Technische Hochschule Aschaffenburg
Fakultät für Ingenieurwissenschaften und Informatik

Modulhandbuch

für den Bachelor-Studiengang

Software Design

Erlassen für den Studiengang „Software Design“ der Technischen Hochschule Aschaffenburg durch Eilentscheidung des Dekans vom 26.08.2026 sowie Beschluss des Fakultätsrats der Fakultät Ingenieurwissenschaften und Informatik am xx.xx.2026.

Dieses Modulhandbuch gilt in Verbindung mit der Studien- und Prüfungsordnung vom 01.08.2023 (SPO12), geändert mit Satzung vom 15.07.2025.

Prof. Dr. Vaupel, Dekan

Stand: 26.08.2026

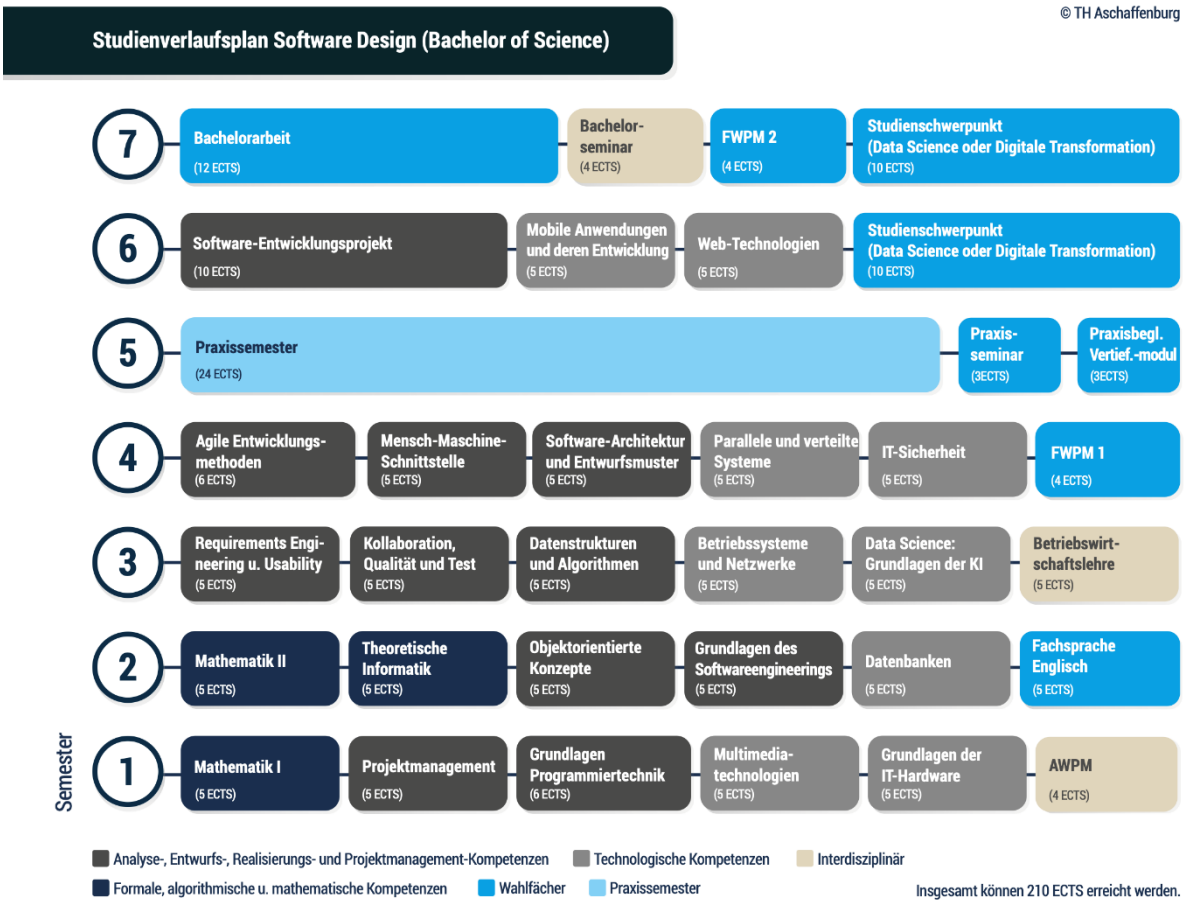
Weitere Informationen zu den Modulen, den Fächern und den jeweiligen Prüfungen und Leistungsnachweisen entnehmen Sie bitte der Studienprüfungsordnung und dem Studienplan Ihres Studiengangs in der jeweils gültigen Fassung

Studienprogramm Software Design

Dieses Modulhandbuch soll Dir helfen, die akademische Struktur des Studiengangs Software Design International (B.Sc.) zu verstehen und Dein Studium fundiert zu planen. Es enthält detaillierte Beschreibungen aller Module, einschließlich Lernziele, Inhalte, Lehr- und Prüfungsformen sowie des erwarteten Arbeitsaufwands (ECTS/Gesamtstunden). Wo relevant, werden außerdem empfohlene Vorkenntnisse, Voraussetzungen und die Einordnung der Module in den Gesamtstudienplan dargestellt.

Der Studiengang ist so aufgebaut, dass Du Dich in klar aufeinander aufbauenden Studienphasen von den Grundlagen bis zur eigenständigen Abschlussleistung entwickelst. In den ersten Semestern erwirbst Du Kernkompetenzen in Programmierung, Mathematik und grundlegender Informatik und schaffst damit die Basis für weiterführende Inhalte. Darauf folgen Module, die Dich entlang des Software-Lebenszyklus qualifizieren – von Anforderungen und Spezifikation über Entwurf und Softwarearchitektur bis hin zu Datenbanken und Netzwerken, Qualitätssicherung und Test sowie IT-Sicherheit. Praxisorientierte Lehrformate und Projektarbeit begleiten diese Entwicklung. Ein zentraler Meilenstein ist das Praxissemester, in dem Du Deine Kenntnisse in einem professionellen Umfeld anwendest und reflektierst. In den späteren Semestern vertiefst Du Dein Profil durch Wahlpflichtmodule und einen Studienschwerpunkt, bevor Du das Studium mit dem Software-Entwicklungsprojekt und der Bachelorarbeit abschließt. Auch wenn KI-gestützte Programmierung zunimmt, bleiben Anforderungen, Spezifikation, Teamarbeit und agile Zusammenarbeit zentral; das Curriculum ist so strukturiert, dass Du diese Fähigkeiten schrittweise entwickelst.

Nutze dieses Handbuch sowohl als Nachschlagewerk als auch als Planungsinstrument: Vergleiche Lernziele und Inhalte der Module, beachte empfohlene Reihenfolgen und Voraussetzungen und wähle Wahlpflichtangebote passend zu Deinen Interessen und beruflichen Zielen.



Inhalt

Modul: SD_01, Mathematik I 4
Modul: SD_02, Mathematik II 5
Modul: SD_03, Grundlagen Programmiertechnik 6
Modul: SD_04, Objektorientierte Konzepte 7
Modul: SD_05, Datenstrukturen und Algorithmen 8
Modul: SD_06, Projektmanagement 9
Modul: SD_07, Grundlagen der IT-Hardware 10
Modul: SD_08, Multimediatechnologie 11
Modul: SD_09, Theoretische Informatik 12
Modul: SD_10, Grundlagen des Software Engineerings 13
Modul: SD_11, Datenbanken 14
Modul: SD_12, Requirements Engineering und Usability 15
Modul: SD_13, Kollaboration, Qualität und Test 16
Modul: SD_14, Betriebssysteme und Netzwerke 17
Modul: SD_15, Data Science: Grundlagen der KI 19
Modul: SD_16, Agile Entwicklungsmethoden 20
Modul: SD_17, Mensch-Maschine-Schnittstelle 21
Modul: SD_18, IT-Sicherheit 22
Modul: SD_19, Software-Architektur und Entwurfsmuster 23
Modul: SD_20, Parallele und verteilte Systeme 24
Modul: SD_21, Fachsprache Englisch 25
Modul: SD_22, Allgemeinwissenschaftlich Wahlpflichtmodul 26
Modul: SD_23, Betriebswirtschaftslehre 27
Modul: SD_24, Fachwissenschaftliches Wahlpflichtmodul 28
Modul: SD_25, Mobile Anwendungen und deren Entwicklung 29
Modul: SD_26, Web-Technologien 30
Modul: SD_27, Software-Entwicklungsprojekt 31
Modul: SD_28, Fachwissenschaftliches Wahlpflichtmodul 32
Modul: SD_29, Praxissemester 33
Modul: SD_30, Praxisbegleitendes Vertiefungsmodul 34
Modul: SD_31, Praxisseminar 35
Modul: SD_32, Seminar zur Bachelorarbeit 36
Modul: SD_33, Bachelorarbeit 37

Modul: SD_01, Mathematik I

Modulbezeichnung	Mathematik I
Kürzel	SD_01
Lehrveranstaltung(en)	LV1: Mathematik I (SU) LV2: Übungen zu Mathematik I (Ü)
Dozierende	Prof. Dr. Galia Weidl
Verantwortliche	Prof. Dr. Klaus Radke, Prof. Dr. Galia Weidl
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 1, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 15h Vorbereitung, 50h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul vermittelt die mathematischen Grundbegriffe und Verfahren, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Diskrete Strukturen und Lineare Algebra Die Studierenden … • formulieren und analysieren Aussagen in der Aussagenlogik und der Logik erster Stufe und übertragen einfache Problemstellungen in formale Modelle. • wenden grundlegende Beweisverfahren (direkter Beweis, Widerspruch, vollständige Induktion) in überschaubaren Kontexten an. • nutzen Mengen, Relationen und Funktionen sicher und untersuchen deren zentrale Eigenschaften. • wenden Konzepte der elementaren Zahlentheorie und Kombinatorik zur Lösung diskreter Problemstellungen an. • führen Rechenoperationen mit Vektoren und Matrizen durch und lösen lineare Gleichungssysteme mit Standardverfahren. • beschreiben grundlegende Eigenschaften von Vektorräumen und linearen Abbildungen in einfachen Beispielen. • modellieren Anwendungsprobleme mithilfe von Graphen und wenden elementare graphentheoretische Methoden in überschaubaren Kontexten an. • erkennen diskrete mathematische Strukturen in informatischen Fragestellungen und stellen Lösungswege formal korrekt dar.
Inhalte	LV1: Diskrete Strukturen und Lineare Algebra: • Aussagenlogik und Logik erster Stufe • Mengen, Relationen, Funktionen • Beweisverfahren • Elementare Zahlentheorie • Elementare Kombinatorik • Vektorräume, Matrizen, Gleichungssysteme • Graphentheorie LV2: • Übungen zu den Inhalten in LV1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: Bearbeitung von Übungsaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Folien
Literatur	• Gerald Teschl, Susanne Teschl: Mathematik für Informatiker, Band 1, Diskrete Mathematik und lineare Algebra, Springer, eXamen.press Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_02, Mathematik II

