



Modulhandbuch zum Studiengang

Informatik
(Bachelor of Science)

Fakultät Informatik

(Stand: 11.11.2021)

Inhalt

1. Studienabschnitt.....	1
Modul 1: Mathematik I	1
Lineare Algebra	1
Modul 2: Mathematik II	3
Analysis	3
Modul 3: Mathematik III	6
Numerische Mathematik	6
Modul 4: Statistik.....	8
Modul 5: Grundlagen der Informationsverarbeitung.....	10
Grundlagen der Informationsverarbeitung.....	10
Modul 6: Theoretische Informatik	12
Modul 7: Programmierung I.....	15
Einführung in die Programmierung.....	15
Datenstrukturen und Algorithmen.....	17
Modul 8: Programmierung II	18
Fortgeschrittene Techniken der Programmierung	18
Modul 9: Informationsmanagement	19
Information Engineering.....	19
Modul 10: Technische Grundlagen I.....	20
Digitaltechnik und IT-Mobiltechnik.....	20
Modul 11: Technische Grundlagen II.....	22
Mikroprozessortechnik & Eingebettete Systeme	22
Rechnerarchitektur.....	24
Modul 12: Wirtschaftswissenschaften.....	26
Wirtschaftswissenschaften.....	26
Modul 13: Fächerübergreifende Kompetenzen	28
Einführung in das wissenschaftliche Arbeiten	28
2. Studienabschnitt.....	29
Pflichtmodule.....	29
Wissensverarbeitung	29
Software Engineering.....	30
Computergestützte Intelligenz.....	31
Datenbanksysteme	33
Rechnernetze	35
Betriebssysteme	36
Graphische Datenverarbeitung (GDV)	37
Projektmanagement.....	39
Paradigmen der Softwareentwicklung.....	41
IT-Sicherheit	43
Wahlmodule	45
Schlüsselqualifikation.....	45
Unternehmensplanspiel	46
Wahlpflichtmodule	47
Compilerbau	47
Data-Warehouse-Systeme und NoSQL-Systeme	48
Datenbankprogrammierung mit PL/SQL	50
Deep Learning Architectures.....	52
Grundlagen der Webtechnik	53
IoT-Security	55
Mobile Anwendungsentwicklung	56
Mobile Web-Anwendungen.....	58
Netzwerkplanung und -konfiguration.....	60

Scientific Programming	61
Softwarequalität	63
Spieleentwicklung	65
Vertiefung Software-Entwurf	66
XR Medien	67
Praxismodul	68
Bachelorarbeit	69

1. Studienabschnitt

Modul 1: Mathematik I

Modulname	Lineare Algebra
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Math. Gerd Recknagel
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • die formale mathematische Notationen zu lesen zu interpretieren und zu verstehen. • lineare Gleichungssysteme und Eigenwertprobleme zu lösen, Determinanten, Matrizenprodukte, Faktorisierungen von Matrizen und Projektionen zu berechnen sowie kleinste Quadrate Approximationen durchzuführen. • Grundlagenwissen in angewandter höherer Mathematik sowie geeignete Methoden der linearen Algebra bei wissenschaftlichen, technischen bzw. wirtschaftlichen Fragestellungen anzuwenden. • mathematische Modellierungen zur Lösung von Problemen der Berufspraxis einzusetzen. • mathematische Problemstellungen zu analysieren und zu strukturieren sowie diese unter Anwendung der eingeführten Techniken und Methoden zu lösen. • mathematische Denkweisen auf andere Gebiete zu übertragen und abstrakte Zusammenhänge zu verstehen. • eigene Denkansätze und Lösungen zu entwickeln.
Modulinhalte	<ul style="list-style-type: none"> • Vektorrechnung <ul style="list-style-type: none"> ○ Was sind Vektoren? ○ Zweidimensionale Vektoren (Definition, geometrische Deutung, Arithmetik, Rechenregeln, Linearkombinationen, Skalarprodukt, Winkel zwischen Vektoren) ○ Drei- und mehrdimensionale Vektoren (Definition, geometrische Deutung, Arithmetik, Rechenregeln, Linearkombinationen, Skalarprodukt, Winkel zwischen Vektoren) • Lineare Gleichungssysteme mit zwei Unbekannten <ul style="list-style-type: none"> ○ Sichtweisen ○ Eliminationsverfahren ○ Lösbarkeit • Lineare Gleichungssysteme mit m Gleichungen und n Unbekannten... <ul style="list-style-type: none"> ○ Definition, Äquivalente Umformungen, Lösbarkeit ○ Gaußsches Eliminationsverfahren • Matrizen/Matrizenrechnung <ul style="list-style-type: none"> ○ Einführung und Definitionen ○ Rechenregeln für Matrixoperationen ○ Multiplikation von Matrizen ○ Inverse Matrizen, Gauß-Jordan-Verfahren ○ Faktorisierung einer Matrix ○ symmetrische Matrizen • Vektorräume und Untervektorräume <ul style="list-style-type: none"> ○ Räume von Vektoren, Spaltenraum einer Matrix ○ Kern und Rang einer Matrix ○ vollständige Lösung eines linearen Gleichungssystems ○ Lineare Unabhängigkeit, Basis und Dimension ○ Dimension der Unterräume einer Matrix, Fundamentalsatz der Linearen Algebra • Orthogonalität

	<ul style="list-style-type: none"> ○ Orthogonalität der Unterräume einer Matrix, Fundamentalsatz der Linearen Algebra ○ Orthogonale Projektion ○ Kleinste Quadrate Approximation ○ Orthonormale Basen, Gram-Schmidt-Verfahren ● Determinanten <ul style="list-style-type: none"> ○ Eigenschaften von Determinanten ○ Berechnung der Determinante ○ Volumen geometrischer Körper, Kreuzprodukt ● Eigenwerte und Eigenvektoren <ul style="list-style-type: none"> ○ Eigenwert, Eigenvektor ○ Lösen von Eigenwertproblemen, charakteristisches Polynom ○ Diagonalisierung einer Matrix, Potenzen von Matrizen ○ Eigenwerte und Eigenvektoren symmetrischer Matrizen
Lehrformen	Vorlesungen (3 SWS), Übungen Besprechung der Lösung der Übungsaufgaben (1 SWS) und Selbststudium anhand der vorgegebenen Übungsaufgaben
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Den Studierenden werden zu Beginn der Veranstaltung ein Lehrbrief inkl. umfangreicher Übungsaufgaben sowie themenspezifische Dokumente in studip zur Verfügung gestellt.</p> <p>Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> ● Strang, G., Lineare Algebra, Wellesley-Cambridge Press, Wellesley ● Manteuffel, K., Lineare Algebra, Teubner Verlag, Leipzig ● Pforr, E., Oehlschlaegel, L., Seltmann, G., Übungsaufgaben zur linearen Algebra und linearen Optimierung., Teubner Verlag, Leipzig ● Leupold, W., u.a., Mathematik ein Studienbuch für Ingenieure, Fachbuchverlag Leipzig – Köln ● Beutelspacher, A., Lineare Algebra, Vieweg Verlag, Braunschweig/Wiesbaden... ● Preuß, W., Lehr- u Übungsbuch Mathematik für Informatiker, Fachbuchverlag, Leipzig
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Informatik“, „Multimedia-Marketing“, „Verwaltungsinformatik“ und „Wirtschaftsinformatik“ ein Pflichtfach.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium: 60 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 120 Minuten
Semester	1. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 2: Mathematik II

Modulname	Analysis
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Math. Gerd Recknagel
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • die formale mathematische Notationen zu lesen zu interpretieren und zu verstehen. • den axiomatischen Aufbau einer Wissenschaft am Beispiel der Analysis zu verstehen, sind mit den fundiert bewiesenen Aufbau von wissenschaftlichen Aussagen vertraut und können dieses Vorgehen auf neue Probleme übertragen. • im Bereich der komplexen Zahlen Berechnungen nachzuvollziehen sowie selbst durchzuführen. • Grenzübergänge und Grenzwertbetrachtungen durchzuführen sowie das Konvergenzverhalten von Folgen und Reihen zu untersuchen. • reelle Funktionen vollständig zu diskutieren und ihren Verlauf grafisch darzustellen. • Extremwertprobleme zur Optimierung von Größen in Wissenschaft, Technik und Ökonomie zu analysieren und zu strukturieren sowie diese unter Anwendung der eingeführten Techniken und Methoden zu lösen. • Grundlegende Integrale zu berechnen sowie diese Technik für die Flächen- und Volumenberechnung einzusetzen. • mathematische Denkweisen auf andere Gebiete zu übertragen und abstrakte Zusammenhänge zu verstehen. • eigene Denkansätze und Lösungen zu entwickeln.
Modulinhalte	<ul style="list-style-type: none"> • Mengenlehre <ul style="list-style-type: none"> ○ Grundbegriffe ○ Mengenrelationen, Mengenoperationen ○ Rechengesetze der Mengenoperationen ○ kartesisches Produkt • Reelle Zahlen <ul style="list-style-type: none"> ○ Zahlenbereiche ○ Ordnungsstruktur der reellen Zahlen, absoluter Betrag, Intervalle ○ Potenzen, Wurzeln, Logarithmen, binomische Formeln, Summen und Produkte ○ Grenzen von Mengen ○ Gleichungen und Ungleichungen, Arten von Gleichungen und Ungleichungen sowie Lösungsstrategien • Komplexe Zahlen <ul style="list-style-type: none"> ○ Einführung, Grundbegriffe, Grundrechenarten ○ Geometrische Darstellung (Normal-, Polarkoordinaten-, trigonometrische sowie trigonometrische Darstellung), Eulersche Formel ○ geometrische Interpretation der Rechenoperationen mit komplexen Zahlen ○ Multiplikation, Division, Potenzieren, Radizieren, Logarithmieren in exponentieller Darstellung, Fundamentalsatz der Algebra • Funktionen <ul style="list-style-type: none"> ○ Abbildungen ○ Funktionsbegriff, Darstellung, Operationen, Symmetrie, Monotonie, Periodizität, Umkehrfunktion ○ Spezielle Funktionen (ganz rationale, gebrochen rationale, Potenz-, Wurzel-, Exponential-, Logarithmus-, trigonometrische, zyklometrische Funktionen, Additionstheoreme) • Zahlenfolgen

	<ul style="list-style-type: none"> ○ Definition und allg. Eigenschaften, Zuordnungsvorschriften, Monotonie, Beschränktheit ○ arithmetische und geometrische Folgen ○ Grenzwert von Zahlenfolgen, Grenzwertsätze ○ spezielle Zahlenfolgen, endliche arithmetische und geometrische Reihen, Beweisverfahren der vollständigen Induktion ● Unendliche Reihen <ul style="list-style-type: none"> ○ Definitionen, einfache Beispiele, Konvergenz, Vergleichskriterien ○ Wurzel- und Quotientenkriterium ○ Potenzreihen, Konvergenzverhalten ○ Umordnung von Reihen ● Funktionen – Stetigkeit und Grenzwerte <ul style="list-style-type: none"> ○ Stetigkeit einer Funktion ○ Grenzwerte von Funktionen, Grenzwertsätze ○ Eigenschaften stetiger Funktionen, Nullstellensatz ● Differentialrechnung <ul style="list-style-type: none"> ○ Ableitung einer Funktion, Differentialquotient, Differenzierbarkeit ○ Differentiationsregeln, Ableitung von Grundfunktionen, Tangente an eine Kurve ○ Mittelwertsatz der Differentialrechnung, Grenzwertberechnung ○ Kurvendiskussion, Monotonie, Krümmung, Extrema ○ Extremwertaufgaben ● Integralrechnung <ul style="list-style-type: none"> ○ Integralbegriff und Integrierbarkeit, unbestimmtes Integral, Grundintegrale ○ bestimmtes Integral, Fläche unter einer Kurve, Eigenschaften integrierbarer Funktionen ○ Hauptsatz der Differential und Integralrechnung, uneigentliche Integrale ○ Integrationsverfahren, Substitution, partielle Integration, Partialbruchzerlegung ○ Anwendung der Integralrechnung, Flächen- und Volumenberechnung
Lehrformen	Vorlesungen (2 SWS), Übungen Besprechung der Lösung der Übungsaufgaben (1 SWS) und Selbststudium anhand der vorgegebenen Übungsaufgaben
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Den Studierenden werden zu Beginn der Veranstaltung umfangreiche Übungsaufgaben und themenspezifische Dokumente in studip zur Verfügung gestellt.</p> <p>Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> ● Leupold, W.: Mathematik. Fachbuchverlag, Leipzig ● Heuser, H.: Lehrbuch der Analysis. Teubner ● Preuß, W. : Mathematik für Informatiker. Fachbuchverlag, Leipzig ● Königsberger, K.: Analysis. Springer, Berlin ● Teschl, G.: Mathematik für Informatiker. Springer, Berlin
Lehrbriefautor	
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Informatik“, „Verwaltungsinformatik“ und „Wirtschaftsinformatik“ ein Pflichtfach.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 45 Stunden; Selbststudium: 30 Stunden; Prüfung und Prüfungsvorbereitung: 15 Stunden
ECTS und Gewichtung der	3 ECTS-Credits

Note in der Gesamtnote	Gewichtung: 3/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 90 Minuten
Semester	2. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 3: Mathematik III

Modulname	Numerische Mathematik
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Math. Gerd Recknagel
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • grundlegende numerische Methoden bei der Lösung wissenschaftlicher, technischer und ökonomischer Fragestellungen insbesondere im Bereich der Modellierung und Simulation anzuwenden. • praktische Probleme am Rechner die sich aus den Hardwarebeschränkungen ergeben einzuschätzen und zu bewerten. • kleinere Berechnungsalgorithmen mit Matlab zu entwerfen und als neue Funktionen umzusetzen. Dabei beachten Sie die theoretischen Grundlagen und wenden diese bei der Umsetzung an. • die verschiedenen Typen linearer Gleichungssysteme unter Anwendung der geeigneten Verfahren und Algorithmen zu lösen, die Lösung anhand vorgegebener Kriterien zu optimieren sowie eine Fehleranalyse zur Bewertung der Lösung durchzuführen. • grundlegende Verfahren und Algorithmen zur Lösung spezieller Aufgabenklassen auf unbekannte Probleme zu übertragen und anzuwenden sowie diese darauf anzupassen. • Nullstellen reeller Funktionen mit geeigneten Verfahren zu ermitteln.
Modulinhalte	<ul style="list-style-type: none"> • Einführung <ul style="list-style-type: none"> ○ Aufgaben, Charakterisierung, Fehlerquellen • Computerarithmetik <ul style="list-style-type: none"> ○ Zahlendarstellung, Gleitkommadarstellung, Maschinenzahlen ○ IEEE-Normen ○ Runden, Rundungsfehler ○ Rechnen mit Maschinenzahlen, Gleitpunktrechnung • Fehleranalyse <ul style="list-style-type: none"> ○ absoluter und relativer Fehler ○ Fehlerfortpflanzung bei arithmetischen Operationen ○ Auslöschung ○ Rundungsfehleranalyse, Vorwärts- und Rückwärtsanalyse, Beurteilung numerischer Algorithmen, Kondition, Stabilität • Vektor- und Matrizennormen <ul style="list-style-type: none"> ○ Vektornormen ○ Matrixnorm • Numerische Betrachtung linearer Gleichungssysteme <ul style="list-style-type: none"> ○ Problemstellung, Lösungsverhalten, Matrizentypen Lösungsmethoden ○ Lösungsverfahren und Lösbarkeit ○ Störungstheorie, Kondition, Genauigkeit ○ Direkte Verfahren, partielle Pivotisierung ○ Iterative Verfahren, Jacobi-Verfahren, Gauß-Seidel-Verfahren, Konvergenzkriterien, Fehlerabschätzung • Lineare Ausgleichsrechnung <ul style="list-style-type: none"> ○ Problemstellung ○ Überbestimmte lineare Gleichungssysteme ○ Orthogonalisierungsmethoden ○ Unterbestimmte lineare Gleichungssysteme ○ Rangdefekte lineare Gleichungssysteme ○ Lösung von Gleichungssystemen mit Matlab • Nullstellen reeller Funktionen <ul style="list-style-type: none"> ○ Problemstellung ○ Allgemeines Iterationsverfahren, Konvergenz,

	<p>Fehlerabschätzung, Konvergenzgeschwindigkeit und -beschleunigung</p> <ul style="list-style-type: none"> ○ Bisektionsverfahren ○ Newtonverfahren ○ Sekantenverfahren
Lehrformen	Vorlesungen (2 SWS), Übungen Besprechung der Lösung der Übungsaufgaben sowie Rechnerübungen mit Matlab (1 SWS) und Selbststudium anhand der vorgegebenen Übungsaufgaben
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Inhaltliche Voraussetzung ist das Modul „Lineare Algebra“.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Den Studierenden werden zu Beginn der Veranstaltung umfangreiche Übungsaufgaben und themenspezifische Dokumente in studip zur Verfügung gestellt.</p> <p>Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> • Gramlich, W.: Numerische Mathematik mit MATLAB, Dpunkt • Opfer, G.: Numerische Mathematik für Anfänger, Vieweg • Biran, Breiner: Matlab für Ingenieure, Addison-Wesley Press • Hämmerlin, Hoffmann: Numerische Mathematik, Springer-Verlag
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Pflichtfach.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 45 Stunden; Selbststudium: 30 Stunden; Prüfung und Prüfungsvorbereitung: 15 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	3 ECTS-Credits Gewichtung: 3/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 90 Minuten
Semester	2. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 4: Statistik

