen mit sich. Als konsekutiver Masterstudiengang wird die einjährige Variante des Masterstudiengangs Computer and Information Science angeboten. Ebenso ist der Direkteinstieg zur Promotion möglich.



## 2.3 Musterstudienplan Bachelor Informatik

Semester (WiSe- Start)	Semester (SoSe- Start*)	Module	Umfang in SWS	ECTS- Credits	Leistung
<b>Basisstudium</b>					
1	1	Informatik 1, bestehend aus: – Konzepte der Informatik (V+Ü) – Programmierkurs 1 (imperative Sprache)	9 5 4	12 6 6	OP P S
1	2	Rechnersysteme und -netze (V+Ü)	5	6	OP
1	2	Analysis 1 (V+Ü)	6	9	P
1	1	Schlüsselqualifikationen	2	3	S
2	3	Informatik 2, bestehend aus: – Konzepte der Programmierung (V+Ü) – Programmierkurs 2 (deklarative Sprache)	9 5 4	9 4 5	OP P S
2	1	Datenbanksysteme (V+Ü)	6	9	OP
2	1	Diskrete Strukturen (V+Ü)	6	9	P
2	2	Schlüsselqualifikationen	2	3	S
3	4	Informatik 3, bestehend aus: – Betriebssysteme, Systemnahe Programmierung (V+Ü) – Programmierkurs 3 (systemnahe Sprache)	8 4 4	7 4 3	P P S
3	4	Software Engineering (V+Ü)	5	5	P
3	4	Algorithmen und Datenstrukturen (V+Ü)	6	9	P
3	2	Lineare Algebra (V+Ü)	6	9	P
4	5	Software-Projekt	4	6	P
4	3	Theoretische Grundlagen der Informatik (V+Ü)	6	9	P
4	3	Computergrafik und interaktive Systeme (V+Ü)	6	9	P
4	4	Statistik** (V+Ü)	4	6	P
Summe Basisstudium			96	120	
<b>Vertiefungsstudium</b>					
5	6	Analyse und Visualisierung von Information (V+Ü)	6	9	P
5–8	5–7	vertiefende Lehrveranstaltungen (Informatik), darunter mind. 1 Seminar (à 4 cr.) und 1 Projekt (à 9 cr.)		30	P
5–8	3, 5–8	Ergänzungsbereich: wahlweise fachfremde Lehrveranstaltungen, SQ-/SLI-Veranstaltungen (max. 6 cr.) und vertiefende Lehrveranstaltungen (max. 18 cr.)		36	S/P
6/7	7	Praktikum oder Auslandsaufenthalt		30	N
8	8	Bachelorarbeit (12 cr.) und Kolloquium (3 cr.)		15	P
Summe Vertiefungsstudium				120	
Gesamtsumme				240	

\* = die Semesteraufteilung für Studierende, die zum Sommersemester beginnen, erfolgt individuell aufgrund von Vorkenntnissen und/oder Neigungen in Rücksprache mit der Fachstudienberatung. Insbesondere die Integration eines Auslandsaufenthaltes ist frühzeitig abzusprechen.

\*\*= Die Statistikvorlesung kann im Sommer- oder im Wintersemester absolviert werden. Alternativ kann auch die Veranstaltung **Numerik** besucht werden.

OP = Module sind Teil der Orientierungsprüfung (vgl. § 15 der Prüfungsordnung)

P = studienbegleitende Prüfungsleistung (vgl. § 13 der Prüfungsordnung)

S = Studienleistung (vgl. §14 der Prüfungsordnung)

N = Nachweis

Die Angabe von Semesterwochenstunden (SWS) ist unverbindlich. Sie dient als Hinweis auf den zu erwartenden Umfang des Präsenzstudiums.

## 3 Department of Computer and Information Science

With its twelve professors, three junior professors, and 80 lecturers and researchers, the department of Computer and Information Science amply covers the spectrum of knowledge required for a degree in computer science, and is uniquely and excellently staffed with respect to its research focus of the **exploration and visualisation of large volumes of data**.

- Prof. Dr. Michael Berthold (Bioinformatics & Information Mining), currently replaced by PD Dr. Christian Borgelt
- Prof. Dr. Ulrik Brandes (Algorithmics)
- Prof. Dr. Oliver Deussen (Computer Graphics & Media Informatics)
- Prof. Dr. Goldlücke (Computer Vision & Image Analysis)
- Prof. Dr. Daniel Keim (Analysis & Visualisation)
- Apl. Prof. Dr. Sven Kosub (Theory of Computing)
- Prof. Dr. Stefan Leue (Software and Systems Engineering)
- Prof. Dr. Harald Reiterer (Human-Computer Interaction)
- Prof. Dr. Dietmar Saupe (Multimedia Signal Processing)
- Prof. Dr. Marc Scholl (Databases and Information Systems)
- Prof. Dr. Falk Schreiber (Computational Life Sciences)
- Prof. Dr. Marcel Waldvogel (Distributed Systems)
- Junior professor Dr. Bela Gipp (Information Science)
- Junior professor Dr. Michael Grossniklaus (Databases & Information Systems)
- Junior professor Dr. Tatjana Petrov (Modelling of Complex Self-organizing Systems)

The department has an international reputation as a centre for cutting-edge research in the field of data analysis and visualisation. The German Research Foundation (DFG)-based transregional collaborative research centre **Quantitative Methods for Visual Computing** also contributes to the excellent repute of the department.

### 3.1 Study programmes

The department offers the following study programmes as majors:

- Bachelor of Science Computer Science
- Bachelor of Science Information Engineering
- Master of Science Computer and Information Science
- Bachelor of Education Computer Science
- Master of Education Computer Science (starting winter term 2017/2018)
- Teacher Education Degree Programme Computer Science (no study start possible any more)

In addition you can choose computer science as a minor or as an elective module.

The Information Engineering Bachelor study programme comprises three years, while the Bachelor degree in Computer Science comprises four years. Both impart the basic knowledge and skills

to enable graduates to take up typical professions in today's information society.

Straight after the Bachelor study programmes, or after a number of years of professional experience, or even after a completely different study programme, in-depth knowledge can be acquired within the framework of the Master study programme. How long the Master study programme takes depends on the Bachelor degree. In combination with a three-year Bachelor study programme, the Master degree takes two years; following a four-year, equivalent Bachelor study programme, the Master degree only takes one year.

Both Bachelor study programmes are generally held in German. However, knowledge of English will be required for advanced undergraduate studies and course-related literature. The Master study programme Computer and Information Science is held in English.

The information in this module directory describes the degree course. **Binding information** is documented in the **examination regulations** for the study programme.

Each year of study is divided into two halves: the winter and summer terms. Apart from the final examination, all other examinations are related to coursework, meaning that the content of the course is tested at the end of the term in which the course module has been taken. Examinations are in the form of either term papers, oral examinations or written examinations.

The work involved in each course is measured in credits in accordance with ECTS (European Credit Transfer System). According to this system, one credit represents 30 hours of work. The degree courses require that you collect 30 ECTS credits per term.

## 3.2 Delimiting Computer Science and Information Engineering

The following sections explain the concepts of Computer Science and Information Engineering, and delineate both concepts to show the differences between the study programmes.

The objective behind classic **Computer Science** is to provide a study programme that is oriented towards the basic principles and scopes of application, involving the entire spectrum of information processing systems and machines. In contrast to other universities and colleges, our study programme also provides knowledge about theoretical concepts and methods which are separate from current trends. In their professional lives, computer scientists will find that this will give them considerable independence from industry trends. The proportion of maths and theory is therefore greater in this study programme. As this is an eight-term study programme, the curriculum provides sufficient time for a practical phase (one term study abroad or a term spent working in industry, also possible abroad).

**Information Engineering** at the University of Konstanz encompasses all aspects of the mining, preparation, selection, extraction, analysis and communication of information. We understand it as applied computer science which addresses all the tasks associated with the data-information-knowledge process chain.

Information Engineering focuses on the conceptual design and construction of information processing systems and the respective methods and applications. Examples of relevant topics are efficient coding methods for transferring multimedia data, query evaluation in XML databases, data mining capabilities in drug production or the visualisation of weather data. Further relevant topics are the user-friendliness of computer programmes and Web design as well as ethical issues relevant to the availability of information. The study programme therefore combines fundamental issues of practical and applied informatics with the user-oriented approaches of information science.

The study programme is based on the three pillars Computer Science, Information Handling and Mathematics. It focuses particularly on the area of data exploration and visualisation. Our teaching is closely in touch with the research conducted in the department and true to the principle of the University of Konstanz: **Teaching from research**. The department's research focus, **Visualisation and Exploration of Large Information Spaces** gives its research a unique breadth in Germany and enables the department to enjoy international standing.

The study programme is complemented by additional studies in a field not directly related to the student's studies (non-specialist). The intention here is to prepare students for the fact that their postgraduate work will include handling information requirements from other areas of application. The need to manage increasing volumes of information emerging from business, public services and society is opening a wide range of opportunities for graduates to put their acquired skills into practice. In addition to the many jobs directly related to computer science, typical professions include work in the area of information services.

### **3.3 Development and accreditation of the study programmes**

Information Engineering has been offered at the University of Konstanz as the first consecutive Bachelor and Master study programme since the winter term 1999/2000. In the course of the accreditation process, conducted by ASIIN, a professional agency, in 2005 (this was another first at the University of Konstanz), the study programme was reviewed and improved in the winter term 2006/2007, in the winter term 2010/2011 and again in the winter term 2014/2015.

Since the winter term 2010/11, a four-year Bachelor study programme in Computer Science has been offered additionally. The Master study programme Information Engineering was renamed into Master Computer and Information Science in the summer semester 2015. The current accreditation was carried out on 18.02.2015 as part of the system accreditation and is valid until 28.02.2019.

## 4 Bachelor Study Programme Computer Science

### 4.1 Qualification targets of the Bachelor study programme Computer Science

The Bachelor study programme Computer Science conveys a wide range of expertise and the necessary foundations for starting a professional career. The aim of the Computer Science study programme is to teach the principled methods of applied computer science and impart a solid foundation in theoretical and mathematical skills.

Computer science plays a key role in a steadily increasing number of industries. The high level of relevance of this area can be seen by the fact that leading companies in all industries have identified computer science as a strategic focus of their activities. The Computer Science study programme portfolio concentrates on the expertise that is required by potential employers.

The biggest target group for this course is school-leavers who are interested in a Bachelor study programme in Computer Science and would like to benefit from the research focus of the Department of Computer and Information Science and the external practical course-component, spent either abroad or in industry. The degree enables direct employment in a promising segment of the employment market or admission to a higher degree or PhD course.

The students of the study programme Computer Science distinguish themselves from students of the Information Engineering Bachelor study programme by the fact that they will have a greater interest in mathematical and system-related principles.

The study programme

1. qualifies students for jobs (rather than merely enabling them),
2. comprises four years including a one-term external practical component,
3. offers a fast-track option with subsequent admission to a PhD course of study,
4. includes a mentor system, which intends to help make studying easier for students, and
5. has a high degree of permeability to and from the existing Bachelor study programme in Information Engineering and the Bachelor of Education in Computer Science.

**Skills.** The graduates learn to apply methods and approaches of computer and information science and to manage projects. The study programme conveys subject-specific competences (such as formal, algorithmic, and mathematical competences, competences in analysis, design, realisation, and project management, as well as technological competences), and also interdisciplinary competences and key qualifications. The research focus of the department is also reflected in the design of the study programme. Notwithstanding individual specialisation, graduates will have competences in the area of methods and systems of visualisation, analysis, exploration, and processing of large amounts of information. The obligatory internship or alternatively a study abroad semester equips graduates with additional professional skills as well as e.g. social and language competences.

The academic training profile of graduates is therefore not only an academic research profile but one that responds to the demands placed by industry. This kind of profile is also required in the research and development labs of the manufacturing industry and information markets.

## 4.2 Structure of the Bachelor study programme Computer Science

The Computer Science Bachelor study programme comprises four years of studies and is divided into basic undergraduate studies followed by advanced undergraduate studies. It focuses on a wide basis in the areas of mathematics and informatics principles. In addition to this, the eight terms enable students to include a practical component or term spent abroad, which earns the student 30 ECTS in accordance with the examination regulations. In the advanced undergraduate studies, an additional 30 ECTS are gained by completing work in the research areas of the Department of Computer and Information Science.

Admission to the study programme is regulated by the Admissions and Matriculations policies of the University of Konstanz. As there are no further restrictions over and above the specified conditions, the general higher education entrance qualification is all that is required. International students need to provide documentation of sufficient German language skills.

The four-term basic undergraduate studies encompass modules with a student work-load of 120 ECTS, including:

- module computer science 1 (concepts of computer science, programming course 1)\*,
- module computer science 2 (concepts of programming, programming course 2)\*,
- module computer science 3 (operating systems and system-relevant programming, programming course 3),
- computer systems and networks\*,
- database systems\*,
- algorithms and data structures,
- software engineering,
- software project,
- theoretical computer science,
- computer graphics and interactive systems,
- analysis,
- linear algebra,
- discrete structures,
- statistics or numerics,
- and key qualifications.

Two of the courses marked with an \* have to be taken and passed by the end of the second term in order to pass the **orientation test**.

The advanced undergraduate studies also encompass four terms and include modules with a workload of 120 ECTS. In addition to the obligatory module "Analysis and Visualisation of Information" advanced courses are chosen from the different areas of specialization in the department. Additionally, courses from other departments, key qualifications, and/or language classes can be chosen. The obligatory six-months internship or the study semester abroad takes place during the sixth or seventh semester. The Bachelor's project and seminar prepare for the three-months Bachelor's thesis (15 ECTS) in the eighth semester. The final assessment is based on the examination results achieved during the studies and the Bachelor's thesis plus an oral colloquium on the thesis.

Successful students are awarded the academic title of **Bachelor of Science (B.Sc.) in the field of Computer Science**. Graduates are prepared for a wide range of tasks in today's global information society. The four-year degree qualifies for jobs and gives students initial work experience thanks to the practical component of the course. For admission to a Master study programme, this four-year degree entitles graduates to be admitted to the shorter Master study programme, which comprises one year. It is also possible to be admitted directly to a PhD course.

### 4.3 Sample timetable Bachelor study programme Computer Science

Semester (Start Winter term)	Semester (Start Summer term*)	Courses	Scope in SWS	ECTS Credits	Examination
<b>Basic Undergraduate Studies</b>					
1	1	Computer Science 1, consisting of:	9	12	OP
		– Concepts of Computer Science (lecture+tutorial)	5	6	P
		– Programming Course 1 (Imperative Programming)	4	6	S
1	2	Computer Systems and Networks (I+t)	5	6	OP
1	2	Analysis 1 (I+t)	6	9	P
1	1	Key Skills 1	2	3	S
2	3	Computer Science 2, consisting of:	9	9	OP
		– Concepts of Programming (I+t)	5	4	P
		– Programming Course 2 (Declarative Programming)	4	5	S
2	1	Database Systems (I+t)	6	9	OP
2	1	Discrete Structures (I+t)	6	9	P
2	2	Key Skills 2	2	3	S
3	4	Computer Science 3, consisting of:	8	7	P
		– Operating Systems and Systems Programming (L+T)	4	4	P
		– Programming Course 3 (System Programming)	4	3	S
3	4	Software Engineering (I+t)	5	5	P
3	4	Algorithms and Data Structures (I+t)	6	9	P
3	2	Linear Algebra 1 (I+t)	6	9	P
4	5	Software-Project	4	6	P
4	3	Theoretical Computer Science (I+t)	6	9	P
4	3	Computer Graphics and Interactive Systems (I+t)	6	9	P
4	4	Statistics** (I+t)	4	6	P
Total Basic Undergraduate Studies			96	120	
<b>Advanced Undergraduate Studies</b>					
5	6	Data Analysis and Visualisation (L+T)	6	9	P
5–8	5–7	Advanced Courses (Computer Science), of these, at least 1 seminar (à 4 cr.) and 1 project (à 9 cr.)		30	P
5–8	3, 5–8	Supplementary Area of Studies: optional courses from other subjects key skills/language courses (max. 6 cr.) advanced courses (max. 18 cr.)		36	S/P
6/7	7	Internship or Study Abroad		30	N
8	8	Bachelor's thesis (12 cr.) and Colloquium (3 cr.)		15	P
Total Advanced Undergraduate Studies				120	
Overall Total				240	

\* = Term organisation for students who start their course in the summer term is handled individually according to prior knowledge and/or interests. Please consult with the study advisory service of the department. Especially the integration of a term spent studying abroad should be discussed early in the course of studies.

\*\*= The statistics lecture can be successfully completed in the Winter or Summer term. Alternatively you can visit the **Numeric** lecture.

OP = these courses are part of the orientation test (see § 15 of the exam regulations)

P = course-related examination (credit points) (see § 13 of the exam regulations)

S = study credits (see §14 of the exam regulations)

N = evidence

The specification of semester week hours (SWS) is not binding. It merely serves as a means of indicating the extent of time spent on the course that is expected.

## 5 Modulverzeichnis / Module directory

In diesem Kapitel sind detaillierte Beschreibungen der Lehrveranstaltungen aufgeführt. Die Veranstaltungen des Pflichtbereichs werden grundsätzlich mindestens einmal im Studienjahr angeboten. Bei den Vertiefungsveranstaltungen ist das Angebot veränderlich, die hier aufgeführten zählen jedoch zu den immer wieder angebotenen und können in dieser oder ähnlicher Form erwartet werden. Sie sind eng an den jeweiligen Forschungsschwerpunkt der Dozentinnen und Dozenten gekoppelt.

Die Lehrveranstaltungen des Grundstudiums sind in deutscher Sprache beschrieben, da sie normalerweise auch auf Deutsch stattfinden. Die Lehrveranstaltungen des Vertiefungsstudiums sind in englischer Sprache beschrieben, da sie normalerweise auf Englisch stattfinden.

Die Studienkommission des Fachbereichs ist das verantwortliche Gremium für die Qualitätssicherung des Lehrangebots. Sie stellt sicher, dass in jedem Semester eine ausreichende Auswahl an Veranstaltungen aus allen Forschungsgebieten zur Verfügung steht. Rechtzeitig vor Beginn eines Semesters wird das jeweils aktuelle Veranstaltungsangebot im elektronischen Vorlesungsverzeichnis veröffentlicht.