Modulbezeichnung	Mathematik II
Kürzel	SD_02
Lehrveranstaltung(en)	LV1: Mathematik II (SU) LV2: Übungen zu Mathematik II (Ü)
Dozierende	Prof. Dr. Galia Weidl
Verantwortliche	Prof. Dr. Klaus Radke, Prof. Dr. Galia Weidl,
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 2, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 15h Vorbereitung, 50h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	Empfohlen: SD_01 Mathematik I
Verwendbarkeit des Moduls	Das Modul vermittelt die mathematischen Grundbegriffe und Verfahren, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • erläutern grundlegende Begriffe der Analysis (Konvergenz, Stetigkeit, Differenzierbarkeit, Integrierbarkeit) und wenden diese auf typische Beispiele an. • berechnen Ableitungen und Integrale von Funktionen einer Variablen und interpretieren die Ergebnisse im Anwendungskontext. • bestimmen partielle Ableitungen multivariater Funktionen und nutzen diese zur Lösung einfacher Optimierungsprobleme. • modellieren einfache Anwendungsprobleme mithilfe der Differential- und Integralrechnung. • erläutern den Wahrscheinlichkeitsbegriff einschließlich bedingter Wahrscheinlichkeit und wenden ihn in überschaubaren Kontexten an. • berechnen und interpretieren grundlegende statistische Maße (z. B. Mittelwert, Varianz, Standardabweichung) für gegebene Datensätze. • beschreiben grundlegende diskrete und stetige Wahrscheinlichkeitsverteilungen und setzen sie in einfachen Fragestellungen ein. • führen einfache Verfahren der Parameterschätzung durch und interpretieren die Ergebnisse im jeweiligen Kontext.
Inhalte	LV1: Analysis: • Konvergenz, Stetigkeit, Differenzierbarkeit, Integrierbarkeit • Differential- und Integralrechnung • Multivariate Funktionen Wahrscheinlichkeitstheorie und Statistik: • Wahrscheinlichkeit und bedingte Wahrscheinlichkeit • Statistische Maße • Wahrscheinlichkeitsverteilungen • Parameterschätzung LV2: • Übungen zu den Inhalten in LV1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung:Bearbeitung von Übungsaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Folien
Literatur	• Gerald Teschl, Susanne Teschl: Mathematik für Informatiker, Band 2, Analysis und Statistik, Springer, eXamen.press Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_03, Grundlagen Programmiertechnik

Modulbezeichnung	Grundlagen Programmiertechnik
Kürzel	SD_03
Lehrveranstaltung(en)	LV1: Grundlagen Programmiertechnik (SU) LV2: Übungen zu Grundlagen Programmiertechnik (Ü / Pr)
Dozierende	Prof. Dr. Martin Sträßer
Verantwortliche	Prof. Dr. Alison McNamara
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 1, WiSe
Arbeitsaufwand	Gesamtaufwand: 180 h (davon: Präsenz: 60h, Selbststudium: 120h (davon: 40h Vorbereitung, 40h Nachbereitung, 40h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	6
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul vermittelt die Grundbegriffe und Verfahren der Programmierung, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • erläutern die Bedeutung der Informatik im Kontext der Digitalisierung und des digitalen Wandels • erklären grundlegende Konzepte von Programmiersprachen wie Syntax, Semantik, Datentypen, Speicherstrukturen, Methodenaufrufe und Parameterübergabe am Beispiel einer Programmiersprache. • schreiben syntaktisch und semantisch korrekte Programme nach einem vorgegebenen Algorithmus in der erlernten Programmiersprache. • Erklären wesentliche Entwicklungsschritte für die OO Sprache bis hin zu ausführbaren Programmen. • Führen die Entwicklungsschritte in einer Umgebung (IDE) für die Programmiersprache durch. • implementieren Aufgabenstellungen in eigene Programme in der Programmiersprache. • implementieren einen gegebenen einfachen Such- oder Sortieralgorithmus in der Programmiersprache. • erklären grundlegende objektorientierte Konzepte, wie Klassen, Vererbung und Polymorphie. • implementieren Aufgabenstellungen in eigene Programme in der Programmiersprache.
Inhalte	LV1: • Grundlagen der Informatik (Historie, Bedeutung der Informatik im digitalen Zeitalter, Gebiete der Informatik) • Kodierung von Informationen (ASCII, Unicode), Zahlensysteme (dual, oktal, dezimal, hexadezimal) • Einführung in die Programmierung am Beispiel von Java • Syntax und Semantik der Programmiersprache Java, Kontrollstrukturen, einfache Datentypen und Referenzdatentypen • Grundprinzipien der Objektorientierung (Generalisierung, Vererbung, Kapselung, Polymorphie) • Einfache Datenstrukturen und Algorithmen LV2: • Einführung in die Entwicklungsumgebung Eclipse zur Programmierung • Erstellung von Programmen in o.g. Entwicklungsumgebung in o.g. Programmiersprache • Intensive praktische Anwendung aller vorgestellten Konzepte im Rahmen von vielen praktischen Übungen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung:Bearbeitung von Übungsaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Vorführung, Beispielprogramme, Arbeitsblätter
Literatur	• Bates, K. (2021). Head First Java, 3rd Edition. O'Reilly Media, Inc, Usa. • hristian Ullenboom: Java ist auch eine Insel, 16. Auflage, Rheinwerk-Verlag, Bonn. • Bernhard Steppan (2020). Einstieg in Java mit Eclipse. Hanser. • Deitel, P.J. and Deitel, H.M. (2018). Java: how to program : early objects. New York, Ny Pearson. Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_04, Objektorientierte Konzepte

Modulbezeichnung	Objektorientierte Konzepte
Kürzel	SD_04
Lehrveranstaltung(en)	LV1: Objektorientierte Konzepte (SU) LV2: Übungen zu Objektorientierte Konzepte (Ü / Pr)
Dozierende	Prof. Dr. Alison McNamara
Verantwortliche	Prof. Dr. Alison McNamara
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 2, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	Empfohlen: SD_03 Grundlagen Programmiertechnik
Verwendbarkeit des Moduls	Das Modul vermittelt vertiefende Verfahren der Programmierung, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • skizzieren Programmierparadigmen wie imperative, objektorientierte und deklarative Programmierung und erklären diese an Sprachkonstrukten. • bewerten die Eignung unterschiedlicher Programmierparadigmen und Programmiersprachen für verschiedene Anwendungsaufgaben. • erklären erweiterte objektorientierte Konzepte wie Interfaces, abstrakte Klassen, dynamisches Binden anhand der OO Programmiersprache. • schreiben syntaktisch und semantisch korrekte OO-Programme in der Sprache Java nach einem vorgegebenen Algorithmus und testen dieses. • implementieren eigenständig generische Typen und Collections in eigenen Java-Programmen. • bilden UML-Diagramme auf passende Elemente in Java ab. • schreiben einfache Unit-Tests und entwickeln testgetrieben. • führen Code-Reviews durch, wenden „Clean Code“ Prinzipien und Pair Programming an.
Inhalte	LV1: • Programmiersprachen und Konzepte • Einführung in erweiterte Konzepte der objektorientierten Programmierung am Beispiel von Java (Abstrakte Klassen, Interfaces, Collections, Exception Handling, etc.) • Deployment • Softwaretest, Testgetriebene Entwicklung, Unit Tests • Clean Code Prinzipien LV2: • Softwaretechnische Lösung einer Aufgabenstellung von der Anforderung bis hin zur Umsetzung (inkl. Test) • Praktische Anwendung aller Konzepte im Rahmen von Aufgabenstellungen • Team-Übungen zu den Inhalten der LV 1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung:Bearbeitung von Übungsaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Vorführung, Beispielprogramme, Arbeitsblätter
Literatur	• RRZN Hannover (Hrsg.), Programmierung - Grundlagen (mit Beispielen in Java), Hannover • Bates, K. (2021). Head First Java, 3rd Edition. O'Reilly Media, Inc, Usa. • Christian Ullenboom: Java ist auch eine Insel, 16. Auflage, Rheinwerk-Verlag, Bonn. • Bernhard Steppan (2020). Einstieg in Java mit Eclipse. Hanser. • RRZN Hannover (Hrsg.), Programmierung - Grundlagen (mit Beispielen in Java), Hannover • Deitel, P.J. and Deitel, H.M. (2018). Java : how to program : early objects. New York, Ny Pearson. Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_05, Datenstrukturen und Algorithmen