Modulname	Statistik
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. T. Heimrich (gehalten durch externen Lehrbeauftragten)
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • die spezielle Axiomatik der Stochastik und insbesondere der Statistik richtig einzuschätzen. • beschreibende Statistiken zu lesen, auszuwerten und korrekt zu deuten. • statistische Erhebungen selbst durchzuführen, zu analysieren und Zusammenhänge in den Datenerhebungen zu erkennen sowie Vorhersagen daraus abzuleiten. • statistische Maßzahlen zur Charakterisierung von Häufigkeitsverteilungen zu ermitteln und Schlüsse auf Grundgesamtheiten zu übertragen sowie Auswertungen durchzuführen. • statistische Verfahren und Vorgehensweisen sicher anzuwenden sowie Auswertungen rechnergestützt mit entsprechender Statistiksoftware durchzuführen. • Begriffe der Wahrscheinlichkeitsrechnung wie z.B. Ereignis, Wahrscheinlichkeit und Verteilungsfunktion anzuwenden und Bezüge zur Statistik herzustellen. • statistische Tests durchzuführen, deren Ergebnisse zu analysieren und korrekte Schlussfolgerungen daraus zu ziehen.
Modulinhalte	<ul style="list-style-type: none"> • Deskriptive Statistik <ul style="list-style-type: none"> ○ Grundlagen der Statistik (Aufgaben, beschreibende Statistik, schließende Statistik) ○ Datenerhebungen (Abgrenzung, Form, Umfang) ○ Klassifikation von Merkmalstypen (qualitativ, intensitätsmäßig, stetig/diskret quantitativ), statistische Messskalen (nominalskaliert, ordinalskaliert, kardinalskaliert) • Methoden zur Auswertung eindimensionaler Daten <ul style="list-style-type: none"> ○ Häufigkeiten (absolut, relativ, Häufigkeitsverteilungen, Grafische Darstellung, Summenhäufigkeiten, Klasseneinteilung) ○ Lageparameter (Modalwert, Median, arithmetisches Mittel) ○ Streuungsparameter (Spannweite, empirische Varianz, Standardabweichung) • Methoden zur Auswertung zweidimensionaler Daten <ul style="list-style-type: none"> ○ Kreuztabellen und bedingte Häufigkeiten (absolute und relative Häufigkeit, empirische Abhängigkeit und Unabhängigkeit von Merkmalen) ○ Korrelationsrechnung (Korrelationskoeffizient nach Bravais und Pearson, Rangkorrelationskoeffizient nach Spearman) ○ Lineare Regression (Regressionskoeffizienten und -gerade, Prognosen, Bestimmtheitsmaß) • Wahrscheinlichkeitsrechnung <ul style="list-style-type: none"> ○ Zufällige Ereignisse und ihre Wahrscheinlichkeit <ul style="list-style-type: none"> ▪ Zufallsexperiment, Ereignis, Elementarereignis, Ereignisraum, sicheres Ereignis, unmögliches Ereignis, zusammengesetzte Ereignisse, Relationen und Operationen mit Ereignissen ▪ Wahrscheinlichkeitsbegriff, statistische, axiomatische und klassische Definition der Wahrscheinlichkeit ▪ bedingte Wahrscheinlichkeit, Multiplikationssatz, Satz von der totalen Wahrscheinlichkeit, Formel von Bayes, Diagnostik seltener Ereignisse ▪ Unabhängigkeit von Ereignissen, Multiplikationssatz für

	<ul style="list-style-type: none"> unabhängige Ereignisse <ul style="list-style-type: none"> ○ Zufallsgrößen und Verteilungsfunktionen <ul style="list-style-type: none"> ▪ Verteilungsfunktion, Dichtefunktion, Parameter von Verteilungsfunktionen, Erwartungswert, Varianz ▪ Standardnormalverteilung (Gaußsche Glockenkurve) ▪ Normalverteilung mit Parameter μ und σ^2, Sigmaregeln ▪ Binomialverteilung ▪ Hypergeometrische Verteilung • Schließende Statistik <ul style="list-style-type: none"> ○ Einführung <ul style="list-style-type: none"> ▪ Grundlagen (Grundgesamtheit, Stichprobe, t-Verteilung) ▪ statistische Testtheorie (Nullhypothese, Alternativhypothese, Signifikanzniveau) ▪ Fehler 1. und 2. Art ○ Statistische Testverfahren <ul style="list-style-type: none"> ▪ Anpassungstest (Prüfung Erwartungswert bei bekannter und unbekannter Varianz) ▪ Unterschiedstests für unabhängige Stichproben (doppelter t-Test, Chi-Quadrat-Test) ▪ Unterschiedstests für abhängige Stichproben (t-Test, Wilcoxon-Test)
Lehrformen	Vorlesungen (2 SWS), Übungen Besprechung der Lösung der Übungsaufgaben sowie Arbeit mit einer Auswahl an gängiger Statistik-Software (1 SWS) und Selbststudium anhand der vorgegebenen Übungsaufgaben
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Den Studierenden werden umfangreiche Übungsaufgaben und themenspezifische Dokumente in studip zur Verfügung gestellt.</p> <p>Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> • Hartung, Joachim., Statistik, Oldenbourg Verlag, München • Dufner, J., Jensen, U., Schumacher, E., Statistik mit SAS, Teubner-Verlag, Wiesbaden • Puhani, Josef, Statistik, LexikaVerlag, Eibelstadt • Matthäus, W-G, Schulze, J., Statistik mit EXCEL, Vieweg-Teubner, Wiesbaden
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Pflichtfach.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 45 Stunden; Selbststudium: 30 Stunden; Prüfung und Prüfungsvorbereitung: 15 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	3 ECTS-Credits Gewichtung: 3/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 90 Minuten
Semester	2. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 5: Grundlagen der Informationsverarbeitung

Modulname	Grundlagen der Informationsverarbeitung
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Thomas Heimrich
Qualifikationsziele	Die Studierenden sollen in der Lage sein, <ul style="list-style-type: none"> • Zahlencodierungen und Rechenoperationen für das Binär- und Hexadezimalsystem anzuwenden • logische Formeln zu vereinfachen und zu normalisieren • einfache Linux-Shell-Scripte zu schreiben • die Komplexität von Algorithmen abzuschätzen • elementare Datenstrukturen (Bäume) anzuwenden
Modulinhalte	<p>1 Informationsdarstellung</p> <ul style="list-style-type: none"> • Text, ASCII • Zahlendarstellung (Dual, Oktal, Hexadezimal) • Rechnen in verschiedenen Zahlensystemen <p>2 Aussagenlogik</p> <ul style="list-style-type: none"> • Syntax und Semantik der Aussagenlogik • elementare logische Formeln und Verknüpfungen • Wertetabellen • Umformen logischer Formeln • Konjunktive- und Disjunktive-Normalform <p>3 Linux</p> <ul style="list-style-type: none"> • elementarer Umgang mit Linux (grundlegende Befehle) • erstellen von Shell-Scripten mit den entsprechenden Programmierkonstrukten wie , z.B. Schleifen und Verzweigungen <p>4 Datenstrukturen und Komplexität</p> <ul style="list-style-type: none"> • Komplexität von Algorithmen • Bäume. Traversieren von Bäumen und HeapSort als Anwendung von Binärbäumen • Suchbäume • B-Bäume
Lehrformen	Vorlesungen (2 SWS), Übung (1 SWS) und Selbststudium anhand der vorgegebenen Literatur
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	Den Studierenden wird zu Beginn der Veranstaltung ein umfangreicher Foliensatz mit weiterführenden Hinweisen zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> • Heinz-Peter Gumm, Manfred Sommer: Einführung in die Informatik, Oldenbourg Verlag München • Helmut Herold, Bruno Lurz, Jürgen Wohlrab: Grundlagen der Informatik, Pearson Studium
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Pflichtmodul.

Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 45 Stunden; Selbststudium: 45 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	3 ECTS-Credits Gewichtung: 3/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) von 90 Minuten
Semester	1. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Pflichtmodul</i>
Besonderes	

Modul 6: Theoretische Informatik

Modulname	Theoretische Informatik
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Math. Gerd Recknagel
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • Eigenschaften wie z.B. Reflexivität, Symmetrie, Transitivität von Relationen auf speziellen Mengen zu analysieren, binäre Relationen mit Hilfe von Matrizen bzw. Graphen darzustellen sowie Produkte binärer Relationen zu bilden. Speziell sind sie in der Lage Potenzen von Alphabeten zu bilden und Sprachen über einem Alphabet zu beschreiben. • logische Operationen und Syllogismen anzuwenden, komplexe Zusammenhänge mittels logischer bzw. prädikatenlogischer Formeln darzustellen sowie die unterschiedlichen Beweisverfahren sicher anzuwenden. • die unterschiedlichen Algorithmenmodelle und die sich daraus abgeleiteten Programmierparadigmen zu unterscheiden sowie die Grenzen algorithmisch lösbarer und nicht lösbarer Probleme kennen. Weiterhin sind die Studierenden mit dem Halteproblem, dem Begriff der Rekursion und der Ackermann-Funktion vertraut. Sie kennen die grundlegenden Ideen der funktionalen Programmierung am Beispiel von Lisp und können rekursive Probleme selbständig lösen. • die einzelnen Automatenmodelle zu vergleichen und auf die entsprechenden grundlegenden Aufgabenklassen anzuwenden. Sie differenzieren deterministische und nichtdeterministische Automaten. • den deklarativen Charakter von Kalkülen zu verstehen. Sie können deren synthetische und analytische Funktionsweise erklären. • den Grammatikbegriff sicher anzuwenden. Sie verstehen den hierarchischen Aufbau von Automaten, formalen Sprachen und Grammatiken die Chomsky-Hierarchie. • die Äquivalenz endlicher Automaten regulärer Sprachen und linearer Grammatiken sowie regulärer Ausdrücke zu beschreiben sowie die unterschiedlichen Darstellungsformen ineinander zu überführen. • Kellerautomaten als Akzeptoren kontextfreier Sprachen zu abstrahieren und praktisch effiziente tabellengesteuerte Kellerautomaten zur Syntaxanalyse zu generieren. • auf der Grundlage von erworbenem langfristig gültigem Grundlagenwissen, das unabhängig von Methoden und Technik ist, dieses auf IT-Sachverhalte sowie IT-Probleme zu übertragen und anzuwenden um diese hinsichtlich ihrer Lösbarkeit und Komplexität zu analysieren und zu bewerten sowie neue Herangehensweisen und Lösungen zu entwerfen.
Modulinhalte	<ul style="list-style-type: none"> • Diskrete Mathematik <ul style="list-style-type: none"> ○ Mengenlehre ○ Relationen (Eigenschaften, Typen, Hüllenbildung), Funktionen ○ Alphabete, Worte, Sprachen • Logische Grundlagen <ul style="list-style-type: none"> ○ Aussagenlogik, Beweisverfahren ○ Logisches Schließen (Syllogismen, Resolution) ○ Prädikatenlogik (Quantoren, Schlussfolgern) • Formalisierung von Algorithmen durch Funktionen <ul style="list-style-type: none"> ○ Grundlagen der Algorithmenformalisierung, Vergleich der Algorithmenmodelle ○ Entscheidungsprobleme ○ PRF, Ackermann-Funktion, allg. und partiell rekursive Funktionen ○ Funktionale Programmierung, Rekursion, Halteproblem in Lisp • Automaten

	<ul style="list-style-type: none"> ○ endliche Automaten (Darstellung, Notation, Typologie, Minimierung) ○ Kellerautomaten (Notation, Darstellung, NKA) ○ Turingmaschinen (Aufbau, Notation, universelle TM, Halteproblem) ● Kalküle <ul style="list-style-type: none"> ○ wortzeugende Kalküle ○ wortverarbeitende Kalküle, KPK, NPK ○ Markow-Algorithmen ● Formale Sprachen und Grammatiken <ul style="list-style-type: none"> ○ Regelgrammatiken ○ Chomsky-Hierarchie ● Endliche Automaten, Reguläre Sprachen und lineare Grammatiken <ul style="list-style-type: none"> ○ Endliche Automaten - Akzeptoren regulärer Sprachen, Transformation von Grammatiken in Zustandsgraphen ○ Eigenschaften regulärer Sprachen, Pumping Lemma ○ Deterministische und nichtdeterministische endliche Automaten ○ Reguläre Ausdrücke, Ableitung RA aus DEA ● Kontextfreie Sprachen und Kellerautomaten <ul style="list-style-type: none"> ○ Definition kontextfreier Grammatiken ○ Ableitungs- bzw. Syntaxbaum ○ Kellerautomaten - Akzeptoren kontextfreier Sprachen ○ Top-down und bottom-up Parsing ○ Top-Down-Analyse, S-, Q-, LL(1)-Grammatik, Funktionen FIRST und FOLLOW, Entscheidungsmengen ○ Bottom-Up-Analyse, LR(k)-Parser
Lehrformen	Vorlesungen (3 SWS), Übungen Besprechung von Übungsaufgaben und Umsetzung von Konzepten am Rechner mit Lisp (1 SWS) und Selbststudium anhand der vorgegebenen Übungsaufgaben
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	Den Studierenden werden umfangreiche Übungsaufgaben und themenspezifische Dokumente in studip bzw. der Übung zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> ● Haggarty: Diskrete Mathematik für Informatiker, Pearson Studium ● Hopcroft / Motwani / Ullman: Einführung in Automatentheorie, formale Sprachen und Berechenbarkeit, Springer ● Asteroth / Beier: Theoretische Informatik, Pearson Studium ● Horn / Kerner / Forbrig: Lehr- und Übungsbuch der Informatik Bd. 2: Theorie der Informatik, Fachbuchverlag Leipzig
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium: 60 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 120 Minuten am Rechner
Semester	1. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung	Pflichtmodul

(Pflicht, Wahl, etc.)	
Besonderes	

Modul 7: Programmierung I

Modulname	Einführung in die Programmierung
Modulverantwortlicher/ Modulverantwortliche	<i>Prof. Hartmut Seichter, PhD</i>
Qualifikationsziele	<ul style="list-style-type: none"> • Studierende kennen grundlegende Konzepte der Digitaltechnik • Kursteilnehmer*Innen kennen Strategien Probleme algorithmisch zu lösen • Studierende erhalten vertiefende Kenntnisse in Bezug auf Datentypen, Datenfluss und Kontrollstrukturen • Kursteilnehmer*Innen können Theoreme und Konzepte praktisch in den Programmiersprachen C# und Python anwenden • Studierende bekommen einführende Kenntnisse zu programmiersprachenspezifischen Paradigmen
Modulinhalte	<ul style="list-style-type: none"> • Basis Datentypen und Kodierung • Einfache Datentypen, Variablen und Zuweisungen • Ausdrücke • Kontrollstrukturen • Funktionen, statische Methoden, Methodensignaturen • zusammengesetzte Datentypen • Klassen und Objekte • Pakete und Module • Container
Lehrformen	<i>Seminaristische Vorlesung mit Übung in Hybridform (flipped classroom) mit Lernplenum (Vorlesung), Gruppenarbeiten, Übungen im Team</i>
Voraussetzungen für die Teilnahme	<i>Keine formalen Voraussetzungen</i>
Literatur/ multimediale Lehr- und Lernprogramme	<p><i>Grundlagen als interaktive H5P Module, Vorlesungsmaterialien</i></p> <p><i>Auszug aus der Literaturliste:</i></p> <ul style="list-style-type: none"> • Bonacina, Michael. C# Programmieren für Einsteiger. 2. Auflage. Einfach Programmieren lernen. Landshut: BMU Media GmbH, 2020. • Dijkstra, Edgar. "Edgar Dijkstra: Go To Statement Considered Harmful." <i>Communications of the ACM</i> 11, no. 3 (March 1968): 147–48. • Goldberg, David. "What Every Computer Scientist Should Know about Floating-Point Arithmetic." <i>ACM Computing Surveys</i> 23, no. 1 (March 1991): 5–48. https://doi.org/10.1145/103162.103163. • Nassi, I., and B. Shneiderman. "Flowchart Techniques for Structured Programming." <i>SIGPLAN Not.</i> 8, no. 8 (August 1973): 12–26. https://doi.org/10.1145/953349.953350. • Oram, Andrew, and Greg Wilson, eds. <i>Beautiful Code</i>. 1st. ed. Theory in Practice Series. Beijing ; Sebastopol, Calif: O'Reilly, 2007. • Theis, Thomas. <i>Einstieg in Python: ideal für Programmierneinsteiger</i>. 6., Aktualisierte Auflage. Rheinwerk Computing. Bonn: Rheinwerk Verlag, 2020. • Yoder, Cornelia M., and Marilyn L. Schrag. "Nassi-Shneiderman Charts an Alternative to Flowcharts for Design." <i>In Proceedings of the Software Quality Assurance Workshop on Functional and Performance Issues</i>, 79–86. New York, NY, USA: Association for Computing Machinery, 1978. https://doi.org/10.1145/800283.811104.