In this chapter, detailed descriptions of the offered courses are listed. The obligatory courses of the basic studies take place at least once a year. For the advanced courses, no fixed programme exists, but the ones listed are the ones that are offered quite regularly and can be expected to take place as announced or in similar form. They are closely linked to the respective research focus of the lecturers.

The courses of the basic studies are described in German, as they are normally offered in German. The courses of the advanced studies are described in English, as they are normally offered in English.

The study commission of the department is the responsible body for the quality assurance of the courses offered. It ensures that a sufficient selection of courses from all research areas is available every semester. Before the start of the semester, the current course offer will be published in the electronic course catalogue.



## 5.1 Studiumsvorbereitung / Preparatory Course

### 5.1.1 Brückenkurs Mathematik / Preparatory Course Mathematics

**Modulverantwortlicher:**

Apl. Prof. Dr. Kosub

**Credits:**

keine

**Inhalte:**

Dieser einwöchige Kurs soll Studienanfängerinnen und -anfänger dabei unterstützen, sich auf die Arbeitsweise in den Bachelor-Studiengängen Information Engineering und Informatik einzustellen. Da erfahrungsgemäß viele Studienanfängerinnen und -anfänger mit mathematischen Inhalten Schwierigkeiten haben, werden Themen aus der Schulmathematik aufgegriffen und wiederholt, wobei hinsichtlich der Vorlesungen in den ersten Semestern Themen aus der diskreten Mathematik im Vordergrund stehen werden. Der Kurs gliedert sich in zwei Teile: einen (größeren) Vorlesungsteil und einen Übungsteil, in dem auf die vorher in der Vorlesung angesprochene Themen mit Aufgaben und Beispielen näher eingegangen wird. Dabei wird auch das eigenständige Lösen von Aufgaben Bestandteil der Übungen sein.

Schwerpunkte:

- Arithmetik
- Polynome
- Folgen
- Reihen
- Kombinatorik
- Induktion
- Lineare Gleichungssysteme

**Lernziele:**

Auffrischung und Angleichung mathematischer Kenntnisse und Arbeitsweisen aus der Schule am Beispiel von Fragestellungen aus der Algorithmik sowie der diskreten Mathematik.

**Voraussetzung:**

Keine.

**Literatur:**

Es wird keine Literatur vorausgesetzt.

## 5.2 Grundstudium - 1. Semester / Basic studies - 1. term

### 5.2.1 Konzepte der Informatik / Principles of Computer Science 1

**Modulverantwortlicher:** Prof. Dr. Deussen, Dr. Pampel (z.B. INF-11700-20152)

**Credits:** 5 SWS, 6 ECTS

**Inhalte:**

Das Modul Informatik 1 besteht aus der Vorlesung "Konzepte der Informatik" und dem "Programmierkurs 1". Es empfiehlt sich deswegen auch für Nebenfächler, beide Veranstaltungen zu belegen. Kern der Vorlesung "Konzepte der Informatik" sind die folgenden Themen:

- Informationscodierung und -speicherung - Codierung von Zahlen und Zeichen, Speicherbereiche, elementare Datentypen, Streuspeicherung
- Übersicht über die verschiedenen Programmierparadigmen, ausführlich den Kern imperativer Sprachen und Objektorientierung
- Algorithmen und Datenstrukturen - häufig verwendete Datenstrukturen wie Listen, Arrays, Stapel und Warteschlangen, Bäume und allg. Graphen; Eigenschaften von Algorithmen, insbesondere Algorithmenkomplexität und Korrektheit, sowie die algorithmische Konzepte Iteration und Rekursion, Teile und Herrsche, am Beispiel verschiedener Sortierverfahren
- Theoretische Grundlagen - Einführung in die Automatentheorie sowie formale Sprachen und Grammatiken; Fragen der Berechenbarkeit von Problemen, Komplexität und Korrektheit von Algorithmen
- Einführung in die Informationswissenschaft - Grundlagen, Information Retrieval, Wissensrepräsentation

**Lernziele:**

Die Absolventinnen und Absolventen kennen und verstehen die Grundlagen der Informationscodierung, -speicherung und -verarbeitung.

**Voraussetzung:**

Keine

**Arbeitsaufwand:**

180 Stunden, davon 75 Stunden Präsenzstudium und 105 Stunden Eigenstudium.

**Prüfungsleistung:**

- Studienleistung: 60% der Punkte aus den Übungen, mindestens 40% pro Aufgabenblatt
- Prüfungsleistung: Klausur von 120 Minuten Dauer, Teilnahmevoraussetzung ist das Absolvieren der Studienleistung
- Die Note entspricht der Klausurnote

**Literatur:**

Leider gibt es kein einzelnes Buch, das alle Inhalte der Vorlesung behandelt. Deswegen ist die folgende Liste eine Sammlung an Büchern, die die meisten relevanten Themen enthalten. Grundlagen der Informatik:

- Herold, Lurz & Wohlrab: *Grundlagen der Informatik* (Ibs 830/h27, ISBN 3-8273-7305-2)
- Grumm & Sommer: *Einführung in die Informatik* (Ibs 830/g95(7), ISBN 3-486-58115-7)
- Küchlin & Weber: *Einführung in die Informatik* (Ibs 843/k92(3), ISBN 3-540-20958-1)

Algorithmen und Datenstrukturen:

- Cormen: *Algorithmen - eine Einführung* (lbs 840/a53, ISBN 3-486-58262-8) oder das englische Original *Introduction to Algorithms* (lbs 830/c67(28), ISBN 0-262-03293-7)
- Sedgewick: *Algorithmen in Java* (Teil 1-4) (kid 112:n/s26-1/4, ISBN 3-8273-7072-8) oder das englische Original inklusive Graphalgorithmen: *Algorithms in Java, Part 1-4* (ISBN 0-201-36120-9), *Algorithms in Java, Part 5* (ISBN 0-201-36121-6)
- Ottmann & Widmayer: *Algorithmen und Datenstrukturen* (lbs 830/o99(4), ISBN 3-8274-1029-0)

## 5.2.2 Programmierkurs 1 (Imperative Sprache) / Programming Course 1 (Imperative Programming)

---

**Modulverantwortlicher:** Prof. Dr. Deussen, Dr. Fuchs (z.B. INF-11930-20152)

**Credits:** 4 SWS, 6 ECTS

**Inhalte:**

Das Modul Informatik 1 besteht aus der Vorlesung "Konzepte der Informatik" und dem "Programmierkurs 1". Es empfiehlt sich deswegen auch für Nebenfächler, beide Veranstaltungen zu belegen. Kern des "Programmierkurs 1" sind die folgenden Themen:

- Objektorientierte Programmierung - die in der Vorlesung "Konzepte der Informatik" vorgestellten Konzepte objektorientierter Programmiersprachen wie Klassen, Vererbung, Polymorphismus, Ausnahmebehandlung oder generische Programmierung werden praktisch mit Java an Hand verschiedenster Beispiele geübt
- Imperative Programmierung - Befehlsorientierte Programmierung mit Methoden, Schleifen und Auswahlbefehle.
- Angewandte Programmierung - Programmqualität, Dokumentation und Testen von Programmen

**Lernziele:**

Die Absolventinnen und Absolventen haben ein grundlegendes Verständnis der imperativen und objektorientierten Programmierung mit Java. Grundlegende Modellen können selbstständig implementiert werden.

**Voraussetzung:**

Keine

**Arbeitsaufwand:**

180 Stunden, davon 60 Stunden Präsenzstudium und 120 Stunden Eigenstudium.

**Prüfungsleistung:**

- nur unbenotete Studienleistung möglich
- 2 von 3 Programmieraufgaben bestanden
- 50% der Punkte aus dem Programmieretest

**Literatur:**

- Barnes, Kölling: Object first with Java. A practical introduction using BlueJ. oder die deutsche Übersetzung: Java lernen mit BlueJ: Eine Einführung in die objektorientierte Programmierung

## 5.2.3 Rechnersysteme und -netze / Computer Systems

**Modulverantwortlicher:**

Prof. Dr. Berthold (z.B. INF-11880-20152)

**Credits:**

5 SWS, 6 ECTS

**Inhalte:**

Die Vorlesung behandelt Grundlagen der Technischen Informatik wie Digitale Schaltungstechnik, Boolesche Algebra, Sequentielle Logik, Maschinensprache, Computerarchitektur, Assembler, Virtuelle Maschinen, Höhere Programmiersprachen, Compiler, Betriebssysteme und Netzwerktechnik.

**Lernziele:**

Die Absolventinnen und Absolventen haben ein grundlegendes Verständnis der kombinatorischen und sequentiellen Schaltungstechnik. Die grundlegenden Konzepte von Rechnerarchitekturen und Betriebssystemen (z.B. von Neumann) sind ebenso verstanden wie Techniken des Compilerbaus, der Virtuelle Maschinen und Assembler. Des Weiteren sollen die Studierenden Modelle der Netzwerktechnik (z.B. 5-Schichten-Modell) erörtern und verschiedene Protokolle (z.B. HTTP, SMTP, TCP, IP, ...) erläutern und in die besprochenen Modelle einordnen können.

**Voraussetzung:**

Keine

**Arbeitsaufwand:**

180 Stunden, davon 75 Stunden Präsenzstudium und 105 Stunden Eigenstudium.

**Prüfungsleistung:**

- Studienleistung: 60% der Punkte aus den Quizzes
- Prüfungsleistung: Klausur von 120 Minuten Dauer, Teilnahmevoraussetzung ist das Absolvieren der Studienleistung.
- Die Note entspricht der Klausurnote

**Literatur:**

**Rechnersysteme:**

- The Elements of Computing Systems: Building a Modern Computer from First Principles von Noam Nisan und Shimon Schocken, 2008, MIT Press (ISBN: 978-0-262-14087-4, lbs 846/n48:a)
- David A. Patterson, John L. Hennessy: Rechnerorganisation und -entwurf – Die Hardware/Software-Schnittstelle (3. Auflage 2005). Englische Ausgabe: Computer Organization and Design. Morgan Kaufmann Publishers
- Andrew S. Tanenbaum: Computerarchitektur, Strukturen - Konzepte - Grundlagen (5. Auflage 2006), Pearson/Prentice Hall; Englische Ausgabe: Structured Computer Organization
- Viele andere Bücher, z.B. durch eine Suche auf <http://www.ub.uni-konstanz.de> nach "Betriebssysteme" oder "Operating Systems".

**Netzwerke:**

- Kurose/Ross. Computer Networking – A Top-Down Approach Featuring the Internet. Addison-Wesley 2001
- Walter Proebster. Rechnernetze: Technik, Protokolle, Systeme, Anwendungen. Oldenbourg 1998

## 5.2.4 Analysis I / Analysis I

**Modulverantwortlicher:**

FB Mathematik (z.B. MAT-20000-20162)

**Credits:**

6 SWS, 9 ECTS

**Inhalte:**

Zentral sind Zahlen, Folgen, Reihen, metrische und normierte Räume, Stetigkeit, Differenzierbarkeit, Integration, Reihen von Funktionen

**Lernziele:**

Erwerb grundlegender Kenntnisse in den Grundlagen der Analysis, insbesondere in den Methoden der Differential- und Integralrechnung von Funktionen einer Veränderlichen

**Voraussetzung:**

Keine

**Arbeitsaufwand:**

Präsenzstudium: 84 Stunden, Eigenstudium: 186 Stunden

**Prüfungsleistung:**

schriftliche Prüfung von 90 Minuten Dauer. Die Note ergibt sich aus der schriftlichen Prüfungsleistung.

**Literatur:**

R. Denk, R. Racke: Kompendium der Analysis, Band 1. Vieweg+Teubner (2011).

## 5.2.5 Schlüsselqualifikation für Informatiker / Key Competences for Undergraduate Students of Information Engineering and Computer Science

---

**Modulverantwortlicher:**

Dr. Fuchs (z.B. INF-10175-20152)

**Credits:**

2 SWS, 3 ECTS

**Inhalte:**

Einführung in die Textverarbeitung mit  $\text{\LaTeX}$ , Einführung in Unix sowie ein Überblick über gängige Tools und Editoren:

- Unix-artige Umgebung und die Shell
- Textverarbeitung mit  $\text{\LaTeX}$
- Subversion
- Umgang mit der Shell, z.B. GNU coreutils
- Secure Shell
- ...

**Voraussetzung:**

Dieser Kurs richtet sich an Erstsemesterstudierende der Studiengänge „Information Engineering“ und „Informatik“.

**Arbeitsaufwand:**

90 Stunden, davon 30 Stunden Präsenzstudium und 60 Stunden Eigenstudium

**Prüfungsleistung:**

Regelmäßige Abgabe der Übungen während des Semesters

**Literatur:**

Wird im Kurs bekannt gegeben.

## 5.3 Grundstudium - 2. Semester / Basic studies - 2. term

### 5.3.1 Konzepte der Programmierung / Programming Concepts

**Modulverantwortlicher:** Jun.-Prof. Dr. Grossniklaus (z.B. INF-12070-20151)

**Credits:** 5 SWS, 4 ECTS

**Inhalte:**

Das Modul Informatik 2 besteht aus der Vorlesung "Konzepte der Programmierung" und dem "Programmierkurs 2" (deklarative Sprache). Kern des Moduls ist eine Einführung in deklarative Programmierung. Im Unterschied zur imperativen Programmierung wird dabei durch den Programmierer idealerweise nur vorgegeben, was berechnet werden soll, aber nicht wie genau die Berechnung durchgeführt wird. Am Beispiel der rein funktionalen Programmiersprache Haskell soll dieses Konzept eingeführt werden. Dabei werden Konzepte wie z.B. Seiteneffekte, Typsysteme, Auswertestrategien und Datenstrukturen erläutert und aus formaler Sicht betrachtet. Mit einer Einführung in den lambda-Kalkül wird die einfachste formale Grundlage fast aller Programmiersprachen vorgestellt, viele Haskell-Konstrukte lassen sich leicht darauf zurückführen. Vorlesungsbegleitend gibt der "Programmierkurs 2" eine praktische Einführung in die Programmierung mit Haskell. Da Vorlesung und Programmierkurs inhaltlich sehr eng verzahnt sind, werden die Übungen zu beiden Veranstaltungen zusammengelegt.

**Lernziele:**

Die Absolventinnen und Absolventen haben ein grundlegendes Verständnis von Programmierparadigmen und von funktionaler Programmierung. Sie sind in der Lage, selbständig kleinere Projekte in Haskell zu definieren und zu implementieren. Konzepte von Programmiersprachen sollen bewusst gemacht werden.

**Voraussetzung:**

keine

**Arbeitsaufwand:**

120 Stunden, davon 70 Stunden Präsenzstudium und 50 Stunden Eigenstudium.

**Prüfungsleistung:**

Leistungsnachweis: schriftliche Prüfung; die erfolgreiche Teilnahme an den Übungen ist Voraussetzung für die Zulassung zur Prüfung. Die Note ergibt sich aus der Note der schriftlichen Prüfung



### 5.3.2 Programmierkurs 2 (Deklarative Sprache) / Programming Course 2 (Declarative Programming)

---

**Modulverantwortlicher:** Jun.-Prof. Dr. Grossniklaus, Dr. Fuchs (z.B. INF-12120-20151)

**Credits:** 4 SWS, 5 ECTS

**Inhalte:**

Das Modul Informatik 2 besteht aus der Vorlesung "Konzepte der Informatik" und dem Programmierkurs 2. Kern des Moduls ist eine Einführung in deklarative und logische Programmierparadigmen. Im Unterschied zur imperativen Programmierung wird dabei durch den Programmierer nur vorgegeben, was berechnet werden soll, aber nicht, wie genau die Berechnung durchgeführt wird. Die erste Hälfte der Vorlesung wird sich mit Haskell beschäftigen. Haskell ist ein Beispiel für eine rein funktionale Programmiersprache, die ohne Seiteneffekte (wie etwa Variablenzuweisungen) auskommt. In der zweiten Hälfte wird Prolog als Beispiel einer logischen Programmiersprache behandelt. Vorlesungsbegleitend gibt der "Programmierkurs 2" eine praktische Einführung in diese beiden Programmiersprachen. Da Vorlesung und Programmierkurs inhaltlich eng verzahnt sind, werden die Übungen zu beiden Veranstaltungen zusammengelegt.

**Lernziele:**

Die Absolventinnen und Absolventen haben ein grundlegendes Verständnis von Programmierparadigmen und von deklarativer und logischer Programmierung. Sie sind in der Lage, selbständig kleinere Projekte in Haskell und Prolog zu definieren und zu implementieren

**Voraussetzung:**

Gleichzeitiger Besuch von "Konzepte der Programmierung"

**Arbeitsaufwand:**

150 Stunden, davon 56 Stunden Präsenzstudium und 94 Stunden Eigenstudium.

**Prüfungsleistung:**

schriftliche Prüfung wie angegeben in "Konzepte der Programmierung"

**Literatur:**

wird im Kurs bekannt gegeben

### 5.3.3 Datenbanksysteme / Database Systems

**Modulverantwortlicher:**

Prof. Dr. Scholl (z.B. INF-12040-20151)

**Credits:**

6 SWS, 9 ECTS

**Inhalte:**

Die Veranstaltung vermittelt einen grundlegenden Überblick über Funktionalität, Architektur und Realisierungskonzepte von Datenbanksystemen als Grundlage für computergestützte Informationssysteme. Charakteristisch für Datenbanksysteme ist, dass Informationen gemäß irgendeinem Modell in strukturierter Form dargestellt, gespeichert und aufbewahrt werden, die mittels Operationen einer geeigneten Sprache abgefragt (wiedergewonnen) und manipuliert werden können. Im Vordergrund stehen die Schnittstellen, d.h. die Nutzersicht, Implementierungsaspekte werden nur angerissen. In dieser Veranstaltung werden sowohl die Modellierungs- wie auch die Nutzungsaspekte von Datenbanksystemen vermittelt: z.B. Entity-Relationship- und Relationale Datenmodellierung, Relationale Entwurfstheorie und Normalformen, Datenbanksprachen (insbes. Algebra, Kalkül, SQL), ACID-Transaktionen. Die Lehrveranstaltung liefert Grundlagen für weiterführende Lehrveranstaltungen aus den Gebieten Datenbanken, Informationssysteme und Information Retrieval.