Modulbezeichnung	Datenstrukturen und Algorithmen
Kürzel	SD_05
Lehrveranstaltung(en)	LV1: Datenstrukturen und Algorithmen (SU) LV2: Übungen zu Datenstrukturen und Algorithmen (Ü / Pr)
Dozierende	Prof. Dr. Barbara Sprick
Verantwortliche	Prof. Dr. Barbara Sprick
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 3, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	Empfohlen: SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte
Verwendbarkeit des Moduls	Das Modul vermittelt vertiefende Algorithmen und Datenstrukturen, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • können die wichtigsten grundlegenden Datenstrukturen und Algorithmen der Informatik erläutern • können das Laufzeitverhalten und Speicherplatzanforderungen von Algorithmen bestimmen und die Effizienz von Algorithmen vergleichen • können grundlegende Datentypen (Listen, Stacks, Queues, Bäume) erklären und diese in Java implementieren. • können den Einfluss einer gewählten Datenstruktur auf die Realisierung eines Algorithmus erläutern • kennen ausgewählte Entwurfsprinzipien und können diese erklären und in kleineren Beispielen anwenden • kennen Strategien zur Auswahl von Algorithmen und können für ausgewählte Standardprobleme in der Informatik geeignete Algorithmen identifizieren und zur Lösung des Problems einsetzen
Inhalte	LV1: • Begriff Algorithmus, O-Notation, Komplexität • Grundlegende Datenstrukturen (Array, Stack, Queue, Listen, …) • Suchalgorithmen • Sortieralgorithmen • Hashverfahren • Bäume • Graphen und Graphalgorithmen LV2: • Team-Übungen und PBL zu den Inhalten der LV 1 • Implementierung der vorgestellten Algorithmen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung, Beispielprogramme, Arbeitsblätter
Literatur	• Thomas H. Cormen, Charles E. Leiserson et al.: Introduction to Algorithms, The MIT Press • Robert Sedgewick, Kevin Wayne: Algorithmen: Algorithmen und Datenstrukturen, Pearson Studium • Martin Dietzfelbinger, Kurt Mehlhorn, Peter Sanders: Algorithmen und Datenstrukturen: Die Grundwerkzeuge, eXamen.press Alle Bücher jeweils in der aktuellen Auflage.
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_06, Projektmanagement

Modulbezeichnung	Projektmanagement
Kürzel	SD_06
Lehrveranstaltung(en)	LV1: Projektmanagement (SU) LV2: Projektmanagement (Ü/Pr)
Dozierende	Prof. Dr. Timea Illes-Seifert
Verantwortliche	Prof. Dr. Timea Illes-Seifert
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 1, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul vermittelt die Grundbegriffe und Verfahren des Projektmanagements, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • kennen die wesentlichen Merkmale eines Projekts und verstehen den Unterschied zwischen einem Projekt und einer Routinetätigkeit. • beschreiben die Produktvision in einem Product Vision Board und wenden die SMART-Schablone zur Zielformulierung an. • diskutieren den Unterschied zwischen klassischer und agiler Projektplanung und -steuerung. • identifizieren Teilaufgaben und fassen sie in einem Projektstrukturplan zusammen. • identifizieren Stakeholder, erkennen deren Einfluss auf den Projekterfolg und erarbeiten Maßnahmen zur deren Steuerung. • ermitteln Risiken und schlagen Maßnahmen zur Risikominderung oder -vermeidung vor. • wenden unterschiedliche Kreativitätstechniken an um Ideen zu entwickeln oder Probleme zu lösen und präsentieren ihre (Zwischen-)Ergebnisse. • planen und steuern ein Beispielprojekt in einem Team und wenden die erlernten Methoden an. • wenden unterschiedliche Methoden zur Retrospektive an.
Inhalte	• Einführung Projektmanagement: Begriffe, Standards, Kennzeichen von Projekten, Projekterfolg • Agiles vs. Klassisches Projektmanagement, Vorgehensmodelle, SCRUM • Projektinitialisierung und -definition: Auftragsklärung, Kick-Off und Startbrainstorming, Produktvision, Projektumfeld, Stakeholderanalyse, Projektziele, Risikomanagement, Anforderungsanalyse • Projektplanung: Projektmanagementphasen, Phasenplan, Meilensteinplanung, Projektstrukturplan, Aufwandsschätzung • Projektsteuerung: Stufen des Projektcontrollings, Projektstatus und –fortschritt, Meilenstein- und Kostentrendanalyse • Projektabschluss: Dokumentation, Lessons Learned • Konfliktmanagement, Zusammenarbeit im Team, Feedback, Kreativitätstechniken • Agiles Projektmanagement am Beispiel SCRUM
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Vorführung, Laborexperimente
Literatur	• Jürg Kuster, Christian Bachmann et al.: Handbuch Projektmanagement: Agil • Klassisch – Hybrid (Deutsch), Springer Gabler • Christian Aichele, Marius Schönberger: IT-Projektmanagement, Springer Vieweg • Manfred Broy, Marco Kuhrmann: Projektorganisation und Management im Software • Engineering, Xpert.press • Kompetenzbasiertes Projektmanagement (PM3), Handbuch für die Projektarbeit, • Qualifizierung und Zertifizierung auf Basis der IPMA, GPM Deutsche Gesellschaft für Projektmanagement • Andreas Johannsen, Anne Kramer et al.: Basiswissen für Softwareprojektmanager im klassischen und agilen Umfeld: Aus- und Weiterbildung zum ASQF® Certified Professional for Project Management (CPPM), dpunkt.verlag GmbH Alle Bücher jeweils in der aktuellen Auflage.
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_07, Grundlagen der IT-Hardware

Modulbezeichnung	Grundlagen der IT-Hardware
Kürzel	SD_07
Lehrveranstaltung(en)	LV1: SD_07a: Grundlagen der IT-Hardware (SU) LV2: SD_07b: Übungen Grundlagen der IT-Hardware (Ü)
Dozierende	Prof. Dr.-Ing. Alexander Biedermann, Prof. Dr. Jörg Abke
Verantwortliche	Prof. Dr.-Ing. Alexander Biedermann
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 1, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul vermittelt die Grundbegriffe und Methoden der IT-Hardware, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • erläutern und bezeichnen Grundkomponenten eines Computers. • ordnen Leistungsmerkmale von Computern und Prozessoren ein. • verstehen und charakterisieren IT-Hardware hinsichtlich ihres inneren Aufbaus (Durchsatz, Latenz, Parallelisierbarkeit, …). • stellen Komponenten einer Hardwareschaltung dar. • vollziehen Übersetzungsschritte eines hochsprachlich verfassten Programms in Maschinencode nach. • stellen ein eigenes Maschinenprogramm auf. • vollziehen die Ausführung von Maschinencode auf einem Prozessor nach. • Skizzieren die Kommunikation zwischen Hardware- und Softwarekomponenten. • nutzen Boolesche Algebra zur Beschreibung und Optimierung von Hardwareschaltungen.
Inhalte	• Komponenten, Funktionsweise und Aufbau von Computersystemen • Struktur von Mikroprozessoren und Speicher • Grundbausteine der Digitaltechnik • Entwurf von Hardwarebausteinen mittels Hardwarebeschreibungssprachen • Speicherorganisation und -hierarchie im Computer • Zusammenhang von Softwareerstellung und Ausführung der Software auf der Hardware • Zusammenspiel von Rechnerkomponenten in heterogenen Architekturen (eingebettete Systeme mit Aktorik/Sensorik, Client-Server-Architekturen, …) • Optimierungsstrategien für Hardware: Caching, Pipelining, Parallelisierung LV1: • Erarbeiten der o.g. Kenntnisse im seminaristischen Unterricht LV 2: • Praktische Anwendung der in LV1 erarbeiteten Konzepte anhand von Arbeitsblättern und Rechnerübungen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Arbeitsblätter, Arbeiten am Rechner
Literatur	• Patterson, Hennessy: Rechnerorganisation und Rechnerentwurf: Die Hardware/Software-Schnittstelle, Walter de Gruyter GmbH & Co KG • Harris, Harris: Digital Design and Computer Architecture, Morgan Kaufmann Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_08, Multimediatechnologie

Modulbezeichnung	Multimediatechnologie
Kürzel	SD_08
Lehrveranstaltung(en)	LV1: SD_08a: Multimediatechnologie LV2: SD_08b: Übungen zu Multimediatechnologie
Dozierende	N. N.
Verantwortliche	Prof. Dr. Alison McNamara
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 1, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul vermittelt die Grundbegriffe und Verfahren der Multimediatechnologie wie sie in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • definieren den Begriff Multimedia und benennen unterschiedliche Typen digitaler Medien sowie deren Spezifika. • beschreiben den Prozess der Digitalisierung und seine Auswirkung auf die Aufnahme und Optimierung digitaler Medien. • referieren und bewerten verschiedene Bild-, Ton- und Videoformate und die bei deren Erstellung verwendeten Codecs und Kompressionsverfahren. • erläutern verfügbare Tools und Workflows zur Konzeption, Gestaltung und Erstellung von multimedialen Inhalten und Anwendungen. • wenden Grundprinzipien der nutzerzentrierten Gestaltung auf den Prozess der Umsetzung von Webanwendungen und mobilen Applikationen an: von der Konzeption über die Aufnahme, Aufbereitung bis zur Publikation und Evaluation. • setzen agile Strategien sowie kreative Tools für den Entwurf von multimedialen Softwareanwendungen für unterschiedliche Zielplattformen ein (Design Thinking und Rapid Prototyping anhand von Scribbles, Wireframes, Click Dummys). • präsentieren überzeugend eigene Konzepte und Umsetzungen und diskutieren selbstkritisch die Stärken und Schwächen von Gestaltungslösungen. • analysieren und diskutieren systematisch eigene und fremde Gestaltungslösungen im Hinblick auf die Anforderungen des Nutzers an Multimedia-Anwendungen und die Mensch-Maschine-Schnittstelle. • meistern die Herausforderungen bei der Entwicklung von Gestaltungslösungen in Projektteams durch Selbstreflexion und aktive Kommunikation innerhalb von ersten Anwendungsprojekten.
Inhalte	LV1: • Definition des Begriffes Multimedia und Überblick über Typen digitaler Medien • Überblick über den Prozess der Digitalisierung: Der Weg von analogen zu digitalen Bild- und Tonsignalen • Überblick über digitale Bild-, Ton- und Videoformate und die Rolle und Funktion von Komprimierung und Codecs • Bedeutung von Eingabe- und Ausgabegeräten für Multimedia-Anwendungen (Controller, Touch-Input, Gesten, Displaytechnologien) • Nutzerzentrierter Einsatz von Multimedia-Inhalten in Webanwendungen und mobilen Applikationen: Prozess der Gestaltung/Aufnahme, Aufbereitung, Publikation und Evaluation • Agile Strategien, Tools und Schritte beim nutzerzentrierten Entwurf von Softwareanwendungen für unterschiedliche Zielplattformen: Design Thinking, Scribbles, Wireframes, Rapid Prototyping LV2: • Nutzerzentrierte und agile Planung, Konzeption, Gestaltung, Umsetzung und Evaluation von Multimedia-Anwendungen für unterschiedliche Zielplattformen • Kritische Analyse und Evaluation der im Rahmen der Übungen erarbeiteten Umsetzungen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: Bearbeitung von Übungsaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Arbeitsblätter, digitale Lehrinhalte
Literatur	• Peter Bühler et al.: Bibliothek der Mediengestaltung, Springer Vieweg Verlag, Bände „Digitales Bild“, „AV-Medien“, „Digital Publishing“, „Präsentation“ Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_09, Theoretische Informatik

