

Modulverzeichnis

des Master-Studiengangs

Angewandte Informatik (M.Sc.) – MIN

an der Hochschule Hannover

Stand: 5.1.2016

Hochschule Hannover
Fakultät IV – Wirtschaft und Informatik
Abteilung Informatik
Ricklinger Stadtweg 120
30459 Hannover
<http://f4.hs-hannover.de/>



Module und Dozentinnen/Dozenten

Master-Studiengang Angewandte Informatik (M.Sc.) – MIN				
Lfd. Nr.	Modul	Abkürzung	Modulverantwortlichkeit	Dozentinnen/Dozenten
Gemeinsamer Block				
1	Softwarearchitekturen	MIN-SWA	Dunkel	Bruns, Dunkel, Koschel
2	Projekt- und Qualitätsmanagement	MIN-PMQM	Koschel	Bruns, Hovestadt, Koschel
3	Visualisierung und HCI	MIN-VISH	Ahlers	Ahlers, Ginkel
4	Algorithmen und Komplexität	MIN-AK	Ginkel	Garmann, Ginkel, Heine, Sprengel, Kleiner
5	Seminar	MIN-MSEM	Wohlfeil	Alle Lehrenden
6	Masterprojekt	MIN-MSPR	Dunkel	Alle Lehrenden
7	Masterarbeit	MIN-MA	Wohlfeil	Alle Lehrenden
Schwerpunkt Computergrafik				
G8	Geometrische Modellierung	MIN-GM	Sprengel	Ginkel, Sprengel
G9	Computergrafik und Interaktion	MIN-CGI	Ahlers	Ahlers, Ginkel, Sprengel
G10	Visual Computing	MIN-VC	Sprengel	Ahlers, Dennert-Möller, Ginkel, Sprengel
G11	Computational Geometry	MIN-COG	Ginkel	Ahlers, Ginkel, Sprengel
G12	Computer Vision	MIN-CV	Dennert-Möller	Dennert-Möller, Sprengel
Schwerpunkt IT-Sicherheit				
T8	IT-Sicherheit I	MIN-ITS1	Wohlfeil	Hovestadt, Peine, Wohlfeil
T9	IT-Sicherheit II	MIN-ITS2	Wohlfeil	Hovestadt, Peine, Wohlfeil
T10	Spezialthema IT-Sicherheit	MIN-SSI	Hovestadt	Hovestadt, Peine, Wohlfeil
T11	Secure Software Engineering	MIN-SSE	Peine	Peine, Wohlfeil Gleiches Modul wie S12, kann entweder in IT-Sicherheit oder in Softwaretechnik belegt werden
Schwerpunkt Informationssysteme				
I8	Datenbankparadigmen	MIN-DBP	Heine	Dennert-Möller, Heine, Kleiner, Koschel
I9	Verteilte Informationssysteme	MIN-VIS	Koschel	Heine, Hovestadt, Koschel
I10	Datenanalyse	MIN-DA	Heine	Dennert-Möller, Heine, Kleiner, Koschel
I11	Geo- und Multimedia Informationssysteme	MIN-GMI	Kleiner	Dennert-Möller, Heine, Kleiner
Schwerpunkt Softwaretechnik				
S8	Programmierparadigmen	MIN-PPD	Peine	Garmann, Kleiner, Peine, Wohlfeil
S9	Intelligente Systeme	MIN-INS	Bruns	Bruns, Dunkel, Wohlfeil
S10	Fortgeschrittene Aspekte verteilter Systeme	MIN-FAVS	Koschel	Heine, Hovestadt, Koschel
S11	Software-Test und Requirements	MIN-STR	Garmann	Garmann, Koschel, Peine
S12	Secure Software Engineering	MIN-SSE	Peine	Peine, Wohlfeil Gleiches Modul wie T11, kann entweder in IT-Sicherheit oder in Softwaretechnik belegt werden

Modulbeschreibungen

Vorlesungsmodule im gemeinsamen Block werden mit allen Master-Studierenden durchgeführt, also ca. 35 Studierenden in den Vorlesungen und typischerweise zwei Übungsgruppen mit je ca. 18 Studierenden.

Die Vorlesungsmodule in den einzelnen Schwerpunkten werden mit entsprechend anteiligen Master-Studierendenzahlen durchgeführt, also ca. 10 bis 35 Studierenden in den Vorlesungen und ein oder zwei Übungsgruppe mit je ca. 10 bis 20 Studierenden.

1. Gemeinsamer Block

Modulbezeichnung:	Softwarearchitekturen
Kürzel	MIN-SWA
Studiensemester:	1. Semester (SoSe), Turnus: Jährlich
Modulverantwortliche(r):	Jürgen Dunkel
Dozent(in):	Ralf Bruns, Jürgen Dunkel, Arne Koschel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, Pflichtfach, 1. Semester
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Keine
Angestrebte Lernergebnisse:	<p>Technologische Kompetenz: es sollen tiefgehende Fachkenntnisse im Gebiet Softwarearchitekturen vermittelt werden. Dabei soll insbesondere ein umfassendes Verständnis für die Anwendbarkeit der vorgestellten Techniken und Methoden sowie deren Grenzen entwickelt werden.</p> <p>Methodenkompetenz: Anwendung und kritische Bewertung der vorgestellten innovativen Methoden für die Entwicklung komplexer Softwarearchitekturen.</p> <p>Analyse-/Design und Realisierungskompetenz: die Fähigkeit, Anforderungen sowie die Erfolgsfaktoren von Softwarearchitekturen definieren zu können; Verwendung formaler Methoden und Modellierungssprachen zur Beschreibung von Softwarearchitekturen; Einsatz von Architekturmodellen zur Umsetzung von Systemarchitekturen.</p>
Inhalt:	<p>Grundkonzepte und Eigenschaften moderner Softwarearchitekturen, Entwurfskriterien, fortgeschritten Techniken zur Entwicklung komplexer Softwaresysteme. Dabei sollen insbesondere aktuelle Themen und Entwicklungen berücksichtigt werden. fortgeschrittene Ansätze zur Modellierung und Realisierung komplexer Systeme, z.B. Design-By-Contract mit OCL, Domain-Driven-Design, Modellgetriebene Softwareentwicklung (Domain Specific Languages), Geschäftsprozessmodellierung.</p> <p>Forschungsnahe Themen können ggf. in Masterarbeiten fortgeführt werden.</p>
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung</p>
Literatur:	Aktuell nach Lehrinhalten