Verwendbarkeit	<i>Dieses Modul findet Verwendung als Pflichtmodul in den Bachelor-Studiengängen Informatik und Wirtschaftsinformatik</i>
Arbeitsaufwand/ Gesamtworkload	<ul style="list-style-type: none"> • Präsenzzeit: 45h • Selbststudium: 60h • Prüfungsvorbereitung: 45h
ECTS und Gewichtung der Note in der Gesamtnote	<i>5 ECTS (5/180)</i>
Leistungsnachweis	<i>Schriftliche Prüfung am PC (90min)</i>
Semester	<i>1. Semester</i>
Häufigkeit des Angebots	<i>Einmal im Studienjahr</i>
Dauer	<i>Ein Semester</i>
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Pflichtmodul</i>

Modulname	Datenstrukturen und Algorithmen
Modulverantwortlicher/ Modulverantwortliche	Dr.-Ing. David Sommer, Vertretungsprofessor
Qualifikationsziele	<ul style="list-style-type: none"> • Kenntnis der Grundlagen der Analyse von Algorithmen und Datenstrukturen • Kenntnis paradigmatischer Algorithmen und Datenstrukturen und insbesondere ihrer Eigenschaften • Fähigkeit zur eigenständigen Analyse von Algorithmen und Datenstrukturen und zur Bewertung von Alternativen • Fähigkeit zum Entwurf und der Implementierung von Datenstrukturen und Algorithmen
Modulinhalte	<ul style="list-style-type: none"> • Analyse von Algorithmen: Grundlagen • Algorithmen und Komplexitätsklassen: O-Notation • Spezifikation und Implementierung von Datenstrukturen • Rekursive und dynamische Datenstrukturen • Suchen und Sortieren • Bäume • Graphen
Lehrformen	Vorlesungen (2 SWS) und Übungen (2 SWS) am Rechner
Voraussetzungen für die Teilnahme	Grundlegende Programmierkenntnisse (möglichst in Java), Programmieren 1
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> • Ottmann, Widmayer: <i>Algorithmen und Datenstrukturen</i>, Spektrum Verlag • Saake, Sattler: <i>Algorithmen und Datenstrukturen (mit Java)</i>, dpunkt.Verlag • Sedgewick: <i>Algorithmen in Java: Teil 1 - 4</i>, Pearson Studium • Cormen, Leiserson, Rivest, Stein: <i>Introduction to Algorithms</i>, MIT Press
Lehrbriefautor	
Verwendbarkeit	Bachelorstudiengänge Informatik und Wirtschaftsinformatik
Arbeitsaufwand/ Gesamtworkload	Gesamt 150 Stunden: Präsenzzeit 60 Stunden, Selbststudium 60 Stunden und Prüfungsvorbereitung sowie Prüfung 30 Stunden.
ECTS und Gewichtung der Note in der Gesamtnote	5 CP (5/180)
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) am Rechner (90 Minuten)
Semester	2. Semester (Sommersemester)
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 8: Programmierung II

Modulname	Fortgeschrittene Techniken der Programmierung
Modulverantwortlicher/ Modulverantwortliche	Dr.-Ing. Sommer
Qualifikationsziele	<ul style="list-style-type: none"> • Kenntnis fortgeschrittener Konzepte der objektorientierten Programmierung (mit Fokus auf Java 8) • Engineering-Aspekte der Programmierung: die Fähigkeit, diese Programmierkonzepte selbständig in Projekten anzuwenden • Programmieren gegen APIs von Drittanbietern (z.B. Spieleprogrammierung, Robotik) • Fähigkeit, objektorientierte Entwurfskonzepte praktisch anzuwenden
Modulinhalte	<ul style="list-style-type: none"> • Objektorientierte Entwurfs- und Programmierkonzepte: Grundideen und Ziele der Objektorientierung, Grundlagen des OO-Entwurfs, Klassen und Objekte, Vererbung, <i>Generics</i>, Nebenläufige Programmierung) • Erlernen und Einüben der zielgerichteten praktischen Anwendung der Programmierkonzept • Einbindung von Komponenten und Programmieren gegen externe APIs • Funktionale Programmierkonzepte (z.B. Stream-basierte Programmierung)
Lehrformen	Vorlesung, Übungen, einzelne Projekte
Voraussetzungen für die Teilnahme	Grundlegende Programmierkenntnisse (z.B. aus Programmieren 1)
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> - Ullenboom, Christian, <i>Java ist auch eine Insel: Java programmieren lernen mit dem umfassenden Standardwerk für Java-Entwickler</i>, Bonn/Rheinwerk 2018 - Eckel, Bruce, <i>Thinking in Java</i>, Prentice Hall, 2006 - Oracle, <i>The Java Tutorials</i>, https://docs.oracle.com/javase/tutorial/ - Habelitz, Hans-Peter, <i>Programmieren lernen mit Java: der leichte Einstieg</i>, Bonn, Rheinweg, 2017
Lehrbriefautor	
Verwendbarkeit	Bachelorstudiengänge Informatik und Wirtschaftsinformatik
Arbeitsaufwand/ Gesamtworkload	150 Stunden (Präsenzzeit: 60 Stunden, Selbststudium: 60 Stunden, Prüfung und Prüfungsvorbereitung: 30 Stunden)
ECTS und Gewichtung der Note in der Gesamtnote	5 CP (5/180)
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) am Rechner (90 Minuten)
Semester	2. Semester (Sommersemester)
Häufigkeit des Angebots	Jedes Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	Da die Programmierung eine praktische Tätigkeit ist, geht es in der Lehrveranstaltung nicht nur darum, Konzepte zu vermitteln, sondern deren Verwendung auch praktisch einzuüben. Aus diesem Grund sind Vorlesung und Übungsbetrieb in dieser Lehrveranstaltung in besonderer Weise verzahnt.

Modul 9: Informationsmanagement

Modulname	Information Engineering
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Regina Polster
Qualifikationsziele	Die Studierenden entwickeln ein Verständnis dafür, wie aus der Fülle von Techniken die ausgewählt und so eingesetzt werden können, die aus unstrukturierten organisatorischen Gegebenheiten Schritt für Schritt ein funktionierendes Informationssystem entwickeln helfen. Durch Fallstudienbearbeitung werden die Anwendung ausgewählter formaler Methoden für die Planung (planning), die Analyse (analysis), den Entwurf (design) und die Realisierung (construction) von Informationssystemen auf unternehmensweiter Basis oder in wesentlichen Unternehmensbereichen vermittelt.
Modulinhalte	<ul style="list-style-type: none"> • Strategisches Information Engineering: u.a. Portfolioanalyse; Erfolgsfaktorenanalyse; Kennzahlensysteme; Wirtschaftlichkeitsanalyse; Nutzwertanalyse. • Administratives Information Engineering: unter anderem Methoden der Benutzerbeteiligung; Methoden des Geschäftsprozessmanagements; Requirements Engineering; Qualitätsmodelle. • Operatives Information Engineering: u.a. Hardware- und Software-Monitoring, Risikoanalyse; IT-Service-Management und Serviceebenen-Vereinbarungen.
Lehrformen	Vorlesungen (3 SWS) und Übungen (1 SWS)
Voraussetzungen für die Teilnahme	Keine formalen Voraussetzungen
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> • Martin, J.(1990): Information Engineering I – Introduction, Prentice Hall. • Martin, J.(1990): Information Engineering II – Planning & Analysis, Prentice Hall. Martin, J.(1990): Information Engineering, III – Design & Construction. Prentice Hall. • Krcmar, H. Einführung in das Informationsmanagement, 2014 • Herrmann, T. (2011): Kreatives Prozessdesign: Konzepte und Methoden zur Integration von Prozessorganisation, Technik und Arbeitsgestaltung. • Peter Hruschka, Business Analysis und Requirements Engineering: Produkte und Prozesse nachhaltig verbessern, 2019 • Ernst Tiemeyer et al.: Handbuch IT-Management, Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis 2020
Lehrbriefautor	
Verwendbarkeit	Bachelorstudiengänge Informatik und Wirtschaftsinformatik
Arbeitsaufwand/ Gesamtworkload	Gesamt 150 Stunden: Präsenzzeit 60 Stunden, Selbststudium 60 Stunden und Prüfungsvorbereitung sowie Prüfung 30 Stunden.
ECTS und Gewichtung der Note in der Gesamtnote	5 CP (5/180)
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur 90 Minuten)
Semester	2. Semester (Sommersemester)
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 10: Technische Grundlagen I

Modulname	Digitaltechnik und IT-Mobiltechnik				
Modulverantwortlicher / Modulverantwortliche	Prof. Dr. Martin Golz				
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> - sich fachbezogene Kompetenzen im Selbststudium anzueignen, - den interdisziplinären Wissenstransfer, insbesondere in Problemfeldern von Elektroingenieuren, zu stützen. <p>Die Studierenden sollen in der Lage sein, ausgewählte Hardware-Grundlagen der</p> <ul style="list-style-type: none"> - Mikrocontrollertechnik, - Internet-of-Things-Technologien, - Rechnerarchitektur, - Rechnernetz- und Mobilfunk-Kommunikation <p>zu kennen und zu verstehen.</p>				
	Inhalt	Kennen, Verstehen	Anwenden	Analysieren, Bewerten	Synthe- tisieren
	Grundlagen der Booleschen Algebra	X		X	X
	Grundlagen der Mikroelektronik	X			
	Digitalelektronische Grundelemente	X		X	
	Grundsaltungen der digitalen Schaltungstechnik	X		X	
	Mathematisch-physikalische Grundlagen der freien und der drahtgebundenen Wellenausbreitung	X			
	Grundlagen der Informationsübertragung	X			
Modulinhalte	<ol style="list-style-type: none"> 1. Grundkonzepte: analoge und digitale Signale, technische Realisierung der Booleschen Algebra, Pegelbereiche, Störabstand 2. CMOS-Technologie: elektronische Schalter der MOS-Technik, CMOS-Prinzip 3. Grundelemente: Logikgatter, Datenbus-Prinzipien 4. Programmierbare Hardware, Schaltnetze 5. Schaltwerke: asynchrone / synchrone, ungetaktete / getaktete; Metastabilität 6. Datentechnik: Register, Schieberegister, Zähler, Code-Wandler 7. Rechentechnik: Addierer / Subtrahierer, Arithmetisch-Logische Einheit 8. Speichertechnik: Kategorien, dynamische & statische RAM, Flash-EEPROM 9. Elektromagnetische Felder und Wellen: Maxwell-Gleichungen, Wellengleichung, Polarisation, ebene Wellen, Kugelwellen, Fresnel-Gleichungen 10. Leitungstheorie: Telegraphengleichung, Wellenwiderstände, Augendiagramm, Wellenleiter: Koaxialleitung, Zweidrahtleitung 11. Antennen: Nah- / Fernfeld, Hertzscher Dipol, Antennentypen 12. Modulation von analogen und digitalen Trägerwellen mit analogen und digitalen Informationen 				
Lehrformen	Vorlesungen und Online-Vorlesungen unterstützt durch Vorlesungsskript				
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen für die Teilnahme bestehen nicht. Schulkenntnisse der Physik, insbesondere Elektrizitätslehre und Schwingungen / Wellen, sowie der Booleschen Algebra werden vorausgesetzt.				
Literatur	<p>Vorlesungsskript</p> <p>Schiffmann & Schmitz (2005) Technische Informatik. Springer</p> <p>Fricke (2018) Digitaltechnik. Springer-Vieweg</p> <p>Gehrke, Winzker, Urbanski, Woitowitz (2016) Digitaltechnik. Springer-Vieweg</p> <p>Gustrau (2019) Hochfrequenztechnik. Hanser</p>				

	<i>Sauter (2018) Grundkurs Mobile Kommunikationssysteme. Springer-Vieweg</i>
Lehrbriefautor	
Verwendbarkeit	<i>Das Modul ist ein Pflichtmodul im Bachelorstudiengang Informatik. Eine Verwendung in anderen Studiengängen der Fakultät ist nicht möglich.</i>
Arbeitsaufwand	<i>150 Stunden, davon 60 Präsenzstunden und 90 Stunden Selbststudium</i>
ECTS, Gewichtung der Note in der Gesamtnote	<i>5 Leistungspunkte Gewichtung der Note in der Gesamtnote: 1 / 36 (5/180)</i>
Leistungsnachweis	<i>Schriftliche Prüfung: Dauer 120 Minuten</i>
Semester	<i>1. Semester</i>
Häufigkeit des Angebots	<i>Einmal im Studienjahr</i>
Dauer	<i>Ein Semester</i>
Art d. Lehrveranstaltung	<i>Pflichtmodul</i>
Besonderes	

Modul 11: Technische Grundlagen II

Modulname	Mikroprozessortechnik & Eingebettete Systeme
Modulverantwortlicher/ Modulverantwortliche	Dr.-Ing. Sommer
Qualifikationsziele	<p>70% Fach-/ 10% Methoden-/ 20% Systemkompetenz</p> <p><i>Fachkompetenz:</i> Architektur von Mikroprozessoren, Mikrocontrollern und Digitalen Signalprozessoren (DSP), Grundlagen und fachspezifisches Wissen zu aktuellen eingebetteten Systemen, Fachwissen zur Funktion von Peripherieelementen, Fachwissen zur seriellen Kommunikation und zur Multiprozessorkommunikation, Kennenlernen der maschinennahen Programmierung</p> <p><i>Methodenkompetenz:</i> Erwerb von Wissen und teilweise von praktischen Erfahrungen zur Entwurfsmethodik, zum Test, insbesondere Simulation und Emulation von Mikrocontrollern und Eingebettete Systemen</p> <p><i>Systemkompetenz:</i> Eingliederung der Einzelelemente im Gesamtzusammenhang, Wissen zu Problemen der Systemkommunikation und der Synchronizität, Wissen zu Multiprozessorsystemen und zur Einbindung in Host-Systeme</p>
Modulinhalte	<p>Mikroprozessor-Architekturen, Speicher-Verwaltung, Stack-Verwaltung, Interruptsystem, Entwicklungswerkzeuge (Simulatoren, Debugger, Emulatoren, Logikanalysator), Maschinenbefehlssatz 8051/52, Maschinensprache-Programmierung (Flags, Funktionsregister, Programmlaufzeiten, Interruptsystem), Zeitverhalten bei externen Speicherzugriffen, Peripherieelemente: Timer / Counter, Ports, PWM-Unit, Schnittstellen (RS232, I²C, CAN, SPI, USB1/2/3, Firewire, Bluetooth, PCI-Express), Eingebettete Systeme (ARM, AVR, 8051), spezialisierte Betriebssysteme, modernen Sensorik- und Aktorik-Komponenten, Einsatzgebiete</p>
Lehrformen	Vorlesung (1 SWS) und Übung (1 SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramm	<p>Es ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> • Walter J (2008) Mikrocomputertechnik mit der 8051-Controller Familie : Hardware, Assembler, C. Springer Berlin Heidelberg. • Berg B, Groppe P, Klein J (2006) C-Programmierung für 8051er 1-3: 3 Bdo. Elektor-Verlag. • Marwedel P (2007) Eingebettete Systeme, Springer Berlin Heidelberg.
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang Informatik ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 30 Stunden; Selbststudium und selbständige Tätigkeit: 30 Stunden;
ECTS und Gewichtung der Note in der Gesamtnote	2 ECTS-Credits Gewichtung: 2/180

Leistungsnachweis	alternative Prüfungsleistung: Bearbeitung einer Projektaufgabe (Teamarbeit) mit den Schritten: Konzeption, Implementation, Test, Verifikation, Präsentation
Semester	1. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Pflichtmodul</i>
Besonderes	

Modulname	Rechnerarchitektur
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Math. Gerd Recknagel
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • moderne Architekturen zu analysieren und können somit die neuen Architekturen bezüglich ihrer Leistungsgrenzen und Einsatzmöglichkeiten fundiert klassifizieren und bewerten. • traditionelle Architekturen mit aktuellen Entwicklungen zu vergleichen. • moderne Benchmarks zur Leistungsmessung und -bewertung gezielt und mit Augenmaß einzusetzen und deren Ergebnisse zu analysieren und beurteilen. • die Arbeitsweise und Implementierung unterschiedlicher Rechnersysteme auch im mobilen Bereich zu verstehen. Sie können auf dieser Grundlage geeignete Entscheidungen zur Beschaffung und Einsatz der aktuellen am Markt befindlichen Systeme treffen. • Chancen und Möglichkeiten neuer Entwicklungen zu erkennen und bewerten. • die einzelnen Rechnerkomponenten zu benennen. Sie kennen den grundlegenden Aufbau und die Arbeitsweise der einzelnen Komponenten des von-Neumann-Rechners sowie ihr Zusammenspiel im Ganzen. Sie sind mit den Konzepten des Interrupthandlings sowie den Standarderweiterungen aktueller Systeme vertraut. • anhand von ausgewählten Kriterien mögliche Maßnahmen zur Leistungsverbesserung zu beurteilen. Sie kennen die Vor- und Nachteile von Pipelining sowie mögliche Konflikte und deren Behandlung.
Modulinhalte	<ul style="list-style-type: none"> • Grundlagen <ul style="list-style-type: none"> ○ Was ist Rechnerarchitektur? ○ Meilensteine ○ Sichtweisen ○ Praxisorientierte Interpretation der Rechnerarchitektur ○ Aufgaben und Ziele, Amdahlsches Gesetz ○ Entwurfsverfahren ○ Rechnerklassifikation ○ Leistungsbewertung, MIPS, MFLOPS, Benchmark • Digitale Logik und Schaltnetze <ul style="list-style-type: none"> ○ Boolesche Algebra ○ Schaltnetze ○ arithmetische Schaltungen ○ Speicherschaltungen ○ ALU • von-Neumann-Rechner <ul style="list-style-type: none"> ○ von-Neumann-Architektur ○ Aufbau und Arbeitsweise des Prozessors ○ Befehlszyklus ○ Interruptbehandlung ○ Speicherwerk und Speicherorganisation ○ Ein- und Ausgabewerk, Kommunikationswege ○ Standarderweiterungen • Befehlssatzentwurf <ul style="list-style-type: none"> ○ Aufbau Maschinenbefehl ○ Adressformate ○ Adressierungsarten ○ Befehlsklassen ○ Stackorganisation ○ Kriterien für den Befehlssatzentwurf • Ausgewählte Aspekte der Prozessorimplementierung <ul style="list-style-type: none"> ○ Steuerwerksimplementierung ○ fest verdrahtetes Steuerwerk