**Lernziele:**

Die Absolventinnen und Absolventen des Kurses haben grundlegendes Verständnis über den Aufbau und die Funktionsweise von Datenbanksystemen und deren Nutzung. Sie haben fundiertes Wissen über konzeptionelle Datenmodellierung mit Hilfe des Entity-Relationship-Modells und die Abbildung auf relationale Datenbankschemata. Sie können die grundlegenden Sprachkonstrukte von SQL mittels mathematisch präziser formaler Sprachen (Algebra, Kalkül) analysieren und können SQL-Anfragen und -Änderungsoperationen selbstständig formulieren und anwenden. Sie haben die prinzipiellen Realisierungstechniken solcher deklarativer Sprachen kennen gelernt und können bestehende SQL-Anwendungen analysieren und bewerten. Sie sind in der Lage, grundlegende Informationssystem-Funktionalitäten selbstständig zu realisieren. Die Funktionsweise und Abstraktionsmechanismen der transaktionsorientierten Verarbeitung sind ihnen bekannt, sie können Synchronisations- und Recovery-Probleme erkennen und grundsätzliche Lösungsmöglichkeiten aufzeigen.

**Voraussetzung:**

Entsprechend den Modulen Mathematik: Diskrete Strukturen (paralleler Besuch) und Konzepte der Informatik, elementare Programmierkenntnisse.

**Arbeitsaufwand:**

270 Stunden, davon 84 Stunden Präsenzstudium und 186 Stunden Eigenstudium.

**Prüfungsleistung:**

Prüfung: Klausur von 120min Dauer. Die erfolgreiche Teilnahme an den Übungen ist Voraussetzung für die Zulassung zur Klausur. Die Note ergibt sich aus der Klausurnote.

**Literatur:**

Es gibt eine große Auswahl an guten Standard-Lehrbüchern über Datenbanken, sowohl in deutscher wie auch in englischer Sprache (siehe auch DB-Literatur). Die Veranstaltung folgt einem eigenen Aufbau und orientiert sich nicht an einem einzelnen Buch. Als Einstieg kann dennoch bspw. dienen:

- Kemper, A. und A. Eickler (2009). Datenbanksysteme: Eine Einführung. Oldenbourg, 7. Aufl.

### **5.3.4 Mathematik: Diskrete Strukturen / Mathematics: Discrete Structures**

**Modulverantwortlicher:**

Apl. Prof. Dr. Kosub (z.B. INF-12090-20151)

**Credits:**

6 SWS, 9 ECTS

**Inhalte:**

- Algebraische Strukturen (Algebren, Gruppen, endliche Körper)
- Kombinatorik (Abzählprinzipien, Urnenmodelle, Binomialkoeffizienten, Permutationen, Partitionen)
- Graphentheorie (Graphen, Bäume und Wälder, Kreise, planare Graphen, Färbungen, Matchings)
- Rekursionen (Lineare Rekursionsgleichungen, erzeugende Funktionen, höhere Rekursionsgleichungen)
- Diskrete Wahrscheinlichkeitstheorie (Wahrscheinlichkeitsräume, bedingte Wahrscheinlichkeit, Unabhängigkeit, Zufallsvariablen, Gesetz der großen Zahlen, wichtige diskrete Verteilungen)

**Lernziele:**

Beherrschung der grundlegenden Konzepte der Diskreten Mathematik, wie sie für Informatik und Information Engineering benötigt und vorausgesetzt werden.

**Voraussetzung:**

Kenntnisse entsprechend der Vorlesung Mathematische Grundlagen der Informatik

**Arbeitsaufwand:**

270 Stunden, davon 84 Stunden Präsenzstudium und 186 Stunden Eigenstudium.

**Prüfungsleistung:**

Klausur, 120 Minuten, sowie 50% der Punkte aus den wöchentlichen Übungen

**Literatur:**

- Angelika Steger. *Diskrete Strukturen 1. Kombinatorik, Graphentheorie, Algebra*. 2., Auflage, Springer-Verlag, Berlin, 2007.
- Thomas Schickinger, Angelika Steger. *Diskrete Strukturen 2. Wahrscheinlichkeitstheorie und Statistik*. 2., Auflage, Springer-Verlag, Berlin, 2007.

## **5.4 Grundstudium - 3. Semester / Basic studies - 3. term**

### **5.4.1 Betriebssysteme und systemnahe Programmierung / Operating Systems and Systems Programming**

---

**Modulverantwortlicher:** Prof. Dr. Waldvogel (z.B. INF-11740-20152)

**Credits:** 4 SWS, 4 ECTS

**Inhalte:**

Die Veranstaltung Betriebssysteme, welche zusammen mit dem Programmierkurs III das Modul Informatik III bildet, beinhaltet eine grundlegende Einführung in Konzepte zum Aufbau und Methoden zum Betrieb von Betriebssystemen. Die Vorlesung bietet sowohl einen theoretischen Einblick in die allgemeinen Konzepte und Problemstellungen als auch praktische Anwendungsbeispiele. Studierende sollen durch direktes Umsetzen der vorgestellten Konzepte und Methoden zum einen ein tieferes Verständnis erlangen, zum anderen praktische Erfahrung im Bereich der systemnahen Programmierung sammeln. Der Programmierkurs III, welcher durch die Prüfungsordnung (BA Informatik) gefordert wird, dient neben dem Erlernen der Programmiersprache C zusätzlich dem praktischen Umsetzen der in der Veranstaltung Betriebssysteme erlernten Konzepte und Methoden, welche mit direktem Bezug auf aktuelle Hardwarekomponenten umgesetzt werden können. Ein Besuch wird durch die Lehrenden dringend empfohlen.

**Lernziele:**

Die Verwendung, Spezifikation und Dokumentation der Systemprogrammierschnittstelle (natives API, system call interface) von Betriebssystemen findet (fast immer) mittels der Programmiersprache C statt. Eine grundlegende Einführung in diese Sprache ist daher Bestandteil der ersten Übungsstunden der Veranstaltung. Die Vorlesung kann als Grundlagenveranstaltung für zahlreiche weiterführende Veranstaltungen verstanden werden und soll Studenten befähigen, Bachelor-/Masterprojekte an system- und/oder hardwarenahen Forschungsprojekten im Fachbereich annehmen zu können.

**Voraussetzung:**

Modul Informatik I, Rechnersysteme (Computerarchitektur, Assembler)

**Arbeitsaufwand:**

120 Stunden, davon 60 Stunden Präsenzstudium und 60 Stunden Eigenstudium.

**Prüfungsleistung:**

- Wöchentliche Übungen in Zweiergruppen: Je Übungsblatt mindestens 50% der Punkte, "three strikes and you're out"(d.h. zwei Übungsblätter mit unter 50% der Punkte werden noch toleriert)
- 3 Quizzes über das Semester verteilt über den gesamten bisherigen Stoff von BS+PK3: Bestehen von mindestens zwei davon; die besseren zwei zählen zu je 10% zur Klausurnote
- Bestehen der Veranstaltung Programmierkurs 3
- Gemeinsame Klausur über BS (70% der Punkte) und PK3 (30%) am Ende des Semesters

**Literatur:**

- William Stallings: Operating Systems, Internals and design principles, 6th / 7th ed.  
<http://williamstallings.com/OS/OS6e.html>  
<http://williamstallings.com/OperatingSystems/>
- Andrew S. Tanenbaum: Operating Systems Design & Implementation 3rd Edition

## 5.4.2 Programmierkurs 3 (Systemnahe Programmierung) / Programming Course 3 (System Programming)

---

**Modulverantwortlicher:**

Prof. Dr. Waldvogel (z.B. INF-11850-20152)

**Credits:**

4 SWS, 3 ECTS

**Inhalte:**

Der Programmierkurs „Systemnahe Programmierung“ betrachtet anhand von Unix/Linux Betriebssysteme aus einer praktischen Perspektive heraus. Es wird davon ausgegangen, dass die Studierenden die Vorlesung „Betriebssysteme und systemnahe Programmierung“ parallel hören oder schon Vorkenntnisse in diesem Bereich haben.

Der Programmierkurs beinhaltet:

1. Die Programmiersprache C
2. Systemnahe Programmierung in Linux

Die Inhalte sind: Einführung in die Programmiersprache C, Zeigerarithmetik, Präprozessor, "make"; einfaches I/O, Prozesssteuerung, IPC über Pipes, Shared Memory, Socket und Signale; Synchronisation mit Semaphoren, Linker, Relocation, statische und dynamische Bibliotheken.

**Lernziele:**

Für fortgeschrittene Programmierung in der Linux / Unix-Umgebung ist es wichtig, die Dienste des Linux-Kernels (System Calls) und der C-Bibliothek zu kennen. Zu den Themen gehören File, Buffered und Advanced I/O (z. B. scatter / gather, poll, memory mapped I/O), Prozessmanagement, Datei- und Verzeichnisverwaltung, Speicherverwaltung, Signale und Zeitfunktionen. Die Linux Kernel-Entwicklung bietet einen Einblick in das Lesen und Verbessern von Kernel-Code. Themen sind die Entwicklung von Linux-Kernel-Modulen und einfachen Gerätetreibern. Aspekte wie Kernel-Threads und grundlegende Kernstrukturen werden diskutiert.

**Voraussetzung:**

keine

**Arbeitsaufwand:**

90 Stunden, davon 60 Stunden Präsenzstudium und 30 Stunden Eigenstudium.

**Prüfungsleistung:**

- Wöchentliche Übungen in Zweiergruppen: Je Übungsblatt mindestens 50% der Punkte, "three strikes and you're out"(d.h. zwei Übungsblätter mit unter 50% der Punkte werden noch toleriert)
- 3 Quizzes im Rahmen von BS über das Semester verteilt über den gesamten bisherigen Stoff von BS+PK3: Bestehen von mindestens zwei davon; die besseren zwei zählen zu je 10% zur Klausurnote
- Bestehen der Veranstaltung Betriebssysteme und Systemnahe Programmierung
- Gemeinsame Klausur über BS (70% der Punkte) und PK3 (30%) am Ende des Semesters

**Literatur:**

- Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. 274 pages, April 1988, 2nd ed., Prentice Hall. ISBN-10: 0131103628, ISBN-13: 978-0131103627.
- W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment. 927 pages, February 2008, 2nd revised ed., Addison-Wesley Professional. ISBN-10: 0321525949, ISBN-13: 978-0321525949.

### 5.4.3 Software Engineering / Software Engineering

**Modulverantwortlicher:**

Prof. Dr. Leue (z.B. INF-11890-20152)

**Credits:**

5 SWS, 5 ECTS

**Inhalte:**

Die Veranstaltung Software Engineering führt in Verfahren, Methoden und Werkzeuge zum ingenieurmäßigen Entwurf von Softwaresystemen ein. Sie beschäftigt sich mit Software-Prozessmodellen, dem objektorientierten Entwurf von Software und ihrer Architektur, der Spezifikation, Verifikation und dem Testen von Software, der Planung und Durchführung von Softwareprojekten und der quantitativen Bewertung von Software-Artefakten. Gleichzeitig bereitet sie die TeilnehmerInnen auf die prozessorientierte Durchführung von Softwareprojekten vor.

**Lernziele:**

Die TeilnehmerInnen werden in die Lage versetzt, Software-Entwicklungsmethoden zu bewerten, ihren Einsatz zu strukturieren, und selber Software-Entwicklungsprojekte zu leiten.

**Voraussetzung:**

Erfolgreiche Teilnahme an den Lehrveranstaltungen Modul Informatik I, Modul Informatik II, Diskrete Strukturen.

**Arbeitsaufwand:**

150 Stunden, davon 75 Stunden Präsenzstudium und 75 Stunden Eigenstudium.

**Prüfungsleistung:**

Klausur, 90 Minuten, plus Studienleistungen. Details werden während der Veranstaltung bekannt gegeben.

**Literatur:**

- Ian Sommerville. Software Engineering. 9th edition. Pearson Education Ltd. 2010.
- Bernd Brügge, Allen H. Dutoit. Object-Oriented Software Engineering. Using UML, Patterns and Java. 3rd edition, Prentice-Hall, 2009.
- Weitere Literatur wird während der Veranstaltung bekannt gegeben.

## 5.4.4 Algorithmen und Datenstrukturen / Algorithms and Data Structures

**Modulverantwortlicher:**

Prof. Dr. Brandes (z.B. INF-20250-20152)

**Credits:**

6 SWS, 9 ECTS

**Inhalte:**

In der Vorlesung werden Standardalgorithmen und grundlegende Datenstrukturen behandelt. Darstellungsformen und Spezifikation von Algorithmen, elementare und höhere Datenstrukturen, Suchbäume, Hash-Tabellen, rekursive Algorithmen, Algorithmen zum Suchen und Sortieren, grundlegende Graphenalgorithmen und Zeichenkettenalgorithmen. In theoretischen Übungen wird der Vorlesungsstoff vertieft, in praktischen Übungen werden Algorithmen und Datenstrukturen in Java implementiert.

- Einführung
- Sortieren
- Suchen
- Streuen
- Ausrichten
- Graphen

**Lernziele:**

Kenntnis elementarer Algorithmen und Datenstrukturen. Auswahl und Effizienzbeurteilung von alternativen Verfahren und Implementationen in Anwendungsszenarien. Fähigkeit zu deren Anpassung an gegebene Umstände sowie Entwurf neuer Algorithmen und Datenstrukturen.

**Voraussetzung:**

Für die praktischen Übungsaufgaben: elementare Programmierkenntnisse in der Programmiersprache Java

**Arbeitsaufwand:**

270 Stunden, davon 90 Stunden Präsenzstudium und 180 Stunden Eigenstudium.

**Prüfungsleistung:**

Teilnahme an den Übungen (50% der Punkte sind Zulassungsvoraussetzung für die Klausur); Klausur (120min)

**Literatur:**

Es wird ein Skript mit allen Inhalten zur Verfügung gestellt. Weitere Literatur:

- N. Blum: Algorithmen und Datenstrukturen. Oldenbourg, 2004
- T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein: Algorithmen - Eine Einführung. Oldenbourg, 2007 (2. Aufl.)
- T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen. Spektrum Akademischer Verlag, 2002 (4. Aufl.)
- P. Sanders, K. Mehlhorn: Algorithms and Data Structures. Springer, 2008
- R. Sedgwick, K. Wayne: Algorithmen - Algorithmen und Datenstrukturen. Pearson, 2014 (4. Aufl.)
- U. Schöning: Algorithmik. Spektrum Akademischer Verlag, 2001
- M.A. Weiss: Data Structures and Algorithm Analysis in Java. Pearson, 2007 (2nd ed.)

## 5.4.5 Lineare Algebra I / Linear Algebra I

**Modulverantwortlicher:**

FB Mathematik (z.B. MAT-20005-20162)

**Credits:**

6 SWS, 9 ECTS

**Inhalte:**

Algebraische Grundlagen, Vektorräume, Lineare Abbildungen und ihre Darstellbarkeit durch Matrizen, Dual und Bidual, der Transponierte Operator, Determinanten.

Algebraic foundations, vector spaces, linear maps between vector spaces and their representability by matrices, Dual and Double dual space, the transpose of a linear operator, Determinants.

**Lernziele:**

Die Studierenden sollen instand gesetzt werden, selbstständig und souverän mit abstrakten Konzepten umzugehen, einfache Beweise eigenständig zu führen, mathematisch exakt darzustellen und vorgelegte Beweisversuche kritisch auf ihre Korrektheit zu überprüfen.

The students are to be enabled to independently handle abstract notions, find easy proofs, present those in accordance with the standards of mathematical rigor and to critically check proof attempts for their correctness.

**Voraussetzung:**

Grundkenntnisse im Umgang mit logischen Zeichen, Mengen, Funktionen und einfachen Beweisformen (direkt/indirekt/induktiv), wie sie z.B. im Vorkurs Mathematik vor Semesterbeginn vermittelt werden.

Basic skills in handling logical symbols, sets, functions and proofs (direct/indirect/inductive) as taught in the preparatory class for mathematics prior to the start of the semester.

**Arbeitsaufwand:**

Präsenzstudium: 84 Stunden, Eigenstudium: 186 Stunden

**Prüfungsleistung:**

Es gibt eine Klausur Anfang Februar und eine zweite Anfang März. Die Endnote ist die bessere der folgenden beiden Noten:  $1/5 \cdot \text{Note 1. Klausur} + 4/5 \cdot \text{Note 2. Klausur}$  bzw. Note 2. Klausur. Außerdem sind zum Bestehen mindestens 50% der Punkte bei Bearbeitung der Übungsblätter zu erreichen. Ist das der Fall und ist die Endnote mindestens 4.0, ist das Modul bestanden. Der Leistungsnachweis erfolgt in Form eines benoteten Scheins.

There will be a first exam early in February and a second exam early in March. The final grade is the better one of the following two grades:  $1/5 \cdot \text{first exam grade} + 4/5 \cdot \text{second exam grade}$  and second exam grade. For passing, it is also required to score at least 50% on the exercise sheets. If this is the case and the final grade is at least 4.0, the student has passed the module and receives a graded certificate.



## 5.5 Grundstudium - 4. Semester / Basic studies - 4. term

### 5.5.1 Software-Projekt / Software Project

**Modulverantwortlicher:** Prof. Dr. Schreiber (z.B. INF-12160-20151)

**Credits:** 4 SWS, 6 ECTS

**Inhalte:**

Die Studierenden entwickeln in Projektgruppen ein Softwaresystem mittlerer Komplexität. Dabei erstellen sie eine Analyse, einen Entwurf und schliesslich die Implementierung des Systems.

**Lernziele:**

Im Softwareprojekt sollen Teilnehmerinnen und Teilnehmer im Team ein realistisches Projekt in allen Schritten (Planung, Analyse, Umsetzung und Implementierung) durchführen. Lernziele sind u.a. das Setzen von Projektzielen, die Auswahl und Verwendung einer geeigneten Software-Entwicklungsmethode, die Erstellung und Weiterführung eines Projektplans, das Erkennen von Problemen und deren Behandlung, die Selbstorganisation im Team (inklusive Aufgabenverteilung, Kommunikation, Teammotivation), die Programmierung von Projektteilen im Team (mit Abhängigkeiten von anderen Teammitgliedern), sowie die Präsentation des Softwareprojekts.