Modulbezeichnung:	Projekt- und Qualitätsmanagement
Kürzel:	MIN-PMQM
Studiensemester:	1. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Arne Koschel
Dozent(in):	Ralf Bruns, Arne Koschel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, Pflichtfach, 1. Semester
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Abgeschlossenes Bachelorstudium mit Informatikanteil, der mindestens den Programmieren- und Softwareengineering-Veranstaltungen vergleichbar ist.
Angestrebte Lernergebnisse:	Analyse, Design- und Methodenkompetenz: Die Studierenden erwerben die Fähigkeit zu eigenverantwortlicher Tätigkeit im Berufsfeld. Sie sollen Methoden und Konzepte wesentlicher Aspekte des Projekt- und Qualitätsmanagement erlernen um auch als Leiter eines Teams, das aus unterschiedlichen Disziplinen und Niveaus bestehen kann, arbeiten zu können.
Inhalt:	Es werden grundlegende Organisationsformen und Abläufe zur Durchführung von Projekten vorgestellt. Dazu gehören u.a. <ul style="list-style-type: none"> • Projekt-Akquisition, • Projektorganisation und -planung, • Maßnahmen zum Projektcontrolling und Risiko-Management, • Konfigurations-Management • Qualitätssicherungsmaßnahmen Darüber hinaus werden Vorgehensmodelle zur Sicherstellung der Prozessqualität diskutiert, wie bspw. ISO 9000 oder CMM.
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Seminarthemen, Ausarbeitung, Ergebnispräsentation, Begutachtung, experimentelle Arbeit, Gesprächsführung
Literatur:	DeMarco, T., Lister, T.: „Wien wartet auf Dich“, Hanser. DeMarco, T.: „Der Termin“, Hanser. Balzert, H.: „Lehrbuch der Software-Technik 1/2“. Spektrum. Cockburn, A.: „Agile Software-Entwicklung“, Mitp (bzw. Engl. Original). Beck, K., Andres, C.: „Extreme Programming Explained“ (und: Kent Beck: Extreme Programming). Weitere Quellen zu speziellen Themen.

Modulbezeichnung:	Visualisierung und HCI
Kürzel:	MIN-VISH
Studiensemester:	2. Semester (SoSe), Turnus: Jährlich
Modulverantwortliche(r):	Volker Ahlers
Dozent(in):	Volker Ahlers, Ingo Ginkel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, Pflichtfach, 2. Semester
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung/Projekt
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Keine
Angestrebte Lernergebnisse:	<p>Algorithmische und mathematische Kompetenzen: Kenntnis der Grundlagen der Mensch-Computer-Interaktion (HCI); Verständnis von Visualisierungsalgorithmen; Verständnis unterschiedlicher Datenrepräsentationen</p> <p>Analyse-, Design- und Realisierungskompetenzen: Analyse von Datensätzen und Visualisierungsanforderungen; Design und Realisierung von Visualisierungslösungen, Implementierung von Algorithmen</p> <p>Technologische Kompetenzen: Kenntnis des Einsatzes von Visualisierungstechniken in unterschiedlichen Anwendungsgebieten</p> <p>Methodenkompetenzen: Kenntnis der Möglichkeiten, des Nutzens und der Grenzen des Einsatzes von Visualisierungstechniken, Erkennen von Fehlern in Visualisierungslösungen</p>
Inhalt:	<ul style="list-style-type: none"> • Grundlagen: Mensch-Computer-Interaktion (HCI), Wahrnehmung und Kognition, Gestaltgesetze, Farbmodelle, Datenrepräsentation • Skalare Daten: Diagramme (z.B. Linien-, Säulen-, Streudiagramm, Histogramm), Farbkodierung, Konturlinien, multivariate Daten • Volumendaten: Isoflächen, Volume Rendering, Ray Casting • Vektorfelder: Glyphen, Stromlinien, Strombahnen • Beziehungen: Visualisierung von Hierarchien (Bäumen) und Netzwerken (Graphen), z.B. Tree Map, kräftebasiertes (force-directed) Layout • Software: Aktuelle Visualisierungsbibliotheken und grafische Entwicklungsumgebungen, Anwendung auf reale Daten
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung von Aufgaben in Gruppen, Projektarbeit mit Ergebnispräsentation, Begutachtung und Diskussion der Lösungen, Gesprächsführung</p>
Literatur:	<p>Skript zur Vorlesung.</p> <p>Telea, A.C.: Data Visualization. A K Peters, neueste Auflage</p> <p>Ward, M., Grinstein, G.G., Keim, D.: Interactive Data Visualization. A K Peters, neueste Auflage</p> <p>Ware, C.: Information Visualization. Morgan Kaufmann, neueste Auflage</p>

Modulbezeichnung:	Algorithmen und Komplexität
Kürzel:	MIN-AK
Studiensemester:	3. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Ingo Ginkel
Dozent(in):	Ingo Ginkel, Frauke Sprengel, Felix Heine, Carsten Kleiner, Robert Garmann
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, Pflichtfach, 3. Semester
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Datenstrukturen und Algorithmen, Theoretische Informatik aus dem Bachelor-Studium
Angestrebte Lernergebnisse:	Theoretische und mathematische Kompetenzen: Charakterisierung des Schwierigkeitsgrads vorgegebener Probleme sowie deren Klassifizierung bezüglich effizienter algorithmischer Lösbarkeit. Analyse-, Design- und Realisierungskompetenzen: Kenntnis verschiedener algorithmischer Lösungsstrategien zur Bearbeitung praxisrelevanter Probleme von hoher Komplexität (Zeit/Speicher) Technologische Kompetenzen: Implementierung dieser Algorithmen
Inhalt:	<ul style="list-style-type: none"> Theorie: Berechenbarkeit, Klassen P und NP, NP-Vollständigkeit, Polynomielle Reduktion, Satz von Cook-Levin, Graphenalgorithmen Lösungsstrategien: Divide and Conquer (Master Theorem), Backtracking, Branch-and-Bound, lokale Verbesserung, lineare Programmierung, genetische Algorithmen, Simulated Annealing, Randomisierung, Bin-Packing Probleme: z.B. Max-Cut Problem, Travelling Salesman Problem, Rucksackproblem, Eulerkreise, Minimum Spanning Tree, schnelle Potenzierung
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung von Aufgaben in Zweiergruppen, Projektarbeit mit Ergebnispräsentation, Begutachtung und Diskussion der Lösungen, Gesprächsführung
Literatur:	Folien/Skript zur Vorlesung Hromkovic: Algorithmics for Hard Problems: Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics, Springer 2004 Wegener: Komplexitätstheorie - Grenzen der Effizienz von Algorithmen. Springer, 2003

Modulbezeichnung:	Masterseminar
Kürzel	MIN-MSEM
Studiensemester:	2. Semester (SoSe), Turnus: Jährlich
Modulverantwortliche(r):	Stefan Wohlfeil
Dozent(in):	Alle Dozentinnen und Dozenten
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, Pflichtfach, 2. Semester
Lehrform/SWS:	2 SWS Seminar mit ca. 5-15 Studierenden
Arbeitsaufwand:	Eigener Vortrag = 2 h; Teilnahme an anderen Vorträgen = 28 h Eigenstudium = 70 h; Erstellung der Seminararbeit = 80 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	
Empfohlene Voraussetzungen:	Alle Master Module des ersten Semesters
Angestrebte Lernergebnisse:	<p>Fachliche Kompetenz: Im Seminar vertiefen die Studierenden exemplarisch die Inhalte der vorherigen Veranstaltungen und erweitern ihre analytischen Kompetenzen indem sie neue und in der Forschung/Entwicklung befindliche Themen bearbeiten. Da diese aktuellen Themen auch aus angrenzenden Fachgebieten stammen vertiefen die Studierenden ihre fachübergreifenden Sachkompetenzen.</p> <p>Methodenkompetenz: Die Studierenden vertiefen ihre Kenntnisse im selbständigen Arbeiten mit anspruchsvoller wissenschaftlicher Literatur.</p> <p>Soziale Kompetenz und Selbstkompetenz: Die Studierenden lernen, fachwissenschaftliche Inhalte überzeugend mündlich und schriftlich zu präsentieren. In Diskussionen wird die Fähigkeit zur kritischen Reflektion geübt. Durch die Pflicht zur Einhaltung von Terminen unter nur begrenzt zur Verfügung stehenden Ressourcen wird die Projektmanagementkompetenz geschult sowie die Lern- und Leistungsbereitschaft gefördert.</p>
Inhalt:	Die Teilnehmer erarbeiten selbstständig ein anspruchsvolles wissenschaftliches Thema, erstellen einen schriftlichen Bericht und präsentieren die Ergebnisse. Dabei werden wissenschaftliche Methoden und Techniken angewendet.
Studien-/Prüfungsleistungen:	Schriftliche Ausarbeitung, eigener Vortrag, aktive Teilnahme an den anderen Vorträgen, Beteiligung an der Diskussion.
Medienformen:	
Literatur:	Wird zu Beginn der Lehrveranstaltung von den Dozenten bekannt gegeben.