	<ul style="list-style-type: none"> ○ mikroprogrammiertes Steuerwerk ○ Kostenreduktion und Leistungssteigerung ○ Pipelining und Pipeline-Hazards ● CISC- und RISC-Architekturen <ul style="list-style-type: none"> ○ CISC ○ von CISC zu RISC ○ RISC ● Entwurf Speicherhierarchie <ul style="list-style-type: none"> ○ Speicherhierarchie, Lokalität, Grundaufgaben der Steuerung der Speicherhierarchie ○ Caches ○ Virtuelle Speicherhierarchie
Lehrformen	Vorlesungen (2 SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Den Studierenden werden zu Beginn der Veranstaltung umfangreiche themenspezifische Dokumente in studip zur Verfügung gestellt.</p> <p>Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> ● Hennessy / Patterson: Rechnerarchitektur. Vieweg ● Tanenbaum / Goodmann: Rechnerarchitektur. Pearson ● Hermann: Rechnerarchitektur. Vieweg ● Martin: Rechnerarchitektur. Hanser ● Erhard: Rechnerarchitektur. Teubner ● Oberschelp / Vossen: Rechneraufbau und Rechnerstrukturen. Oldenbourg ● Schiffmann / Schmitz: Technische Informatik, Bd. 2. Springer
Lehrbriefautor	
Verwendbarkeit	Das Modul ist in den Bachelorstudiengang Informatik ein Pflichtfach.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 30 Stunden; Selbststudium: 40 Stunden; Prüfung und Prüfungsvorbereitung: 20 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	3 ECTS-Credits Gewichtung: 3/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 90 Minuten
Semester	2. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 12: Wirtschaftswissenschaften

Modulname	Wirtschaftswissenschaften
Modulverantwortlicher/ Modulverantwortliche	Professor Dr. Dr. Thomas Urban
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> - betriebswirtschaftliche Problemstellungen in einen unternehmerischen Zusammenhang einzuordnen - Kenntnisse über die grundlegenden Charakteristika der Rechtsformen der Personen- und Kapitalgesellschaften zu erlangen - die grundlegenden Organisationsformen zu kennen - betriebswirtschaftliche Zielformulierungen und Entscheidungen aus Sicht des Managements zu treffen sowie deren Risiken einzuschätzen - grundlegende volkswirtschaftliche Kenntnisse zu erwerben
Modulinhalte	<p>1 Betriebswirtschaftliche Grundlagen</p> <p>2 Rechtsformen und Unternehmensgründung</p> <ul style="list-style-type: none"> - Standort des Unternehmens - Unternehmenszusammenschlüsse und -kooperationen - Insolvenz, Sanierung und Liquidation <p>3 Unternehmensstrategien und -ziele</p> <ul style="list-style-type: none"> - Zielbildung - Zielinhalt - Zieldimension - Zielbeziehung <p>4 Personal</p> <ul style="list-style-type: none"> - Personalbedarfsermittlung und -beschaffung - Personalmotivation und -entlohnung - Personalentwicklung und -führung <p>5 Controlling</p> <ul style="list-style-type: none"> - Wertorientierte Unternehmensführung - Controllingbereiche - Controllinginstrumente <p>6 Organisation</p> <ul style="list-style-type: none"> - Aufbau- und Ablauforganisation - Organisationsformen in der Praxis - Organisationsentwicklung <p>7 Management</p> <ul style="list-style-type: none"> - Managementfunktionen - Strategisches Management - Risikomanagement <p>8 Grundlagen der Mikroökonomie</p> <ul style="list-style-type: none"> - Volkswirtschaftliche Grundlagen - Marktformen und -mechanismen
Lehrformen	<p>In der seminaristischen Vorlesung (4 SWS) werden die grundlegenden theoretischen Aspekte der Wirtschaftswissenschaften vermittelt.</p> <p>Anhand von Übungsaufgaben und Fallstudien erfolgt eine praxisorientierte Darstellung und Vertiefung des Stoffes. Ferner werden offene Fragen diskutiert.</p>

Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	folgende Literatur ist empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> - Schierenbeck, H.: Grundzüge der Betriebswirtschaftslehre, Oldenbourg Wissenschaftsverlag, München. - Thommen, J.-P./Achleitner A.-K.: Managementorientierte Betriebswirtschaftslehre, Gabler, Wiesbaden - Töpfer, A.: Betriebswirtschaftslehre – anwendungs- und prozessorientierte Grundlagen, Springer-Verlag, Berlin - Wöhe, G. et. al.: Einführung in die allgemeine Betriebswirtschaftslehre, Vahlen-Verlag, München.
Lehrbriefautor/en	
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Wirtschaftsinformatik“, „Multimedia Marketing“ und „Informatik“ ein Pflichtfach.
Arbeitsaufwand/ Gesamtworkload	150 Stunden, davon 60 Präsenzstunden, 60 Stunden Selbststudium und 30 Stunden Prüfung und Prüfungsvorbereitung
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 90 Minuten
Semester	1. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modul 13: Fächerübergreifende Kompetenzen

Modulname	Einführung in das wissenschaftliche Arbeiten
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Elisabeth Scherr
Qualifikationsziele	<ul style="list-style-type: none"> - Die Studierenden sind in der Lage, sich zielsicher einen Überblick zum wissenschaftlichen Diskussionsstand eines Forschungsgebiets zu verschaffen - Sie können eine Problemstellung formulieren und daraus eine Forschungsfrage ableiten, die einen Beitrag zum jeweiligen Forschungsgebiet verspricht - Sie kennen verschiedene Forschungsmethoden und können diese zur Beantwortung einer Forschungsfrage entsprechend umsetzen - Sie können die Vorgehensweise im Forschungsprojekt und die daraus abgeleiteten Forschungsergebnisse in Wort und Schrift strukturiert und intersubjektiv nachprüfbar darstellen - Sie wissen die Kriterien der guten wissenschaftlichen Praxis und können diese für eine kritische Würdigung eines Forschungsprojekts anwenden - Sie können eine Projektskizze für ein Forschungsvorhaben erstellen - Sie sind befähigt, im Forschungsprojekt technische Hilfsmittel (z.B. Textverarbeitungsprogramme, Literaturverwaltungsprogramme) einzusetzen
Modulinhalte	Durchführung von Literaturrecherchen, Entwicklung von Forschungsfragen, Anwendung und kritische Würdigung verschiedener Forschungsmethoden, Darstellung von Forschungsergebnissen, Organisation des Projekts „Wissenschaftliche Arbeit“
Lehrformen	Vorlesung (1 SWS) + Übung (1 SWS)
Voraussetzungen für die Teilnahme	Keine
Literatur/ multimediale Lehr- und Lernprogramme	Vorbereitende und weiterführende Literatur wird im Rahmen der Lehrveranstaltung bekannt gegeben.
Lehrbriefautor	
Verwendbarkeit	Die Lehrveranstaltung „Einführung in das wissenschaftliche Arbeiten“ ist dem Modul „Fächerübergreifende Kompetenzen“ zugeordnet. Sie ist Bestandteil der Studiengänge Informatik (B.Sc.), Multimedia Marketing (B.Sc.) und Wirtschaftsinformatik (B.Sc.).
Arbeitsaufwand/ Gesamtworkload	<p>Gesamtworkload 90 Stunden, davon:</p> <ul style="list-style-type: none"> - Präsenzzeit: 30 Stunden - Selbstlernphase: 30 Stunden - Prüfungsaufwand: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	3 von 180 ECTS
Leistungsnachweis	Schriftliche Prüfungsleistung (Exposé mit 5-7 Seiten Textumfang)
Semester	1. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

2. Studienabschnitt

Pflichtmodule

Modulname	Wissensverarbeitung
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. T. Heimrich (gehalten durch externen Lehrbeauftragten)
Qualifikationsziele	Die Studierenden gewinnen einen Überblick über die grundlegenden Ansätze zur Wissensrepräsentation, Wissensverarbeitung, Wissensmanagement und Semantic Web als Bestandteil der Künstlichen Intelligenz. Sie erlernen Methoden zur Entwicklung von Programmen, die grundlegende Aufgaben der Wissensverarbeitung modellhaft lösen: Wissensrepräsentation, Expertensysteme, Inferenzmaschinen, Maschinelles Lernen, Verarbeitung natürlicher Sprache mittels PROLOG In den Übungen gewinnen sie grundlegende Fähigkeiten zur Anwendung der logischen Programmierung zur Lösung der Aufgaben der Wissensverarbeitung.
Modulinhalte	Künstliche Intelligenz und Wissensverarbeitung, Formen der Wissensrepräsentation: Zustandsraum, Logik, Regeln, Taxonomien, Ontologien, Arten der Inferenzmechanismen, , Deduktion und Induktion, Expertensysteme, Verarbeitung natürlicher Sprache, Maschinelles Lernen, Data Mining.
Lehrformen	Vorlesungen, Übungen, Projekt
Voraussetzungen für die Teilnahme	Theoretische Informatik, prozedurale Programmierung, Datenbanken
Literatur/ multimediale Lehr- und Lernprogramme	Levesque: Thinking as Computation Norvig: Artificial Intelligence Programming Russel/Norvig: Künstliche Intelligenz
Lehrbriefautor	
Verwendbarkeit	Bachelorstudiengang Informatik
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden Selbststudium: 45 Stunden Prüfung und Prüfungsvorbereitung: 45 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 CP
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) von 90 Minuten
Semester	3. Semester
Häufigkeit des Angebots	jedes Studienjahr
Dauer	Das Modul erstreckt sich über ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modulname	Software Engineering
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Erwin Neuhardt
Qualifikationsziele	Die Studierenden kennen die Phasen der ingenieurmäßigen Software-Entwicklung. Sie kennen Methoden zur Ermittlung und Darstellung von Anforderungen und Prinzipien des objektorientierten Entwurfs. Sie kennen Methoden zur Ermittlung von Testfällen und zur Qualitätssicherung von Arbeitsergebnissen. Sie kennen zwei Vorgehensmodelle zur Umsetzung des Softwareentwicklungsprozesses. Sie können diese Methoden anwenden, um ausgehend von einem vorgegebenen Text funktionsfähige Software zu entwickeln.
Modulinhalte	Aufgabenstellung und Ziele des Software Engineering, Phasen der Software-Entwicklung: Anforderungsermittlung, logische Architektur, objektorientierter Entwurf, Implementierung, Testen, Qualitätssicherung mit Inspektionen, Vorgehensmodelle. Beschreibungsmöglichkeiten für die Ergebnisse der Anforderungsermittlung und des Entwurfes: Anwendungsfalldiagramm, Anwendungsfallbeschreibungen, Klassendiagramm, Sequenzdiagramm, Zustandsdiagramm, Paketdiagramm.
Lehrformen	Vorlesung (3 SWS) und Übung (1 SWS)
Voraussetzungen für die Teilnahme	Module Programmierung I und II
Literatur/ multimediale Lehr- und Lernprogramme	Balzert, H.: Lehrbuch der Software-Technik Band 2, Spektrum Verlag, 1998 Binder, R.: Testing Object-Oriented Systems, Addison-Wesley, 2000 Larman, C.: Applying UML and Patterns, Upper Saddle River NJ., 2001 Oestereich, B.: Objektorientierte Software Entwicklung, Oldenbourg, 2009 Rupp, C.: Requirements-Engineering und –Management, Hanser Verlag, 2007
Lehrbriefautor	
Verwendbarkeit	Pflichtmodul in den Bachelorstudiengängen Informatik und Wirtschaftsinformatik
Arbeitsaufwand/ Gesamtworkload	Präsenzzeit: 60 Stunden Selbststudium: 60 Stunden Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS Gewichtung 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) 90 Minuten
Semester	3. Semester
Häufigkeit des Angebots	einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	-

Modulname	Computergestützte Intelligenz				
Modulverantwortlicher / Modulverantwortliche	Prof. Dr. Martin Golz				
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> - Problemstellungen der Daten- und Wissenstechnologien zu analysieren, - Prozessschritte der computergestützten Intelligenz zu konzipieren, - Methoden einzelner Prozessschritte zu verstehen und anzuwenden, - Methoden der Validierungsanalyse zu verstehen und anzuwenden, - Grundannahmen und Modelle der empirischen Inferenz zu kennen, - Einige mathematische Hintergründe der empirischen Inferenz zu kennen. 				
Inhalt	Kennen, Verstehen	Anwenden	Analysieren, Bewerten	Synthesisieren	
Grundlagen der computergestützten Intelligenz	X				
Prozesskette der computergestützten Intelligenz	X	X	X		
Vorverarbeitung und Merkmalsextraktion	X	X			
Methoden der Klassifikationsanalyse	X	X	X		
Methoden der Gruppierungsanalyse	X	X	X		
Methoden der Validierungsanalyse	X	X			
Neuroinformatische Methoden	X	X	X		
Modulinhalte	<ol style="list-style-type: none"> 1. Einführung, Künstliche Intelligenz und Gesellschaft, Zeichenerkennung, Objekterkennung, mobile und eingebettete Intelligenz 2. Definitionen und Postulate 3. Vorverarbeitung 4. Merkmalsextraktion, Ähnlichkeitsmaße 5. Unüberwachtes Lernen, Gruppierungsanalyse, Fuzzy-Cluster-Analyse 6. Überwachtes Lernen, Bayes-Theorem, Lineare Diskriminanzanalyse 7. Stützvektormethode: Maximalspanne, weiche Spanne, Kernfunktionssubstitution, Hyperparameter-Optimierung 8. Neuronale Netze: Neurobiologische Grundlagen, Perzeptronen, Netztopologien, Vorwärtsrechnung, Fehlerrückführung 9. Lernende Vektorquantisierung, Automatische Relevanzdetermination 10. Selbstorganisierende Merkmalskarten, Inkrementelle Netze 11. Massendaten und tiefe Lernarchitekturen (Deep Learning) 				
Lehrformen	Vorlesungen und Online-Vorlesungen unterstützt durch Vorlesungsskript, Übungen unterstützt durch Übungsskript, Rechnergestütztes Praktikum unterstützt durch Rahmenprogramme.				
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen für die Teilnahme bestehen nicht. Kenntnisse in Linearer Algebra, Analysis, Statistik, Numerischer Mathematik und Software-Entwicklung sind für eine erfolgreiche Teilnahme erforderlich.				
Literatur	Vorlesungsskript Übungsskript Duda, Hart, Stork (2001) Pattern Classification. Wiley Haykin (2008) Neural Networks and Learning Machines, Prentice Hall Nielsen (2015) Neural Networks and Deep Learning. Determination press Theodoridis & Koutroumbas (2008) Pattern Recognition. Elsevier				
Lehrbriefautor					
Verwendbarkeit	Das Modul ist ein Pflichtmodul im Bachelorstudiengang Informatik. Eine Verwendung im Studiengang Bachelor Wirtschaftsinformatik ist möglich.				
Arbeitsaufwand	150 Stunden, davon 60 Präsenzstunden und 90 Stunden Selbststudium				

ECTS, Gewichtung der Note in der Gesamtnote	<i>5 Leistungspunkte Gewichtung der Note in der Gesamtnote: 1 / 36 (5 / 180)</i>
Leistungsnachweis	<i>Mündliche Prüfung (25 Minuten)</i>
Semester	<i>3. Semester</i>
Häufigkeit des Angebots	<i>Einmal im Studienjahr</i>
Dauer	<i>Ein Semester</i>
Art d. Lehrveranstaltung	<i>Pflichtmodul</i>
Besonderes	<i>Prüfungsvorbereitungskurs für Studierende, die regelmäßig an den Lehrveranstaltungen teilnehmen</i>

Modulname	Datenbanksysteme
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Thomas Heimrich
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • Daten aus der Anwendungswelt in ein Entity-Relationship-Modell (ERM) zu überführen • aus dem ERM einen logischen Datenbankentwurf (relational) abzuleiten • den logischen Datenbankentwurf zu normalisieren • die theoretischen Grundlagen der Normalisierung beherrschen • Tabellen mit SQL zu erstellen, Daten in Tabellen einzufügen und zu ändern • komplexe select-Anfragen mit SQL auszuführen • eine Verbindung zwischen Oracle und einer Java-Anwendung herzustellen
Modulinhalte	<ol style="list-style-type: none"> 1 Datenmodellierung und Entity Relationship Modell <ul style="list-style-type: none"> • Merkmale von Datenbanksystemen und die verschiedenen Arten von Datenunabhängigkeit • der Begriff „Schlüssel“ in der Datenbankwelt • graphische Elemente (Notationen) im ERM • Funktionalität und Kardinalität im ERM • einfache und mehrwertige Relationen • spezielle ERM-Elemente (IST-Beziehung, schwache Entitytypen) • Überblick zur Datenmodellierung mit UML 2 Das relationale Modell <ul style="list-style-type: none"> • theoretische Grundlagen zu Relationen • Überführung der ERM-Elemente in Relationen • Überführung der verschiedenen Arten von Beziehungen in Relationen • Überführung spezieller ERM-Elemente in Relationen 3 Normalisierung (Relationale Entwurfstheorie) <ul style="list-style-type: none"> • mögliche Anomalien bei der relationalen Modellierung • Funktionale Abhängigkeiten (formale Betrachtung) • 1. bis 3. Normalform und deren Eigenschaften • Algorithmus für die Normalisierung • Boyce-Codd Normalform 4 Relationale Algebra und Relationenkalkül <ul style="list-style-type: none"> • relationale Algebra mit ihren Grundoperationen (Selektion, Projektion, verschiedene Join-Operationen, Mengenoperationen) • der Operatorbaum als Mittel der Repräsentation von Algebra-Ausdrücken • Anfragen mit dem Relationenkalkül 5 Schemadefinition und Datenintegrität <ul style="list-style-type: none"> • Einführung zu SQL • SQL-Anweisungen zum Erstellen von Tabellen • Möglichkeiten Integritätsbedingungen in SQL zu definieren • SQL-Anweisungen um Daten in Tabellen einzufügen und zu ändern 6 select-Anfragen mit SQL <ul style="list-style-type: none"> • Grundstruktur der select-Anweisung • einfache und komplexe (geschachtelte) select-Anweisungen 7 Sichten in SQL <ul style="list-style-type: none"> • Grundkonzept von Sichten