**Voraussetzung:**

Kenntnisse entsprechend der Veranstaltungen *Konzepte der Informatik* und *Konzepte der Programmierung* und *Algorithmen und Datenstrukturen*. Erfolgreiche Teilnahme an der Veranstaltung *Software Engineering*, mit der gemeinsam dieses Praktikum das Modul *Software Engineering* darstellt. Die Veranstaltung *Software Engineering* legt die notwendigen methodischen Grundlagen, ohne die dieses Projekt nicht erfolgreich absolviert werden kann.

**Arbeitsaufwand:**

180 Stunden, davon 40 Stunden Präsenzstudium (inklusive Konsultationszeit mit Gruppenbetreuern) und 140 Stunden Eigenstudium (Erstellung der Projektarbeiten, Implementierung).

**Prüfungsleistung:**

Projektarbeiten (Deliverables) und Implementierung des Systems sowie deren Präsentation, Kopfnoten, mündliche Prüfung

## 5.5.2 Theoretische Grundlagen der Informatik / Theoretical Computer Science

**Modulverantwortlicher:**

Prof. Dr. Saupe (z.B. INF-22101-20151)

**Credits:**

6 SWS, 9 ECTS

**Inhalte:**

Die Vorlesung gibt eine Einführung in die theoretischen Grundlagen der Informatik. Folgende Themen werden u.a. behandelt.

- Formale Sprachen und Automatentheorie: Chomsky-Hierarchie (reguläre, kontextfreie, kontext-sensitive, und Typ0-Sprachen, reguläre Ausdrücke), Grammatiken (Typen, Eindeutigkeit, Abgeschlossenheit), Automatenmodelle (endliche Automaten, Kellerautomaten, Turingmaschinen).
- Entscheidbarkeit und Berechenbarkeit: Entscheidbarkeit, Berechenbarkeit, Aufzählbarkeit, Universelle Turingmaschine, Diagonalisierung, Halteproblem,  $\mu$ -rekursive Funktionen, Church/Turing-These, Gödels Unvollständigkeitstheorem.
- Komplexitätstheorie: Entscheidungsprobleme, Reduzierbarkeit, Klassen P und NP, NP-Vollständigkeit.

**Lernziele:**

Der/die Studierende

- besitzt einen Einblick in die Grundlagen der Theoretischen Informatik und beherrscht deren Berechnungsmodelle und Beweistechniken,
- hat die Fähigkeit, die Standardkonstruktionen aus dem Bereich endlicher Automaten, regulärer Ausdrücke und Grammatiken auszuführen,
- hat ein Verständnis für die Unterscheidung von Berechenbarkeit und Unberechenbarkeit, sowie ein Grundverständnis des Begriffs der NP-Vollständigkeit und seiner Motivation.

**Voraussetzung:**

Keine. Es wird jedoch empfohlen, folgende Vorlesungen zuvor gehört zu haben: Algorithmen und Datenstrukturen, sowie die mathematischen Grundvorlesungen Diskrete Strukturen, Analysis I und Lineare Algebra (im Studiengang Informatik) bzw. Diskrete Strukturen und Mathematische Grundlagen der Informatik (im Studiengang Information Engineering).

**Arbeitsaufwand:**

270 Stunden, davon 84 Stunden Präsenzstudium und 186 Stunden Eigenstudium.

**Prüfungsleistung:**

Voraussetzung zur Zulassung zur Klausur: Regelmäßige und erfolgreiche Bearbeitung von Übungsaufgaben. Erreichen von 60% der Punkte aus den Übungsaufgaben.

**Literatur:**

- U. Schöning, Theoretische Informatik - kurzgefasst, Spektrum Akademischer Verlag, 4. Auflage, 2001.
- J. E. Hopcroft, R. Motwani, J. D. Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, Pearson Studium, 2. Auflage, 2002.
- I. Wegener, Theoretische Informatik - eine algorithmische Einführung, B. G. Teubner Verlag, 2. Auflage, 1999.

## 5.5.3 Computergrafik und interaktive Systeme / Computer Graphics and Interactive Systems

---

**Modulverantwortlicher:** Prof. Dr. Reiterer, Prof. Dr. Deussen (z.B. INF-12020-20151)

**Credits:** 6 SWS, 9 ECTS

**Inhalte:**

Die Veranstaltung besteht aus zwei Teilen: Im Teil Computergrafik wird eine Einführung in die interaktive Computergrafik unter OpenGL gegeben. Die Studierenden lernen den Weg von den Eingabedaten (geometrischen Beschreibungen der Objekte) bis hin zum Pixel des Ausgabebildes kennen

- Daten-Vorverarbeitung (Transformation, Projektion, Clipping)
- Rasterisierung (Scanline-Verfahren, Tiefenpuffer)
- Schattierungsmethoden (Gourand shading, Phong shading)
- lokale vs. globale Beleuchtungsverfahren
- Raytracing, Radiosity und bildbasiertes Rendering
- Texturierung

Im Teil Interaktive Systeme wird ein umfassender Überblick über die Ziele und Forschungsfragen der Disziplin Mensch-Computer Interaktion gegeben. Die Studierenden erhalten ein Grundverständnis, wie man interaktive Systeme unter besonderer Berücksichtigung der Benutzeranforderungen entwickeln kann.

- Grundlagen der menschlichen Wahrnehmung
- Methoden und Techniken des User-Interface Designs
- Design von gebrauchstauglichen Produkten, deren Benutzung auch Spaß bereiten soll
- grundlegende Gestaltungsprinzipien, Methoden und Werkzeuge werden vorgestellt und in praktischen Übungen vertieft
- Techniken zur Evaluation von Benutzerschnittstellen

**Lernziele:**

Die Absolventinnen und Absolventen des Kurses haben grundlegendes Verständnis über den Aufbau graphisch-interaktiver Systeme und deren Realisierung mit OpenGL. Sie haben fundiertes Wissen über die Rasterisierungs-Pipeline und können sie in unterschiedlichen Kontexten anwenden und einsetzen. Sie haben Grundlagen der menschlichen Informationsverarbeitung (Wahrnehmung, Kognition, Motorik, Mentale Modelle und Fehler) kennen gelernt. Sie haben Grundregeln der User Interface Gestaltung kennen gelernt und können diese auf etablierte Interaktionsstile (Kommandos, Dialoge, Direkte Manipulation, Suchen und Browsen, interaktive Visualisierungen) anwenden.

Sie sind mit den Grundideen des User Centered Design vertraut und kennen grundlegende Methoden und Techniken zur Entwicklung interaktiver Systeme (z.B. Erheben von Benutzeranforderungen, Erstellen von Skizzen und Prototypen, Evaluationstechniken). Sie können existierende interaktive Systeme analysieren und bewerten. Sie sind in der Lage, grundlegende Interaktionsdesigns selbständig zu realisieren. Sie kennen verschiedene Einsatzformen ausgewählter Interaktionsformen im Bereich GUIs, WWW, Multitouch Oberflächen und mobile Interaktion.

**Voraussetzung:**

Entsprechend den Modulen Mathematische Grundlagen des Information Engineering und Methoden der Praktischen Informatik, elementare Programmierkenntnisse.

**Arbeitsaufwand:**

270 Stunden, davon 84 Stunden Präsenzstudium und 186 Stunden Eigenstudium.

**Prüfungsleistung:**

Prüfung: Klausur von 120min Dauer. Die erfolgreiche Teilnahme an den Übungen ist Voraussetzung für die Zulassung zur Klausur. Die Note ergibt sich aus der Klausurnote.

**Literatur:**

Teil Computergrafik:

- Edward Angel: Interactive Computer graphics, a top-down approach with OpenGL (in der Bibliothek mehrfach vorhanden - Die Vorlesungsfolien reichen nicht als Vorbereitung zur Prüfung!)

Teil Interaktive Systeme:

- Butz A., Krüger A.: Mensch-Maschine Interaktion, Oldenburg, 2014
- Preim, B., Dachsel, R.: Interaktive Systeme - Band 1, (2. Auflage), Springer Verlag, 2010
- Richter, M., Flückiger, M. D.: Usability Engineering kompakt: Benutzbare Software entwickeln, (2.Auflage), Springer Verlag, 2010

## 5.5.4 Statistik I / Statistics I

### **Anmerkung:**

Alle an der Universität Konstanz angebotenen Veranstaltungen zu Statistik I werden für dieses Modul anerkannt.

### **Modulverantwortlicher:**

z.B., FB Psychologie (z.B. PSY-10094-20162)

### **Credits:**

4 SWS, 6 ECTS

### **Lernziele:**

Die Studierenden kennen die Grundlagen und Grundbegriffe der Methodologie und Methodik. Die Studierenden können einen Zusammenhang herstellen zwischen verschiedenen Spezialgebieten der psychologischen Methodenlehre untereinander sowie mit substanzwissenschaftlichen Inhalten der Psychologie. Die Studierenden verbessern ihr methodisches Denken und ihre Argumentationsfähigkeit. Die Studierenden können Strategien zur Aufgabenlösung entwickeln und anwenden. Die Studierenden können besser planen und Probleme lösen. Die Studierenden orientieren sich auf Anwendungen. Die Studierenden sind leistungsbereit, frustrationstolerant und belastbar.

### **Arbeitsaufwand:**

Insgesamt 180 Stunden, davon 56 Stunden Präsenzstudium und 124 Stunden Eigenstudium

### **Prüfungsleistung:**

Klausur

### 5.5.5 Numerik I / Numeric I

**Modulverantwortlicher:**

FB Mathematik (z.B. MAT-10265-20152)

**Credits:**

6 SWS, 10 ECTS

**Inhalte:**

Diese Lehrveranstaltung gibt eine Einführung in grundlegende numerische Algorithmen.

Es werden folgende Themen behandelt:

- Nullstellenverfahren (ein- und mehrdimensional),
- Interpolation
- numerische Integration
- lineare Gleichungssysteme
- linearer Ausgleich
- lineare Optimierung
- Lösung von Anfangswertaufgaben
- Iterationsverfahren für lineare bzw. nichtlineare Gleichungssysteme
- Eigenwertaufgaben
- Minimierung
- Stabilitäts- und Störungsfragen

Im Rahmen der Übungen lernen die Studierenden, selbständig Aufgaben am Computer zu lösen (mit Matlab).

Diese Übungen finden im Rechnerraum V 203 statt, für den ein User-Account erforderlich ist. Bitte besorgen Sie sich rechtzeitig eine solche Zugangsberechtigung (über die Homepage <http://www.phyma.uni-konstanz.de>)

**Voraussetzung:**

Analysis I und II, Lineare Algebra I und Computereinsatz in der Mathematik.

## **5.6 Vertiefungsstudium / Advanced studies**

### **5.6.1 Analyse und Visualisierung von Information / Analysis & Visualization**

**Modulverantwortlicher:** Prof. Dr. Keim (z.B. INF-11710-20152)

**Credits:** 6 SWS, 9 ECTS

**Inhalte:**

- Datenaufbereitung und Datenvorverarbeitung
- Grundlegende Methoden des Data Mining
- Analysealgorithmen und -methoden
- Grundlagen der Informationsvisualisierung
- Menschliche Wahrnehmung und Visualisierungsdesign
- Visualisierungsmöglichkeiten verschiedener Datentypen

**Lernziele:**

Die Studierenden erhalten grundlegende theoretische Kenntnisse und machen erste praktische Erfahrungen im Bereich der Datenanalyse. Sie sind in der Lage, Voraussetzungen und Parameter für die Anwendung verschiedener, gängiger Analysealgorithmen zu benennen, diese praktisch anzuwenden und die erreichten Ergebnisse eigenständig zu beurteilen. Im Bereich der Visualisierung lernen die Studierenden, ausgehend von verschiedenen Datentypen, unterschiedliche geeignete visuelle Abbildungen kennen. Anhand von Beispielen erlernen sie die qualitative Analyse der Visualisierungsgüte, sowie die autonome Beurteilung von Designentscheidungen hinsichtlich der menschlichen Wahrnehmung. Dadurch können sie selbstständig Visualisierungen entwickeln und beurteilen.

**Voraussetzung:**

Modul Informatik 1, Mathematische Grundlagen der Informatik

**Arbeitsaufwand:**

Insgesamt 270 Stunden, verteilt auf 90 Stunden Präsenzstudium und 180 Stunden Eigenstudium

**Prüfungsleistung:**

Klausur sowie erfolgreiches Absolvieren von mind. 50% der Übungsaufgaben

**Literatur:**

- Han J., Kamber M., Data Mining: Concepts and Techniques, 2006, Morgan Kaufmann Publishers, Second Edition
- Berthold M., Borgelt C., Höppner F., Klawonn F., Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data: Making Practical Sense of Real Data (Texts in Computer Science), 2010 Springer
- Ester M., Sander J., Knowledge Discovery in Databases. Techniken und Anwendungen, 2000 Springer Verlag
- Hand D.J., Mannila H., Smyth P., Principles of Data Mining, 2001, MIT Press
- Spence R., Information Visualization, 2007, ACM Press Books, Second Edition
- Ward M. and Grinstein, G. and Keim D. A., Interactive Data Visualization: Foundations, Techniques, and Application, 2010, A.K. Peters, Ltd, ISBN: 978-1-56881-473-5, <http://www.idvbook.com>

## **5.6.2 Allgemeine Seminarbeschreibung / General description of seminars**

**Credits:**

2 SWS, 4 ECTS

**Semesterzuordnung:**

Seminare werden in jedem Semester angeboten.

**Inhalte:**

Im Seminar wird unter Anleitung ein wissenschaftlicher Vortrag über ein gegebenes Thema vorbereitet und gehalten. Von den Teilnehmerinnen und Teilnehmern des Seminars werden Fragen gestellt. Darüber hinaus wird eine schriftliche Arbeit, z.B. in Form einer schriftlichen Ausarbeitung des Vortrags, verlangt. Dazu erhalten die Studierenden Anleitung im wissenschaftlichen Schreiben und üben Literaturarbeit.

Seminare werden von allen Arbeitsgruppen des Fachbereichs angeboten. Die Themen stammen beispielhaft aus den Bereichen:

- Algorithmen
- Bioinformatik
- Computergrafik und Medieninformatik
- Datenbanksysteme
- Datenanalyse und -visualisierung
- Data Mining
- Formale Grundlagen
- Mensch-Computer-Interaktion
- Multimedia Signalverarbeitung
- Analyse sozialer Netzwerke
- Software Engineering
- Visual Analytics
- Verteilte Systeme

**Lernziele:**

Die Studierenden sind in der Lage, eigenständig eine wissenschaftliche Präsentation auszuarbeiten, vorzutragen und Fragen zu beantworten. Sie haben ein grundlegendes Verständnis über das wissenschaftliche Arbeiten und das Verfassen von wissenschaftlichen Veröffentlichungen inklusive des richtigen Umgangs mit Literatur.

**Voraussetzung:**

In der Regel ein bis zwei der relevanten Vorlesungen aus dem Basis- und/oder Vertiefungsstudium. Ein Seminar kann begleitend zu einem Bachelor- oder Masterprojekt angeboten werden.

**Arbeitsaufwand:**

120 Stunden, davon ca. 30 Stunden Präsenzstudium und ca. 90 Stunden Selbststudium

**Literatur:**

Wird im jeweiligen Seminar individuell ausgegeben

**Assignment to semester:**

Seminars are offered every semester.

**Contents:**

In a seminar, an academic presentation on a given topic is prepared and held under guidance. Questions will be asked by the participants in the seminar. Additionally, a term paper is required, which can for example have the form of a written elaboration of the presentation.



For this, students receive guidance in scientific writing and practice how to use academic sources.

Seminars are offered by all working groups of the faculty. The topics are taken from the following areas:

- Algorithmics
- Bioinformatics
- Computer Graphics & Media Informatics
- Databases and Information Systems
- Data Analysis & Visualization
- Distributed Systems
- Human-Computer Interaction
- Information Mining
- Multimedia Signal Processing
- Social Network Analysis
- Software Engineering
- Theory of Computing
- Visual Analytics

**Learning objectives:**

The students are able to work out a scientific presentation independently, to present their findings and to answer questions. They have a basic understanding of scientific practices and the writing of scientific publications, including the correct handling of literature.

**Prerequisites:**

As a rule, one or two of the relevant lectures from the basic and / or advanced studies. A seminar can be offered accompanying a Bachelor or Master project.

**Workload:**

120 hours, approximately 30 hours in class and 90 hours self-study

**Literature:**

Relevant literature will be distributed in the respective seminar.

### 5.6.3 Allgemeine Bachelor-Projektbeschreibung / General description of the Bachelor's project

---

**Credits:** 9 ECTS

**Semesterzuordnung:**

BA-Projekte werden in jedem Semester angeboten.

**Inhalte:**

Das Projekt bereitet auf das Schreiben der Bachelorarbeit vor. Dazu arbeiten sich die Studierenden eigenständig unter Anleitung in das Themengebiet der Bachelorarbeit ein, z.B. durch Literaturrecherche, Evaluation bestehender Systeme oder eigener Implementationen. Projekte werden von allen Arbeitsgruppen des Fachbereichs angeboten.

**Lernziele:**

Absolventinnen und Absolventen sind in der Lage, unter Anleitung grundlegende wissenschaftliche Arbeit zu verrichten wie z.B.:

- Literaturrecherche und systematisches Lesen von Literatur
- Analyse und Vergleich von Forschungsansätzen und -ergebnissen
- Entwicklung eigener Forschungsansätze
- Projektmanagement
- Design und Implementation von neuen Methoden oder Systemen
- Evaluationen dieser Methoden und Systeme
- Schreiben wissenschaftlicher Arbeiten
- Präsentieren ihrer Forschung
- Beantworten von Fragen und Führen von Diskussionen über ihre Arbeit

**Voraussetzung:**

Vorlesungen des Pflichtbereiches, gegebenenfalls Vertiefungsvorlesungen im entsprechenden Bereich

**Arbeitsaufwand:**

270 Stunden

**Literatur:**

Wird im jeweiligen Projekt individuell ausgegeben.