Modulbezeichnung:	Masterprojekt
Kürzel:	MIN-MSPR
Studiensemester:	3. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Jürgen Dunkel
Dozent(in):	Alle Lehrenden
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, Pflichtfach, 3. Semester
Lehrform/SWS:	6 SWS Projektarbeit
Arbeitsaufwand:	Projektarbeit = 180 h Eigenstudium = 180 h
Kreditpunkte:	12 CP (= 360 h)
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Voraussetzungen:	alle Lehrveranstaltungen der jeweiligen Vertiefungsrichtung, insbesondere Projekt- und Qualitäts-Management
Angestrebte Lernergebnisse:	Analyse-Kompetenzen: die gegebene Fragestellung unter Einsatz wissenschaftlicher Methoden und Erkenntnisse eigenständig analysieren können. Design- /Realisierungs-Kompetenzen: Lösungsstrategien entwickeln und diese mit den erworbenen Kenntnissen umzusetzen können. Technologische Kompetenzen: Wissen aus verschiedenen Bereichen kombinieren und zielgerichtet einsetzen können. Methoden-Kompetenzen: verschiedene innovative Methoden zur Lösung praktischer Probleme anwenden können. Projektmanagement-Kompetenz: Fähigkeit zur Projektplanung, zum Aufbau einer Organisationsstruktur und zur gezielten Steuerung von Projekten. Risiko-Management Soziale Kompetenzen: Anwendung von Konflikt-Lösungsstrategien
Inhalt:	Im Masterprojekt wird eine komplexe und aktuelle Fragestellung der Informatik bearbeitet, die einem der Schwerpunkte entstammen kann. Kennzeichnend ist dabei, dass die Fragestellung in einem Projektteam eigenständig mit projektspezifischer Organisation, Durchführung und Planung bearbeitet wird. Die konkreten Inhalte, Verfahren und Methoden ergeben sich aus der jeweiligen Fragestellung.
Studien-/ Prüfungsleistungen:	Experimentelle Arbeit
Medienformen:	Projektarbeit: Präsentation, Tafel, Beispielaufgaben, Diskussion Selbständige Bearbeitung der Aufgaben, Begutachtung der Lösungen, Gesprächsführung
Literatur:	Projektspezifisch

Modulbezeichnung:	Masterarbeit
Kürzel:	MIN-MA
Studiensemester:	4. Semester (SoSe), Turnus: Jährlich (bei Bedarf auch jedes Semester)
Modulverantwortliche(r):	Stefan Wohlfeil
Dozent(in):	Alle Dozentinnen und Dozenten
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, Pflichtfach, 4. Semester
Lehrform/SWS:	
Arbeitsaufwand:	Eigenstudium = 810 h; Vorbereitung und Durchführung des Kolloquiums = 90 h
Kreditpunkte:	30 CP (= 900 h)
Voraussetzungen nach Prüfungsordnung:	Zulassung zum Kolloquium nur wenn sämtliche Fachprüfungen bestanden
Empfohlene Voraussetzungen:	Alle Master-Module des 1.-3. Semesters gemäß gewählten Schwerpunkten.
Angestrebte Lernergebnisse:	Fähigkeit ein komplexes, praxisbezogenes Informatik-Thema selbstständig und mit wissenschaftlichen Methoden zu bearbeiten, d.h. Problemstellung analysieren, Lösungsmöglichkeiten aufzeigen, in den Stand der Wissenschaft/Technik einordnen und evtl. implementieren und abschließend bewerten. Fähigkeit zum Verfassen einer anspruchsvollen wissenschaftlichen Ausarbeitung zum Thema. Im Kolloquium zeigen die Studierenden, dass sie komplexe Sachverhalte in kurzer Zeit vor Fachpublikum verständlich präsentieren können.
Inhalt:	Im Gegensatz zur Bachelorarbeit wird hier ein anspruchsvolleres und evtl. umfangreicheres Thema auf einem wissenschaftlich höheren Niveau über einen längeren Zeitraum bearbeitet.
Studien-/Prüfungsleistungen:	Schriftliche Abschlussarbeit, eigener Vortrag und Fachdiskussion über das Thema der Arbeit / Verteidigung der Arbeit.
Medienformen:	
Literatur:	Wird abhängig vom Thema der Arbeit von den Dozenten vorgeschlagen und dann von den Studierenden selbstständig aktualisiert, erweitert und ergänzt.

2. Schwerpunkt Computergrafik

Modulbezeichnung:	Geometrische Modellierung
Kürzel	MIN-GM
Studiensemester:	2. Semester (SoSe), Turnus: Jährlich
Modulverantwortliche(r):	Frauke Sprengel
Dozent(in):	Frauke Sprengel, Ingo Ginkel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 2. Semester, Modul im Schwerpunkt Computergrafik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Computergrafik 1, Mathematik 3 aus dem Bachelor-Studium
Angestrebte Lernergebnisse:	Theoretische und methodische Kompetenz: erwerben grundlegende Fachkenntnisse im Bereich Geometrische Modellierung, entwickeln Verständnis für anwendbare Techniken und Methoden und deren Grenzen Analyse-, Design- und Realisierungskompetenz: sind fähig, Modellierungsprobleme zu formulieren, zu formalisieren und zu lösen (d.h. entsprechende Algorithmen zu implementieren)
Inhalt:	Freiformkurven und -flächen (Bezier, B-Splines, NURBS), Unterteilungsflächen, Körpermodellierung (Constructive Solid Geometry), Geometrische Stetigkeit, Interpolation, Approximation
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispiele, Diskussion Übung: Selbständige Bearbeitung praktischer Aufgaben in Gruppen, Begutachtung der Lösungen, Gesprächsführung
Literatur:	Skript / Folien zur Vorlesung Gerald Farin: Curves and Surfaces in CAGD, neueste Auflage C. de Boor. A Practical Guide to Splines. <i>Springer</i> , Heidelberg 1987 H. Prautzsch: Bézier and B-Spline Techniques, Springer 2002