Modulbezeichnung	Theoretische Informatik
Kürzel	SD_09
Lehrveranstaltung(en)	LV1: SD_09a: Theoretische Informatik LV2: SD_09b: Übungen zu Theoretische Informatik
Dozierende	Prof. Dr. Barbara Sprick
Verantwortliche	Prof. Dr. Barbara Sprick
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 2, SoSe
Arbeitsaufwand	Gesamtaufwand: 150h (davon Präsenz: 60h, Selbststudium: 90h (davon: 30 Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	empfohlen: Mathematik 1
Verwendbarkeit des Moduls	Das Modul vermittelt die theoretischen Grundlagen der Informatik, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden… • beschreiben und wenden formale Sprachen an. • beschreiben die Grundlagen der Automatentheorie und analysieren gegebene Automaten. • beschreiben Turingmaschinen und wenden diese auf gegebene Problemstellungen an. • beschreiben und wenden reguläre Sprachen an. • wenden grundlegende Prinzipien der Regular Expressions auf gegebene Problemstellungen an. • beschreiben und wenden Grammatiken an. • benennen Laufzeit- und Komplexitätsklassen. • erkennen Laufzeitunterschiede verschiedener Algorithmen. • treffen Aussagen zur Berechenbarkeit und Entscheidbarkeit von Problemstellungen.
Inhalte	• Formale Sprachen • Automatentheorie und Turingmaschinen • Reguläre Sprachen • Grammatiken • Berechenbarkeit und Entscheidbarkeit • Komplexitätstheorie und Komplexitäts-/Laufzeitklassen LV1: Erarbeitung der o.g. Inhalte im seminaristischen Unterricht LV2: Praktische Einübung der in LV1 vermittelten Inhalte anhand von Übungen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Arbeitsblätter
Literatur	• Lutz Priese, Katrin Erk: Theoretische Informatik, Eine umfassende Einführung, Springer Vieweg • Dirk W. Hoffmann, Theoretische Informatik, Hanser • Hopcroft, Motwani, Ullmann: Introduction to Automata Theory, Languages and Computation • Michael Sipser: Introduction to the Theory of Computation Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_10, Grundlagen des Software Engineerings

Modulbezeichnung	Grundlagen des Software Engineerings
Kürzel	SD_10
Lehrveranstaltung(en)	LV1: SD_10a: Grundlagen des Software Engineerings (SU) LV2: SD_10b: Projekt zu Grundlagen des Software Engineerings (Pr/Ü)
Dozierende	Prof. Dr. Timea Illes-Seifert, Prof. Dr. Jóakim von Kistowski
Verantwortliche	Prof. Dr. Timea Illes-Seifert
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 2, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Projekt/Übung)
Kreditpunkte	5
Voraussetzungen	SD_03 Grundlagen Programmiertechnik, SD_06 Projektmanagement
Verwendbarkeit des Moduls	Das Modul vermittelt die Grundbegriffe und Verfahren der Software Engineerings, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • charakterisieren traditionelle Vorgehensmodelle (Wasserfall, V-Modell, V-Modell XT) und agile (Scrum, XP) Vorgehensmodelle. • planen ein Vorgehen für eine Entwicklungsproblemstellung unter Anwendung eines Vorgehensmodells und lösen in Teams Aufgabenstellungen der Softwareentwicklung. • erläutern und wenden Methoden, Sprachen und Werkzeuge für die einzelnen Disziplinen: Spezifikation, Grobentwurf, Feinentwurf, Implementierung und Test an. • ziehen Schlüsse über die Wirksamkeit der erprobten Methoden. • benennen Modellierungselemente und deren korrekte in UML-Syntax. • wenden objektorientierte Analyse und Design an konkreten Beispielen an. • zeigen auf warum Qualitätssicherung eine dauerhafte Aufgabe in der Softwareentwicklung ist und erläutern unterschiedliche Methoden der analytischen und konstruktiven Qualitätssicherung. • moderieren Reviews und geben konstruktives Feedback zum „Testobjekt". • grenzen präventive, perfektive und korrektive Wartung voneinander ab und erläutern warum Testautomation essenziell für eine effiziente Wartung ist. • führen Retrospektiven im Team durch.
Inhalte	LV1: • Grundlagen des Softwareengineerings, Grundbegriffe, Modellierung • Vorgehensmodelle, Prozessmodelle, Bewertung und Verbesserung des Software-Prozesses, agile Softwareentwicklung • Disziplinen des Software Engineerings: Anforderungsanalyse, Architektur und Design, Implementierung, Test • Softwaremanagement: Qualitätsmanagement, Software-Projektmanagement, Konfigurationsmanagement, der Mensch in der Softwareentwicklung, Metriken • Wiederverwendung, Softwarewartung, Reengineering LV2: • Team-Übungen zu den Inhalten der LV 1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Vorführung
Literatur	• Jochen Ludewig, Horst Lichter: Software Engineering: Grundlagen, Menschen, Prozesse, Techniken, dpunkt.verlag GmbH • Ian Sommerville: Software Engineering, Pearson Studium - IT • Andreas Spillner und Tilo Linz: Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB®-Standard (iSQI-Reihe), dpunkt.verlag GmbH • Manfred Broy, Marco Kuhrmann: Projektorganisation und Management im Software • Engineering, Xpert.press • Christine Rupp, Stefan Queins et al.: UML glasklar, Carl Hanser Verlag GmbH & Co. KG • Martin Fowler: Refactoring: Improving the Design of Existing Code, Pearson Addison-Wesley Signature Series Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_11, Datenbanken

Modulbezeichnung	Datenbanken
Kürzel	SD_11
Lehrveranstaltung(en)	LV1: SD_11a: Datenbanken (SU) LV2: Übungen zu Datenbanken (Ü)
Dozierende	Prof. Dr. Barbara Sprick
Verantwortliche	Prof. Dr. Barbara Sprick
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 2, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul ist in den Studiengängen SD, SDI und MEDS verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • können Grundkonzepte und theoretische Grundlagen relationaler Datenbanksysteme und relationaler Anfragesprachen mit Fachbegriffen erklären • können verschiedene Datenbankmodelle voneinander abgrenzen • können den prinzipiellen Aufbau von Datenbankmanagementsystemen mit Fachbegriffen erläutern • diskutieren Datenschutzmechanismen und gesellschaftliche Auswirkungen großer Datensammlungen. • erläutern verschiedene Arten von Wissen und grenzen Wissensverarbeitung von Datenbanken ab. • Können konzeptionelle Datenbankentwürfe entwerfen und diese in normalisierte relationale Datenbankschemata überführen • können komplexe SQL-Anfragen und SQL-Änderungsoperationen durchführen • können kleine Datenbankanwendungen erstellen • schätzen den Wert eines entworfenen Datenbankschemas für einen • Nutzer ein.
Inhalte	LV1: • Grundkonzepte relationaler Datenbanken • Relationale Anfragesprachen, insbesondere SQL • Datenbankmodellierung • Datenbankmanagementsysteme • Normalisierte relationale Datenbankschemata • Mehrbenutzerbetrieb, Transaktionsmanagement, Scheduling • Rechtevergabe in SQL, Zugriffskontrollmodelle • NoSQL LV2: • Übungen zu den Inhalten der LV 1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Vorführung
Literatur	• Datenbanksysteme: Alfons Kemper, André Eickler, De Gruyter Studium, • Übungsbuch Datenbanksysteme: Alfons Kemper, Martin Wimmer, Oldenbourg Verlag • Grundlagen Datenbanksysteme: Ramez Elmasri, Shamkant Navathe, Pearson Studium • Edwin Schicker: Datenbanken und SQL, Springer • Frank Geisler: Datenbanken Grundlagen und Design, mitp.professional • Rene Steiner: Grundkurs Relationale Datenbanken, Springer, Vieweg • Ralf Adams: SQL Der Grundkurs für Ausbildung und Praxis, Hanser Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_12, Requirements Engineering und Usability