**Assignment to semester:**

Bachelor's projects are offered every semester.

**Contents:**

The project prepares the students for the writing of their Bachelor's thesis. For this purpose the students familiarize themselves independently under guidance with the subject area of their Bachelor's thesis, e.g. through literature search and reading, evaluation of existing systems or own implementations. Projects are offered by all working groups of the faculty.

**Learning objectives:**

Graduates are able to perform basic scientific work under supervision. This may include:

- finding, organizing, and systematically reading relevant research papers
- analyzing, comparing, and contrasting research approaches and findings
- developing research approaches
- managing a comprehensive project
- designing and implementing a novel method or system
- conducting performance evaluations

- writing an academic paper
- presenting their research
- answering questions and discussing their work

**Prerequisites:**

Lectures of the basic studies, possibly courses from advanced studies in the respective area.

**Workload:**

270 hours

**Literature:**

Relevant literature will be distributed in the respective project.

## **5.6.4 Information Visualization II**

**Responsible for module:**

Prof. Dr. Keim (z.B. INF-20230-20151)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Advanced topics and problems in Information Visualization. The following topics are planned:

- Advanced visualization techniques for spatial data, geo-oriented data, multivariate data, trees, graphs, networks.
- Text and document visualization
- Comparison and evaluation of visualization techniques
- Visual Analytics
- Visualization toolkits
- Visualization systems and applications

**Learning objectives:**

The participants understand advanced problems and approaches in Information Visualization and visual data analysis, including selected current research problems. The participants learn, by means of group work, to implement a visualization approach and apply it to a real example data set and analysis problem.

**Prerequisites:**

The lectures Database Systems, Module Computer Science 1 and 2 are mandatory.

“Analysis and Visualization”, “Information Visualization I” or a comparable course on Information Visualization in another course of study is mandatory. Good programming skills and basic knowledge of databases and query languages are mandatory.

An initial test may be applied at the beginning of the course to verify that the necessary preconditions for the course are met.

**Workload:**

Total of 180 hours, 56 hours of lectures (2 SWS lecture + 2 SWS tutorial) and 124 hours of self study and practical assignments (development of an InfoVis technique)

**Credit requirements:**

Colloquium: The course Information Visualizaiton II is typically accompanied by a mandatory project assignment, which is to be worked on in teams. A colloquium about the project results will approve the projects' success. The colloquium will be graded.

Depending on the number of participants, oral exam (of 30 minutes duration), or written exam (of 60 minutes duration). The colloquium is a necessary precondition to take part in the course exam.

The final grade corresponds to the average of the result from the project assignment and from the exam.

**Literature:**

- Ward, M., Grinstein, G. and Keim, D. Interactive Data Visualization: Foundations, Techniques, and Applications, 2010 ISBN 9781568814735
- Ware, C. Information Visualization: Perception for Design, Morgan Kaufmann, 2nd edition, 2004
- Andrienko, N. and Andrienko, G. Exploratory Analysis of Spatial and Temporal Data, A Systematic Approach, Springer, 2006
- MacEachren, A. M., How Maps Work: Representation, Visualization, and Design, The Guilford Press, 2004
- Aigner, W., Miksch, S., Schumann, S. and Tominski, C., Visualization of Time-

Oriented Data, Human-Computer Interaction Series, Springer, 2011

## **5.6.5 Applied Visual Analytics**

**Responsible for module:**

Prof. Dr. Keim (z.B. INF-10795-20151)

**Credits:**

2 SWS, 4 ECTS

**Contents:**

- Applied Methods of Information Visualization
- Applied Methods of Data Mining
- Interactive integration of automatic and visual methods
- Application and adaption of recent research work to important application scenarios such as Exploration of Social Web data, or security-related applications (e.g., disease control, criminal investigations, and homeland security) .
- Processing and exploration of different data types like geo-coordinates, gene sequences, text documents, data from social networks and internet traffic, each with respect to temporal developments.
- Participation in an international challenge, e.g., IEEE VAST Challenge.

**Learning objectives:**

Contents and first of all methodical expertise in the area of Visual Analytics:

- Project planning: Development of strategies for the efficient and effective exploration of large data sources with respect to diffuse tasks and to generate hypotheses and recommendations involving uncertainty.
- From research to application: Systematic application of recent research work to real-world problems.
- Systematic evaluation of different Visual Analytics tools and methods with respect to their applicability to real-world problems.
- Presentation: Reporting of analysis results to decision makers as well as a thorough description of the analysis steps.
- Students have deep theoretical knowledge and are capable to evaluate advantages and drawbacks certain methods have in application.

**Prerequisites:**

Practical experiences in programming are expected requirements for successful participation. Additionally, previous knowledge about Information Visualization and Data Mining, or parallel participation in a respective lecture is encouraged.

**Workload:**

2 SWS lecture/seminar, in total 120 hours, split into 28 hours course of study with attendance and 92 hours of self-study time

**Credit requirements:**

The course requires students to prepare a solution for a given analysis and visualization task. The task may come from a contest in the area of analysis and visualisation, such as the IEEE VAST Challenge. The grade results from the final presentation and oral examination about the prepared solution.

**Literature:**

- James J. Thomas and Kristin A. Cook. Illuminating the Path: The Research and Development Agenda for Visual Analytics. National Visualization and Analytics Ctr, 2005.
- Matthew Ward, Georges Grinstein and Daniel A. Keim, Interactive Data Visualization: Foundations, Techniques, and Application, 2010, A.K. Peters, Ltd, ISBN: 978-1-56881-473-5, <http://www.idvbook.com>.
- Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann (Editors):

Mastering the Information Age - Solving Problems with Visual Analytics. E-Book  
verfügbar unter: <http://www.vismaster.eu/> IEEE

- VAST Challenge: <http://www.vacommunity.org/VAST+Challenge>

## **5.6.6 Geographic Information Systems (GIS)**

**Responsible for module:**

Prof. Dr. Keim (z.B. INF-10615-20131)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

- Introduction to Geographic Information Systems
- Abstract Data Models
- Mapping to Relational Data Model
- Spatial Index Structures
- Spatial Query Processing
- Algorithm of Computer Geometry
- Raster Data
- Combinations of Raster and Vector Data
- Spatial Data Mining
- Visual Analytics for Geospatial Data

**Learning objectives:**

The students will be able to deal with geospatial data sets and apply the specific methods for spatial indexing and computer geometry. Students will be enabled to understand the basic foundations of geographical information systems by introducing relevant concepts, application areas, data types and analysis operations. The students will afterwards be capable of dealing with complex GIS-scenarios by reducing them to sub-tasks introduced in the lecture. Students are able to analyze existing GIS systems with respect to their performance, and devise and implement GIS systems based on application demands.

**Prerequisites:**

It is necessary to have skills from the Information Visualization, Database Systems, and Data Mining courses in order to take part in this lecture. Furthermore, we require strong programming skills.

**Workload:**

Altogether 180 hours (90 hours lecture and 90 hours exercises), partitioned into 56 hours presence study and 124 hours own study

**Credit requirements:**

Depending on the number of participants an oral exam (20 minutes) or written exam (90 minutes) will be held; The grading consists of the project grading (one third) and the final examination (two thirds).

**Literature:**

We will provide a script based on lecture slides. Further literature will be announced at the beginning of the lecture and additionally for each lecture part.

Basic literature:

- Natalia Andrienko, Gennady Andrienko, *Exploratory analysis of spatial and temporal data: A systematic approach*, 2006, Springer
- J. Dykes, A.M. MacEachren, M.-J. Kraak, *Exploring geovisualization*, 2005, Elsevier
- Harvey J. Miller (Hrsg.), *Geographic data mining and knowledge discovery*, 2009, CRC Press
- Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind, *Geographic Information Systems & Science*, 2011, John Wiley & Sons, Inc.



## **5.6.7 Multimedia Database Systems**

**Responsible for module:**

Prof. Dr. Keim (z.B. INF-13160-20151)

**Credits:**

6 SWS, 9 ECTS

**Contents:**

- Basics of multimedia databases
- Similarity models for images
- Geometric similarity models
- 3D-Model-Retrieval
- Excursus for further multimedia data types and specialized applications, e.g., for audio data, time series data, or applications in biometry (e.g., recognition of fingerprints and faces)
- Algorithms for k-nearest-neighbor- and area-queries
- Index structures for high dimensional vector spaces
- Metrical index structures

**Learning objectives:**

Students will learn about several different multimedia data types, e.g., images, videos or 3D-objects, as well as about specialized methods for their content-based description. Students will be able to analyze existing methods with respect to their applicability. After taking the course, the participants will be able to apply, develop and refine effective algorithms for the processing of similarity search queries in multimedia databases. Furthermore, students will learn about specialized index structures for efficient query processing and will be able to evaluate their applicability in Multimedia Database applications.

**Prerequisites:**

Principles of Computer Science 1, Information Management, Information Systems.

**Workload:**

270 hours in total, 70 hours presence study, 200 hours self-study

**Credit requirements:**

Requirement: Active participation in the exercises. Active participation in programming project (several weeks) during the semester.

Exam: depending on the number of attendees of the lecture an oral exam (duration: 30 minutes) or a written exam (duration: 120 minutes) will take place. For the exam permission, 50% of the assignment points are needed.

Grade: Exam grade.

**Literature:**

The lecture is based on state-of-the-art conference and journal publications. For preparation the following literature is recommended:

- R ger: Multimedia Information Retrieval. Morgan and Claypool Publishers 2010.
- I. Schmitt:  hnlichkeitssuche in Multimedia-Datenbanken, Oldenbourg, 2006.
- Blanken, de Vries, Blok, Feng: Multimedia Retrieval. Springer 2007
- Volker Gaede, Oliver G nther: "Multidimensional Access Methods". In: ACM Computing Surveys, 30 (2): 170-231 (1998)
- Christian B hm, Stefan Berchtold, Daniel A. Keim: "Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases". In: ACM Computing Surveys, 33(3): 322-373 (2001)

Literature for the chapters of the course will be announced during the lecture.

## **5.6.8 Database System Architecture and Implementation**

**Responsible for module:** Jun.-Prof. Dr. Grossniklaus (z.B. INF-20210-20152)

**Credits:** 5 SWS, 7 ECTS

### **Contents:**

Modern database systems use advanced data structures and algorithms to manage and process data. Building on the basic database system course, this course looks at the internals of database systems in terms of their architecture and implementation. The lectures are structured according to the levels of a general layered software architecture for database systems, which is introduced at the beginning of the course. The course starts at the lowest level with the I/O system and the management of buffers and storage, i.e., disk and file management. Moving up a level of the architecture, the course then introduces different kinds of index structures, such as tree-based and hash-based indexes. Continuing from this groundwork, the course then focuses on query processing in terms of query execution and optimization. Finally, the course will teach how benchmarks can be used to qualitatively and quantitatively assess the performance of different database systems. Gathering and understanding results of database system benchmarks is the basis for selecting and tuning a database system for a given application. Throughout the course, students have a chance to get first-hand practical experience with the presented techniques and implementation concepts through a series of mandatory programming exercises.

### **Learning objectives:**

In this course, students will learn to understand and apply the data structures and algorithms that are used to realize the storage, indexing, and query processing functionality of contemporary relational database systems. In doing so, students will acquire the ability to differentiate between operating and database system functionality. Finally, the course imparts the knowledge required to assess the qualitative and quantitative differences between different relational database systems. Therefore, the course syllabus will enable students to make informed choices when selecting or tuning a database system for a given application.

### **Prerequisites:**

The following skills, knowledge and courses are mandatory prerequisites to attend and successfully complete this course.

- Basics of database systems: database design, query languages, database application programming, etc. (INF-12040 or equivalent)
- Principles of database systems: relational model, relational algebra, normal forms, etc. (INF-12040 or equivalent)
- Computer systems: computer architecture, operating systems, networks, etc. (INF-11740, INF-11880, or equivalent)
- System programming: students must have the ability to program in a language appropriate for system implementation, such as C/C++, C# or Java. (INF-11930 or equivalent)
- Key competences: Subversion, LaTeX, etc. (INF-10175 or equivalent)

### **Workload:**

210 hours, of which 75 hours are spent in class and another 135 hours are spent on exercises and project work.

### **Credit requirements:**

The final exam of this course will be conducted as a 90-minute written exam. The exam will be weighted 50% in the final grade of this course with exercises and project work contributing another 50% of the grade.

### **Literature:**

The following study materials are relevant for this course.

- Textbook: *Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems (3rd Edition)*, McGraw-Hill, 2002
- Course slides: Copies of the slide decks used in the course will be made available online
- Reading list: A bibliography of the references used in the course as well as further readings will be available on the course website

## **5.6.9 Data Stream Management Systems**

**Responsible for module:** Jun.-Prof. Dr. Grossniklaus (z.B. INF-20300-20151)

**Credits:** 4 SWS, 6 ECTS

### **Contents:**

Data Stream Management Systems (DSMS) process queries over continuous data, so-called data streams. Data streams are potentially infinite and the arrival rate and order are out of control of that data management system. Therefore, traditional Database Management Systems (DBMS) are ill-equipped to handle this type of data. Nevertheless, several interesting and important types of applications work on streaming data, for example to monitor network traffic, to manage traffic, or to analyze social media data in real-time. This course will cover a wide range of topics in the area of data stream management systems. First, the course will look at typical data stream applications and their requirements. Then, example data stream management systems will be studied in terms of their architecture as well as the query language and the optimization techniques that they support. Whether and how these systems support the processing of disordered data streams is an interesting question that will be examined in detail. Finally the course will provide insights into the performance analysis and benchmarking of data stream management systems.

### **Learning objectives:**

In this course, students will learn how the characteristics and requirements of data stream management differ from traditional data management. Starting from a general understanding of the architecture of data stream management systems, students will study the algorithms and data structures that these systems use to process data streams. By doing so, students will acquire the ability to differentiate between data stream management systems and other data processing approaches. Finally, the course also imparts the knowledge required to qualitatively and quantitatively distinguish between different types of data stream management systems. Therefore, the course syllabus will enable students to make informed choices when selecting or tuning such a system for a given application.

### **Prerequisites:**

The following skills, knowledge and courses are mandatory prerequisites to attend and successfully complete this course.

- Basics of database systems: database design, query languages, database application programming, etc. (INF-12040 or equivalent)
- Principles of database systems: relational model, relational algebra, normal forms, etc. (INF-12040 or equivalent)
- Database system architecture and implementation: data storage, memory management, query processing and optimization, etc. (INF-20210 or equivalent)
- Computer systems: computer architecture, operating systems, networks, etc. (INF-11740, INF-11880, or equivalent)
- System programming: students must have the ability to program in a language appropriate for system implementation, such as C/C++, C# or Java. (INF-11930 or equivalent)
- Key competences: Subversion, LaTeX, etc. (INF-10175 or equivalent)

### **Workload:**

180 hours, of which 56 hours are spent in class, 34 hours are spent on reading assignments, and another 90 hours are spent on the programming project.

### **Credit requirements:**

The final exam of this course will be conducted as a 90 minutes written exam. The exam will be weighted 50% in the final grade of this course with exercises and project work contributing another 50% of the grade.

**Literature:**

The following study materials are relevant for this course.

- Course slides: Copies of the slide decks used in the course will be made available online
- Reading list: Publications referenced in the course as well as further readings will be available on the course website

## **5.6.10 Data Warehousing und OLAP**

**Responsible for module:**

Prof. Dr. Scholl (z.B. INF-10365-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

The course deals with the development and operation of data warehouse systems, the differences of database systems (for OLTP: Online Transaction Processing), tools for online analytical processing (OLAP) and the Extract-Transform-Load (ETL) process from operational systems into a data warehouse. The lecture is structured roughly as follows:

- Introduction & Terminology
- Data Warehouse Architecture
- Multidimensional Data Model
- Modeling Methodologies
- Extraction, Transformation and Loading
- Data Processing and Transformation
- Index and Storage Structures
- Queries and OLAP
- Query Processing and Optimization
- Materialized Views

**Learning objectives:**

Graduates know the functions of Data Warehouse and OLAP-Systems and are able to differentiate them from Database systems (OLTP-). They know how to build complex data models for OLAP-Applications with state-of-the art techniques and to implement them on relational data warehouse systems. Furthermore they are able to perform ETL processes and write OLAP-Queries with graphical tools and SQL.

**Prerequisites:**

Basic knowledge of databases, as presented in the module Database systems of the Bachelor studies in Information Engineering or Computer Science; overall programming skills and knowledge of data structures and algorithms.

**Workload:**

180 hours, 60 hours in class and 120 hours self-study

**Credit requirements:**

There will be an oral exam. The practical assignments can lead to a bonus for the exam.

**Literature:**

The lecture is oriented at the book

- V. Köppen, G. Saake, K.-U. Sattler. Data Warehouse Technologien. mitp-Verlag, Heidelberg, 2012

More literature will be presented in the lecture.

## **5.6.11 XML Technologies**

**Responsible for module:**

Prof. Dr. Scholl (z.B. INF-12630-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Modern applications build on XML technology require a thorough understanding of a family of W3C standardized technologies. An insight on how database systems and query processors deal with semi-structured data and how they differ from the relational approach is also mandatory. In this course you will . . .

- get to know XML technologies and languages, such as XPath, XQuery, and XSLT
- gain insight into XML databases, XQuery processors and experience how they differ from relational databases
- learn how markup and query languages are parsed, compiled and evaluated

Throughout the course, students have a chance to get first-hand practical experience with the presented techniques and implementation concepts through a series of programming exercises.

**Learning objectives:**

In this course, students will learn to apply XML technologies to a set of data management problems. They will gain an insight on how modern database systems store, index, and query semi-structured data and learn about current approaches and techniques in the particular field of semi-structured database development. By practically applying W3C technologies and studying internal system architecture students will acquire the ability to differentiate between relational, semi-structured and unstructured data management problems and to choose apt systems for different data scenarios.

**Prerequisites:**

The following lectures may be helpful to successfully complete this course:

- Basics of database systems: database design, query languages, database application programming, etc. (INF-12040 or equivalent)
- System architecture of database systems: index structures, query processing, etc. (INF-20210 or equivalent)
- Computer systems: computer architecture, operating systems, networks, etc. (INF-11740, INF-11880, or equivalent)
- System programming: students must have the ability to program in a language appropriate for system implementation, such as C/C++, C# or Java. (INF-11930 or equivalent)
- Key competences: Subversion, LaTeX, etc. (INF-10175 or equivalent)

**Workload:**

180 hours, of which approx. 60 hours are spent in class and another 120 hours are spent on exercises and self-study.