Modulbezeichnung:	Computergrafik und Interaktion
Kürzel:	MIN-CGI
Studiensemester:	1. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Volker Ahlers
Dozent(in):	Volker Ahlers, Frauke Sprengel, Ingo Ginkel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, 1. Semester, Modul im Schwerpunkt Computergrafik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Einführungsvorlesung zur Computergrafik im Bachelorstudium
Angestrebte Lernergebnisse:	<p>Algorithmische Kompetenzen: Verständnis der mathematischen und algorithmischen Grundlagen der Computergrafik, insb. des Echtzeit-Renderng, sowie der virtuellen und erweiterten Realität</p> <p>Analyse-, Design- und Realisierungskompetenzen: Design und Realisierung interaktiver Grafikanwendungen mit aktuellen Grafikbibliotheken; Design und Realisierung natürlicher Benutzerschnittstellen</p> <p>Technologische Kompetenzen: Verständnis der Funktionsweise moderner Grafikprozessoren; Kenntnis des Einsatzes der Computergrafik in unterschiedlichen Anwendungsgebieten</p> <p>Methodenkompetenzen: Kenntnis der Möglichkeiten, des Nutzens und der Grenzen des Einsatzes von Techniken der Computergrafik sowie der virtuellen und erweiterten Realität</p>
Inhalt:	<ul style="list-style-type: none"> • Grundlagen: Echtzeit-Renderng, Architektur moderner Grafikprozessoren, Rendering Pipeline, programmierbare Shader, Konzepte der virtuellen und erweiterten Realität • Fortgeschrittene Rendering-Techniken: Spiegelung, Schatten, Image-based Rendering, Partikelsysteme, Kollisionserkennung • Modellierung: Szenengraphen, räumliche Datenstrukturen • Interaktion: natürliche Benutzerschnittstellen, Stereo Rendering, Motion Tracking, Ansteuerung entsprechender Peripheriegeräte • Software: Aktuelle Grafik-, Szenengraph- und VR-Bibliotheken, Anwendungen der Computergrafik
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung von Aufgaben in Gruppen, Projektarbeit mit Ergebnispräsentation, Begutachtung und Diskussion der Lösungen, Gesprächsführung.</p>
Literatur:	<p>Skript zur Vorlesung</p> <p>Akenine-Möller, T., Haines, E., Hoffman, N.: Real-Time Rendering. A K Peters, neueste Auflage</p> <p>Nischwitz, A., Fischer, M., Haberäcker, P., Socher, G.: Computergrafik und Bildverarbeitung, Band 1. Vieweg+Teubner, neueste Auflage</p>

Modulbezeichnung:	Visual Computing
Kürzel	MIN-VC
Studiensemester:	3. Semester (WiSe), Turnus: Jährlich, wechselnd mit MIN-COG und MIN-CV
Modulverantwortliche(r):	Frauke Sprengel
Dozent(in):	Volker Ahlers, Elisabeth Dennert-Möller, Frauke Sprengel, Ingo Ginkel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 3. Semester, Modul im Schwerpunkt Computergrafik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Grundkenntnisse in digitaler Bildverarbeitung und Computergrafik
Angestrebte Lernergebnisse:	Technologische Kompetenz: tiefgehende Fachkenntnisse in einem ausgewählten Spezialgebiet des Visual Computing, Verständnis für anwendbare Methoden sowie deren Grenzen Methodenkompetenz: Anwendung innovativer Methoden im ausgewählten Bereich Analyse-, Design- und Realisierungskompetenz: Fähigkeiten, Probleme aus einem neuen und in der Entwicklung begriffenen Bereich der Computergrafik, Computer Vision und Visualisierung zu formulieren, zu formalisieren und zu lösen
Inhalt:	Es wird ein ausgewähltes Thema aus dem Bereich Visual Computing vorgestellt, z.B. Medizinische Visualisierung, digitale Bilderzeugung, Mustererkennung und maschinelles Lernen, Künstliche Intelligenz, Robotik, GPU-Computing, Bilddatenbanken, Auffinden von Ereignissen in Bildfolgen, Modellierung und Simulation. Dabei sollen vor allen Dingen aktuelle Themen und Entwicklungen Berücksichtigung finden. Forschungsnahe Fragestellungen legen dabei ggf. eine Basis für die Master-Arbeit.
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, Projektarbeit, Gesprächsführung
Literatur:	Aktuell nach Lehrinhalten

Modulbezeichnung:	Computational Geometry
Kürzel	MIN-COG
Studiensemester:	3. Semester (WiSe), Turnus: Jährlich, wechselnd mit MIN-VC und MIN-CV
Modulverantwortliche(r):	Ingo Ginkel
Dozent(in):	Ingo Ginkel, Volker Ahlers, Frauke Sprengel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 3. Semester, Modul im Schwerpunkt Computergrafik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Geometrische Modellierung, Computergrafik und Interaktion
Angestrebte Lernergebnisse:	Theoretische und mathematische Kompetenzen: Charakterisierung des Schwierigkeitsgrads vorgegebener geometrischer Probleme sowie deren Klassifizierung bezüglich effizienter algorithmischer Lösbarkeit. Analyse-, Design- und Realisierungskompetenzen: Kenntnis verschiedener algorithmischer Lösungsstrategien zur Bearbeitung praxisrelevanter geometrischer Fragestellungen. Technologische Kompetenzen: Implementierung dieser Algorithmen
Inhalt:	Es werden ausgewählte Themen aus dem Bereich Computational Geometry vorgestellt, z.B. Art-Gallery-Problem, Polygon-Triangulation, Voronoi-Diagramme, Delaunay-Triangulation, Windowing, Point Location, Effiziente Kollisionsvermeidungsstrategien, Raumdatenstrukturen wie Octrees oder kd-Trees, Robot Motion-Planning, Graphentheorie und Graphen-Algorithmen, Point-based Graphics, Netzdatenverarbeitung, Geometrische Filteralgorithmen, etc. Dabei sollen vor allen Dingen aktuelle Themen und Entwicklungen Berücksichtigung finden. Forschungsnahe Fragestellungen legen dabei ggf. eine Basis für die Master-Arbeit.
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, Projektarbeit, Gesprächsführung
Literatur:	De Berg: Computational Geometry, Springer 2008, sowie weitere Literatur aktuell nach Lehrinhalten

Modulbezeichnung:	Computer Vision
Kürzel	MIN-CV
Studiensemester:	3. Semester (WiSe), Turnus: Jährlich, wechselnd mit MIN-COG und MIN-VC
Modulverantwortliche(r):	Elisabeth Dennert-Möller
Dozent(in):	Elisabeth Dennert-Möller, Frauke Sprengel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 3. Semester, Modul im Schwerpunkt Computergrafik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung, Projektarbeit in Kleingruppen
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Projektarbeit und Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Grundlagen der digitalen Bildverarbeitung (siehe CG2 aus Bachelor – Studiengang)
Angestrebte Lernergebnisse:	Algorithmische und mathematische Kompetenz: profundes Wissen und Verständnis für die mathematischen und algorithmischen Prinzipien der digitalen Bildverarbeitung, Analyse – und Design - Kompetenz: Lösen von unüblich oder unvollständig definierten Problemen der Bildanalyse in verschiedenen Anwendungsszenarien Technologische Kompetenz durch Kombinieren von Wissen aus Bildverarbeitung, Mathematik und Informatik und Erkennen der Grenzen von Bildanalyseverfahren
Inhalt:	Vertiefte Kenntnisse über Verfahren zur Erkennung und Extraktion logisch zusammenhängender Bildinhalte in Bildern oder Bildfolgen und ihrer Beschreibung auf höherem Abstraktionsniveau, um bspw. abgebildete Objekte zu erkennen <ul style="list-style-type: none"> - Segmentierungsverfahren - Repräsentation und Beschreibung von Segmenten - Objekterkennung - Charakterisierung und Vergleich von ähnlichen Bildern
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbstständige Bearbeitung von Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
Literatur:	Skript zur Vorlesung Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Prentice Hall Jähne, B. Digitale Bildverarbeitung, Springer - Verlag Baggio, Escriva, Mahmood et al.: Mastering OpenCV with Practical Computer Vision Projects, [packt]publishing