Modulbezeichnung	Requirements Engineering und Usability
Kürzel	SD_12
Lehrveranstaltung(en)	LV1: Requirements Engineering und Usability (SU) LV2: Requirements Engineering und Usability (Ü)
Dozierende	N.N.
Verantwortliche	Prof. Dr. Barbara Sprick
Unterrichtssprache	Deutsch/Englisch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 3, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	Empfohlen: SD_10 Grundlagen des Software Engineerings
Verwendbarkeit des Moduls	Das Modul vermittelt Grundbegriffe, Konzepte und Verfahren des Requirements Engineering und Usability Engineering, die in der Softwareentwicklung benötigt und eingesetzt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • definieren und erläutern Kernbegriffe und Fachterminologie aus dem Bereich Requirements Engineering, Requirements Management und Usability Engineering. • benennen, erläutern und unterscheiden Techniken und Methoden zur Ermittlung, Prüfung, Abstimmung, Detaillierung, Verwaltung und Dokumentation von Anforderungen an Softwaresysteme. • wählen für gegebene Projektsituationen eigenständig geeignete Techniken, Methoden und Tools des Requirements Engineering aus und wenden diese an. • setzen den menschzentrierten Gestaltungsprozess im Lebenszyklus von Softwareprojekten planvoll ein, um die Usability, User Experience und Accessibility zu optimieren. • interpretieren, erheben und evaluieren Nutzungskontext und -anforderungen, Effektivität, Effizienz und Zufriedenheit für Softwareprodukte.
Inhalte	LV1: Requirements Engineering und Usability • Grundlagen des Requirements Engineering und der Unternehmensmodellierung • Techniken der Anforderungsermittlung und der Anforderungsdokumentation • Prüfung und Abstimmung von Anforderungen in Softwareprojekten, Methoden und Werkzeuge zur Verwaltung von Anforderungen (Requirements Management) • Definition und Bedeutung von Usability (Gebrauchstauglichkeit), User Experience (UX) und Accessibility (Zugänglichkeit) • Der menschzentrierte Gestaltungsprozess nach DIN EN ISO 9241-210 • Zielgruppen-Analyse, Nutzungsanforderungen und Nutzungskontext • Usability-Tests: Methoden und Einsatz im Entwicklungsprozess und Produktlebenszyklus • Usability-Regelkreis und Qualitätssicherung LV 2: Übungen zu Requirements Engineering und Usability • Team-Übungen zu den Inhalten der LV 1 • Übung der Inhalte aus LV1 anhand eines eigenen Projektes • Einsatz von Design Thinking und Softwaretools
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Vorführung, digitale Lehrinhalte, Arbeiten am Rechner
Literatur	• Klaus Pohl: Requirements Engineering. Grundlagen, Prinzipien, Techniken, dpunkt.verlag • Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering. Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt.verlag • Michael Richter: Usability Engineering: Benutzbare Produkte gezielt entwickeln, Springer Vieweg • Florian Sarodnick, Henning Brau: Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung, Hans Huber Verlag Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_13, Kollaboration, Qualität und Test

Modulbezeichnung	Kollaboration, Qualität und Test
Kürzel	SD_13
Lehrveranstaltung(en)	LV1: Kollaboration, Qualität und Test (SU) LV2: Übungen zu Kollaboration, Qualität und Test (Pr/Ü)
Dozierende	Prof. Dr. Timea Illes-Seifert
Verantwortliche	Prof. Dr. Timea Illes-Seifert
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 3, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Praktikum/Übung)
Kreditpunkte	5
Voraussetzungen	Empfohlen: SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte, SD_05 Datenstrukturen und Algorithmen, SD_06 Projektmanagement, SD_10 Grundlagen des Software Engineerings
Verwendbarkeit des Moduls	Das Modul vermittelt die Grundbegriffe und Verfahren aus den Bereichen Kollaboration, Qualitätsmanagement und Test, die in der angewandten Informatik benötigt werden. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • erläutern Kernbegriffe aus dem Bereich Test und Qualitätssicherung und fassen Aufgaben des Test- und Qualitätsmanagements zusammen. • erklären die Charakteristika unterschiedlicher Qualitätsmanagementsysteme. • erklären die Beziehungen zwischen Softwareentwicklungs- und Testaktivitäten und benennen typische Ziele des Testens. • erläutern und wenden unterschiedliche Black-Box, White-Box und erfahrungsbasierte Testverfahren an. • erklären den Unterschied zwischen statischen und dynamischen Verfahren. • erläutern Zweck von Fehlernachtests und Regressionstests. • fassen Aktivitäten des Reviewprozesses zusammen und wenden ein Reviewverfahren auf ein Arbeitsergebnis an. • fassen Zweck und Inhalt des Testkonzepts zusammen und benennen Faktoren, die den Testaufwand beeinflussen. • wählen für gegebene Projektsituationen eigenständig geeignete Techniken, Methoden und Tools der Qualitätssicherung aus und wenden diese an. • benennen, vergleichen und wenden Methoden und Werkzeuge der kollaborativen Softwareentwicklung an.
Inhalte	LV1 : Kollaboration, Qualität und Test • Grundlagen des Qualitätsmanagements • Qualitätsmodelle und Messen von Softwarequalität • Grundlagen des Testens (Begriffe, Testprozess, Psychologie des Testens) • Testen im Softwareentwicklungszyklus • Statischer Test (Grundlagen, Review-Prozess) • Testverfahren (Black-Box, White-Box, Erfahrungsbasiert) • Test nicht-funktionaler Eigenschaften • Testmanagement • Agiles Testen LV 2: Übungen zu Kollaboration, Qualität und Test • Übungen zu Lehrinhalten aus LV1 • Optionale und freiwillige Zertifizierung zum Certified Tester Foundation Level
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Vorführung, Praktische Arbeiten
Literatur	• Andreas Spillner und Tilo Linz: Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB®-Standard (iSQI-Reihe), dpunkt.verlag GmbH • Kurt Schneider: Abenteuer Softwarequalität. Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement. dpunkt.verlag • Glenford Myers: The Art of Software Testing, John Wiley & Sons, New York NY • Peter Rössler, Maud Schlich, Ralf Kneuper: Reviews in der System- und Software-entwicklung: Grundlagen, Praxis, kontinuierliche Verbesserung, dpunkt.verlag Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_14, Betriebssysteme und Netzwerke

Modulbezeichnung	Betriebssysteme und Netzwerke I
Kürzel	SD_14
Lehrveranstaltung(en)	LV1: Betriebssysteme und Netzwerke (SU) LV2: Übungen zu Betriebssysteme und Netzwerke (Ü/Pr.)
Dozierende	Prof. Dr.-Ing. Jörg Abke
Verantwortliche	Prof. Dr.-Ing. Jörg Abke
Unterrichtssprache	Deutsch/Englisch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 3, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte
Verwendbarkeit des Moduls	Das Modul ist auf die Kompetenzen im Studiengang Software Design abgestimmt. Es ist daher ausschließlich im Studiengang Software Design verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • skizzieren Grundkonzepte und theoretische Grundlagen von Betriebssystemen mit Fachbegriffen. • beschreiben Betriebssystemspezifische Programme (z.B. in einer Shell) und erläutern deren Funktion. • bedienen betriebssystemspezifische Programme und setzen diese ein. • erläutern grundlegende Programmierschnittstellen (API) zu einem Betriebssystem, sie zählen deren Syntax auf und sprechen die Funktionen mit der Programmiersprache Java an. • erläutern die Aufgaben und Funktionen der Schichten des ISO/OSI-Modells und ordnen die wichtigsten Dienstvertreter der jeweiligen Schicht zu. • beschreiben die Funktionsweise des Internetprotokolls im Kern und in den Endsystemen. Sie ordnen die Netzwerkkomponenten den Schichten zu. • geben die Konzepte der Protokolle (z.B. TCP, IP, UDP) wieder und beschreiben deren Funktionsweise z.B. mit Message Sequence Charts. • beschreiben systemspezifische Netzwerkprogramme und deren Verwendung und wenden diese für gegebene Fragestellungen an. • erläutern die Anforderungen an wichtige Protokolle, z.B. für Synchronisation und wechselseitigen Ausschluss, für Konsistenz und Replikation von Daten, für Fehlertoleranz und für Sicherheit. • bewerten Netzwerkprotokolle bzgl. der Anforderungen. • ordnen Kriterien und Größenordnungen für Leistungsmerkmale (z.B. Bandbreite, Latenz) zu Protokollen zu und ermitteln sie mit systemspezifischen Programmen. • entscheiden für ein gegebenes Anwendungsproblem, welche Netztechnologien in den verschiedenen Schichten eingesetzt werden.
Inhalte	LV1: Betriebssysteme: • Grundlegende Konzepte Rechnerarchitektur und Betriebssysteme • Prozesse und Threads • Ressourcenverwaltung (CPU(s), Speicher, Dateisystem, ext. HW) • Benutzungsschnittstellen, inklusive Einführung in die Verwendung von Kommandozeilen (Shells) Rechnernetze: • Struktur von Rechnernetzen • Netzwerkkomponenten • Leistungskriterien (wie z.B. Bandbreite, Latenz, Fehlertoleranz) • Ethernet, Wireless LAN nach IEE 802.11 • wichtige Netzwerk-Protokolle (z. B. IP, TCP, UDP, http, https, DNS, DHCP) • Kommunikationssicherheit: Firewalls, Virtuelle Private Netzwerke LV2: • Übungen zu den Inhalten der LV 1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung, Arbeiten am Computer
Literatur	• Christian Baun: Betriebssyteme kompakt, Springer Vieweg, • Erich Ehses, Lutz Köhler, Petra Riemer, Horst Stenzel, Frank Victor: Systemprogrammierung in UNIX/ Linux, Springer Vieweg

	• Andrew S. Tanenbaum , Todd Austin: Rechnerarchitektur: Von der digitalen Logik zum Parallelrechner, Pearson Studium • Andrew S. Tanenbaum: Moderne Betriebssysteme, Pearson Studium • Eduard Glatz: Betriebssysteme: Grundlagen, Konzepte, Systemprogrammierung, dpunkt Verlag, • Christian Baun: Computernetze kompakt, Springer Vieweg, • Andrew S. Tanenbaum: Computernetzwerke, Pearson Studium • James F. Kurose, Keith W. Ross: Computernetzwerke: Der Top-Down-Ansatz Pearson Studium • Claudia Eckert: IT-Sicherheit: Konzepte - Verfahren – Protokolle, De Gruyter Verlag Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_15, Data Science: Grundlagen der KI