	<ul style="list-style-type: none"> • Definition von Sichten • Daten in Sichten einfügen und ändern • Verwendung von Sichten für die Datenmodellierung <p>8 Datenbankbindung an Programmiersprachen</p> <ul style="list-style-type: none"> • Arten der Einbettung von SQL in eine Programmiersprachen • Anbindung einer ORACLE-Datenbank an eine JAVA-Anwendung • Anfragen an die Datenbank aus der JAVA-Anwendung heraus und verarbeiten der Anfrageergebnisse. <p>9 Transaktionen</p> <ul style="list-style-type: none"> • Grundlagen zu Transaktionen • Probleme bei der Nebenläufigkeit von Anfragen • Sperrmodelle und Sperrprotokolle
Lehrformen	Vorlesungen (3 SWS), rechnergestütztes Praktikum (1 SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramme	Den Studierenden wird zu Beginn der Veranstaltung ein umfangreicher Foliensatz mit weiterführenden Hinweisen zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> • Alfons Kemper, Andre Eickler: „Datenbanksysteme: Eine Einführung“, Oldenbourg Wissenschaftsverlag • Gunter Saake, Kai-Uwe Sattler, Andreas Heuer: „Datenbanken – Konzepte und Sprachen“, mitp
Lehrbriefautor	
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Informatik“ und „Wirtschaftsinformatik“ ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium: 60 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) von 90 Minuten
Semester	3. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Pflichtmodul</i>
Besonderes	

Modulname	Rechnernetze
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr.-Ing. Heinz-Peter Höller
Qualifikationsziele	<ul style="list-style-type: none"> • Kenntnis der Grundlagen der Kommunikationstechnik • Verwendung des OSI-Referenzmodells und der Protokollschichtung als Ordnungsrahmen • Überblick über die Aufgabenstellungen und gefundenen Lösungen in den Protokollschichten • Befähigung zur kritischen Beurteilung von Protokollfunktionalitäten • Befähigung dazu, die spezifischen Wirkungen von Protokollsets auf Anwendungen einzuschätzen
Modulinhalte	<ul style="list-style-type: none"> • OSI-Referenzmodell und Protokollschichtung • Aspekte der Datenübertragungsebene (Topologie, Multiplexing, Leitungskodierung) • Schichtspezifische und schichtenübergreifende Aufgabenstellungen (Rahmenbildung, Zugriffssteuerung, Fehlerbehandlung, Flusskontrolle, Verbindungsverwaltung) • Protokolle der Schicht 2 (Ethernet, WLAN, etc.) • Signalisierung, Routing, Peering • Protokolle der Schicht 3 (IP, ICMP, ARP, etc.) • Spezifische Aufgaben der Transportschicht (Prozessadressierung, Überlastbehandlung, Vorrangdaten) • Protokolle der Schicht 4 (TCP, UDP, DCCP, RTP) • Prozesszugriff auf das Kommunikationssystem: Sockets
Lehrformen	Vorlesungen (3 SWS), Übung (1 SWS)
Voraussetzungen für die Teilnahme	Keine formalen Voraussetzungen
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> • Kurose, J. & Ross, K. (2014). Computernetzwerke. 6. Auflage. Pearson • Tanenbaum, A. S. & Wetherall, D. J. (2012). Computernetzwerke. 5. Auflage. Pearson. • Badach, A. & Hoffmann, E. (2019). Technik der IP-Netze. 4. Auflage. Hanser.
Lehrbriefautor	
Verwendbarkeit	Bachelorstudiengänge Informatik und Wirtschaftsinformatik
Arbeitsaufwand/ Gesamtworkload	150 Stunden (Präsenzzeit: 60 Stunden, Selbststudium: 60 Stunden, Prüfung und Prüfungsvorbereitung: 30 Stunden)
ECTS und Gewichtung der Note in der Gesamtnote	5 CP
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) von 90 Minuten
Semester	3. Semester (Wintersemester)
Häufigkeit des Angebots	Jedes Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modulname	Betriebssysteme
Modulverantwortlicher/ Modulverantwortliche	Prof. Ralf C. Staudemeyer, Ph.D.
Qualifikationsziele	Dieser Grundlagenkurs ist eine Einführung in die Arbeitsweise und Anwendung von klassischen und modernen Betriebssystemen. Absolventen verstehen wie aus einer Zusammenstellung von Programmen eine sinnvolle Abstraktion aus unabhängigen Hardwarekomponenten erstellt wird. Dabei können die Studenten nachvollziehen wie Betriebssystem und Systemhardware zusammenwirken, um die verfügbaren Ressourcen den Anwendungsprogrammen zur Verfügung zu stellen. Themen dieser Vorlesung sind unter anderem Speicherverwaltung, Dateisysteme und Virtualisierung. Die allgemeinen Betriebssystemkonzepte werden durch praktische Übungen gefestigt.
Modulinhalte	Dies Modul ist wie folgt strukturiert: <ul style="list-style-type: none"> - Einführung in Betriebssysteme (Entwicklung, Klassifikation, Betriebsarten) - Speicherverwaltung (Speicherhierarchien, Speicherzugriff, Virtueller Speicher) - Festplatten (Hardware, Adressierung, Dateisysteme) - Prozesse und Threads (Prozessverwaltung, Scheduling, Synchronisation, Interprozesskommunikation) - Virtualisierung und Virtuelle Maschinen
Lehrformen	Vorlesungen (3 SWS), Übungen (1 SWS)
Voraussetzungen für die Teilnahme	Keine formalen Voraussetzungen
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> - Baun, C.: "Betriebssysteme kompakt". Springer. 2017. - Tanenbaum, A.: "Modern Operating Systems". Pearson. 2015. - Stallings, W.: "Operating Systems: Internals and Design Principles". Pearson, 2018. - prüfungsrelevante Literatur wird bekanntgegeben (Änderungen möglich)
Lehrbriefautor	
Verwendbarkeit	Pflichtmodul in den Bachelorstudiengängen Informatik und Wirtschaftsinformatik.
Arbeitsaufwand/ Gesamtworkload	Total: 150h. Präsenzstudium: 60h; Selbststudium incl. Prüfungsvorbereitung: 90h.
ECTS und Gewichtung der Note in der Gesamtnote	5 (5/180)
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) 90min
Semester	3
Häufigkeit des Angebots	einmal im akademischen Jahr (Wintersemester)
Dauer	ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modulname	Graphische Datenverarbeitung (GDV)
Modulverantwortlicher/ Modulverantwortliche	Prof. Hartmut Seichter, PhD
Qualifikationsziele	<ul style="list-style-type: none"> • Studierende kennen Grundprinzipien visueller Medien • Kursteilnehmer*Innen kennen wesentliche Bestandteile der Renderingpipeline • Studierende können Transformationen herleiten und anwenden • Sie kennen wesentliche Modelle in der Computergrafik • Studierende können Beleuchtungsmodelle und deren Approximationen einordnen und bewerten • Kursteilnehmer*Innen sind in der Lage die theoretischen Grundlagen anhand einer webbasierten 3D Engine anzuwenden.
Modulinhalte	<p>In diesem Modul werden folgende Inhalte vermittelt:</p> <ul style="list-style-type: none"> • Grundkenntnisse Wahrnehmungspsychologie und Physiologie des Sehapparates • Konzepte der Bilderzeugung, Speicherung und Transformation • 3D Modelle, insbesondere symbolische Surface und Volumenmodelle • Transformationspipeline: Homogene Räume und Transformationen • Szenengraphen und 3D im Web • Bildsyntheseverfahren • Geometrie und Bild: Samplingverfahren und Anti-Aliasing Strategien • Lichttransport: Texturing, Surfaces and Materials • Anwendung von visuellen Darstellungstechniken in der Praxis
Lehrformen	Online-Vorlesungen, Vorlesung (2SWS), Übungen (1SWS)
Voraussetzungen für die Teilnahme	<p>Keine formalen Voraussetzungen</p> <p>Kenntnisse in den folgenden Bereichen sind wünschenswert:</p> <ul style="list-style-type: none"> • Programmierkenntnisse in objektorientierten Programmiersprachen • Grundkenntnisse lineare Algebra und Vektorräume
Literatur/ multimediale Lehr- und Lernprogramme	<p>Interaktive H5P Module, Vorlesungsfolien</p> <p>Auszug aus der Literaturliste:</p> <ul style="list-style-type: none"> • Ashikhmin, Michael, and Peter Shirley. "An Anisotropic Phong BRDF Model." <i>Journal of Graphics Tools</i> 5, no. 2 (January 2000): 25–32. https://doi.org/10.1080/10867651.2000.10487522. • Blinn, James F. "Models of Light Reflection for Computer Synthesized Pictures." In <i>Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '77</i>, 192–98. San Jose, California: ACM Press, 1977. https://doi.org/10.1145/563858.563893. • Burley, Brent. "Physically-Based Shading at Disney." In <i>ACM SIGGRAPH</i>, 2012:1–7, 2012. • Cohen, Jonathan, Marc Olano, and Dinesh Manocha. "Appearance-Preserving Simplification." In <i>Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98</i>, 115–22. Not Known: ACM Press, 1998. https://doi.org/10.1145/280814.280832. • Cook, R. L., and K. E. Torrance. "A Reflectance Model for Computer Graphics." <i>ACM Transactions on Graphics (TOG)</i> 1, no. 1 (January 1982): 7–24. https://doi.org/10.1145/357290.357293. • Cook, Robert L., Loren Carpenter, and Edwin Catmull. "The Reyes Image Rendering Architecture." <i>ACM SIGGRAPH Computer Graphics</i> 21, no. 4 (August 1, 1987): 95–102. https://doi.org/10.1145/37402.37414.

	<ul style="list-style-type: none"> • <i>Foley, James D, Andries Van Dam, Steven K Feiner, John F Hughes, and Richard L Phillips. Introduction to Computer Graphics. Vol. 55. Addison-Wesley Reading, 1994.</i> • <i>Goldstein, E. Bruce. Sensation and Perception. 3rd ed. Belmont, Calif.: Wadsworth Pub. Co., 1989.</i> • <i>Hughes, John F. Computer Graphics: Principles and Practice. Third edition. Upper Saddle River, New Jersey: Addison-Wesley, 2014.</i> • <i>Pharr, Matt, Wenzel Jakob, and Greg Humphreys. Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann, 2016. http://pbr-book.org.</i> • <i>Schlick, Christophe. "An Inexpensive BRDF Model for Physically-Based Rendering." Computer Graphics Forum 13, no. 3 (August 1994): 233–46. https://doi.org/10.1111/1467-8659.1330233.</i> • <i>Walter, Bruce, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. "Microfacet Models for Refraction through Rough Surfaces." In Proceedings of the 18th Eurographics Conference on Rendering Techniques, 195–206. EGSR'07. Goslar, DEU: Eurographics Association, 2007.</i> • <i>Williams, Lance. "Pyramidal Parametrics." In Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques, 1–11. SIGGRAPH '83. New York, NY, USA: Association for Computing Machinery, 1983. https://doi.org/10.1145/800059.801126.</i>
Verwendbarkeit	<i>Pflichtmodul im Bachelor Informatik</i>
Arbeitsaufwand/ Gesamtworkload	<ul style="list-style-type: none"> • <i>Präsenzzeit: 45h</i> • <i>Selbststudium: 60h</i> • <i>Prüfungsvorbereitung: 45h</i>
ECTS und Gewichtung der Note in der Gesamtnote	<i>5 CP (5/180)</i>
Leistungsnachweis	<i>Schriftliche Prüfungsleistung (90min)</i>
Semester	<i>3. Semester</i>
Häufigkeit des Angebots	<i>Einmal im Studienjahr</i>
Dauer	<i>Ein Semester</i>
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Pflichtmodul</i>

Modulname	Projektmanagement
Modulverantwortlicher	Prof. Dr. Englmeier
Qualifikationsziele	<p>Kennen/Verstehen: In diesem Kurs werden Art und Nutzen von Projekten erläutert und warum es vorteilhaft ist Projektmanagement einzusetzen. Es beginnt mit der Erläuterung der theoretischen und praktischen Grundlagen des Projektmanagements und zeigt die Rolle von Projektmanagern und wie sie die Erwartungen von Unternehmen erfüllen können, die die Vorteile des Wandels nutzen wollen. Eine Reihe von Quizfragen runden den Kurs ab. Die Quizfragen ermöglichen es den Studierenden, ihr Wissen zu überprüfen und ihnen Hinweise zu geben, wo sie ihr Wissen verbessern könnten.</p> <p>Anwenden: Der Inhalt dieses Kurses folgt dem vom Project Management Institute (PMI) entwickelten Wissensfundus für das Projektmanagement (Project Management Body of Knowledge (PMBOK)). Der PMBOK kategorisiert das Wissen im Bereich Projektmanagement in zehn Wissensbereichen, von Integrations- und Umfangmanagement, Kosten- und Terminplanungsmanagement, Risiko- und Qualitätsmanagement bis hin zu Kommunikationsmanagement und Stakeholder-Management. Der Kurs strukturiert daher seinen Inhalt entlang dieser Wissensbereiche.</p> <p>Analysieren/Bewerten: Die Studierenden erlernen den Umgang mit einem Werkzeug für die Projektplanung und -kontrolle.</p> <p>Synthetisieren: Die Studierenden lernen die Ausgestaltung des Projekt-Lebenszyklus und wie alle Wissensbereiche in die jeweiligen Phasen eines Projektes in der Praxis integriert werden</p> <p>Die Veranstaltung wird auf Englisch gehalten. Damit soll der Umgang mit der fachbezogenen Sprache in Deutsch und Englisch erlernt und geübt werden.</p>
Modulinhalte	<ul style="list-style-type: none"> • Grundlagen der Projektmanagements • Integrationsmanagement • Umfang- und Inhaltsmanagement • Terminplanung • Kostenmanagement • Qualitätsmanagement • Ressourcenmanagement • Kommunikationsmanagement • Risikomanagement • Beschaffungsmanagement • Stakeholdermanagement • Projektmanagement-Tool
Lehrformen	Vorlesung (3 SWS), Übung (1 SWS)
Voraussetzungen für die Teilnahme	keine
Literatur / multimediale Lehr- und Lernprogramme	Digitaler Kurs, vom Modulverantwortlichen bereitgestellt. Project Management Institute, PMBOK Guide, 6. Ausgabe, 2017
Verwendbarkeit	Bachelor Informatik, Wirtschaftsinformatik, Multimedia-Marketing

Arbeitsaufwand/ Gesamt-Workload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium 60 Stunden; Praxisarbeitszeit: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 CP, Gewichtung der Note in der Gesamtnote 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) 90 Minuten
Semester	5. Semester
Häufigkeit des Angebots	Einmal pro Studienjahr (Sommersemester)
Dauer	Ein Semester
Art der Lehrveranstaltung	Pflichtveranstaltung

Modulname	Paradigmen der Softwareentwicklung
Modulverantwortlicher/ Modulverantwortliche	Dr.-Ing. David Sommer, Vertretungsprofessor
Qualifikationsziele	<p>60% Fach- / 20% Methoden- / 20% Systemkompetenz</p> <p>Fachkompetenz: Grundlegende Programmierparadigmen und Besonderheiten. Grundbegriffe, Historische Entwicklung. Paradigmen wie Imperative, Deklarative, Objektorientierte Programmierung. Strukturierte, Prozedurale, Modulare; Funktionale, Logische und Constraintprogrammierung</p> <p>Methodenkompetenz: Zielgerichtete Anwendung von Programmiermethoden mit Objektorientierten, Parallelen, Funktionalen, Logischen oder Web-basierten Paradigmen. Programmierung übersichtlicher und effizienter Lösungen mit zielgerichteten Herangehensweisen</p> <p>Systemkompetenz: Kennenlernen neuer Programmiersprachen, schnelle Klassifizierung der Eigenschaften, Überblicken der wichtigsten Merkmale und in dieser Sprache innerhalb kurzer Zeit Programme entwickeln können (wenige Tage).</p>
Modulinhalte	<p>Grundbegriffe: Syntax, Semantik, Pragmatik, Typen, Typsysteme usw. Geschichte der Programmiersprachen, Compiler und Interpreter</p> <p>Programmierparadigmen und ihre Besonderheiten</p> <p>Imperatives, Strukturiertes, Prozedurales, Modulares Paradigma Deklaratives, Funktionales, Logisches, Relationales Paradigma Objektorientiertes, Komponentenorientiertes Paradigma</p> <p>Beispiele, wichtige Sprachkonzepte in ausgewählten Programmiersprachen zu den Programmierparadigmen, Referentielle Transparenz, Lambdakalkül</p>
Lehrformen	Vorlesung (2 SWS) und Übung (2 SWS)
Voraussetzungen für die Teilnahme	Empfehlung: Programmierung I und II
Literatur/ multimediale Lehr- und Lernprogramme	<p>Es ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage):</p> <ul style="list-style-type: none"> • Clausen A (2011) Programmiersprachen - Konzepte, Strukturen und Implementierung in Java, Spektrum Akademischer Verlag • Tate A. (2010) Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages (Pragmatic Programmers), O'Reilly UK Ltd.
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 75 Stunden; Selbststudium und selbständige Tätigkeit: 75 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS – Credits Gewichtung: 5 / 180