4. Schwerpunkt IT-Sicherheit

Modulbezeichnung:	IT-Sicherheit I
Kürzel:	MIN-ITS1
Studiensemester:	1. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Stefan Wohlfeil
Dozent(in):	Stefan Wohlfeil, Holger Peine, Matthias Hovestadt
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, 1. Semester, Modul im Schwerpunkt IT-Sicherheit
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Kenntnisse von Betriebssystem- und Netz-Technologien
Angestrebte Lernergebnisse:	<p>Technologische Kompetenzen: Die Studierenden besitzen fundierte, tiefgehende Kenntnisse von Sicherheitsmaßnahmen u. -Mechanismen. Sie können Wissen aus den Bereichen Mathematik, Rechnernetze und IT-Sicherheit kombinieren und mit der deren Komplexität umgehen. Sie kennen die Grundlagen der wichtigsten Techniken im Bereich IT-Sicherheit und können diese selbständig erweitern und vertiefen.</p> <p>Design- und Realisierungskompetenzen: Die Studierenden können unübliche, unvollständig definierte Probleme aus dem Bereich IT-Sicherheit lösen und die Lösungen auch technisch umsetzen.</p> <p>Soziale Kompetenzen: Die Studierenden erwerben in den Übungen die Fähigkeit zu eigenverantwortlicher Tätigkeit (Einarbeitung, Analyse, Entwurf und Umsetzung) sowie zur Zusammenarbeit in Teams.</p>
Inhalt:	<p>Fachterminologie: Bedrohungen, Risiken, Schwachstellen, Schutzziele.</p> <p>Sicherheitstechniken: Authentisierung, Autorisierung, Identity Management; Kryptographie; Hash-Funktionen; Message Authentication Codes; Digitale Signaturen; X.509 Zertifikate; Anwendung der Techniken bei E-Mail (PGP, S/MIME), WWW (SSL), Anmeldung auf entfernten Rechnern (SSH); Kerberos; Sicherheitsstandards/-prozesse; Forensik; Web-Sicherheit.</p>
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispiele, Diskussion</p> <p>Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung</p>
Literatur:	<p>C. Eckert: IT- Sicherheit. Konzepte, Verfahren, Protokolle; Oldenbourg</p> <p>C. Kaufman, R. Perlman, M. Speciner: Network Security – Private Communication in a PUBLIC World; Prentice Hall</p> <p>W. Stallings: Cryptography and Network Security; Prentice-Hall</p> <p>S. Wohlfeil: "Kurs 1866 - Sicherheit im Internet", FernUniversität Hagen</p>

Modulbezeichnung:	IT-Sicherheit II
Kürzel	MIN-ITS2
Studiensemester:	2. Semester (SoSe), Turnus: Jährlich
Modulverantwortliche(r):	Stefan Wohlfeil
Dozent(in):	Stefan Wohlfeil, Holger Peine, Matthias Hovestadt
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 2. Semester, Modul im Schwerpunkt IT-Sicherheit
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	IT-Sicherheit 1
Angestrebte Lernergebnisse:	<p>Technologische Kompetenzen: Die Studierenden besitzen fundierte und tiefgehende Kenntnisse von Sicherheitsmaßnahmen und -Mechanismen. Sie können Wissen aus den Bereichen Mathematik, Rechnernetze und IT-Sicherheit kombinieren und mit der entstehenden Komplexität umgehen. Sie kennen die Grundlagen der wichtigsten Techniken im Bereich IT-Sicherheit und können diese selbständig erweitern und vertiefen.</p> <p>Design- und Realisierungskompetenzen: Die Studierenden können unübliche, unvollständig definierte Probleme aus dem Bereich IT-Sicherheit lösen und die Lösungen auch technisch umsetzen.</p> <p>Soziale Kompetenzen: Die Studierenden erwerben in den Übungen die Fähigkeit zu eigenverantwortlicher Tätigkeit (Einarbeitung, Analyse, Entwurf und Umsetzung) sowie zur Zusammenarbeit in Teams.</p>
Inhalt:	Sicherheit im E-Commerce, sichere Bezahlfverfahren; Biometrie; Anonymität und Privacy (Mixe); Intrusion Detection (Prinzipien und netzbasierte sowie hostbasierte Systeme); Virtual Private Networks (Funktionsweise, Eigenschaften, Systeme wie IPSEC, OpenVPN), Sicherheitsarchitekturen, Firewallarchitekturen (Paketfilter, ALG, DMZ, etc.); Virtualisierung;
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispiele, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
Literatur:	S. Northcutt, J. Novak: Network Intrusion Detection; New Riders J. Snader: VPNs Illustrated; Addison-Wesley S. Wohlfeil: "Kurs 1867 - Sicherheit im Internet 2"; FernUniversität Hagen.

Modulbezeichnung:	Spezialthema IT-Sicherheit (IT-Sicherheit III)
Kürzel:	MIN-SSI
Studiensemester:	3. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Matthias Hovestadt
Dozent(in):	Matthias Hovestadt, Holger Peine, Stefan Wohlfeil
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, 3. Semester, Modul im Schwerpunkt IT-Sicherheit
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	IT-Sicherheit I und II
Angestrebte Lernergebnisse:	Technologische Kompetenz: tiefgehende Fachkenntnisse in einem ausgewählten Spezialgebiet der IT-Sicherheit erwerben; Verständnis für anwendbare Techniken und Methoden und deren Grenzen entwickeln. Methodenkompetenz: innovative Methoden des ausgewählten Bereichs anwenden können; Voraussetzungen für die Weiterentwicklung der IT-Sicherheit als wissenschaftliche Disziplin erwerben Übergreifend: Transferkompetenz; komplexe Zusammenhänge erfassen und zusammenfassen können
Inhalt:	Ausgewählte aktuelle Themen aus dem Bereich der IT-Sicherheit, z.B. Trusted Computing, Kryptographie, Entwicklung sicherer Software, Sicherheitsbewertung, -überprüfung und -management, Bedrohungs- und Risikoanalysen, Penetrationstests.
Studien-/Prüfungsleistungen:	Mündliche Prüfung und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Demonstration, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, teilweise in Zusammenarbeit mit anderen Gruppen, Begutachtung der Lösungen, Zusammenfassung der Ergebnisse, Gesprächsführung
Literatur:	Aktuell nach Lehrinhalten.