Modulbezeichnung	Data Science: Grundlagen der KI
Kürzel	SD_15
Lehrveranstaltung(en)	LV1: Data Science: Grundlagen der KI (SU) LV2: Übungen zu Data Science: Grundlagen der KI (Ü/Pr.)
Dozierende	Prof. Dr. Klaus Radke
Verantwortliche	Prof. Dr. Klaus Radke
Unterrichtssprache	Deutsch/Englisch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 3, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	keine
Verwendbarkeit des Moduls	Das Modul ist in den Studiengängen SD, SDI und MEDS verwendbar.
Modulziele/Angestrebte Lernergebnisse	**Kenntnisse:** Die Studierenden • unterscheiden Lernverfahren hinsichtlich überwachtem oder unüberwachtem Lernen. • kennen maschinelle Lernverfahren für Klassifikations-, Regressions- und Clusteringprobleme. • können den Data-Science-Prozess beschreiben und die erlernten Methoden den jeweiligen Phasen zuordnen. **Fertigkeiten:** Die Studierenden • wenden Datentransformationsmethoden, deskriptive Statistik und Datenvisualisierung an. • führen das Training eines lernenden Modells aus. • bewerten Leistung und Aussagekraft eines maschinentrainierten Modells. **Kompetenzen:** Die Studierenden sind in der Lage • in Data Science Projekten konstruktiv mitzuarbeiten. kleine Data Science-Projekte eigenständig durchzuführen.
Inhalte	• Grundbegriffe von Data Science: Daten, Modelle, Features, Label, usw. • Vorbereitung der Daten: Einführung in Python, Data Wrangling, Feature Engineering • Daten beschreiben und visualisieren • Ausblick auf weitere Verfahren der Datenvorverarbeitung • Supervised Learning: lineare Regression, k-nächste-Nachbarn-Klassifikation, Entscheidungsbäume, Random Forests, neuronale Netzwerke • Unsupervised Learning: k-Means-Clustering • Ausblick auf weitere Lernverfahren • Auswertungsmetriken: Fehlerrate, Precision, Recall, F1-Score, … Methoden der Reproduzierbarkeit von Datenexperimenten
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung
Literatur	• James, Witten, Hastie, Tibshirani: An Introduction to Statistical Learning, Springer • Raschka et al.: Machine Learning with PyTorch and Scikit-Learn, Packt Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_16, Agile Entwicklungsmethoden

Modulbezeichnung	Agile Entwicklungsmethoden
Kürzel	SD_16
Lehrveranstaltung(en)	LV1: Agile Entwicklungsmethoden (SU) LV2: Übungen zu Agile Entwicklungsmethoden (Ü/Pr.)
Dozierende	Prof. Dr. Timea Illes-Seifert
Verantwortliche	Prof. Dr. Timea Illes-Seifert
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 4, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	6
Voraussetzungen	SD_10 Grundlagen des Software Engineerings, SD_06 Projektmanagement
Verwendbarkeit des Moduls	Das Modul ist auf die Kompetenzen im Studiengang Software Design abgestimmt. Es ist daher ausschließlich im Studiengang Software Design verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • benennen agile und nicht-agile Softwareentwicklungsprozesse. • benennen die Schritte zur agilen Releaseplanung eines SW-Projekts. • wenden Planungsschritte eines agilen SW-Entwicklungsprozesses an • nennen Rollen und Verantwortlichkeiten in einem agilen Prozess und wenden sie an • kennen und nutzen Methoden und Werkzeuge zur Unterstützung agiler Prozessabläufe.
Inhalte	• Überblick Softwareentwicklungsprozesse mit ihren Vor- und Nachteilen • Ausprägungen von Agilität • SW-Projektplanung nach agilen Entwicklungsmethoden wie Scrum und Kanban sowie Möglichkeiten der Skalierung • Rollen und Verantwortlichkeiten in agilen Projekten • Kommunikation und Interaktion in agilen Projektteams • Agile Release- und Sprintplanung, Agile Schätzung und Priorisierung, Akzeptanzkriterien, User Stories, Tasks • Tools zur agilen Projektplanung und -bearbeitung
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung
Literatur	• Boris Gloger: Scrum: Produkte zuverlässig und schnell entwickeln, Carl Hanser Verlag GmbH & Co • Stefan Roock, Henning Wolf: Scrum – verstehen und erfolgreich einsetzen, dpunkt.verlag GmbH • Mike Cohn: Succeeding with Agile: Software Development Using Scrum, Addison Wesley Signature Series • Esther Derby, Diana Larsen: Agile Retrospectives: Making Good Teams Great, Pragmatic Bookshelf Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_17, Mensch-Maschine-Schnittstelle

Modulbezeichnung	Mensch-Maschine-Schnittstelle
Kürzel	SD_17
Lehrveranstaltung(en)	LV1: Mensch-Maschine-Schnittstelle (SU) LV2: Übungen zu Mensch-Maschine-Schnittstelle (Ü/Pr.)
Dozierende	Prof. Dr.-Ing. Alexander Biedermann
Verantwortliche	Prof. Dr.-Ing. Alexander Biedermann
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 4, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	Keine
Verwendbarkeit des Moduls	Das Modul ist auf die Kompetenzen im Studiengang Software Design abgestimmt. Es ist daher ausschließlich im Studiengang Software Design verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • benennen die Grundmuster menschlicher Interaktion mit Maschinen. • identifizieren Affordances und Signifier von Interfaces. • beschreiben das Verhalten von Interfaces über Zustandsmodelle. • erklären Pattern und Dark-Pattern in der Interfacegestaltung. . • ordnen Interfaces hinsichtlich der Verwendung von Pattern ein. • erarbeiten Verbesserungen existierender Interfaces durch Pattern. • vollziehen die Bedeutung responsiven Designs nach. • legen exemplarische Interfaces nach etablierten Gestaltungsprinzipien an. • kennen die Bedeutung guter Interfacegestaltung für die Sicherheit und Ergonomie der Bedienung.
Inhalte	• Historie und Anfänge der Mensch-Maschine-Interaktion • Grundmuster menschlicher Interaktion mit Maschinen • Fehlerquellen und Risikofaktoren der Mensch-Maschine-Schnittstelle • Verhaltensmodellierung von User Interfaces über Zustandsautomaten • Pattern und Anti-Pattern in der Interfacegestaltung • User Interaction, User Experience, hedonische Qualitäten der Mensch-Maschine-Interaktion • Design und Gestaltungsgrundlagen, Skeuomorphismus, Flat Design, Organic Design • Ergonomische Gestaltung von Mensch-Maschine-Schnittstellen, Barrierefreiheit • Zugänglichkeit von Mensch-Maschine-Schnittstellen in Abhängigkeit von kultureller Prägung, Vorerfahrungen, Alter
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung
Literatur	• Lee, John D., Wickens, Christopher D. , Liu, Yili , Boyle, Linda Ng: Designing for people an introduction to human factors engineering, CreateSpace • Don Norman: The Design of Everyday Things, Basic Books Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_18, IT-Sicherheit

Modulbezeichnung	IT-Sicherheit
Kürzel	SD_18
Lehrveranstaltung(en)	LV1: IT-Sicherheit (SU) LV2: Übungen zu IT-Sicherheit (Ü/Pr.)
Dozierende	Prof. Dr. Marie Oetzel
Verantwortliche	Prof. Dr. Marie Oetzel
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 4, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	SD_14 Betriebssysteme und Netzwerke
Verwendbarkeit des Moduls	Das Modul ist auf die Kompetenzen im Studiengang Software Design abgestimmt. Es ist daher ausschließlich im Studiengang Software Design verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • verstehen die Bedeutung der IT-Sicherheit. • erklären Integrität, Vertraulichkeit, Verfügbarkeit. • nennen kryptographische Verfahren zur Verschlüsselung von Informationen. • kennen Maßnahmen zum Schutz von IT-Systemen • kennen die Bestandteile eines Secure Software Development Lifecycle (SDLC). • führen Bedrohungsanalysen (Threat Modeling) durch. • wenden im Rahmen einer CI/CD Pipeline die relevanten Sicherheitsmethoden und -tools an.
Inhalte	• Secure SDLC • Bedrohungsanalysen (Threat Modeling) • Kryptographische Verfahren: Symmetrische und asymmetrische Verschlüsselung, Public Key-Kryptographie • Angriffsvektoren und -methoden auf IT-Systeme • Maßnahmen zum Schutz von IT-Systemen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: Projektarbeit
Medienformen	Tafel, Beamer, Vorführung, Übungen
Literatur	• Howard and Lippner: The Security Development Lifecycle, Microsoft Press. • Long et al.: The CERT Oracle Secure Coding Standard for Java, Addison-Wesley. • Shostack: Threat Modeling: Designing for Security, Wiley. • Anderson: Security Engineering, Wiley. Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_19, Software-Architektur und Entwurfsmuster

Modulbezeichnung	Software-Architektur und Entwurfsmuster
Kürzel	SD_19
Lehrveranstaltung(en)	LV1: Software-Architektur und Entwurfsmuster (SU) LV2: Übungen zu Software-Architektur und Entwurfsmuster (Ü)
Dozierende	Prof. Dr. Jóakim von Kistowski
Verantwortliche	Prof. Dr. Jóakim von Kistowski
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 4, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte, SD_10 Software Engineering, SDI_13 Kollaboration, Qualität und Test
Verwendbarkeit des Moduls	Das Modul ist auf die Kompetenzen im Studiengang Software Design abgestimmt. Es ist daher ausschließlich im Studiengang Software Design verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • beschreiben mindestens eine Modellierungs- und Entwurfssprache (z.B. UML) und deren Inhaltskategorien (wie Diagrammtypen). • beschreiben architekturprägende nicht-funktionale Anforderungen • erläutern das Vorgehen beim Erstellen und Analysieren mittels Entwurfsmustern mithilfe einer Modellierungs- und Entwurfssprache. • verknüpfen Elemente zur Beschreibung von Entwurfsmustern zur Darstellung von Algorithmen. • setzen Entwurfsmustern in eigenen Programmen in einer Programmiersprache um. • benennen Anforderungskriterien und -Regeln an eine Software Architektur (z.B. Redundanzfreiheit, Eindeutigkeit, Kohäsion). • unterscheiden Software-Architekturmuster und Architekturstile anhand der Anforderungskriterien. • klassifizieren Software Architekturen bzgl. der Erfüllung der Anforderungen. • Wägen Architekturentscheidungsalternativen systematisch anhand technischer und organisatorischer Auswirkungen ab • formen eine Software-Architektur unter Verwendung von Entwurfsmustern in einer Programmiersprache.
Inhalte	LV1: • Conway's Law • Software-Qualitätsszenarien • Software Modellierung • Modellierung von Software mittels z.B. UML • Objekt Design Patterns • Architekturentscheidungen und Entscheidungsdokumentation • Software-Architekturmuster • Software-Architekturstile LV2: • Übungen zu den Inhalten der LV 1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: Bearbeitung von Übungsaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Vorführung, Beispielprogramme
Literatur	• Gernot Starke: Effektive Softwarearchitekturen: Ein praktischer Leitfaden, Hanser-Verlag • Oliver Vogel: Software-Architektur, Spektrum Verlag • Joachim Goll: Architektur- und Entwurfsmuster der Softwaretechnik, Springer Verlag • Frank Buschmann, Kevlin Henney, Douglas C. Schmidt: Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing, Wiley Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_20, Parallele und verteilte Systeme