Leistungsnachweis	Schriftliche Prüfungsleistung 90 Minuten
Semester	4. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	

Modulname	IT-Sicherheit
Modulverantwortlicher/ Modulverantwortliche	Prof. Ralf C. Staudemeyer, Ph.D.
Qualifikationsziele	Ziel dieses Kurses vermittelt die Grundlagen der IT-Sicherheit, Kryptographie und datenschutzfördernder Technologien, sowie deren wirtschaftliche und gesellschaftliche Bedeutung. Absolventen können Risiken besser einschätzen und verstehen die Prinzipien der derzeit verfügbaren Abwehrmechanismen. Sie können IT-Systeme und Netze bezüglich ihrer Sicherheit einstufen und Schutzverfahren erfolgreich in die täglichen Praxis einbinden.
Modulinhalte	Dies Modul ist grundsätzlich wie folgt strukturiert: - Einführung IT-Sicherheit und Privatheit (Fallstudien, Terminologie, Schutzziele, Sicherheitsprobleme) - Angriffe (Schwachstellen, Angriffstypen, Sicherheitsproblem Mensch) - Authentifikation (Biometrie, Challenge-Response-Verfahren, Zugriffskontrolle, Authentifikation in verteilten Systemen) - Grundlagen Kryptographie (Prinzipien, Symmetric- und Public-key Kryptographie, Hashfunktionen, Signaturen, Blockchain) - Privatphäre-fördernde Technologien (Sichere und Anonyme Kommunikationssysteme, Datenbanksicherheit) - Anwenderprogramme für IT-Sicherheit und Privatheit in der täglichen Praxis (Email, Web, Chat, Dateisysteme)
Lehrformen	Vorlesungen (3 SWS), Übungen (1 SWS), Lehrsprache ist Englisch
Voraussetzungen für die Teilnahme	–
Literatur/ multimediale Lehr- und Lernprogramme	- Eckert, C. (2014). IT-Sicherheit. Berlin, München, Boston. De Gruyter. - Paar, C., & Pelzl, J. (2010). Understanding Cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg. - Stallings, W. (2016). Cryptography and network security, principles and practices (7th edition). Prentice Hall. - prüfungsrelevante Literatur wird bekanntgegeben (Änderungen möglich)
Lehrbriefautor	–
Verwendbarkeit	Pflichtmodul im Bachelorstudiengang Informatik. Verwendbar auch für die Bachelorstudiengänge Wirtschaftsinformatik und Multimedia Marketing
Arbeitsaufwand/ Gesamtworkload	Total: 150h. Präsenzstudium: 60h; Selbststudium incl. Prüfungsvorbereitung: 90h
ECTS und Gewichtung der Note in der Gesamtnote	5 (5/180)
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) 90min
Semester	4
Häufigkeit des Angebots	jährlich
Dauer	ein Semester (möglicherweise in Teilen Blockveranstaltung)
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul
Besonderes	Lehrsprache: Englisch (course language: english) Für deutschsprachiges Kursangebot siehe „IT-Sicherheit und Datenschutz“ im Studiengang Multimedia Marketing.

Modulname	Projekt Informatik
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Thomas Heimrich
Qualifikationsziele	Die Studierenden sollen in der Lage sein, <ul style="list-style-type: none"> • ein Problem zu Analysieren • einen Software-Entwurf durchzuführen • eine verteilte Anwendung zu implementieren und zu testen
Modulinhalte	1 Client-Server-Konzept bzw. Konzepte verteilter Systeme 2 Grafische Benutzeroberfläche 3 Datenbankentwurf 4 Datenbankzugriff 5 Testkonzepte
Lehrformen	Projekt
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht.
Literatur/ multimediale Lehr- und Lernprogramm	Es ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> • Balzert, H.: „Lehrbuch der Softwaretechnik“
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 30 Stunden; praktische Tätigkeit: 90 Stunden;
ECTS und Gewichtung der Note in der Gesamtnote	3 ECTS-Credits Gewichtung: 3/180
Leistungsnachweis	alternative Prüfungsleistung
Semester	5. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Pflichtmodul</i>
Besonderes	

Wahlmodule

Modulname	Schlüsselqualifikation
Modulverantwortlicher/ Modulverantwortliche	Rickes Matthias Wohlfarth, Anna
Qualifikationsziele	<ul style="list-style-type: none"> • Studierende erhalten Kompetenzen, welche über die rein fachlichen Kenntnisse und Fähigkeiten hinaus gehen und gesellschaftliche Teilhabe in einem umfassenden Sinne ermöglichen. • Sie sind in der Lage, fachliche Inhalte in einen größeren Kontext einzuordnen und eigene Persönlichkeitsmerkmale hinsichtlich der Berufseignung kritisch zu reflektieren und weiterzuentwickeln. • Sie erhalten konkret Fertigkeiten hinsichtlich Handlungskompetenz, Medienkompetenz, Methodenkompetenz, Selbstkompetenz und Sozialkompetenz.
Modulinhalte	<p>Für Studierende besteht eine freie Auswahl hinsichtlich der zu belegenden Kurse. Das aktuelle Angebot findet sich auf der Internetseite der Hochschule.</p> <p>Das Angebot umfasst beispielsweise:</p> <ul style="list-style-type: none"> • Gender and Diversity • Gesprächsführung • Höflichkeit und Etikette • Interkulturelles Management • Kompetenztraining zur erfolgreichen Beruflichen Orientierung • Konfliktmanagement • Mediation, Konfliktlösung und Verhandlungsmanagement • Motivation und Selbstmanagement • Problemlösungskompetenz • Selbstorganisation • Teamarbeit • Teamfähigkeit • Verhandlungsmanagement • Zeitmanagement • ...
Lehrformen	seminaristische Vorlesung
Voraussetzungen für die Teilnahme	keine formalen Voraussetzungen.
Literatur/ multimediale Lehr- und Lernprogramme	Literaturempfehlungen zu den einzelnen Kursen finden sich auf der Homepage der Hochschule.
Verwendbarkeit	Schlüsselqualifikationen sind in den Bachelorstudiengängen Informatik und Wirtschaftsinformatik entsprechend der Studienordnung als Wahlmodul zu belegen.
Arbeitsaufwand/ Gesamtworkload	je nach Kurs 2 oder 3 ECTS, Präsenzzeit: 30 h , Selbststudium: 30 h bzw. 60 h
ECTS und Gewichtung der Note in der Gesamtnote	Je Kurs 2 ECTS bzw. 3 ECTS an der ECTS-Gesamtanzahl des jeweiligen Studiengangs
Leistungsnachweis	Klausur
Semester	Je nach Studienablaufplan
Häufigkeit des Angebots	Kursangebot in jedem Semester
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlmodul

Modulname	Unternehmensplanspiel
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Regina Polster
Qualifikationsziele	<p>Umsetzung der im Grundstudium erworbenen betriebswirtschaftlichen Kenntnisse, Umgang mit komplexen Entscheidungssituationen in den Bereichen Forschung & Entwicklung, Einkauf, Produktion, Logistik, Marketing, Vertrieb und Rechnungswesen/Controlling, Lösen von Zielkonflikten in der Unternehmensführung, Behebung von Unsicherheiten bei der Entscheidungsfindung.</p> <p>Wissenselemente aus des Grundstudium müssen im Rahmen von simulierten Entscheidungssituationen eingebracht werden. Aufgrund der Begleitung der Veranstaltung werden Defizite und Fragen der Studierenden aufgegriffen und es erfolgt eine problembezogene Wissensvermittlung.</p> <p>Durch das teamorientierte Arbeiten mit Gruppenprojekten unter strikten Zeitvorgaben werden Softskills hinsichtlich Zeitmanagement, Präsentationstechnik und Konfliktmanagement gefördert</p>
Modulinhalte	<ul style="list-style-type: none"> - Simulation von Unternehmensentscheidungen, in einer Wettbewerbssituation - bis zu 5 Gruppen/Unternehmen spielen gegeneinander und messen sich am Unternehmenserfolg. - Periodenbezogen müssen Unternehmensentscheidungen auf Wirtschaftsprognosen und die Spielergebnisse der Vorperiode angepasst werden. - Der Veranstaltungsleiter vermittelt situativ theoretische Wissensinhalte zur Unternehmensführung (Bilanzanalyse, Konkurrenzanalyse, Marketing-Mix, ggf. Insolvenzverfahren, etc.)
Lehrformen	Übung (2 SWS)
Voraussetzungen für die Teilnahme	Formale Voraussetzungen bestehen nicht
Literatur/ multimediale Lehr- und Lernprogramme	Handbücher, Dokumente und Ergebnisse der aktuellen Spielstands des Planspiels TOPSIM General Management
Verwendbarkeit	Pflichtfach in den Studiengängen Wirtschaftsinformatik und Informatik
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 30 Stunden Selbststudium: 15Stunden Prüfung und Prüfungsvorbereitung: 15 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	2 ECTS; einfache Gewichtung 2/180
Leistungsnachweis	Mündliche Prüfungsleistung
Semester	5. Semester
Häufigkeit des Angebots	Jedes Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul

Wahlpflichtmodule

Modulname	Compilerbau
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. T. Heimrich (gehalten durch externen Lehrbeauftragten)
Qualifikationsziele	Die Studierenden lernen die Prinzipien des Compilerbaus und der Konzepte von Spezifikationssprachen kennen. Damit erlangen sie die Fähigkeit mittels Programmgeneratoren effizient, fehlerfreie und wartungsfreundliche Programme für ein breites Problemspektrum syntaxgesteuerter Datentransformationen und Datenextraktionen zu generieren.
Modulinhalte	Lexikalische und syntaktische Analyse von Programmen, semantische Analyse, Typisierung und Scoping, Interpretation und abstrakte Maschinen, Codegenerierung und Optimierung, Fehlerbehandlung, Programmgeneratoren, Spezifikationssprachen, FLEX, BISON
Lehrformen	Vorlesungen, Übungen
Voraussetzungen für die Teilnahme	Keine formalen Voraussetzungen Vorkenntnisse aus dem Modul „Theoretische Informatik“ sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	Ravi Sethi, Monica Lam, Alfred Aho: Compiler. Prinzipien, Techniken und Tools, Pearson Studium Herold: Die Profitools zur lexikalischen und syntaktischen Textanalyse, Pearson Studium
Lehrbriefautor	
Verwendbarkeit	Wahlpflichtmodul im Studiengang Bachelor Informatik.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden Selbststudium: 60 Stunden Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 CP Gewichtung 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) 90 Minuten
Semester	Wahlpflicht im 4. oder 6. Semester
Häufigkeit des Angebots	jedes Studienjahr
Dauer	Das Modul erstreckt sich über ein Semester.
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	Data-Warehouse-Systeme und NoSQL-Systeme
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Thomas Heimrich
Qualifikationsziele	Die Studierenden sollen in der Lage sein, <ul style="list-style-type: none"> • multidimensionale Datenmodellierungen durchzuführen • ein Data Warehouse physisch zu erstellen • Anfragen an ein Data Warehouse zu stellen • Datenmodellierungen für NoSQL-Datenbanken vorzunehmen • Daten in NoSQL-Datenbanksystemen zu speichern und abzufragen
Modulinhalte	<p>1 Data-Warehouse-Systeme</p> <ul style="list-style-type: none"> • multidimensionale Datenmodellierung • Extraktion Transformation und Laden • physischer Entwurf (Partitionierung, Indexierung – B-Baum, <ul style="list-style-type: none"> ○ Bitmap-Indexe, Mehrdimensionale Indexe) • Anfrageverarbeitung und materialisierte Sichten • Data-Mining-Verfahren (Algorithmen) <p>2 NoSQL – Datenbanken (MongoDB)</p> <ul style="list-style-type: none"> • Theoretische Grundlagen zu NoSQL-Datenbanken • Speichern, Anfragen, Ändern von Daten in MongoDB • Anbindung an PHP • JavaScript als serverseitige Programmiersprache für MongoDB <p>3 Verteilte Datenbanken</p> <ul style="list-style-type: none"> • Arten der Fragmentierung • verteilte Anfragen
Lehrformen	Vorlesungen (3 SWS), rechnergestützte Praktika (1 SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse aus dem Modul „Datenbanksysteme“ sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	Den Studierenden wird zu Beginn der Veranstaltung ein umfangreicher Foliensatz mit weiterführenden Hinweisen zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> • V. Köppen, G. Saake, K.-U. Sattler: „Data Warehouse Technologien“ • A. Bauer, H. Günzel: „Data Warehouse Systeme – Architektur, Entwicklung, Anwendung“ • W. Lehner: „Datenbanktechnologie für Data-Warehouse-Systeme – Konzepte und Methoden“
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Informatik“ und „Wirtschaftsinformatik“ ein Wahlpflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium: 60 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) von 90 Minuten
Semester	4. oder 6. Semester

Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Wahlpflichtmodul</i>
Besonderes	

Modulname	Datenbankprogrammierung mit PL/SQL
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Thomas Heimrich
Qualifikationsziele	Die Studierenden sollen in der Lage sein, <ul style="list-style-type: none"> • gespeicherte Prozeduren (Stored Procedures) und nutzerdefinierte Funktionen zu erstellen und zu nutzen • eigene PL/SQL Funktionen in select-Anfragen anzuwenden • Datenbanktrigger zu erzeugen und zu nutzen • gespeicherte Prozeduren und Funktionen in Java-Anwendungen zu verwenden
Modulinhalte	<ol style="list-style-type: none"> 1 Ausführbare Anweisungen erstellen <ul style="list-style-type: none"> • Lexikalische Einheiten in PL/SQL-Blöcken identifizieren • Built In-SQL-Funktionen in PL/SQL verwenden • Verschachtelte Blöcke erstellen und Variablen mit Labels kennzeichnen • Lesbaren Code mit entsprechender Einrückung erstellen • Sequences in PL/SQL-Ausdrücken verwenden 2 Kontrollstrukturen erstellen <ul style="list-style-type: none"> • IF-Anweisungen erstellen • CASE-Anweisungen und CASE-Ausdrücke verwenden • Schleifenanweisungen erstellen und identifizieren • Richtlinien für bedingte Kontrollstrukturen anwenden 3 Stored Procedures und Stored Functions erstellen <ul style="list-style-type: none"> • Zwischen anonymen Blöcken und Unterprogrammen unterscheiden • Einfache Prozeduren erstellen und mit anonymen Blöcken aufrufen • Einfache Funktionen erstellen • Einfache Funktionen erstellen, die Parameter annehmen • Zwischen Prozeduren und Funktionen unterscheiden • Fehlerbehandlung 4 Funktionen erstellen <ul style="list-style-type: none"> • Zwischen Prozeduren und Funktionen unterscheiden • Verwendungsmöglichkeiten von Funktionen beschreiben • Stored Functions erstellen • Funktionen aufrufen • Funktionen entfernen 5 Dynamisches SQL <ul style="list-style-type: none"> • Ausführungsablauf von SQL-Anweisungen beschreiben • SQL-Anweisungen dynamisch mit nativem dynamischen SQL (NDS) erstellen und ausführen • Situationen erkennen, in denen an Stelle von NDS das DBMS_SQL-Package verwendet werden muss, um SQL-Anweisungen dynamisch zu erstellen und auszuführen 6 Trigger erstellen <ul style="list-style-type: none"> • Datenbank-Trigger und ihre Verwendung beschreiben • Verschiedene Trigger-Typen beschreiben • Datenbank-Trigger erstellen • Regeln für das Auslösen von Triggern beschreiben • Datenbank-Trigger entfernen • Trigger-Informationen anzeigen

	<p>7 PL/SQL in Java-Anwendungen nutzen</p> <ul style="list-style-type: none"> • Aufruf von Stored Procedures und Functions aus Java mit Konvertierung der In- und Out-Parameter • Fehlerbehandlung und Fehlerweiterleitung zwischen Oracle und Java
Lehrformen	Vorlesungen (3 SWS), Praktika (1 SWS) und Selbststudium anhand der vorgegebenen Literatur
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse aus dem Modul „Datenbanksysteme“ sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	Den Studierenden wird zu Beginn der Veranstaltung ein umfangreicher Foliensatz mit weiterführenden Hinweisen zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> • Jürgen Sieben: „Oracle PL/SQL: Das umfassende Handbuch für Datenbankentwickler“ • Boobal Ganesan: „Advanced PLSQL Programming: The Definitive Reference“
Lehrbriefautor	
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Informatik“ und „Wirtschaftsinformatik“ ein Wahlpflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium: 60 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) von 90 Minuten
Semester	4. oder 6. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	<i>Wahlpflichtmodul</i>
Besonderes	