Modulbezeichnung:	Secure Software Engineering
Kürzel	MIN-SSE
Studiensemester:	1., 2. oder 3., in der Regel Jährlich
Modulverantwortliche(r):	Holger Peine
Dozent(in):	Holger Peine, Stefan Wohlfeil
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 1.-3. Semester, Modul im Schwerpunkt IT-Sicherheit
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	IT-Sicherheit 1, gute Kenntnisse der typischen Aktivitäten eines Softwareentwicklungsprozesses (einschließlich Programmierung in Java und C), Kenntnisse der Funktion verteilter Anwendungen
Angestrebte Lernergebnisse:	<ul style="list-style-type: none"> • Analysekompetenzen und Technologische Kompetenzen: Studierende können die wichtigsten Klassen von Software-Sicherheitsschwachstellen in realen Systemen erkennen und ihr Schadenspotential einschätzen • Design-, Realisierungs- und Methodenkompetenzen: Studierende beherrschen Techniken und Methoden für verschiedene Aktivitäten im Softwareentwicklungsprozess, um dort Schwachstellen zu vermeiden oder aufzudecken
Inhalt:	<ul style="list-style-type: none"> • Klassen von Sicherheitsschwachstellen in Software und ihre Ursachen • Erhebung und Formulierung von Sicherheitsanforderungen • Bedrohungen Finden, Einschätzen und Abwehren • Software-Sicherheit auf Architektur-Ebene • Software-Sicherheit auf Quelltext-Ebene • Testen und Inspektion auf Sicherheitsschwachstellen • Benutzerfreundlichkeit und Prozessmodelle bzgl. Sicherheit
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, experimentelle Projektarbeit, Gesprächsführung
Literatur:	Michael Howard, David LeBlanc: Writing Secure Code, 2 nd ed. 2003 Gary McGraw: Software Security - Building Security In, 2006 John Viega, Gary McGraw: Building Secure Software, 2001 Michael Howard, Steve Lipner: The Security Development Lifecycle, 2006 Sachar Paulus: Basiswissen Sichere Software, 2011.

5. Schwerpunkt Informationssysteme

Modulbezeichnung:	Datenbankparadigmen
Kürzel	MIN-DBP
Studiensemester:	1. oder 3. Semester (WiSe), Turnus: Jährlich
Modulverantwortliche(r):	Felix Heine
Dozent(in):	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner, Arne Koschel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 1.-3. Semester, Modul im Schwerpunkt Informationssysteme
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Keine
Angestrebte Lernergebnisse:	<p>Analyse- und Design-Kompetenz: Verwendungsmöglichkeiten alternativer DB-Paradigmen erkennen; das richtige Modell bzw. das richtige System für das gegebene Problem erkennen; Anwendungen auf dieser Basis konzeptionieren können.</p> <p>Algorithmische Kompetenzen: moderne DB-Algorithmen, insbesondere Speicherorganisation und Anfrageverarbeitung, kennen und beurteilen können. Vor- und Nachteile der Algorithmen für gegebene Problemstellungen erkennen.</p> <p>Technologische Kompetenzen: Abfragesprachen beherrschen, Zugriffsschnittstellen beherrschen.</p>
Inhalt:	<p>Schwerpunktmäßig werden in dem Modul Alternativen zu relationalen Datenbank behandelt, sowie fortgeschrittene Erweiterungen relationaler Datenbanken. Dabei wird sowohl auf die Datenmodellierung und die Abfragesprachen als auch auf Implementierungsaspekte wie z.B. Anfragebearbeitung eingegangen. Ggf. wird auch die Anwendungsprogrammierung auf Basis dieser DB-Paradigmen betrachtet. Themen können dabei z.B. sein:</p> <ul style="list-style-type: none"> • Objektrelationale- und Objektorientierte Datenbanken • XML-Datenbanken und Abfragesprachen • Graphdatenbanken • Key/Value-Datenbanken • Dokumenten-Orientierte Datenbanken • Spaltenorientierte Speicherung von Daten • In-Memory-Datenbanken • Embedded Datenbanken
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation & Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit</p>
Literatur:	<p>Edlich, Friedland, Hampe, Brauer: „N*SQL. Einstieg in die Welt Nichtrelationaler Web 2.0 Datenbanken“, Hanser</p> <p>Dietrich, Urban: „An Advanced Course in Database Systems: Beyond Relational Databases“, Prentice Hall</p> <p>Weitere aktuelle Quellen zu den Vorlesungsinhalten.</p>

Modulbezeichnung:	Verteilte Informationssysteme
Kürzel	MIN-VIS
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Arne Koschel
Dozent(in):	Arne Koschel, Felix Heine, Matthias Hovestadt
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 1.-3. Semester, Modul im Schwerpunkt Informationssysteme
Lehrform/SWS:	4 SWS: Vorlesung, Projekt und Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Abgeschlossenes Bachelor-Studium mit Informatikanteil, der mindestens den Programmieren-, Informationssysteme- und Betriebssysteme/Netze-Veranstaltungen vergleichbar ist.
Angestrebte Lernergebnisse:	<p>Analyse und Design-Kompetenzen: Die Studierenden sollen Informationssysteme hinsichtlich behandelter Kriterien beurteilen bzw. entwerfen und umsetzen können. Muster auswählen und anwenden können.</p> <p>Algorithmische Kompetenzen: Verteilte Algorithmen verstehen und einsetzen können.</p> <p>Technologische Kompetenzen: Die Studierenden sollen Kenntnisse über Konzepte und Technologien zu Aufbau, Betrieb und Nutzung verteilter Informationssysteme erlangen. Im Kontext verschiedener Architekturmodelle sollen hierzu für verteilte Informationssysteme notwendige Aspekte betrachtet werden.</p>
Inhalt:	<p>In diesem Modul werden fortgeschrittene Kompetenzen des Aufbaus und Betriebs verteilter Informationssysteme vermittelt. Hierzu werden Kernanforderungen an solche Informationssysteme beleuchtet und Verfahren zur Erfüllung dieser Anforderungen vorgestellt. Theoretische Konzepte werden hierbei im Kontext existierender Softwaresysteme erlernt und angewandt. Wichtige Aspekte hierbei betreffen:</p> <ul style="list-style-type: none"> • Grundlegende Architekturkonzepte zur Realisierung verteilter Informationssysteme (Cluster, Cloud, ...); • Datenintegration (fDBMS, Datenintegrationstools, ...); • Aspekte der Parallelität (Skalierbarkeit, Lastbalancierung, Konsistenz, weiterführenden Transaktionskonzepte, ...); • und weitere aktuelle Themen wie z.B. BigData
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation & Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit</p>
Literatur:	<p>Conrad, S., Hasselbring, W., Koschel, A., Tritsch, R.: „Enterprise Application Integration“, Spektrum/Elsevier.</p> <p>Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., Koschel, A. „Systemarchitekturen für verteilte Anwendungen“, Hanser.</p> <p>Tanenbaum, A.: „Verteilte Systeme: Prinzipien und Paradigmen“, Pearson.</p> <p>Weitere aktuelle Quellen zu den Vorlesungsinhalten.</p>