Modulbezeichnung	Parallele und verteilte Systeme
Kürzel	SD_20
Lehrveranstaltung(en)	LV1: Parallele und verteilte Systeme (SU) LV2: Übungen zu Parallele und verteilte Systeme (Ü)
Dozierende	Prof. Dr. Martin Sträßer
Verantwortliche	Prof. Dr.-Ing. Alexander Biedermann
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 4, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	Empfohlen: SD_14 Betriebssysteme und Netzwerke, SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte, SD_05 Datenstrukturen und Algorithmen
Verwendbarkeit des Moduls	Das Modul ist auf die Kompetenzen im Studiengang Software Design abgestimmt. Es ist daher ausschließlich im Studiengang Software Design verwendbar.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • beschreiben Anforderungen an parallele und verteilte Systeme (wie z.B. Antwortzeit, Datenintegrität, Sicherheit, Verfügbarkeit). • erläutern Konzepte, um Anforderungen für parallele und verteilte Systeme zu erfüllen und erarbeiten diese zur Erfüllung der Anforderungen an ein konkretes System. • zählen verschiedene Programmierschnittstellen in einer Sprache auf und bewerten diese hinsichtlich der gestellten Systemanforderungen. • ordnen Schnittstellen und Elemente vorgegebener Betriebs- und Kommunikationssysteme den geforderten Konzepten zu. • setzen die zugeordneten Systemkomponenten in einer Programmiersprache um. • erläutern parallele Programmverarbeitung. • teilen Konzepte von Parallelverarbeitung in Kategorien ein. • beschreiben Konzepte zur Prozesses-Kommunikation und –Synchronisation. • konfigurieren Betriebs- und Kommunikationssystemkomponenten und Services für verteilte Anwendungen. • erstellen Komponenten und Schnittstellen zur Parallelverarbeitung. • interpretieren den Einfluss von Systemparametern in verteilten Systemen bei eigenen Programmentwicklungen.
Inhalte	LV1: • Semaphoren und Mutexe • Sockets • Remote Procedure Calls • Inter-Process Communication • Threads • Prozesse • Prozessmanagement • Hardware-Parallelität • Mehrprozessor-Systeme LV2: • Übungen zu den Inhalten der LV 1
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung, Beispielprogramme
Literatur	• Comer, Douglas E.: Computer Networks and Internets: Global Edition, Pearson Education, • George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems: Concepts and Design, Addison Wesley • Thomas Springer, Alexander Schill: Verteilte Systeme - Grundlagen und Basistechnologien, Pearson Verlag • Rainer Oechsle: Parallele und verteilte Anwendungen in Java, Hanser Verlag Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_21, Fachsprache Englisch

Modulbezeichnung	Fachsprache Englisch
Kürzel	SD_21
Lehrveranstaltung(en)	Fachsprache Englisch (SU/Ü)
Dozierende	Karine Schubert, Angelika Grigor
Verantwortliche	Prof. Dr. Sylvana Krauße
Unterrichtssprache	Englisch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 2, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 35h Nachbereitung, 25h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS
Kreditpunkte	5
Voraussetzungen	Englischkenntnisse auf Niveau der Fachhochschulreife
Verwendbarkeit des Moduls	Das Modul vermittelt Kenntnisse zum Agieren mit der Fremdsprache im internationalen Umfeld. Dieses Modul wird ausschließlich im Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden • besitzen die erforderlichen sprachlichen und kommunikativen Kenntnisse, um diese im fachlichen Kontext anzuwenden. • kommunizieren komplexe technische Zusammenhänge neben der deutschen, insbesondere in englischer Sprache. • entwickeln die soziale Kompetenz im interkulturellen Umfeld weiter. • beherrschen das erlernte Fachvokabular, können es auf neue technische Themenbereiche übertragen und eigenständig weiterentwickeln. • sind in der Lage, englischsprachiger Fachliteratur die benötigten Informationen zu entnehmen, zu analysieren, mündlich wie schriftlich wiederzugeben und die Inhalte auf jeweilige Problemstellungen zu übertragen. • können gängige betriebliche Schriftstücke verfassen und die erworbenen mündlichen Kommunikationsfertigkeiten spontan und sicher in Englisch abrufen.
Inhalte	• Vokabular aus unterschiedlichen technischen Bereichen (ausführliche Erarbeitung und Einübung für vertieftes Verständnis) • grundlegende grammatikalische Formen der Referenzstufen B2/C1 (Überblick und Einübung für vertieftes Verständnis) • Prozessbeschreibungen, Instruktionen, Berichterstattung, Beschreibung von Diagrammen, Objekten und Materialien, Textanalyse, Betriebliche Korrespondenz, Bewerbung (ausführliche Erarbeitung und Einübung für vertieftes Verständnis) • Hörverständnisübungen, Diskussionstechniken, Ergebnispräsentationen, Konversationsübungen unter Einbeziehung Interkultureller Aspekte (ausführliche Erarbeitung und Einübung für vertieftes Verständnis)
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Folien
Literatur	• Bonamy, David: Technical English 3, Pearson Education Ltd. Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_22, Allgemeinwissenschaftlich Wahlpflichtmodul

Modulbezeichnung	Allgemeinwissenschaftliches Wahlpflichtmodul
Kürzel	SD_22
Lehrveranstaltung(en)	Allgemeinwissenschaftliches Wahlpflichtmodul (SU/Ü)
Dozierende	unterschiedliche
Verantwortliche	Beauftragter für die Studienplanung
Unterrichtssprache	Deutsch, abhängig vom gewählten Fach
Zuordnung zum Curriculum, Semester	Software Design, Sem. 1, WiSe
Arbeitsaufwand	Gesamtaufwand: 120 h (davon: Präsenz: 48h, Selbststudium: 72h (davon: 24h Vorbereitung, 28h Nachbereitung, 20h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (Seminaristischer Unterricht/Übung)
Kreditpunkte	4
Voraussetzungen	Je nach gewähltem Wahlpflichtmodul
Verwendbarkeit des Moduls	Wahlpflichtfächer sind i. d. R. in mehreren Studiengängen nutzbar. Detaillierte Beschreibungen finden sich im Modulhandbuch der Wahlpflichtfächer.
Modulziele/Angestrebte Lernergebnisse	Je nach gewähltem Wahlpflichtmodul
Inhalte	Je nach gewähltem Wahlpflichtmodul
Studien- und Prüfungsleistungen	Leistungsnachweis siehe Studienplan
	Bonusleistung: keine
Medienformen	Tafel, Beamer, Folien, weitere nach gewähltem Wahlpflichtmodul
Literatur	Je nach gewähltem Wahlpflichtmodul
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_23, Betriebswirtschaftslehre

Modulbezeichnung	Betriebswirtschaftslehre
Kürzel	SD_23
Lehrveranstaltung(en)	Betriebswirtschaftslehre
Dozierende	Prof. Dr. Gregor Weiche
Verantwortliche	Prof. Dr. Gregor Weiche
Unterrichtssprache	Deutsch / Englisch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 3, WiSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (Seminaristischer Unterricht/Übung/Praktikum)
Kreditpunkte	5
Voraussetzungen	Keine
Verwendbarkeit des Moduls	Dieses Modul kann in anderen Ingenieurstudiengängen eingesetzt werden.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • haben Grundlagenkenntnisse über das Gebiet der allgemeinen Betriebswirtschaftslehre, insbesondere die zentralen wirtschaftlichen Prozesse in einem Unternehmen. • wenden die Grundlagenkenntnisse aus dem Gebiet der Betriebswirtschaftslehre an. • lösen selbständig Aufgabenstellungen und praktische Anwendungen (z.B. Fallbeispiel) unter betriebswirtschaftlicher Sichtweise.
Inhalte	• Unternehmerische Ziele (Überblick) • Unternehmerische Entscheidungen (Überblick) • Betriebliche Funktionen, die Wertschöpfungskette (Überblick) • Ausgewählte thematische Vertiefungen mit Bezug zum Studiengang
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung:keine
Medienformen	Tafel, Beamer, Folien
Literatur	• Griffin Ebert: Business Essentials, Pearson • Schulten: Integrierte Geschäftsprozesse, Espresso Tutorials Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_24, Fachwissenschaftliches Wahlpflichtmodul

Modulbezeichnung	Fachwissenschaftliches Wahlpflichtmodul I
Kürzel	SD_25
Lehrveranstaltung(en)	Fachwissenschaftliches Wahlpflichtmodul I (SU/Ü/Pr)
Dozierende	Unterschiedliche
Verantwortliche	Beauftragter für die Studienplanung
Unterrichtssprache	Deutsch / Englisch (je nach gewähltem Wahlpflichtmodul)
Zuordnung zum Curriculum, Semester	Software Design, Sem. 4, SoSe
Arbeitsaufwand	Gesamtaufwand: 180 h (davon: Präsenz: 72h, Selbststudium: 108h (davon: 36h Vorbereitung, 36h Nachbereitung, 36h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (Seminaristischer Unterricht/Übung/Praktikum)
Kreditpunkte	4
Voraussetzungen	Je nach gewähltem Wahlpflichtmodul
Verwendbarkeit des Moduls	Wahlpflichtfächer sind i. d. R. in mehreren Studiengängen nutzbar. Detaillierte Beschreibungen finden sich im Modulhandbuch der Wahlpflichtfächer.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden erwerben fachspezifische Kenntnisse über die ausgewählten Fachgebiete bzw. weitere fachsprachliche Kenntnisse je nach der gewählten Sprache. Sie wenden die Kenntnisse in einfachen Anwendungen aus den fachspezifischen Gebieten an oder sind in der Lage, weitere sprachliche Fertigkeiten anzuwenden. Die Studierenden können technische Aufgabenstellungen der modulspezifischen Gebiete unter Berücksichtigung von fachübergreifenden Aspekten bearbeiten.
Inhalte	Die Inhalte werden in der Beschreibung der Wahlpflichtmodule angegeben.
Studien- / Prüfungsleistungen	Leistungsnachweis.
	Bonusleistung: Je nach gewähltem Wahlpflichtmodul
Medienformen	Tafel, Folien, Beamer (weitere je nach gewähltem Wahlpflichtmodul)
Literatur	Abhängig vom gewählten Wahlpflichtmodul
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_25, Mobile Anwendungen und deren Entwicklung