**Credit requirements:**

The final exam of this course will be conducted as a 90-minute written exam. The exam will be weighted 100% in the final grade of this course. Exercises and project work serve as training for the final exam.

**Literature:**

The following study materials are relevant for this course:

Course slides: Copies of the slide decks used in the course will be made available online

Reading list: A bibliography of the references used in the course as well as further readings will be made available.

A preliminary list includes:

- Priscilla Walmsley; XQuery
- Michael Brundage; XQuery: The XML Query Language
- Erik Siegel, Adam Retter; eXist - A NoSQL Document Database and Application Platform
- Howard Katz, Donald Dean Chamberlin; XQuery from the Experts: A Guide to the W3C XML Query Language
- XQuery Portal (<http://docs.basex.org/wiki/XQuery>)
- Selected W3C technical specifications (XQuery, XQuery Full-Text, ...)



## **5.6.12 Transactional Information Systems**

**Responsible for module:**

Prof. Dr. Scholl (z.B. INF-20200-20142)

**Credits:**

5 SWS, 8 ECTS

**Contents:**

Transaction processing is a key technology for centralized and distributed multi-user systems. It guarantees the atomic, persistent and isolated execution of parallel user processes. Such generic system services result in dramatic simplifications of software development of fault-tolerant, multi-user applications that depend on consistent and reliable data management. Thus, transactions are one of the main cornerstones of database management systems, that have also found entrance into operating systems and programming language technology as well as internet, e-commerce, or workflow management applications.

This course discusses theoretical foundations of as well as implementation techniques for high-performance concurrency control and recovery mechanisms that shield applications from transaction, system and media faults in centralized and distributed TP systems.

**Learning objectives:**

Successful student will be able to distinguish several forms of transactional processing and assess their strengths and weaknesses/limitations. They will be able to design and conduct tests and comparisons of system functionality and performance of TP systems and select technologies for specific application scenarios.

**Prerequisites:**

Basic understanding of the workings of complex application systems as well as on data structures & algorithms, principles of computer science. It is recommended (even though not strictly required) to have already passed the Introduction to Database Systems course (e.g., in the Bachelor programmes of our department).

**Workload:**

240 hours, about 75h in classroom and 165h for individual learning.

**Credit requirements:**

Aural or written exam (depending on number of participants)

**Literature:**

Copies of presentation slides will be given to students. The course builds upon the book "Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery" by Gerhard Weikum and Gottfried Vossen, Morgan Kaufmann Publishers, 2001.

Additional material:

- P. A. Bernstein, V. Hadzilacos, N. Goodman: Concurrency Control and Recovery in Database Systems, Addison-Wesley Publishing Company, 1987.
- A. K. Elmagarmid: Database Transaction Models for Advanced Applications, Morgan Kaufmann Publishers, 1992.
- J. Gray, A. Reuter: Transaction Processing: Concepts and Technologies, Morgan Kaufmann Publishers, 1993.
- V. Kumar, M. Hsu: Recovery Mechanisms in Database Systems, Prentice Hall, 1998.

### **5.6.13 Usability Engineering: Design**

**Responsible for module:**

Prof. Dr. Reiterer (z.B. INF-10960-20151)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

This course is about systematic approaches for developing user-centered interactive products (eg, application software, websites, information appliances, etc.). The course thereby focuses on the comparison and the practical application of various popular process models and methods within user-centered design. In addition to these theoretical approaches, the practical part includes the development of innovative product concepts and designs.

The contents of the course are primarily based on four popular models of the usability engineering approach (see literature) in addition to an exploratory design phase.

During the course, the basic concepts and methods of these process models will be taught. Similarities and differences between the process models and techniques will be compared by identifying individual advantages and disadvantages.

The lecture thereby covers the following subjects:

- General process models and practices in usability engineering
- Creative techniques for the exploration of design problems and design solutions (Affinity Diagramming, Sketching)
- Methods for determining the needs and requirements from the context of use (eg User Roles, Task Mnalysis)
- Techniques for the preparation of design studies (eg, conceptual and physical interaction design, Prototyping)

These theoretical foundations are applied in practice for developing ideas and concepts as well as attractive design studies. This practical part will be conducted in groups. Results are presented regularly to the other students for discussion and evaluation.

**Learning objectives:**

- Acquisition and analysis of needs and requirements derived from the context of use
- Design of conceptual models and design alternatives (Modeling & Sketching)
- Development of interactive design studies (Mockups & Prototypes)

**Prerequisites:**

The contents of the course "Computer graphics and interactive systems (Computergrafik und Interaktive Systeme)" should be known.

**Workload:**

A total of 180 hours

- 56 hours compulsory attendance
- 124 hours home work

**Credit requirements:**

- Written midterm examination on the contents of the theoretical part. (Duration: one hour)
- Conduction of a user-centered design process in the group by following a specific process model. Documentation of all steps and intermediate results. Regular presentation of these results using diagrams, sketches and interactive prototypes. Presentation dates: mid to end of the semester. Attendance at these meetings is mandatory for all students!
- An online documentation (in BSCW) of the group's project on the basis of presentation slides, artifacts, models, style guides etc. is required. A comprehensible

documentation of the group's process and design decisions must be delivered in a separate document (eg, slides).

**Literature:**

As an introduction to the topic we recommend:

- Preece J., Rogers Y., Sharp H.: Interaction Design, John Wiley & Sons, 2002

The main contents of the course are based on the following textbooks:

- Buxton B., Sketching User Experiences: Getting the Design Right and the Right Design, Morgan Kaufmann, 2007
- Constantine L., Lockwood L., Software for Use, Addison-Wesley, 1999
- Mayhew D., The Usability Engineering Lifecycle, Morgan Kaufmann, 1998
- Rosson M., Carroll J., Usability Engineering - Scenario-based Development of Human-Computer Interaction, Morgan Kaufmann, 2002
- Beyer H., Holtzblatt K., Contextual Design, Morgan Kaufmann, 1998

## **5.6.14 Usability Engineering: Evaluation**

**Responsible for module:**

Prof. Dr. Reiterer (z.B. INF-10025-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Evaluation is an integral part of Usability Engineering. It serves the purpose to recognize usability problems early in the development phase of interactive products and develop ideas for improvement. There is a broad spectrum of techniques and methods available (e.g. observation, usability tests, surveys, etc.), which differ in many terms, such as when to apply during the development process or whether to include end-users or experts and thereby provide results for different purposes.

During the theoretical part of the lecture, students will learn about the different methods and techniques. This includes the design and conduction of surveys, interviews, focus groups, usability tests, eye-tracking, diary method, observation, etc. During the course, students will also learn the basic principles of controlled experiments, such as the formulation of hypotheses, dependent and independent variables, within- or between-subjects designs, statistical analysis of data, etc.

During the practical part of the lecture, students will work on a small project in groups. They will have to conduct a study and apply the learned methods to evaluate an interactive product. This allows them to gain first hands-on experiences and also use our usability lab.

**Learning objectives:**

The learning objectives of the lecture is to provide the knowledge in terms of the use of methods and techniques for the following activities:

- Evaluation of interactive products with regards to usability and user experience
- Deduction of change requirements and re-design recommendations

**Prerequisites:**

Students should be familiar with the contents of the course "Computer Graphics an Interactive Systems".

**Workload:**

A total of 180 hours

- 56 hours compulsory attendance
- 124 hours home work

**Credit requirements:**

The proof of academic achievement consists of the following parts:

- written exam about the theoretical part of the lecture (1 hour)
- the participation in practical project and a rating of project milestones by the lecturer
- presentation of practical project
- term-paper in extended format of the presentation (e.g. Powerpoint)

**Literature:**

Main Textbooks:

- Lazar, Jonathan; Feng, Jinjuan Heidi; Hochheiser, Harry (2010): *Research Methods in Human-Computer Interaction*. Chichester, UK: John Wiley & Sons Ltd.
- Hartson, Rex; Pyla, Pardha S. (2012): *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Waltham, US: Morgan Kaufmann. Chapters 12-18.
- Rogers, Yvonne; Sharp, Helen; Preece, Jenny (2012): *Interaction Design: Beyond*

*Human-Computer Interaction*. Chichester, UK: John Wiley & Sons Ltd. Chapters 12-15.

## **5.6.15 New Directions in Human-Computer Interaction: Blended Interaction**

**Responsible for module:** Prof. Dr. Reiterer (z.B. INF-11840-20151)

**Credits:** 6 SWS, 9 ECTS

### **Contents:**

The course will consist of a theoretical and practical part that run in parallel throughout the semester: The theoretical part will consist of 4 hours of lectures & exercises per week that introduce the theories, conceptual foundations, models, principles, and hardware & software technologies for Blended Interaction. The practical part will be taken out in teams of two participants. Each team will be assigned a project in which the knowledge gained from the theoretical part has to be applied on the design and implementation of a prototype. Each week, every team will present its progress as part of 2 hours of project presentations in front of the entire course. This is also the chance for the participants to get recommendations and advice from the lecturers. At the end of the course, the system must fulfill the specified task from the assignment and must be usable as an interactive prototype written in HTML5/CSS3/JavaScript.

### **Learning objectives:**

- Which theories and conceptual frameworks from cognitive science, psychology, and the social sciences can help us to understand and predict the effects of Blended Interaction on the users?
- How to combine stationary displays, physical objects, and mobile devices to create blended work environments?
- How to use large interactive tabletops, large displays, or multiple mobile displays to foster efficient collaboration between multiple users?
- How to combine new modalities (e.g., body or touch gestures, pen or speech input) to afford "natural" interaction? And what does "natural" or "intuitive" interaction actually mean?
- What kind of programming models and languages are useful to support concurrent user activities at multiple points of actions (e.g., simultaneous input from multiple fingers, devices, gestures, physical objects, voices, or computer vision)?

### **Prerequisites:**

To participate in the course, previous participation in the course "Interactive Systems" or equivalent experience from other courses or universities is recommended. Previous experience in programming graphical user interfaces or input event processing in C#/WPF, Java, or HTML5/JavaScript/CSS3 is a must.

### **Workload:**

A total of 270 hours

- 84 hours compulsory attendance
- 186 hours home work

### **Credit requirements:**

- Participation: Frequent presence and active participation in theoretical lectures (VL), practical exercises (Ü), and project presentations (P). Presence during all team presentations and final presentation is mandatory! Participants are requested to sign in every lecture, exercise, and project presentation. Absence can result in exclusion from this course.
- Written Exam: Participants must pass a written exam. The exam will be 1 hour and is based on multiple choice questions or alike.
- Presentations: For the practical part, participants are grouped into teams. Each team has:

- 1 intermediate presentation of their current state of the project. The presentation must include design rationals and a competitive comparative analysis.
- 1 final presentation of the result of the practical project. The presentation must include a brief introduction into design rationals, presentation of their interactive prototype and implementation details, a report on the encountered problems and their solution and lessons learned.

All presentations must discuss related work from scientific journals, books, and conference papers and related designs of commercial products.

- Final Grade: The final grade is calculated from the grades for the written exam (40%), the quality of the intermediate presentation (20%), and the quality of the final presentation and interactive prototype (40%). Each part of the final grade needs to be passed with 4.0 or better in order to pass the entire course.

### **Literature:**

- Jetter, H.-C., Reiterer, H., Geyer, F. (2013). Blended Interaction: Understanding Natural Human-Computer Interaction in Post-WIMP Interactive Spaces. Personal and Ubiquitous Computing, Oct 2013, DOI=10.1007/s00779-013-0725-4, Springer-Verlag [http://hci.uni-konstanz.de/downloads/jetter-et-al\\_final.pdf](http://hci.uni-konstanz.de/downloads/jetter-et-al_final.pdf)
- Reiterer, H. (2014). Blended Interaction - Ein neues Interaktionsparadigma. Informatik-Spektrum, Springer Verlag Berlin Heidelberg, Jun 2014, DOI=10.1007/s00287-014-0821-5 (The final publication is available at <http://link.springer.com/article/10.1007/s00287-014-0821-5>), PDF=<http://hci.uni-konstanz.de/downloads/BlendedInteraction.pdf>
- Imaz, M., Benyon D. (2007). Designing with Blends - Conceptual Foundations of Human-Computer Interaction and Software Engineering. MIT Press. [recommended are chapters 1 to 4]

## **5.6.16 Visual Information Seeking Systems**

**Responsible for module:**

Prof. Dr. Reiterer (z.B. INF-10190-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Today's information spaces such as digital libraries, local file systems, data warehouses, Wikipedia or business data of a company continuously increase in size and complexity. Whereas enhanced search- and indexing techniques allow the access via conventional search queries, user interfaces that are used to search, browse and visualize results are often left unattended. In many cases, user interfaces are still available that were used for decades without being redesigned.

The lecture "Visual Information Seeking Systems" discloses how the application of interactive visualizations may help users to essentially enhance their information seeking process in terms of efficiency, effectiveness but also joy of use. The students will learn how to support users of information systems in the formulation of information needs through the use of appropriate visualizations and elaborate presentation techniques for results. By means of innovative user interfaces, such as animated zoomable user interfaces (ZUI) or new in- and output devices such as multi touch tabletop displays, traditional constraints often become blurred in terms of query formulation, result presentation and navigation.

**Learning objectives:**

- The students will learn the potentials but also the constraints of interactive visualizations to support information seeking in digital information spaces (e.g. WWW, digital libraries, online databases).
- Furthermore, different interaction concepts (e.g. ZUI) and their application on different devices (e.g. smart phones, desktop PCs, multi touch tabletop displays) will be disclosed.
- The students will achieve a deep understanding of theoretical and conceptual issues in conjunction with the application of interactive visualizations and the embedding of appropriate interaction techniques.
- The students will be able to independently design and implement an interactive visualization that inherits a high usability.

**Prerequisites:**

Basic knowledge in Human-Computer Interaction and object-oriented programming skills.

**Workload:**

A total of 180 hours, 56 hours compulsory attendance and 124 hours home work

**Credit requirements:**

The proof of academic achievement consists of the following parts:

- written exam about the theoretical part of the lecture (1 hour, mid-term!)
- the participation in practical project and a rating of project milestones by the lecturer
- presentation of practical project
- term-paper in extended format of the presentation (e.g. Powerpoint)

**Literature:**

- Card, Stuart K.: Information Visualization. In: Jacko Julie A.; Sears Andrew (Eds.): The Human-Computer Interaction Handbook. Mahwah, NJ (Lawrence Erlbaum) 2003, pp.544-582.
- Bederson B. B., Shneiderman B. (Hrsg.): The Craft of Information Visualization. Readings and Reflections. San Francisco, CA, Morgan Kaufmann, 2003



- Card, Stuart K.; Mackinlay, Jock D.; Shneiderman, Ben (Hrsg.): Readings in Information Visualization. Using Vision to Think. San Francisco, CA (Morgan Kaufmann Publishers, Inc.) 1999
- Robert Spence; Information Visualization, Addison-Wesley, 2001
- Reiterer, H.: Visuelle Exploration digitaler Datenbestände. In: Maximilian Eibl; Harald Reiterer; Peter F. Stephan; Frank Thissen (Eds.): Knowledge Media Design. Theorie, Methodik, Praxis., Oldenbourg, München, 2005, siehe: [http://hci.uni-konstanz.de/downloads/Reiterer\\_VisuelleExplorationDigitalerDatenbestaende.pdf](http://hci.uni-konstanz.de/downloads/Reiterer_VisuelleExplorationDigitalerDatenbestaende.pdf)
- Marti Hearst; User Interfaces and Visualization. In: Ricardo Baeza-Yates, Berthir Ribeiro-Neto; Modern Information Retrieval, acm Press, 1999
- Gerken, Jens; Heilig, Mathias; Jetter, Hans-Christian; Rexhausen, Sebastian; Demarmels, Mischa; König, Werner A.; Reiterer, Harald: Lessons Learned from the Design and Evaluation of Visual Information Seeking Systems. In: Adam, Nabil; Furuta, Richard; Neuhold, Erich, Springer, International Journal on Digital Libraries, p. 49–66, Aug 2009

### **5.6.17 Virtual and Augmented Reality**

**Responsible for module:**

Prof. Dr. Deussen (z.B. INF-20410-20151)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

The course gives an introduction to virtual and augmented reality systems, their technology and programming. We will start with defining such systems, will go through the various systems for simulation, presentation and information, will speak about concepts for technology and what is needed to produce the illusion of reality. We will discuss augmented reality, its application in information technology. In the practical course work we will program a virtual reality system using the Oculus development system, we will construct VR scenes using Scenegraps and other programming techniques.

A second part of the course deals with the question of what reality is. We will discuss findings from psychology (perception), philosophy, sociology and other scientific fields to understand how reality is formed in our brain. We will see that it is absolutely not trivial to answer this question and we will discuss how modern VR and AR technology influences our perception of reality.

**Learning objectives:**

Successful participants of the course are able to analyze and evaluate VR and AR systems. They have advanced programming skills to create environments that construct VR and AR scenes using OpenGL, Scenegraps and related programming environments. They have a deep understanding of technology and problems of VR systems. They are able to discuss the formation of reality in a broader perspective.

**Workload:**

180 hours (56 Stunden course, 124 Stunden homework)

**Credit requirements:**

oral examination (20 min) or written test (90min) will be announced at the beginning of the course

**Literature:**

will be given out in the course

## **5.6.18 Illustrative Computer Graphics**

**Responsible for module:**

Prof. Dr. Deussen (z.B. INF-20780-20161)

**Credits:**

6 SWS, 6 ECTS

**Contents:**

- 2D-Methods for the production of non-photorealistic Computer Graphics (Dithering, Halftoning, artistic screening, Stipping)
- Manipulation of 2.5-dimensional Data (Images with depth), edge enhancement, depth differences, Unsharp Masking
- Non-photorealistic rendering of 3D Geometry data (Computation of geometric Features, Minima and Maxima of derivations and curvatures, finding silhouettes, numerical problems)

**Learning objectives:**

Understanding, analyzing and evaluating of algorithms of Non-Photorealistic Computer Graphics. The course focusses on 2D-, 2.5D- and 3D methods to create images aside from standard photo realistic techniques. Participants are able to implement, evaluate and analyze artistic abstraction methods in computer graphics systems. They learn to implement them in Java using the Processing framework.