Modulbezeichnung:	Datenanalyse
Kürzel	MIN-DA
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Felix Heine
Dozent(in):	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner, Arne Koschel
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 1.-3. Semester, Modul im Schwerpunkt Informationssysteme
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Keine
Angestrebte Lernergebnisse:	Analyse- und Design-Kompetenz: Verfahren zum Auffinden und Analysieren von Daten kennen und einsetzen können. Anhand des gegebenen Problems das richtige Verfahren oder die richtige Kombination aus Verfahren auswählen können. Architektur eines Data Warehouse verstehen. Algorithmische Kompetenzen: Analyseverfahren verstehen, Vorbedingungen, Einsatzgebiete sowie Vor- und Nachteile kennen. Technologische Kompetenzen: Beispiel-Systeme wie z.B. Rapid-Miner nutzen können.
Inhalt:	Es werden verschiedene Herangehensweisen zur Organisation und Analyse großer Datenbestände beschrieben. Dabei stehen folgende Bereiche im Vordergrund: <ul style="list-style-type: none"> • Data Warehousing • Data Mining • Datenqualität • Information Retrieval Anhand von exemplarischen Systemen wie z.B. MS SQL Server, Rapid Miner, etc. werden die Verfahren in den Übungen praktisch eingesetzt.
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation & Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit
Literatur:	Bauer, Günzel: "Data Warehouse Systeme", dpunkt.verlag Tan, Steinbach, Kumar: „Introduction to Data Mining“, Addison Wesley Manning, Prabhakar Raghavan, Schütze: "Introduction to Information Retrieval", Cambridge University Press Schmitt: "Ähnlichkeitssuche in Multimedia-Datenbanken", Oldenbourg Weitere aktuelle Quellen zu den Vorlesungsinhalten.

Modulbezeichnung:	Geo- und Multimedia-Informationssysteme
Kürzel:	MIN-GMI
Studiensemester:	2.(SoSe) oder 3. Semester (WiSe), Turnus: in der Regel alle zwei Jahre
Modulverantwortliche(r):	Carsten Kleiner
Dozent(in):	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, 2.-3. Semester, Modul im Schwerpunkt Informationssysteme
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Kenntnisse aus einem Bachelorstudium im Umfang der Module IS1, IS2
Angestrebte Lernergebnisse:	<p>Analyse-, Design- und Realisierungskompetenzen: Stärken und Schwächen verschiedener Möglichkeiten zur persistenten Speicherung zwei- und drei-dimensionaler Daten kennenlernen und bewerten können; Modellierung und Anfragetypen für Multimediadaten kennenlernen, bewerten und vergleichen können; für genannte Daten beste persistente Speichervariante in bestimmtem Anwendungsfeld auswählen und realisieren können</p> <p>Algorithmische Kompetenzen: Verschiedene geometrische Algorithmen verstehen, deren Laufzeiten und Einsatzmöglichkeiten bewerten können sowie diese effizient implementieren können.</p> <p>Technologische Kompetenzen: Speicherung und Abfrage räumlicher Daten unterschiedlicher Modelle in Datenbanksystemen beherrschen und einsetzen können (Oracle Spatial, PostGIS); Speicherung und Abfrage von Multimediadaten in Datenbanksystemen beherrschen und einsetzen können (Oracle, Postgres); alternative Datenbankparadigmen für diese Datentypen kennen.</p>
Inhalt:	Räumliche Datentypen und geometrische Algorithmen, Datenbank- und Informationssysteme für räumliche Daten (z.B. Geoinformationssysteme), drei-dimensionale Datenmodelle und deren Speicherung, Persistente Speicherung und Abfrage von Multimediadaten Zugriffsstrukturen für spezielle Anwendungsbereiche.
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Beispielaufgaben, Diskussion, Visualisierung Übung: Bearbeitung von Übungsaufgaben, Besprechung der Ergebnisse (Tafel, Gespräch), Bearbeitung einer Projektaufgabe mit Präsentation der Ergebnisse, Erstellung eines schriftlichen Berichts
Literatur:	De Berg, van Kreveld, Overmars: Computational Geometry, Springer Rigaux, Scholl, Voissard: Spatial Databases with application to GIS, Morgan Kaufmann T. Brinkhoff: Geodatenbanksysteme in Theorie und Praxis, Wichmann I. Schmitt: Ähnlichkeitssuche in Multimedia-Datenbanken, Oldenbg.

6. Schwerpunkt Softwaretechnik

Modulbezeichnung:	Programmierparadigmen
Kürzel	MIN-PPD
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Holger Peine
Dozent(in):	Holger Peine, Robert Garmann, Carsten Kleiner, Stefan Wohlfeil
Sprache:	Deutsch oder Englisch
Einordnung Curriculum	Master, Wahlpflichtfach, 1.-3. Semester, Schwerpunkt Softwaretechnik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen:	Keine
Empfohlen:	Gute Kenntnisse in objektorientierter Programmierung
Angestrebte Lernergebnisse:	<ul style="list-style-type: none"> • Analyse-, Design- und Realisierungs-Kompetenzen: Einordnen und Einschätzen der Ausdrucksmittel von Programmiersprachen • Methodenkompetenzen: Erweiterung der Problemlösefähigkeiten durch Beherrschung alternativer Denkansätze, Vertiefung des Verständnisses der objektorientierten Programmierung • Technologische Kompetenzen: Erlernen einer funktionalen Programmiersprache (z. B. Scala) und (in begrenztem Umfang) einer logischen Programmiersprache (z. B. Prolog)
Inhalt:	<ul style="list-style-type: none"> • Programmiersprachen und -paradigmen (kurz) • Imperatives Paradigma (kurz) • Objektorientiertes Paradigma (mittellang) <ul style="list-style-type: none"> ○ Kapselung, Vererbung ○ Polymorphismus, Substitutionsprinzip, Varianz • Funktionales Paradigma (lang) <ul style="list-style-type: none"> ○ Unveränderlichkeit und Seiteneffektfreiheit ○ Funktionsobjekte, Funktionen höherer Ordnung, Closures ○ Verzögerte Auswertung, Continuations, funktionaler Zustand • Logisches Paradigma (mittellang) <ul style="list-style-type: none"> ○ Aussagenlogik, Prädikatenlogik, Hornlogik, Resolution ○ Unifikation, negation as failure, prozedurale Semantik ○ Optional: Typisierung, Integration mit anderen Paradigmen
Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Experimentelle Arbeit (einzeln und in Gruppen), Diskussion
Literatur:	Michael L. Scott: Programming Language Pragmatics Bertrand Meyer: Object oriented software construction P. Chiusano, R. Bjarnason: Functional Programming in Scala Ivan Bratko: PROLOG Programming for Artificial Intelligence.

Modulbezeichnung:	Intelligente Systeme
Kürzel:	MIN-IS
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Ralf Bruns
Dozent(in):	Ralf Bruns, Jürgen Dunkel, Stefan Wohlfeil
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Softwarearchitektur
Angestrebte Lernergebnisse:	Technologische Kompetenz: Erwerb tiefgehender Fachkenntnisse im Gebiet Intelligente Systeme (Künstliche Intelligenz), Verständnis entwickeln für anwendbare Techniken und Methoden; Nutzen und Grenzen intelligenter Technologien einschätzen können Methodenkompetenz: Anwendung und kritische Beurteilung innovativer Methoden im Bereich Intelligenter Systeme Analyse-, Design- und Realisierungskompetenz: Fähigkeit, Probleme aus einem neuen Anwendungsbereich zu formulieren, zu strukturieren und mit Methoden intelligenter Systeme zu lösen
Inhalt:	Theoretische Grundlagen und Konzepte für Entwurf und Entwicklung von intelligenten (Anwendungs-)Systemen, z.B. Einführung in Künstliche Intelligenz, Multi-Agentensysteme, Inferenzalgorithmen, Semantic Web, Complex Event Processing Dabei sollen i.W. aktuelle Themen und Entwicklungen Berücksichtigung finden, insbesondere aus dem Bereich webbasierter und mobiler Systeme. Forschungsnahe Fragestellungen legen dabei ggf. eine Basis für die Masterarbeit.
Studien-/ Prüfungsleistungen:	Mündliche Prüfung und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
Literatur:	Aktuell nach Lehrinhalten