Modulname	Deep Learning Architectures				
Modulverantwortlicher	Prof. Dr. Martin Golz				
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> - Problemstellungen der Big-Data-Technologien zu analysieren, - Grundannahmen und Modelle der tiefen Lernarchitekturen zu kennen, - Tiefe Lernarchitekturen zu konzipieren, anzuwenden und zu bewerten, - Training und Deployment tiefer Netze zu verstehen und anzuwenden, - Grundlagen der Regularisierung tiefer Netze zu verstehen, - Deep-Learning-Frameworks anzuwenden. 				
	Inhalt	Kennen, Verstehen	Anwenden	Analysieren, Bewerten	Synthe- tisieren
	Grundlagen künstlicher neuronaler Netze	X			
	Fehlerrückführung	X	X	X	
	Stochastischer und adaptiver Gradientenabstieg	X	X	X	
	Tiefe Auto-Encoder-Netze	X	X	X	
	Tiefe Faltungs-Netze	X	X	X	
	Netze vom Typ des langen Kurzzeit-Gedächtnisses	X	X	X	
Modulinhalte	<ol style="list-style-type: none"> 1. Einführung: Historische Entwicklung des Fachgebiets 2. Künstliche neuronale Netze: Grundbegriffe, Berechnung, Lernregel 3. Tiefe Netze: Überblick zu verschiedenen Architekturen 4. Tiefe Auto-Encoder-Netze: Architektur, Berechnung, Zielfunktion 5. Deep Belief Networks (DBN): Architektur, Berechnung 6. Tiefe Faltungs-Netze (CNN): Architektur, Faltung-, Vereinigungs-, Abbildungs-schichten, Anwendungen 7. Rückgekoppelte neuronale Netze: Topologien, Probleme 8. Langes Kurzzeit-Gedächtnis: Architektur, Berechnung, Gated Recurrent Units 9. Deep-Learning-Frameworks 				
Lehrformen	Vorlesungen und Online-Vorlesungen, Übungen und rechnergestütztes Praktikum				
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen für die Teilnahme bestehen nicht. Kenntnisse in Computergestützter Intelligenz, Linearer Algebra, Analysis, Statistik, Numerischer Mathematik und Software-Entwicklung sind erforderlich.				
Literatur	<p>Mueller, Massaron (2019) Deep Learning. Wiley Goodfellow, Bengio, Courville (2016) Deep Learning. MIT Press Nielsen (2015) Neural Networks and Deep Learning. Determination press</p>				
Lehrbriefautor					
Verwendbarkeit	Bachelorstudiengang Informatik				
Arbeitsaufwand	150 Stunden, davon 60 Präsenzstunden und 90 Stunden Selbststudium				
ECTS, Gewichtung der Note in der Gesamtnote	5 Leistungspunkte Gewichtung der Note in der Gesamtnote: 1 / 36 (5 / 180)				
Leistungsnachweis	Mündliche Prüfung (25 Minuten)				
Semester	4. oder 6. Semester				
Häufigkeit des Angebots	Einmal im Studienjahr				
Dauer	Ein Semester				
Art d. Lehrveranstaltung	Wahlpflichtmodul				
Besonderes					

Modulname	Grundlagen der Webtechnik
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Inform.(FH) René Brothuhn
Qualifikationsziele	Die Studierenden sollen in der Lage sein, <ul style="list-style-type: none"> • den grundsätzlichen Aufbau und die Funktion von modernen (HTML5) Web-Seiten und Web-Anwendungen zu verstehen • Web-Auftritte und Web-Anwendungen zu entwickeln, zu erweitern und zu warten • Web-Oberflächen mit HTML/CSS zu erstellen • mit Hilfe JavaScript/DOM browserseitige Interaktionen zu realisieren • grundlegende Client/Server-Interaktion auf Basis von HTTP verstehen und in Anwendungen nutzen können • geeignete Datenformate für Webanwendungen auswählen können • zukünftige Web-Technologien auswählen und bewerten können
Modulinhalte	<ul style="list-style-type: none"> • Grundlagen von Client/Server-Anwendungen • Entwicklung von Web-Oberflächen mit HTML und CSS • Steuerung der Oberfläche mit JavaScript/DOM und Eventverarbeitung • Web-Kommunikation und die zugrundeliegenden Protokolle, insbesondere HTTP • Einführung in die Serverprogrammierung mit PHP • Erweiterte Web-Interaktionen mit Ajax • Grundlagen des Responsive Webdesigns (RWD) • Grundlagen von Bild-, Audio- und Videoinformationen, sowie deren Datenformate und Standards • Ausblick auf erweiterte Web-Anwendungen mit HTML5-Techniken
Lehrformen	Vorlesungen (3 SWS), Übungen (1 SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Grundlegende Programmierkenntnisse (Prozedural, Objektorientiert), Rechnernetze sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	Den Studierenden wird ein umfangreicher Foliensatz sowie ausführliche Beispiele zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert: <ul style="list-style-type: none"> • http://www.w3schools.com • http://selfhtml.org • "HTML Living Standard" - https://html.spec.whatwg.org • IETF, RFC 2616 "Hypertext Transfer Protocol", https://tools.ietf.org/html/rfc2616 • "PHP Handbuch" - https://www.php.net/manual/de • http://www.selfphp.de • "DIVE INTO HTML5" - Mark Pilgrim - http://diveintohtml5.info • "Computernetzwerke" - Tanenbaum, Andrew S. - Pearson Studium
Lehrbriefautor	
Verwendbarkeit	Das Modul ist in den Bachelorstudiengängen „Informatik“ und „Wirtschaftsinformatik“ ein Wahlpflichtmodul.
Arbeitsaufwand/ Gesamtworkload	150 Stunden, davon Präsenzzeit: 60 Stunden; Selbststudium: 45 Stunden; Prüfung und Prüfungsvorbereitung: 45 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 CP Gewichtung: 5/180
Leistungsnachweis	Mündliche Prüfungsleistung (15min)

Semester	4. oder 6. Semester (Sommersemester)
Häufigkeit des Angebots	Jedes Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	IoT-Security
Modulverantwortlicher/ Modulverantwortliche	Prof. Ralf C. Staudemeyer, Ph.D.
Qualifikationsziele	Dieser Vertiefungskurs vermittelt grundlegende Architekturen, Protokolle und Betriebssysteme im Internet-der-Dinge. Sie machen sich im Anschluss vertraut mit den verschiedenen Bedrohungen die von nicht ordnungsgemäß verwalteten IoT-Geräten ausgehen. Erlernen Sie Tools die gegen IoT-Geräte eingesetzt werden können und implementieren Sie Schutzmassnahmen im Rahmen Ihres Projektes (Gruppenarbeit). Am Ende dieses Kurses werden Sie ein fundiertes Verständnis von IoT-Geräten haben und wissen auf welche Sicherheitsprobleme Sie achten sollten.
Modulinhalte	Die IT-Sicherheit und Privatheit folgender Technologien wird diskutiert: <ul style="list-style-type: none"> - Architekturen des Internet-der-Dinge - Protokolle und Technologien - IoT-Betriebssysteme - Bedrohungen und Schwachstellen - Angriffe und Schutzmassnahmen Die Inhalte dieses Moduls werden fortlaufend aktualisiert und angepasst.
Lehrformen	Vorlesung (2 SWS), Übung (2 SWS), Projekt Lehrsprache ist Englisch.
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse in Rechnerarchitektur, Betriebssysteme und Rechnernetze sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	- Ausgewählte aktuelle Quellen (werden angekündigt)
Lehrbriefautor	
Verwendbarkeit	Wahlmodul im Bachelorstudiengang „Informatik“.
Arbeitsaufwand/ Gesamtworkload	Total: 150h. Präsenzstudium incl. Projekttreffen: 60h; Projektarbeit und Selbststudium: 90h.
ECTS und Gewichtung der Note in der Gesamtnote	5 (5/180)
Leistungsnachweis	Alternative Prüfungsleistung
Semester	6
Häufigkeit des Angebots	jährlich
Dauer	ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlmodul
Besonderes	Lehrsprache: Englisch (course language: english)

Modulname	Mobile Anwendungsentwicklung
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. T. Heimrich (gehalten durch Lehrbeauftragten)
Qualifikationsziele	Die Studierenden sollen in der Lage sein, <ul style="list-style-type: none"> - die spezifischen Anwendungsfälle für mobile Applikationen zu verstehen - die spezifischen Rahmenbedingungen für mobile Anwendungen zu verstehen (z.B. Ressourcenknappheit, hohe Anforderungen an Flexibilität) - den Aufbau eines mobilen Betriebssystems zu verstehen (insbesondere Android) - die Architektur einer mobilen Applikation zu verstehen - eine mobile Anwendung zu entwickeln auf der Basis der Android-API
Modulinhalte	<p>1 Mobile Betriebssysteme</p> <ul style="list-style-type: none"> - Fragmentierung bei Android - Ressourcenmanagement - Schichtenarchitektur von Android <p>2 Entwicklung unter Android</p> <ul style="list-style-type: none"> - Modell View Control (MVC) - Aufbau eines AndroidStudio-Projets - Das Build-System gradle - Virtual Devices <p>3 Views</p> <ul style="list-style-type: none"> - Das View-System - Elementare Views und Container - Listener und Adapter <p>4 Activitys</p> <ul style="list-style-type: none"> - Adaptives Verhalten mobiler Anwendungen - Activity-Lifecycle und Back-Stack - Konfigurationswechsel - Mehrere Activitys und Intents - Intents und Permissions - Fragmente <p>5 Multithreading und Services</p> <ul style="list-style-type: none"> - Nebenläufigkeit in einer MVC-Architektur - Kommunikation mit dem GUI-Thread - Started- und Bound Services <p>6. Aktuelle und weiterführende Themen</p>
Lehrformen	Vorlesung, Übung
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse in Grundlagen in der Java-Programmierung sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> - Kuennenrth, Thomas, <i>Android 8: Das Praxisbuch für Java-Entwickler</i>, Bonn, Rheinwerk 2018 - Marciano, Kristin; Gardner, Brian; Philips, Bill; Steward, Chris, <i>Android Programming: The Big Nerd Ranch Guide</i>, Big Nerd Ranch 2018 - Android Developer, https://developer.android.com/
Lehrbriefautor/en	

Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Wahlpflichtfach.
Arbeitsaufwand/ Gesamtworkload	Gesamt 150 Stunden: Präsenz 60 Stunden, Selbststudium 60 Stunden und Prüfung und Prüfungsvorbereitung 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 90 Minuten
Semester	4. Semester oder 6. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	Mobile Web-Anwendungen
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Inform.(FH) René Brothuhn
Qualifikationsziele	<p>Webanwendungen sind immer häufiger eine alternative gegenüber nativen Anwendungen und Apps, die Studierenden sollen daher,</p> <ul style="list-style-type: none"> • HTML5 zur Entwicklung von plattformunabhängigen (mobilen) Anwendungen nutzen zu können • umfassende Kenntnisse über aktuelle Web-Standards, insbesondere HTML5 haben • komplexe Webanwendungen und Web-Apps entwickeln können • moderne Ansätze wie Single Page Applications (SPA) und Progressive Web Apps (PWA) verstehen und anwenden können • gezielt die spezifischen Eigenschaften (mobiler) Endgeräte nutzen (Displaygröße, Rechen- und Grafikleistung) und darauf ausgerichtete serverseitige Dienste entwickeln • neue Entwicklungen im Bereich der Web-Standards einordnen und für zukünftige Anwendungen nutzen können •
Modulinhalte	<ul style="list-style-type: none"> • erweitertes HTML5 Markup und CSS3 • Erzeugung von geräteunabhängigem Web-Layout (Responsive Webdesign) mit CSS3 Media Queries • erweitertes JavaScript zur Entwicklung von Webanwendungen • Touch-Oberflächen mit HTML5 (Pointer-Events) • Thread-Fähige Clientanwendungen (Web Worker) • Einführung in Web Sockets • Architekturen von Web-Anwendungen, Einführung in Single Page Applications und Progressive Web Apps • JSON Datenformat • HTML5 Offline Anwendungen und Web Storage • Kommunikation zwischen Webanwendungen (Web Messaging und PeerConnection) • Erweiterte Serverprogrammierung mit PHP • serverseitige Verwaltung von Anwendungsdaten, Sessionverwaltung und Cookies • Einführung der erweiterten HTML5-API zum Ansprechen von Kamera, Akku, Kalender, WebRTC, Geolocation usw.
Lehrformen	Vorlesungen (3 SWS), Übungen (1 SWS)
Voraussetzungen für die Teilnahme	<p>Formelle Voraussetzungen bestehen nicht.</p> <p>Kenntnisse zu Grundlagen der Webtechnik sind für eine erfolgreiche Teilnahme erforderlich.</p>
Literatur/ multimediale Lehr- und Lernprogramme	<p>Den Studierenden wird ein umfangreicher Foliensatz sowie ausführliche Beispiele zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert:</p> <p>"HTML Living Standard" - https://html.spec.whatwg.org http://www.w3schools.com http://selfhtml.org https://caniuse.com/ "CSS3" - w3Schools - http://www.w3schools.com/css3 Responsive Patterns – Brad Frost - https://bradfrost.github.io/this-is-responsive/patterns.html</p>
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Wahlpflichtmodul.

Arbeitsaufwand/ Gesamtworkload	150 Stunden, davon Präsenzzeit: 60 Stunden; Selbststudium: 45 Stunden; Prüfung und Prüfungsvorbereitung: 45 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 CP Gewichtung: 5/180
Leistungsnachweis	Alternative Prüfungsleistung
Semester	6. Semester (Sommersemester)
Häufigkeit des Angebots	Jedes Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	Netzwerkplanung und -konfiguration
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr.-Ing. Heinz-Peter Höller
Qualifikationsziele	<ul style="list-style-type: none"> • Spezifische Kenntnisse von Technologien der Schichten zwei und drei, die über das Grundlagenwissen hinausgehen • Analyse der Aufgabenstellung und Abbildung auf ein Netzdesign • Anwenden des Grundlagenwissens zur Planung eines komplexen Netzwerkes, insbesondere Subnetztopologie, Adressplan, Routingkonzept • Konkrete Konfiguration der Netzwerkplanung mithilfe von Switches (Schicht 2), Routern (Schicht 3) und Endsystemen
Modulinhalte	<ul style="list-style-type: none"> • Kabeltechnik • Ethernet-Zugriffsverfahren und Switching • Virtual Local Area Networks (VLANs) • Routing und Inter-VLAN-Routing • Spezialprotokolle (DHCP, NAT) • Netzwerktools (ping, tracer, ipconfig, etc.) • Grundzüge CISCO IOS • Netzwerkplanungsskizzen • Erstellung einer Netzwerkplanung • IP-Subnetzadressplanung
Lehrformen	Vorlesungen und Übungen
Voraussetzungen für die Teilnahme	Rechnernetze
Literatur/ multimediale Lehr- und Lernprogramme	<p>Vorbereitung anhand des Vorlesungsskripts, der angegebenen Literatur und durch die praktischen Prozesse der Planung und späteren Konfiguration</p> <ul style="list-style-type: none"> • Badach, A. & Hoffmann, E. (2019). Technik der IP-Netze. 4. Auflage. Hanser. • Dembowski, K. (2009). Lokale Netze. Handbuch der kompletten Netzwerktechnik. Addison Wesley. • „Cisco 3600 Series - Modular, High-Density Access Routers“, http://www.cisco.com/univercd/cc/td/doc/pcat/3600.htm • "EOL: Catalyst 2926 Series - Multilayer Switches", http://www.cisco.com/univercd/cc/td/doc/pcat/ca2926.htm
Lehrbriefautor	
Verwendbarkeit	Wahlpflichtmodul im Bachelorstudiengang Informatik.
Arbeitsaufwand/ Gesamtworkload	Vorlesung (15 Stunden), Konsultationsstunden und Konfiguration (15 Stunden), Arbeitsgruppen/Selbststudium (60 Stunden), Labor/Konfigurationsvorbereitung (30), Prüfungsvorbereitung und Prüfung (30 Stunden)
ECTS und Gewichtung der Note in der Gesamtnote	5 CP
Leistungsnachweis	Mündliche Prüfung (25 Minuten)
Semester	4. oder 6. Semester (Sommersemester)
Häufigkeit des Angebots	Jedes Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	Scientific Programming
Modulverantwortlicher/ Modulverantwortliche	Prof. Hartmut Seichter, PhD
Qualifikationsziele	<ul style="list-style-type: none"> • Studierende kennen wissenschaftliche Problemstellungen und Analysemethoden um Algorithmen effizient zu implementieren • Kursteilnehmer*Innen haben vertiefende Fähigkeiten Algorithmen mit Hilfe von scientific computing Technologien umzusetzen
Modulinhalte	<ul style="list-style-type: none"> • Einführung C++20 • Einführung in SYCL, CUDA und OpenMP und weiteren Compute-APIs • praktische Introspection und Discovery in heterogenen Systemen • Paradigmen der hardwarenahen Programmierung • Deployment und Profiling • Problemstellungen im Visual Computing, AR Tracking und Computergrafik • Verwendung anderer Programmiersprachen und DSL • Paradigmen im Scientific Computing
Lehrformen	Seminaristische Vorlesung, Übung
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse auch dem Modul Einführung in die Programmierung sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Grundlagen als interaktive H5P Module, Vorlesungsmaterialien</p> <p>Auszug aus der Literaturliste:</p> <ul style="list-style-type: none"> • Alexandrescu, Andrei. Modern C++ Design: Generic Programming and Design Patterns Applied. C++ In-Depth Series. Boston, MA: Addison-Wesley, 2001. • Bast, Radovan, and Roberto Di Remigio. CMake Cookbook. Place of publication not identified: Packt Publishing, 2018. https://www.safaribooksonline.com/library/view/title/9781788470711/?ar?orpq&email=^u. • Hartley, Richard, and Andrew Zisserman. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge, UK ; New York: Cambridge University Press, 2003. • Kirk, David, and Wen-mei Hwu. Programming Massively Parallel Processors: A Hands-on Approach. Third edition. Amsterdam ; Cambridge, MA: Elsevier, 2017. • Meyers, Scott. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. First edition. Beijing ; Sebastopol, CA: O'Reilly Media, 2014. • Pharr, Matt, Wenzel Jakob, and Greg Humphreys. Physically Based Rendering: From Theory to Implementation. Third edition. Cambridge, MA: Morgan Kaufmann Publishers/Elsevier, 2017. • Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs. New York, NY: Springer Science+Business Media, 2019. • Reinders, James, Ben Ashbaugh, James Brodman, Michael Kinsner, John Pennycook, and Xinmin Tian. Data Parallel C++ Mastering DPC++ for Programming of Heterogeneous Systems Using C++ and SYCL, 2021. https://link.springer.com/10.1007/978-1-4842-5574-2.