**Prerequisites:**

experience similar to modules Mathematical Foundations for Computer Science, Informatics I and Computer Graphics and interactive Systems (or similar), elementary programming skills.

**Workload:**

total of 180 hours, consisting of 56 hours compulsory attendance and 124 hours home work

**Credit requirements:**

Course achievement: 50% of the points of the practical exercises

Examination: dependent on the number of participants oral examination (20min) or written examination (60min)

Grade: Result of the examination

**Literature:**

will be given in the course

### **5.6.19 Modelling in Computer Graphics**

**Responsible for module:**

Prof. Dr. Deussen (z.B. INF-12740-20151)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

This course teaches methods for data modelling in computer graphics. Based on a comparison of volume- and surface models, their creation, properties and rendering methods we discuss complex models and their creation. Data driven modeling from scanned data is another topic.

- a comparison of geometric representations
- curves and surfaces, parametrization
- fractals
- botanic descriptions and computer generated plants
- modelling of virtual eco systems
- data-driven modelling (reconstruction from scanned point clouds)
- level-of-detail methods

**Learning objectives:**

Successful participants of the course have an enhanced understanding of computer graphics modeling and are able to create and maintain complex scenes. They have a deep understanding of raster graphics and OpenGL and are able to estimate the developmental complexity of computer graphics.

**Prerequisites:**

Basics on computer graphics, programming skills

**Workload:**

180 hours (56 hours course, 124 hours homework)

**Credit requirements:**

oral examination (20 min) or written test (90min) will be announced at the beginning of the course

**Literature:**

will be announced in the course

## **5.6.20 Image Analysis and Computer Vision**

**Responsible for module:**

Prof. Dr. Goldlücke (z.B. INF-20160-20152)

**Credits:**

6 SWS, 9 ECTS

**Contents:**

Basic topics and methodology in image analysis and computer vision:

- Image filtering, frequency and scale
- Image features
- Coordinate transformations and image alignment
- Camera models and calibration
- Multiple views and motion
- Reconstruction of 3D scene geometry

See course web pages at <http://www.informatik.uni-konstanz.de/cvia/students/teaching/> for more details. If you have previous knowledge about image analysis or machine learning and/or would like to specialize in the field, the seminar "Deep Learning in Computer Vision" is recommended to be attended in parallel.

**Learning objectives:**

- Obtain a broad overview of the (considerable) range of topics in computer vision
- Understand basic challenges and ideas for solutions
- Become familiar with implementing image analysis and computer vision techniques in MATLAB

**Prerequisites:**

Basic mathematical background will be assumed mostly in the following fields, less familiar topics will be covered as necessary.

- Multivariable Calculus
- Linear Algebra

Basic knowledge of MATLAB can be helpful, but will be introduced in exercises.

**Workload:**

270 hours, consisting of 90 hours in class and 180 hours self-study

**Credit requirements:**

Exercises and programming projects (MATLAB), final exam.

**Literature:**

- R. Szeliski „Computer Vision: Algorithms and Applications“
- R. Hartley and A. Zisserman, „Multiple View Geometry in Computer Vision“
- R. Gonzales and R. Woods, „Digital Image Processing“

## **5.6.21 Introduction to Mathematical Image Analysis**

**Responsible for module:**

Prof. Dr. Goldlücke (z.B. INF-20170-20142)

**Credits:**

3 SWS, 6 ECTS

**Contents:**

Introduction to inverse problems and mathematical modeling in image analysis, intended to make students familiar with modern techniques in continuous optimization in this area. Good preparation for a Master's or Bachelor's thesis. Can be attended complementary to the lecture "Image Analysis and Computer Vision", but is completely independent. Planned content:

1) Variational methods crash course:

- Continuous modeling and energy minimization frameworks
- Euler-Lagrange equations
- Descent methods, parallel implementations

2) Convex optimization crash course:

- Introduction to convex analysis
- Proximal gradient methods
- Chambolle-Pock saddle point solver and parallel implementation
- Convex regularizers: total variation and variants
- Calibration method and convex relaxation

3) Applications in image analysis

- Inverse problems: denoising, deconvolution, inpainting, zooming, super-resolution  
...
- Geometric problems: segmentation and labeling

Programming examples will be shown in MATLAB, depending on time and interest maybe a mini introduction to GPU programming and CUDA will be given. See also detailed course web page on <http://www.informatik.uni-konstanz.de/cvia/students/teaching/>

**Learning objectives:**

- Familiarity with variational problem formulations in image analysis
- Theory and implementation of efficient convex optimization methods

**Prerequisites:**

This is an advanced course intended to get to current research topics within only a short amount of time. Thus, solid familiarity with mathematical basics must be assumed, in particular multi-variable calculus (gradient, divergence, integrals on nD domains) and linear algebra. If you do not attend the lecture "Computer Vision and Image Analysis" in parallel, some previous knowledge about image processing and Matlab is helpful, but not required - everything I need will be quickly introduced.

**Workload:**

180 hours, consisting of 45 hours in class and 135 hours self-study.

**Credit requirements:**

Participation in exercise groups, oral or written exam depending on number of participants

**Literature:**

- Scherzer et al. „Variational Methods in Imaging“
- Boyd and Vandenberghe „Convex Optimization“
- Bredies and Lorenz „Mathematische Bildverarbeitung“

## **5.6.22 Digital Signal Processing**

**Responsible for module:**

Prof. Dr. Saupe (z.B. INF-10150-20152)

**Credits:**

6 SWS, 9 ECTS

**Contents:**

This introductory course on digital signal processing covers

- discrete-time signals and systems
- FIR and IIR filters
- discrete-time Fourier analysis
- sampling and resampling of signals
- Fourier analysis using the fast Fourier transform

**Learning objectives:**

The student

- masters the basic methods in digital signal processing
- has an in-depth understanding of the concept and methods of continuous-time and discrete-time Fourier transformations
- is enabled to analyse time-discrete signals and systems
- has experience in the practice of digital signal processing using MATLAB or other programming languages

**Prerequisites:**

None, if taken as a master course. If taken as an advanced course in the bachelor program:

- basic math courses offered in our bachelor programs
- algorithms and data structures
- introduction to computer science including programming

**Workload:**

- ECTS: 9 points.
- SWS: 4h lecture time, 2h practice session.
- Total: 270h, including 78h in class.

**Credit requirements:**

The credit requirements include solving homework problems and a written exam.

**Literature:**

- Alan Oppenheimer, Ronald Schafer, John Buck, Discrete-time Signal Processing, Prentice-Hall, 2010.
- Alan Oppenheimer, Ronald Schafer, John Buck, Zeitdiskrete Signalverarbeitung, 2. Auflage, Pearson Studium, 2004.
- Vinay Ingke, John Proakis, Digital Signal Processing using MATLAB, Third Edition, Cengage Learning, 2012.

### **5.6.23 Numerical Continuation Methods**

**Responsible for module:**

Prof. Dr. Saupe (z.B. INF-20600-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

This course covers:

- Basic principles of continuation methods
- Predictor corrector methods
- Simplicial continuation methods
- Update methods

**Learning objectives:**

The student

- masters the basic methods continuation of solutions to nonlinear systems
- has experience in the practice of continuation methods
- is familiar with programming techniques for scientific computing

**Prerequisites:**

Working knowledge in Analysis (I) and Linear Algebra (I).

**Workload:**

- ECTS: 6 points.
- SWS: 2h lecture time, 2h practice session.
- Total: 180h, including 28h in class.

**Credit requirements:**

Solving homework problems including programming and a written exam.

**Literature:**

- Allgower, Eugene L., and Kurt Georg. Introduction to numerical continuation methods. Vol. 45. SIAM, 2003.
- Gomes, Abel, et al. Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms. Springer Science & Business Media, 2009.
- Seydel, Rüdiger. Practical bifurcation and stability analysis. Vol. 5. Springer Science & Business Media, 2009.
- Seydel, Rüdiger. From equilibrium to chaos. Elsevier, 1988.



## **5.6.24 Eye Tracking: Theory and Practice**

**Responsible for module:**

Prof. Dr. Saupe (z.B. INF-13450-20162)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

This one-semester course provides an introduction to Eye Tracking in the form of a one part theoretical lecture and one part practical lab session.

- Human Visual System
- Eye Tracking Systems
- Eye Tracking Methodology
- Eye Tracking Applications

**Learning objectives:**

Following this course, students will

- learn to set up and handle eye-tracking hardware equipment
- implement simple eye-tracking studies
- analyze the generated eye-tracking data independently
- learn about the principles of eye-tracking, on physiological, theoretical, software and hardware levels.

**Prerequisites:**

None, if taken as a master course. If taken as an advanced course in the bachelor program:

- basic math courses offered in our bachelor programs
- algorithms and data structures
- introduction to computer science including programming

**Workload:**

- ECTS: 6
- SWS: 4
- Total: 180 hours

**Credit requirements:**

Credit will be given based on a written (midterm) exam and a presentation of an experimental study using our eye tracking system. The presentation should include the theoretical foundations, the design of the experiment, the data analysis, and the visualization of the results. In addition a written report containing this material must be provided.

**Literature:**

- Kenneth Holmqvist, Marcus Nystrom, Eye Tracking: A Comprehensive Guide to Methods and Measures, Oxford, 2012.
- Andrew Duchowski, Eye Tracking Methodology: Theory and Practice, second Edition, Springer, 2007.

## **5.6.25 Design and Analysis of Algorithms**

**Responsible for module:**

Prof. Dr. Brandes (z.B. INF-10160-20162)

**Credits:**

6 SWS, 9 ECTS

**Contents:**

Design and analysis of algorithms are fundamental to computer science. In this course, we will have a look at selected tasks and methods in algorithmics, and thereby lay the ground for dealing with algorithmic problems occurring in more specific domains. While exemplary problems arise from various domains, an emphasis is put on graph algorithms.

**Learning objectives:**

Knowledge of advanced algorithms and data structures. Command of advanced means to evaluate and compare their performance and suitability.

**Prerequisites:**

Basic knowledge about algorithms and data structures.

**Workload:**

270 hours: approximately 90 hours in course and 180 hours self-study

**Credit requirements:**

Graded weekly assignments (50% threshold), active participation in the tutorial, and oral examination

**Literature:**

- T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein: Introduction to Algorithms. McGraw-Hill, 2001 (2nd ed.)
- J. Kleinberg, E. Tardos: Algorithm Design. Addison-Wesley, 2006
- M.T. Goodrich, R. Tamassia: Algorithm Design. Wiley, 2002
- T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen. BI-Wissenschaftsverlag, 1993
- D. Jungnickel: Graphen, Netzwerke und Algorithmen. BI-Wissenschaftsverlag, 1994

## **5.6.26 Network Analysis**

**Responsible for module:**

Prof. Dr. Brandes (z.B. INF-22405-20141)

**Credits:**

6 SWS, 9 ECTS

**Contents:**

Networks have become omnipresent. Besides physical networks as, e.g., in electrical engineering or transportation systems, abstract networks such as the structure of the WWW or constellations of political actors are increasingly analysed. Due to the variety of applications and resulting research questions, a wide range of methods is applied leading to interesting coherences between graph theory, linear algebra and probabilistic methods. In this course, some of the used methods and their theoretical foundations are presented. While research questions will be motivated by examples of application, the focus, however, will be on algorithmic approaches to these problems as well as their assumptions and characteristics. Moreover, we will come across a number of open questions that invite to further work on this topic.

**Learning objectives:**

Understanding of foundations and role of formal network analysis. Knowledge of basic and advanced combinatorial techniques to analyze networks. Capability to assess suitability of techniques and to interpret results.

**Prerequisites:**

Basic knowledge in algorithms and discrete mathematics.

**Workload:**

270 hours (about 90 hours in class and 180 hours self-study/homework)

**Credit requirements:**

Oral exam (30min) or - in case of unusually high attendance - written exam (120min).

Successful completion of weekly assignments is required to be admitted for examination.

**Literature:**

- Hennig, Brandes, Pfeffer, Mergel: Studying Social Networks. Campus, 2012.
- Brandes, Erlebach (Hrsg.): Network Analysis. LNCS 3418, Springer, 2005.
- Diestel: Graphentheorie. Springer, 1996.
- Kolaczyk: Statistical Analysis of Network Data. Springer, 2009.
- Steger: Diskrete Strukturen. Springer, 2001.
- Wasserman, Faust: Social Network Analysis. Cambridge Univ. Press, 1994.

## **5.6.27 Graph Drawing**

**Modulverantwortliche:**

Dr. Pampel (z.B. INF-13150-20151)

**Credits:**

6 SWS, 9 ECTS

**Contents:**

Layout algorithms for graphs form the basis of effective and efficient visualization of networks. Thus, automatic graph drawing is important in core areas of computer science like data bases, software engineering, VLSI and network design, and visual user interfaces. It is also relevant in other application areas, for example, in engineering, chemistry and biology, and social or political science. Various algorithmic methods for automatic graph drawing will be treated, for instance, force-directed or flow-based methods.

**Learning objectives:**

Students know and understand effective and efficient methods for visualizing graphs/networks.

**Prerequisites:**

Algorithms and Datastructures

**Workload:**

in class: approx. 80h, self-study: approx. 190h

**Credit requirements:**

- Weekly homework assignments (50% of points required for examination)
- Oral examination

**Literature:**

- G. Di Battista, P. Eades, I.G. Tollis, R. Tamassia: "Graph Drawing: Algorithms for the Visualization of Graphs". Prentice Hall, 1999
- M. Jünger, P. Mutzel (Eds.): "Graph Drawing Software" Mathematics and Visualization Series, Springer-Verlag, 2003
- M. Kaufmann, D. Wagner (Eds.): "Drawing Graphs – Methods and Models". Lecture Notes in Computer Science Tutorial 2025, Springer-Verlag 2001
- K. Sugiyama: "Graph Drawing and Applications for Software and Knowledge Engineers". World Scientific, 2002

## **5.6.28 Big Data and Scripting**

**Responsible for module:**

Dr. Pampel (z.B. INF-12810-20171)

**Credits:**

6 SWS, 8 ECTS

**Contents:**

The term “big data” is often used to describe vast collections of semi-structured data in the range of tera- or even petabytes. Companies like Google and Amazon illustrate that mining and analyzing such collections yields the potential for completely new applications. The lecture provides an overview of motivations to analyze big data and introduces techniques needed in the process. This includes introductions to scripting languages, NOSQL databases and Map/Reduce systems which are accompanied by practical exercises.

Content overview:

- Streaming Algorithms
- Memory Hierarchies
- Parallel Computations
- Storage Area Networks and Distributed File Systems

For implementations the students will learn and use the language Python.

**Learning objectives:**

The students know and understand the basic concepts for dealing with very large data sets and are able to apply them in small projects.

**Prerequisites:**

Module 1: Principles of Computer Science and Programming Course 1

**Workload:**

240 hours

**Credit requirements:**

Students have to pass 50% of the weekly theoretical and practical assignments and a written exam at the end of the semester.

## **5.6.29 Introduction to Data Analysis**

**Responsible for module:**

Prof. Dr. Berthold (z.B. INF-20740-20152)

**Credits:**

4 SWS, 8 ECTS

**Contents:**

The lecture series provides an introduction to Data Mining Methods with an emphasis placed on basic approaches and how they are incorporated into different problem definitions.

- Data Mining: problem definition, motivation, application examples
- Modelling: data-driven concept development, presentation of hypotheses
- Version space and the evaluation of hypotheses
- Clustering methods
- Regression
- Association rules

**Learning objectives:**

Achieve an understanding of data mining. This includes the theoretical concepts which are for example the data mining cycle, but also the algorithms to explore the data.

**Prerequisites:**

Basic mathematical skills (statistics), basic knowledge in programming (JAVA), basic courses in computer science.

**Workload:**

240 hours, which include 60 hours of presence in the lectures, tutorials, and final exam and 180 hours of self-study.

**Credit requirements:**

There will be programming exercises and theoretical exercises which are a requirement for taking the exam. The final grade will only be based on the oral exam (20 minutes) at the end of the semester.

**Literature:**

- Berthold, M.R., Borgelt, C., Höppner, F., Klawonn, F., "Guide to intelligent data analysis: How to intelligently make sense of real data", Springer, 2010
- Han J., Kamber M., "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, August 2000.
- Ester M., Sander J., "Knowledge Discovery in Databases. Techniken und Anwendungen", Springer, 2000.
- Hand D.J., Mannila H., Smyth P., "Principles of Data Mining", MIT Press, 2001.
- Mitchell T. M., "Machine Learning", McGraw-Hill, 1997.
- Witten I. H., Frank E., "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann Publishers, 2000.