Modulbezeichnung:	Fortgeschrittene Aspekte verteilter Systeme
Kürzel	MIN-FAVS
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Arne Koschel
Dozent(in):	Arne Koschel, Felix Heine, Matthias Hovestadt
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
Lehrform/SWS:	4 SWS: Vorlesung, Projekt und Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	Abgeschlossenes Bachelor-Studium mit Informatikanteil, der mindestens den Programmieren-, Informationssysteme- und Betriebssysteme/Netze-Veranstaltungen vergleichbar ist.
Angestrebte Lernergebnisse:	<p>Analyse und Design-Kompetenzen: Herausforderungen verschiedener Formen verteilter Systeme einschätzen und ihnen mithilfe geeigneter Verfahren und Technologien begegnen. Muster auswählen und anwenden können.</p> <p>Algorithmische Kompetenzen: Verteilte Algorithmen verstehen und einsetzen können.</p> <p>Technologische Kompetenzen: Die Studierenden sollen vertiefte Kenntnisse über Herausforderungen, Verfahren und aktuelle Forschungsergebnisse verschiedener Ansätze verteilter Systeme erlangen.</p>
Inhalt:	In diesem Modul werden fortgeschrittene Kompetenzen zu Architektur und Funktionalität verschiedener Formen verteilter Systemen vermittelt. Hierzu werden zunächst grundlegende Modelle für verteilte Systeme - z.B. Cloud Computing, P2P, SOA - beleuchtet und auf charakteristische Eigenschaften in Architektur, Verwaltung und Laufzeitverhalten sowie auf die Themenbereiche Parallelität und Fehlertoleranz eingegangen. Ausgewählte theoretische Konzepte werden mithilfe verschiedener Softwaresysteme praktisch umgesetzt. Hinzu kommen aktuelle Forschungsergebnisse und Fragestellungen.
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation & Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit</p>
Literatur:	<p>Conrad, S., Hasselbring, W., Koschel, A., Tritsch, R.: „Enterprise Application Integration“, Spektrum/Elsevier.</p> <p>Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., Koschel, A. „Systemarchitekturen für verteilte Anwendungen“, Hanser.</p> <p>Tanenbaum, A.: „Verteilte Systeme: Prinzipien und Paradigmen“, Pearson.</p> <p>Weitere aktuelle Quellen zu den Vorlesungsinhalten.</p>

Modulbezeichnung:	Software-Test und Requirements
Kürzel:	MIN-STR
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Robert Garmann
Dozent(in):	Robert Garmann, Arne Koschel, Holger Peine
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
Lehrform/SWS:	4 SWS: Vorlesung, Projekt und Übung
Arbeitsaufwand:	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Voraussetzungen:	Praktische Projekterfahrungen in allen Phasen des Softwareentwicklungsprozesses, insb. bei der Erstellung eines Pflichtenheftes und der Erstellung von verschiedenen Testroutinen Kenntnis verschiedener Testverfahren (z. B. black box / white box) in den verschiedenen Teststufen (Komponenten- bis Abnahmetest) Kenntnis der Grundbegriffe des Projekt- und Qualitätsmanagements
Angestrebte Lernergebnisse:	Analyse- und Design-Kompetenzen: Die Studierenden kennen grundlegende Prinzipien und Verfahren der Anforderungsanalyse und können diese in einem konkreten Anforderungsumfeld anwenden. Sie besitzen einen detaillierten Einblick in alle wesentlichen Testverfahren und können für ein komplexes System Testfälle entwerfen und die erwartete Qualität nachweisen. Sie können ein System hinsichtlich seiner fachlichen und technischen Qualitätsmerkmale bewerten. Realisierungs-Kompetenz: Die Studierenden können technische Testaktivitäten durchführen sowie die notwendigen Werkzeuge und Techniken zum Erreichen der definierten Testziele implementieren.
Inhalt:	Requirements: System und Systemkontext, Anforderungen ermitteln, (natürlichsprachlich) dokumentieren und konsolidieren, Anforderungen prüfen und abstimmen Software-Test: Testprozess, Testverfahren (spezifikationsorientiert, strukturorientiert, fehlerbasiert, erfahrungsbasiert), Test der Softwareeigenschaften bei fachlichen und technischen Tests, Testwerkzeuge und Testautomatisierung
Studien-/ Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Diskussion, Aufsatzbesprechung Übung: Selbständige Bearbeitung von Aufgaben, experimentelle Projektarbeit, Referate, Gesprächsführung
Literatur:	Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering, dpunkt (2010) Andreas Spillner, Tilo Linz: Basiswissen Softwaretest, dpunkt (2005) Graham Bath, Judy McKay: Praxiswissen Softwaretest, dpunkt (2011)

Modulbezeichnung:	Secure Software Engineering
Kürzel	MIN-SSE
Studiensemester:	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
Modulverantwortliche(r):	Holger Peine
Dozent(in):	Holger Peine, Stefan Wohlfeil
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
Lehrform/SWS:	2 SWS Vorlesung, 2 SWS Übung
Arbeitsaufwand:	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
Kreditpunkte:	6 CP (= 180 h)
Voraussetzungen nach Prüfungsordnung	Keine
Empfohlene Voraussetzungen:	IT-Sicherheit 1, gute Kenntnisse der typischen Aktivitäten eines Softwareentwicklungsprozesses (einschließlich Programmierung in Java und C), Kenntnisse der Funktion verteilter Anwendungen
Angestrebte Lernergebnisse:	<ul style="list-style-type: none"> • Analysekompetenzen und Technologische Kompetenzen: Studierende können die wichtigsten Klassen von Software-Sicherheitsschwachstellen in realen Systemen erkennen und ihr Schadenspotential einschätzen • Design-, Realisierungs- und Methodenkompetenzen: Studierende beherrschen Techniken und Methoden für verschiedene Aktivitäten im Software-Entwicklungsprozess, um dort Schwachstellen zu vermeiden oder aufzudecken
Inhalt:	<ul style="list-style-type: none"> • Klassen von Sicherheitsschwachstellen in Software und ihre Ursachen • Erhebung und Formulierung von Sicherheitsanforderungen • Bedrohungen Finden, Einschätzen und Abwehren • Software-Sicherheit auf Architektur-Ebene • Software-Sicherheit auf Quelltext-Ebene • Testen und Inspektion auf Sicherheitsschwachstellen • Benutzerfreundlichkeit und Prozessmodelle bzgl. Sicherheit
Studien-/Prüfungsleistungen:	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
Medienformen:	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, experimentelle Projektarbeit, Gesprächsführung
Literatur:	Michael Howard, David LeBlanc: Writing Secure Code, 2 nd ed. 2003 Gary McGraw: Software Security - Building Security In, 2006 John Viega, Gary McGraw: Building Secure Software, 2001 Michael Howard, Steve Lipner: The Security Development Lifecycle, 2006 Sachar Paulus: Basiswissen Sichere Software, 2011.