	<ul style="list-style-type: none"> • Sanders, Jason, and Edward Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming. Upper Saddle River, NJ: Addison-Wesley, 2011. • Sutter, Herb. Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions. Reading, Mass: Addison-Wesley, 2000.
Verwendbarkeit	Modul im Wahlbereich des Bachelor Informatik
Arbeitsaufwand/ Gesamtwirkload	<ul style="list-style-type: none"> • Präsenzzeit: 45h • Selbststudium: 60h • Prüfungsvorbereitung: 45h
ECTS und Gewichtung der Note in der Gesamtnote	5 CP (5 / 180)
Leistungsnachweis	Alternative Prüfungsleistung
Semester	2. oder 4. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahl

Modulname	Softwarequalität
Modulverantwortlicher/ Modulverantwortliche	Dipl.-Math. Gerd Recknagel
Qualifikationsziele	<p>Die Studierenden sollen in der Lage sein,</p> <ul style="list-style-type: none"> • moderne Methoden und Prinzipien des Software Engineering auf bestehende Software anzuwenden. Sie kennen die Möglichkeiten vorhandene Software umzustrukturieren bei gleichzeitigem Erhalt der Funktionalität. Sie können dieses Vorgehen durch gezielten Einsatz von Testfällen absichern. • grundlegende Qualitätsmetriken auf bestehende Software anzuwenden um das Design von Software zu beurteilen, schlechten Quellcode anhand bekannter Muster zu identifizieren und zu verbessern sowie damit das Softwaredesign gezielt zu verändern. • Softwareeigenschaften wie z.B. Erweiterbarkeit, Wiederverwendbarkeit, Testbarkeit einer Software gezielt zu verbessern sowie Umstrukturierungsmaßnahmen nach konkret festgelegten Abläufen anzuwenden und praktisch umzusetzen. • durchzuführende Umbaumaßnahmen durch geeignete Testfälle abzusichern und diese in automatisierten Abläufen wiederholt anzuwenden. Sie sind mit dem Zusammenhang zwischen Refactoring und Test vertraut. • Software testgetrieben zu entwickeln, d.h. sie schreiben zuerst die Testfälle und entwickeln anschließend die Funktionalität.
Modulinhalte	<ul style="list-style-type: none"> • Was ist Refactoring? <ul style="list-style-type: none"> • Ergebnisse, Risiken, Probleme • Refactoring an einem konkreten Beispiel <ul style="list-style-type: none"> • Ableitung von ersten konkreten Regeln, Problemanalyse, Zerlegung, Umverteilung, Ersetzen von Bedingungen durch Polymorphismus, Anwendung des Zustandsmusters • Prinzipien des Refactoring <ul style="list-style-type: none"> • Definitionen • Gründe, Zeitpunkte, Strategien • Grenzen • PRF, Ackermann-Funktion, allg. und partiell rekursive Funktionen • Funktionale Programmierung, Rekursion, Halteproblem in Lisp • Bad Smells <ul style="list-style-type: none"> • Identifikation von Codestrukturen mit Verbesserungsbedarf • Refaktorisierungen <ul style="list-style-type: none"> • typische Fehler und Probleme sowie deren Lösung mit Refaktorisierungsverfahren u.A. Erläuterung, Motivstion und konkretem Vorgehen • Methoden zusammenstellen • Eigenschaften zwischen Objekten verschieben • Umgang mit Generalisierung und Spezialisierung • Einführung Tests <ul style="list-style-type: none"> • Vorurteile, Begriffe, Unit Tests, Testarten, automatisiertes Testen • Test-First-Ansatz <ul style="list-style-type: none"> • testgetriebene Entwicklung, Vorgehen, Vorteile, offene Tests, Testorganisation und -ausführung • Testideen und Heuristiken <ul style="list-style-type: none"> • Entwicklertests, Testfallüberarbeitung, typische Funktionalität, Grenzwerte und Äquivalenzklassen • Integrations- und Interaktionstests • Testumfang, Testoptimierung, Testabdeckung
Lehrformen	Vorlesungen (3 SWS), Übungen (1 SWS)

Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Inhaltliche Voraussetzung ist das Modul Softwareengineering.
Literatur/ multimediale Lehr-und Lernprogramme	Den Studierenden werden Übungsaufgaben und themenspezifische Dokumente in studip bzw. der Übung zur Verfügung gestellt. Darüber hinaus ist folgende Literatur empfehlenswert (jeweils in der neuesten Auflage): <ul style="list-style-type: none"> • Fowler, M.: Refactoring - Improving the Design of Existing Code, Addison-Wesley • Kerievsky, J.: Refactoring to Patterns, Addison Wesley • Wake, W. C.: Refactoring Workbook, Addison Wesley • Dustin / Rashka / Paul: Automated Software Testing, Addison Wesley • Parrington / Roper: Software Test - Ziele, Anwendungen, Methoden, McGraw-Hill • Link, J.: Unit Tests mit Java, dpunkt-Verlag • Fewster / Graham: Software Test Automation, Addison-Wesley
Lehrbriefautor	
Verwendbarkeit	Das Modul ist im Bachelorstudiengang „Informatik“ ein Wahlpflichtfach.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit/Präsenzstudium: 60 Stunden; Selbststudium: 60 Stunden; Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS-Credits Gewichtung: 5/180
Leistungsnachweis	Schriftliche Prüfungsleistung (Klausur) im Umfang von 120 Minuten am Rechner
Semester	4. oder 6. Semester
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	Spieleentwicklung
Modulverantwortlicher/ Modulverantwortliche	Prof. Hartmut Seichter, PhD
Qualifikationsziele	<ul style="list-style-type: none"> • Studierende kennen die grundlegenden Prinzipien und Konzepte von Spielen • Sie haben ein Verständnis von Interaktion und modernen Spieltechnologien • Kursteilnehmer*Innen haben vertiefende Fähigkeiten in der Implementation von Spielekonzepten • Studierende wenden die theoretischen Konzepte anhand einer Open-Source Game Engine an um einen Prototypen zu erstellen
Modulinhalte	<ul style="list-style-type: none"> • Basic Concepts • Game Genres • Principles of Game Design • Specifics of Genre Designs • Storytelling • Level-Design • Puzzle-Concepts • Team Development and Management • <i>Project Management</i>
Lehrformen	Seminaristische Vorlesung (2SWS) mit Übung (2SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse in Graphischer Datenverarbeitung sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr-und Lernprogramme	<p>Vorlesungsmaterialien, Folien</p> <p>Auszug aus der Literaturliste</p> <ul style="list-style-type: none"> • Bates, B., 2004. Game design, 2nd ed. ed. Premier Press, Boston, Mass. • Nystrom, R., Lorenzen, K., 2015. Design Patterns für die Spieleprogrammierung, 1. Aufl. ed. mitp-Verl, Frechen. • Schell, J., 2015. The art of game design: a book of lenses, Second edition. ed. CRC Press, Boca Raton. • Swink, S., 2009. Game feel: a game designer's guide to virtual sensation. Morgan Kaufmann Publishers/Elsevier, Amsterdam ; Boston.
Verwendbarkeit	Wahlpflichtbereich Bachelor Informatik
Arbeitsaufwand/ Gesamtworkload	<ul style="list-style-type: none"> • Präsenzzeit: 45h • Selbststudium: 60h • Prüfungsvorbereitung: 45h
ECTS und Gewichtung der Note in der Gesamtnote	5CP (5/180)
Leistungsnachweis	alternative Prüfungsleistung
Semester	
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtbereich

Modulname	Vertiefung Software-Entwurf
Modulverantwortlicher/ Modulverantwortliche	Prof. Dr. Erwin Neuhardt
Qualifikationsziele	Sie kennen Methoden, um ein Software-System in Teilsysteme, die unabhängig voneinander entwickelt und getestet werden können, zu zerlegen. Sie kennen die grundlegenden Entwurfsmuster für objektorientierte Systeme. Sie können die Methoden anwenden, größere Software-Systeme so zu entwerfen, dass sie wartbar und erweiterbar sind. Sie können typische Entwurfsprobleme innerhalb von Teilsystemen oder zwischen Teilsystemen durch lose gekoppelte Klassen umsetzen und Alternativen abwägen.
Modulinhalte	Aufteilung von Software-Systemen in Komponenten, Beschreibung von Komponenten, webbasierte Benutzungsoberfläche, Zugriff auf relationale Datenbanken mit Java Persistence API, Fehlerbehandlung in großen Software-Systemen, Software-Architektur für Informationssysteme, typische Entwurfsprobleme und ihre Lösung mit Entwurfsmustern, z. B. Adapter, Befehl, Beobachter, Komposition, Zustand, Model-View-Controller.
Lehrformen	Vorlesung (3 SWS) und Übung (1 SWS)
Voraussetzungen für die Teilnahme	Modul Software Engineering
Literatur/ multimediale Lehr- und Lernprogramme	Freeman, E. / Sierra, K. / Bates, B.: Entwurfsmuster von Kopf bis Fuß, O'Reilly, 2005 Gamma, E. / Helm, R. / Johnson, R. / Vlissides, J.: Entwurfsmuster – Elemente wiederverwendbarer Software, Addison-Wesley, 2001 Siedersleben, J.: Moderne Software-Architektur, dpunkt Verlag, 2004 Müller, B., Wehr, H.: Java Persistence API 2, Hanser, 2012 Ball, J. et al.: The Java EE 5 Tutorial, https://docs.oracle.com/javaee/5/tutorial/doc/ , 2010
Lehrbriefautor	
Verwendbarkeit	Wahlpflichtmodul im Bachelorstudiengang Informatik
Arbeitsaufwand/ Gesamtworkload	Präsenzzeit: 60 Stunden Selbststudium: 60 Stunden Prüfung und Prüfungsvorbereitung: 30 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	5 ECTS Gewichtung 5/180
Leistungsnachweis	alternative Prüfungsleistung
Semester	4. oder 6. Semester
Häufigkeit des Angebots	einmal im Studienjahr
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlpflichtmodul
Besonderes	

Modulname	XR Medien
Modulverantwortlicher/ Modulverantwortliche	Prof. Hartmut Seichter, PhD
Qualifikationsziele	<ul style="list-style-type: none"> • Studierende lernen zentrale Konzepte der Mixed, Augmented und Virtual Reality kennen. • Anhand von Anwendungsszenarien können Sie die Interaktion und die notwendigen Techniken verstehen, analysieren und anwenden. • Kursteilnehmer*Innen lernen Sie wichtige Handhabung von Nutzerstudien und Analysen im Rahmen von Immersiven Systemen kennen. • Studierende können Virtual- und Augmented Reality Applikationen mittels einschlägiger 3D Engines implementieren und testen.
Modulinhalte	<ul style="list-style-type: none"> • Tracking und Rekonstruktionstechnologien in Augmented und Virtual Reality • Displaytechnologien • Multimodale Systeme • Tangible User Interfaces • Interaktionstechniken • Technologien der Virtual und Augmented Reality
Lehrformen	Seminaristische Vorlesung (2SWS) mit Übung (2SWS)
Voraussetzungen für die Teilnahme	Formelle Voraussetzungen bestehen nicht. Kenntnisse aus dem Modul Grafische Datenverarbeitung sind für eine erfolgreiche Teilnahme erforderlich.
Literatur/ multimediale Lehr- und Lernprogramme	<ul style="list-style-type: none"> • Bowman, Doug A, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. 3D User Interfaces: Theory and Practice. Addison-Wesley, 2004. • Übungen an Game Engines und VR und AR Technologien • Laborübungen mit Trackingtechnologien • Folien
Verwendbarkeit	Wahlpflichtmodul im Bachelor Informatik
Arbeitsaufwand/ Gesamtworkload	<ul style="list-style-type: none"> • Präsenzzeit: 45h • Selbststudium: 60h • Projektarbeit und Vorbereitung: 45h
ECTS und Gewichtung der Note in der Gesamtnote	5 CP (5/180)
Leistungsnachweis	alternative Prüfungsleistung
Semester	
Häufigkeit des Angebots	Einmal im Studienjahr
Dauer	Ein Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Wahlbereich

Praxismodul

Modulname	Praxismodul
Modulverantwortlicher/ Modulverantwortliche	Betreuer des Praxismoduls
Qualifikationsziele	Die Studierenden erwerben im Praktikum individuelle Fachkenntnisse aus dem jeweils anstehenden Aufgabenfeld. Thematisch sollte die, im Studium erworbene Fachkompetenz, durch konkrete betriebliche Tätigkeiten gefestigt und erweitert werden. Die Studierenden erwerben die Kompetenz im betrieblichen Team zu arbeiten. Dabei erlangen sie die Fähigkeit eigene Ideen in Gruppen zu kommunizieren und nach Möglichkeit durchzusetzen.
Modulinhalte	<ul style="list-style-type: none"> • Die Studierenden sollen entsprechend ihrer zukünftigen Ausrichtung an Aufgaben mitarbeiten und Teilaufgaben selbstverantwortlich ausführen, deren Schwierigkeitsgrad dem Ausbildungsstand angemessen ist. • Auseinandersetzung mit dem Unternehmen, dessen Kunden, den Aufgaben der Praxisstelle und den Bezügen untereinander sowie den konkreten Problemlösungen vor Ort. • Bearbeitung von praktischen Problemlagen und Aufgaben. • Planung und Erstellung der schriftlichen Ausarbeitung. • Diskussion von erarbeiteten Lösungsansätzen für konkrete praxisbezogene Problemstellungen unter Einbeziehung von Wissensbezügen, beispielsweise zu Vorlesungen.
Lehrformen	Schriftliche Ausarbeitung und Problemlösungen vor Ort im Unternehmen
Voraussetzungen für die Teilnahme	Erster Studienabschnitt, Anerkennung des Praktikumsthemas durch den Betreuer.
Literatur/ multimediale Lehr- und Lernprogramme	Abhängig vom Thema/Betreuer
Verwendbarkeit	Das Modul im Bachelorstudiengang „Informatik“ ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Praxisphase/Präsenzstudium Unternehmen 525 Stunden; Kontaktzeit/Präsenzstudium Hochschule 30 Stunden; Selbststudium: 45 Stunden
ECTS und Gewichtung der Note in der Gesamtnote	20 Leistungspunkte Gewichtung der Note in der Gesamtnote: 20/180
Leistungsnachweis	Schriftliche Ausarbeitung
Semester	5. Semester
Häufigkeit des Angebots	Jedes Semester
Dauer	1 Semester
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul

Bachelorarbeit

Modulname	Bachelorarbeit
Modulverantwortlicher/ Modulverantwortliche	Betreuer der Bachelorarbeit
Qualifikationsziele	<ul style="list-style-type: none"> • Herangehensweise an eine wissenschaftliche Problemstellung. • Strukturierung und Eingrenzung eines Themas. • Aufbau einer Gliederung. • Unterstützung bei der fachlichen Lösungsfindung. • Vermittlung von formalen und weiteren Kriterien für die Erstellung einer wissenschaftlichen Arbeit. • Hinweise für die Gestaltung der Präsentation und Verteidigung der Inhalte im Rahmen des Kolloquiums.
Modulinhalte	<p>In der Bachelorarbeit soll der Studierende zeigen, dass die während des Studiums erlernten Kenntnisse und erworbenen Fähigkeiten erfolgreich in die Praxis umgesetzt werden können.</p> <p>Dazu wird eine projektartige Aufgabe unter Einsatz von ingenieurmäßigen Methoden bearbeitet. Der betreuende Professor begleitet die Studierenden während der Bachelorarbeit und leitet sie zum wissenschaftlichen Arbeiten an. Die Arbeit schließt mit einer schriftlichen Ausarbeitung und einem Vortrag ab.</p>
Lehrformen	Selbständiges Erarbeiten einer schriftlichen Ausarbeitung zu einer mit dem Betreuer abgestimmten Themenstellung innerhalb einer vorgegebenen Frist.
Voraussetzungen für die Teilnahme	Erster Studienabschnitt, Anerkennung des Bachelorthemas durch den Betreuer.
Literatur/ multimediale Lehr- und Lernprogramme	<p>Abhängig vom Thema</p> <p>Grundlegende Werke zum wissenschaftlichen Arbeiten sind, z.B.:</p> <ul style="list-style-type: none"> • Helmut Balzert, C.Schäfer, M. Schröder, U.Kern: Wissenschaftliches Arbeiten, 2. Aufl. 2011. • Werner Sesink, Einführung in das wissenschaftliche Arbeiten inklusive E-Learning, Web-Recherche, digitale Präsentation u.a., 9. Aufl. 2012. • Theo Hug, Gerald Poscheschnik: Empirisch Forschen: Über die Planung und Umsetzung von Projekten im Studium, 2010.
Verwendbarkeit	Das Modul im Bachelorstudiengang „Informatik“ ein Pflichtmodul.
Arbeitsaufwand/ Gesamtworkload	Kontaktzeit Hochschule: 10 Stunden; Selbststudium: 350 Stunden.
ECTS und Gewichtung der Note in der Gesamtnote	12 Leistungspunkte Gewichtung der Note in der Gesamtnote: 12/180
Leistungsnachweis	Bewertung der wissenschaftlichen Arbeit und des Kolloquiums (20 Minuten) durch zwei Prüfer
Semester	6. Semester
Häufigkeit des Angebots	Jedes Semester
Dauer	Die Bearbeitungszeit für die Bachelorarbeit beträgt drei Monate.
Art der Lehrveranstaltung (Pflicht, Wahl, etc.)	Pflichtmodul