### **5.6.30 Data Mining 2: Frequent Pattern Mining**

**Responsible for module:**

PD Dr. Borgelt (z.B. INF-13520-20171)

**Credits:**

3 SWS, 5 ECTS

**Contents:**

- Frequent item set mining and association rule induction
- Frequent sequence mining (discrete and interval data)
- Frequent tree and graph mining
- Efficient basic algorithms and data structures
- Avoiding redundant search when analyzing structured data, especially with the help of canonical forms of the desired patterns
- Approaches to evaluate and filter found patterns
- Extensions of the basic algorithms for special applications
- Application examples, especially for mining frequent graphs and sequences

**Learning objectives:**

- Knowledge of the basic algorithmic schemes and the most common concrete algorithms for finding frequent item sets
- Understanding of the needed efficient data structures and processing methods
- Insight into the special problems occurring in the analysis of structured data (sequences, trees, general graphs) and approaches to solve these problems
- Ability to select an appropriate method to find frequent patterns depending on the application
- Capability to develop efficient specialized algorithms to find frequent patterns

**Prerequisites:**

Data Mining 1 or Analysis and Visualization

**Workload:**

150 hours

**Credit requirements:**

oral examination (30 minutes)

**Literature:**

- lecture slides
- C.C. Aggarwal and J. Han (eds.), Frequent Pattern Mining, Springer-Verlag 2014

## **5.6.31 Data Mining 2: Artificial Neural Networks and Deep Learning**

**Responsible for module:**

PD Dr. Borgelt (z.B. INF-13670-20171)

**Credits:**

3 SWS, 5 ECTS

**Contents:**

- Biological background
- Threshold logic units and their geometric interpretation
- General neural networks
- Multilayer perceptrons
- Deep learning
- Radial basis function networks
- Learning vector quantization and self-organizing maps
- Hopfield networks and associative memory
- Recurrent neural networks

**Learning objectives:**

- Knowledge of the most common types of artificial neural networks
- Suitability of different network types for supervised and unsupervised learning tasks
- Understanding of the advantages and disadvantages of the different types of neural networks depending on the structure of the data
- Training neural networks with gradient descent and its variants
- Ability to select a neural network type for a given problem
- Application of training artificial neural networks in practice

**Prerequisites:**

Data Mining 1 or Analysis and Visualization

**Workload:**

150 hours

**Credit requirements:**

oral examination (30 minutes)

**Literature:**

- lecture slides
- R. Kruse, C. Borgelt, C. Braune, F. Klawonn, C. Moewes und M. Steinbrecher, Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze, Springer/Vieweg, 2. Auflage 2015
- R. Kruse, C. Borgelt, C. Braune, S. Mostaghim und M. Steinbrecher, Computational Intelligence: A Methodological Introduction, Springer-Verlag, 2. Auflage 2016
- S. Haykin, Neural Networks and Learning Machines, Pearson, 3rd edition 2008



## **5.6.32 Web Information Retrieval**

**Responsible for module:**

Jun.-Prof. Dr. Gipp (z.B. INF-20680-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

The lecture will cover the following topics:

1. Core Concepts of Information Retrieval
  - Information Retrieval Tasks, Process, Models and Data Structures
  - Evaluation of Information Retrieval Systems
2. Core Concepts of Web Information Retrieval
  - Web Search
  - Web Crawling
  - Web Indexing
  - Link Analysis
  - Recommender Systems for the Web
3. (Web) Information Retrieval in Practice
  - Introduction to Python for IR
  - Major Information Retrieval Frameworks and Tools
4. Advanced Topics in Web Information Retrieval
  - Semantic Search
  - The Dark Web
  - Human Computation and Crowdsourcing

The tutorial sessions will mix theoretical assignments and programming projects. The theoretical assignments will consolidate the key concepts introduced in the lecture. The programming projects (team work is possible) will address complex information retrieval tasks on the Web and include subtasks such as:

- crawling and traversing Web pages
- scraping HTML pages, forms, or JavaScript
- parsing and processing natural language from large data sets (e.g. Twitter or Wikipedia)
- cleaning, extracting, transforming, and storing information

The programming language Python will be used.

**Learning objectives:**

Students will learn the predominant information retrieval task on the Web. They will understand the conceptual requirements of particular retrieval tasks and will be able to describe retrieval approaches (data structures, algorithms and their usage as part of a retrieval process) to address particular tasks. After completing this course, students can critically evaluate the strengths and weaknesses of retrieval approaches and can apply suitable retrieval approaches to solve complex information retrieval tasks on the Web.

**Prerequisites:**

A strong foundation in at least one object-oriented programming language, preferably Python, is required. Experience with web development technologies and frameworks is a plus.

**Workload:**

Total workload: 6 ECTS = 180 hours

Attendance time for lecture: 28 hours

Attendance time for tutorial sessions: 28 hours

Self-study workload: 124 hours

**Credit requirements:**

- completion of assignments and programming projects in tutorial sessions
- mid-term exam (pass/fail grading, a pass grade is required for admission to the final exam)
- final exam (written or oral depending on number of participants)
- voluntary option: presenting a research paper in class; the grade for the presentation will contribute to the final grade

**Literature:**

- S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali and S. Quarteroni. *Web Information Retrieval*. Springer, 2013. ISBN 3642393136.
- C. D. Manning, P. Raghavan and H. Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England 2009. (free online edition: <http://www-nlp.stanford.edu/IR-book/>).
- B. Ribeiro-Neto and R. Baeza-Yates. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Pearson Education, Ltd., Harlow, England, Addison-Wesley, 2 edition, 2011. ISBN 9780321416919.

### **5.6.33 Computational Methods for Document Analysis**

**Responsible for module:** Jun.-Prof. Dr. Gipp, Dr. Fuchs (z.B. INF-10585-20152)

**Credits:** 4 SWS, 5 ECTS

**Contents:**

- Natural Language Processing
- Information Retrieval
- Structure Analysis and Information Extraction
- Text Data Mining
- Text Visualization

**Learning objectives:**

The students are able to name the problems of document analysis and the basic algorithms for their solution. The students are able to transfer the algorithms to a specific document analysis problem and are able to assess their solution.

**Prerequisites:**

A successful previous participation in either of the courses Analysis & Visualization or Data Mining is mandatory. The necessary prerequisites for participating in this lecture may be tested as part of an initial exam. Programming skills in Java are necessary to solve the practical exercises.

**Workload:**

Overall 150 hours, distributed to 42 hours of attendance study and 108 hours of self-study.

**Credit requirements:**

Working on the exercise is a prerequisite for admission to examination.

Examination: depending on the number of participants oral (individual exam of 20 minutes) or written (written examination 120 minutes).

Grade: The grade results from the grade of the examination.

**Literature:**

None required, suggested are:

- G. Heyer, U. Quasthoff, T. Wittig. Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse. W3I-Verlag. 2006.
- C. D. Manning, P. Raghavan, H. Schütze. Introduction to Information Retrieval. Cambridge University Press. 2008.
- Ruslan Mitkov. The Oxford Handbook of Computational Linguistics. Oxford University Press. 2005.
- Thomas Payne. Exploring Language Structure: A Student's Guide. Cambridge University Press. 2006.
- M. Ward, G. G. Grinstein, D. Keim. Interactive Data Visualization: Foundations, Techniques, and Applications. Taylor & Francis Ltd. 2010.

### **5.6.34 Algorithms in Bioinformatics**

**Responsible for module:**

Prof. Dr. Schreiber (z.B. INF-20780-20161)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Algorithms are important for the analysis of biological data. This course deals with fundamental algorithms from bioinformatics, e.g. for analysis of sequences, evolutionary relationships, protein structures and biological networks.

**Learning objectives:**

Knowledge of fundamental algorithms in bioinformatics.

**Prerequisites:**

Knowledge of the topics presented in the course "Algorithms and Data Structure". Biological knowledge is not required, the first lecture will be an introduction into important terms and processes from biology.

**Workload:**

180 hours

**Credit requirements:**

Active participation in the exercises (50% of the points) and, depending on number of participants, oral examination (20 min) or written exam (60 min)

**Literature:**

- Jones & Pevzner: An Introduction to Bioinformatics Algorithms, MIT Press, 2004
- Zvelebil & Baum: Understanding Bioinformatics, Garland Science, 2007
- Lesk: Introduction to Bioinformatics, Oxford University Press, 2014
- Mount: Bioinformatics, CSHL Press, 2004

## **5.6.35 Computational Systems Biology: Modelling and Simulation**

**Responsible for module:**

Prof. Dr. Schreiber (z.B. INF-20850-20161)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Computational systems biology investigates biological systems with methods from computer science. This course deals with topics of the modelling, analysis and simulation of mainly molecular biological processes and systems. It focuses on a broad range of approaches including model building, topological modelling and analysis of networks, constraint based analysis techniques, Petri net modelling, rule-based systems, and kinetic modelling.

This course is part of the course series

Computational Systems Biology: Modelling and Simulation

Computational Systems Biology: Representation and Visualisation

**Learning objectives:**

Knowledge of methods for modelling and simulation in systems biology.

**Workload:**

180 hours

**Credit requirements:**

Active participation in the exercises (50% of the points) and, depending on number of participants, oral examination (20 min) or written exam (60 min)

**Literature:**

- Klipp et al.: Systems Biology in Practice, Wiley, 2005
- Junker & Schreiber: Analysis of Biological Networks, Wiley, 2008
- Koch, Reisig & Schreiber: Modelling in Systems Biology - The Petri Net Approach, Springer, 2011

## **5.6.36 Computational Systems Biology: Representation and Visualisations**

**Responsible for module:**

Prof. Dr. Schreiber (z.B. INF-13310-20162)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Computational systems biology models and analyses biological systems with methods from computer science. This course deals with topics of the representation and visualization of mainly molecular biological processes and systems. It focuses on computer science aspects of ontologies in the biosciences, standards for the representation of models and knowledge in systems biology, methods for data mapping and data integration, algorithms for the visualisation of biological data, and immersive analytics methods for data visualisation and analytics.

This course is part of the course series

Computational Systems Biology: Modelling and Simulation

Computational Systems Biology: Representation and Visualisation

**Learning objectives:**

Knowledge of algorithms and methods for the representation, visualisation and data integration in systems biology.

**Workload:**

180 hours

**Credit requirements:**

Active participation in the exercises and a project (50% of the points) and, depending on number of participants, oral examination (20 min) or written exam (60 min). The project will consist of some tasks for immersive analytics of life science data.

**Literature:**

Will be announced at the start of the course.

### **5.6.37 Petri Nets**

**Responsible for module:**

Prof. Dr. Schreiber (z.B. INF-13370-20162)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

Petri nets (named after C. A. Petri) are a well-known and often used structure for the modelling and simulation of different processes in engineering, science, economy and other areas. This course presents a theory for the modelling with Petri nets and different techniques for the analysis of Petri nets. A number of examples enrich the theoretical understanding. Beside elementary Petri nets this course will also deal with Petri net extensions.

**Learning objectives:**

Knowledge of fundamental methods for the modelling using Petri nets and for their analysis.

**Workload:**

180 hours

**Credit requirements:**

Active participation in the excercises (50% of the points) and, depending on number of participants, oral examination (20 min) or written exam (60 min)

**Literature:**

- Wolfgang Reisig: Understanding Petri Nets, Springer, 2013
- Rene David, Hassane Alia: Discrete, Continuous, and Hybrid Petri Nets, Springer, 2005
- Ina Koch, Wolfgang Reisig, Falk Schreiber: Modeling in Systems Biology: The Petri Net Approach, Springer, 2011

## **5.6.38 Functional Safety for Embedded Systems**

**Responsible for module:** Prof. Dr. Leue (z.B. INF-13180-20151)

**Credits:** 3 SWS, 6 ECTS

**Contents:**

This directed studies course will introduce into the areas of embedded systems and functional safety engineering for those systems. Selected topics from these areas will be discussed and presented by participants on a weekly basis. Lectures will be accompanied by a tutorial.

**Learning objectives:**

Participants of this course will be enabled to understand and assess methods used in the design of safe embedded systems. This will enable them to employ adequate techniques in practical system engineering projects. Participants will also learn to acquire technical knowledge from textbook sources and to present this knowledge to their peers.

**Prerequisites:**

Bachelor level knowledge of operating systems, distributed systems, software engineering and theory.

**Workload:**

180 hours, consisting of 42 hours in class and 138 hours self-study.

**Credit requirements:**

Final examination. At least one presentation of a course topic (appr. 30 mins.).

**Literature:**

- P. Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, Springer 2010.
- J. Börcsök, Functional Safety - Basic Principles of Safety-related Systems, VDE Verlag, 2007.
- M. Bozzano, A. Villafiorita, Design and Safety Assessment of Critical Systems, CRC Press, 2010.
- H. Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications, Springer 2011.



### **5.6.39 Directed Studies: Testing of Software**

**Responsible for module:**

Prof. Dr. Leue (z.B. INF-20570-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

This directed studies course will introduce in the foundations and applications of techniques to test complex software systems. Testing is one of the most frequently used software quality assurance techniques in practice. We will discuss the nature and inherent limitations of software testing, testing techniques used at different stages of the software development process, and consider tools that support software testing.

**Learning objectives:**

The participants will learn to assess software testing techniques and to design test strategies and procedures when responsible for software testing, for instance as software developers, software project managers or software testing engineers.

**Prerequisites:**

Bachelor degree in Computer Science level knowledge of Software Engineering, Discrete Mathematics, Programming and Theory of Computation.

**Workload:**

180 hours, consisting of 60 hours in class and 120 hours self-study.

**Credit requirements:**

Final examination (written or oral.) A grade bonus can be earned during the tutorials.

**Literature:**

A reader for this course will be compiled and made available at the beginning of the course.

## **5.6.40 Model Checking of Software and Systems**

**Responsible for module:**

Prof. Dr. Leue (z.B. INF-12110-20152)

**Credits:**

6 SWS, 10 ECTS

**Contents:**

The course will introduce into model checking for reactive software and systems. Model checking is an algorithmic, automated technique for the behavioral analysis of soft- and hardware systems. We will illustrate the algorithmic foundations of this technique, present the SPIN tool, and address advanced topics. The accompanying project will aim at the modeling and analysis of an industrial case-study.

**Learning objectives:**

It is the goal of the course to enable students to understand and assess the efficiency and practical applicability of model checking technology. It is a further goal to exemplify to students that model checking is implemented in tools, and that these can be applied to practical, industrial-strength case studies. This will enable them to decide about an adequate use of model checking technology when conducting their own software and systems development projects, or to pursue further research in this area.

**Prerequisites:**

Bachelor level background in programming, algorithms, concurrent systems, automata theory, graph theory and logic.

**Workload:**

300 hours, which includes 90 hours of presence in the lectures and tutorials and 210 hours of self-study.

**Credit requirements:**

Final written examination (90 mins) or oral examination (30 mins.) Participants achieving at least 80% of the marks that can be obtained in the assignments will earn a 0.3 point grade bonus.

**Literature:**

- E. Clarke, O. Grumberg and D. Peled, *Model Checking*, MIT Press, 1999
- C. Baier and J.-P. Katoen, *Principles of Model Checking*, MIT Press, 2008
- G. Holzmann, *The SPIN Model Checker - Primer and Reference Manual*, Addison Wesley, 2003

Further literature will be announced during the course.

## **5.6.41 Advanced Model Checking**

**Responsible for module:**

Prof. Dr. Leue (z.B. INF-13200-20151)

**Credits:**

6 SWS, 10 ECTS

**Contents:**

The course will cover advanced topics in Model Checking. The course will introduce into CTL model checking and into the foundations of BDD based symbolic model checking. The course will further cover model checking of real-time and probabilistic systems. The course Model Checking of Software and Systems is not a prerequisite for the successful completion of this course.

**Learning objectives:**

It is the objective of this course to enable students to assess and use the particular model checking technologies that will be presented in this course. The participants of this course will be enabled to use model checking technologies in practical problems and to estimate and anticipate the possibilities as well as the limitations of using this technology in practical systems engineering projects. The participants will also be enabled to chose adequate tool support for the solution of practical model checking problems.

**Prerequisites:**

- No special model checking knowledge is needed, all foundations will be taught in the course
- Basic programming skills

**Workload:**

300 hours

**Credit requirements:**

- Oral or written examination at the end of the semester
- Active participation in the excercises

**Literature:**

- E. Clarke, O. Grumberg and D. Peled, *Model Checking*, MIT Press, 1999
- C. Baier and J.-P. Katoen, *Principles of Model Checking*, MIT Press, 2008
- Additional literature will be announced during the course

## **5.6.42 High-Performance Systems and Networks**

**Responsible for module:**

Prof. Dr. Waldvogel (z.B. INF-20150-20152)

**Credits:**

4 SWS, 6 ECTS

**Contents:**

For high-performance and real-time systems, it is important to ensure that events are handled efficiently and in due time. This lecture explains strategies to speed up low-level and high-performance operations, including removal of generality or temporal and spatial off-loading. The examples will be taken from functions relevant to implement high-speed networking, but many are generalizable to other fast operations. Topics include:

- Structured overview over general efficiency-enhancing techniques
- Overhead reduction in the operating system (eliminating copies, interrupts, system calls)
- Efficient handling of timers
- Improving protocol processing
- String matching and advanced matching
- Priorities, scheduling, and fair queueing
- Measuring network traffic
- Efficient security filters

**Learning objectives:**

The goals of this lecture are to

- learn mechanisms to improve the speed and efficiency of system functions
- know how to apply these mechanisms
- understand the individual challenges in applying the general mechanisms to specific problems
- be aware of efficient solutions to specific problems and their generalizations

**Prerequisites:**

The following background knowledge is strongly recommended, but not formally required:

- the Operating Systems lecture
- sound knowledge about computer networking
- basic C programming

**Workload:**

180 hours in total, 60 hours in class and 120 hours self-study

**Credit requirements:**

- 50% of points in the exercise sheets
- oral exam at the end of the semester

**Literature:**

- Varghese, George. *Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices*. Morgan Kaufmann, 2005.
- Le Boudec, Jean-Yves. *Performance Evaluation of Computer and Communication Systems*. EPFL Press, 2010.

## **5.6.43 Open Source Software Development**

**Responsible for module:**

Prof. Dr. Waldvogel (z.B. INF-20990-20161)

**Credits:**

3 SWS, 5 ECTS

**Contents:**

Open Source Software plays an important role in many enterprise and private applications. For example, a large part of our Web experience stack is built on Open Source: from Linux over Apache/nginx to Web scripting languages or Web application environments and browsers.

Knowing how to enhance an existing product using plug-ins or modification of the existing code is a key skill in the modern world, as it allows features to be added efficiently and simplifies the life of all users.

This course will therefore not only provide you with a background on open source technologies and project management, you will also get a chance to actively contribute to your favorite open source project under guidance.

To make your choice easier, we have a selection of smaller and larger changes ready. Feel free to bring your own ideas, though!

During real-world experience, you will be using collaboration and will be guided and enhanced using peer instruction.

**Learning objectives:**

You will learn

- History, benefits and disadvantages of Open Source
- Open Source Licensing
- How to efficiently collaborate in a distributed project with many volunteers
- The social and technological foundations of open source project management
- Security aspects of Open Source
- Build processes and tools
- Introductions to web scripting languages

And you will get hands-on experience

- Contributing to an existing Open Source project
- Finding the right spot for your changes
- Coordinating with the other developers and users

**Prerequisites:**

Software project and knowledge of at least one common programming language used in Open Source projects.

**Workload:**

150 hours

**Credit requirements:**

The grade will be determined based on the projects.