Modulbezeichnung	Mobile Anwendungen und deren Entwicklung
Kürzel	SD_25
Lehrveranstaltung(en)	LV1: SD_25a: Mobile Anwendungen und deren Entwicklung LV2: SD_25b: Übungen zu Mobile Anwendungen und deren Entwicklung
Dozierende	Prof. Dr. Alison McNamara
Verantwortliche	Prof. Dr. Alison McNamara
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 6, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60 h, Selbststudium: 90 h (davon: 30 h Vorbereitung, 30 h Nachbereitung, 30 h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte, SD_05 Datenstrukturen und Algorithmen, SD_08 Multimediatechnologie, SD_17 Mensch-Maschine-Schnittstelle
Verwendbarkeit des Moduls	Das Modul vermittelt grundlegende Kenntnisse und Kompetenzen für die Umsetzung von mobilen Anwendungen. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • benennen und erklären die Gemeinsamkeiten und Unterschiede der zentralen mobilen Zielplattformen iOS und Android und der damit verbundenen App-Ökosysteme • analysieren und diskutieren die Vor -und Nachteile von nativen Tools und Cross-Plattform-Frameworks • benennen und unterscheiden die spezifischen Anforderungen und Möglichkeiten mobiler Plattformen und Geräte als Softwareplattform und erklären den damit verbundenen Lebenszyklus von mobilen Anwendungen • wenden die gestalterischen und technischen Vorgaben der Zielplattform Android in eigenen Projekten gezielt an und erarbeiten in Teams eigene App-Prototypen und konzeptionieren beispielhafte Gestaltungslösungen • testen, evaluieren und optimieren die gemeinsam entwickelten Anwendungen anhand der erlernten Tools und Verfahren • präsentieren überzeugend die im Team entwickelten eigenen Anwendungen und analysieren strukturiert und kritisch die Anwendungen anderer Teams und Entwicklerinnen und Entwicklern.
Inhalte	LV1: • Merkmale und Eigenheiten von iOS und Android als zentrale mobile Zielplattformen: Tools, Publishing-Workflows, UI-Guidelines • Spezifische Anforderungen von mobilen Geräten und Plattformen (Performance, Speicherausstattung, Energieverbrauch, Internetanbindung ...) • Überblick zu Cross-Platform-Tools (React Native, Flutter, Xamarin …) hybriden App-Frameworks und deren Vor- und Nachteile • Datenaustausch von mobilen Applikationen mit Webdatenbanken via Restful APIs, lokale/persistente Speicherung von Daten auf mobilen Geräten (SQLite) • Android SDK, Android Studio und die UI-Guidelines und Gestaltungsparadigmen für Android (Material Design) • Sicherheit und Rechteverwaltung von Apps, Signieren von Apps, Zertifikate und Distribution durch App-Stores LV2: • Grundlagen und Beispiele zur Gestaltung und Entwicklung von Android-Apps in Java mittels Android Studio • Umsetzung des Activity Layout Designs und Planung der Lifecycles von Activities • Umgang mit Intents, Broadcasts, Services und Notifications • Zielgerichteter Einsatz von Simulatoren und Endgeräten im mobilen Entwicklungsprozess • Testverfahren und Debugging von mobilen Anwendungen • Kritische Analyse und Evaluation der im Rahmen der Übungen erarbeiteten und präsentierten App-Umsetzungen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: Bearbeitung von Übungsprojekten mit Präsentation
Medienformen	Tafel, Beamer, Arbeitsblätter, digitale Lehrinhalte
Literatur	• Thomas Künneth: Android 11 - Das Praxisbuch für App-Entwickler, Rheinwerk Computing. • Android Developer's Guide: developer.android.com • Material Design Guidelines: material.io Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_26, Web-Technologien

Modulbezeichnung	Web-Technologien
Kürzel	SD_26
Lehrveranstaltung(en)	LV1: SD_26a: Web-Technologien LV2: SD_26b: Übungen zu Web-Technologien
Dozierende	Prof. Dr. Martin Sträßer
Verantwortliche	Prof. Dr. Alison McNamara
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 6, SoSe
Arbeitsaufwand	Gesamtaufwand: 150 h (davon: Präsenz: 60 h, Selbststudium: 90h (davon: 30h Vorbereitung, 30h Nachbereitung, 30h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (LV1: 2 SWS Seminaristischer Unterricht, LV2: 2 SWS Übung)
Kreditpunkte	5
Voraussetzungen	SD_11 Datenbanken, SD_18 IT-Sicherheit
Verwendbarkeit des Moduls	Das Modul vermittelt grundlegende Kenntnisse zu Web-Technologien und Kompetenzen für die Umsetzung von nutzerzentrierten Internetanwendungen. Die Inhalte und Anwendungsbeispiele sind im Wesentlichen auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • benennen und erklären die technischen Grundlagen und Rahmenbedingungen für browserbasierte Anwendungen (Webstandards, Protokolle/Datenaustausch, Client-Server-Prinzip) • grenzen CSS- und JavaScript-Frameworks sowie Backendsprachen voneinander ab und wählen und begründen das adäquate Toolset für Webanwendungen • zeichnen selbst Texte standardkonform in HTML aus, stylen den HTML-Markup per CSS und erstellen (einfache und fortgeschrittene) Interaktionen per JavaScript • setzen Web-Technologien in Rahmen von eigenen Projekten gezielt ein und programmieren in Teams Prototypen und konzeptionieren beispielhafte Gestaltungslösungen unter Berücksichtigung der Best-Practices der Webentwicklung • testen, evaluieren und präsentieren im Team die eigenen Gestaltungslösungen und diskutieren und kritisieren konstruktiv die Lösungen anderer Entwicklerinnen und Entwicklern
Inhalte	LV1: • Überblick zur Internetprotokollfamilie inklusive HTTP/HTTPS, IPv4 und IPv6 • Client-Server-Prinzip als zentrales Element der Internetkommunikation • Einführung Webstandards und Frontend-Stack: HTML, CSS und JavaScript • CSS- und JavaScript-Frameworks (Bootstrap, Angular.js, React) und Publikation von WebApps (PWAs) • Relationale und nicht-relationale Webdatenbanken (MySQL/MongoDB ...) • Restful APIs und JSON als zentrale Schnittstellentechnologien im Web • Backend-Sprachen (Auswahl aus PHP, Perl, Python, Ruby …); Node.js (JavaScript im Backend) • Webbasierte Content-Management-Systeme: Drupal/Typo3/WordPress LV2: • Nutzerzentrierte und agile Planung, Konzeption, Gestaltung, Umsetzung und Evaluation von eigenen Internet-Anwendungen mittels eines ausgewählten Frameworks • Zielgerichtete Anwendung und Vertiefung der in der Vorlesung und in anderen Modulen erarbeiteten Kompetenzen im Rahmen von Web-Projekten
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: Bearbeitung von Projektaufgaben mit Präsentation
Medienformen	Tafel, Beamer, Arbeitsblätter, digitale Lehrinhalte
Literatur	• Philip Ackermann: Professionell entwickeln mit JavaScript - Design, Patterns, Praxistipps, Rheinwerk Computing • Peter Bühler et al.: Bibliothek der Mediengestaltung, Springer Vieweg Verlag, Bände „Webdesign“, „HTML5 und CSS3“, „Webtechnologien“ • Online-Kurse zu HTML, CSS und JavaScript auf www.w3schools.com Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_27, Software-Entwicklungsprojekt

Modulbezeichnung	Software-Entwicklungsprojekt
Kürzel	SD_27
Lehrveranstaltung(en)	LV1: Software-Entwicklungsprojekt LV2: Seminar zu Software-Entwicklungsprojekt
Dozierende	verschiedene
Verantwortliche	Prof. Dr. Timea Illes-Seifert
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 6, SoSe
Arbeitsaufwand	Gesamtaufwand: 300 h (davon: Präsenz: 30h, Projektarbeit: 200h, 40h Seminarvorbereitung (Recherche, Präsentation), 30h Prüfungsvorbereitung))
SWS / Lehrform	2 SWS (LV1: Software-Entwicklungsprojekt, LV2: Seminar zu Software-Entwicklungsprojekt)
Kreditpunkte	10
Voraussetzungen	SD_03 Grundlagen Programmiertechnik, SD_04 Objektorientierte Konzepte, SD_05 Datenstrukturen und Algorithmen, SD_06 Projektmanagement, SD_10 Grundlagen des Software Engineerings, SD_16 Agile Entwicklungsmethoden
Verwendbarkeit des Moduls	Das Modul vertieft Kenntnisse in Spezialdisziplinen des Software Engineerings in Form eines Seminars. Weiterhin verfolgt das Modul die Zielsetzung, erworbene Kenntnisse und Fähigkeiten in einem Softwareentwicklungsprojekt anzuwenden und im Team durchzuführen.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • wenden Kenntnisse über Softwareentwicklung, Softwarequalität und Projektmanagement im Rahmen einer konkreten Problemstellung praktisch an. • entscheiden sich für ein Vorgehensmodell und bestimmen welche Rollen notwendig sind. • planen und steuern alle Aufgaben, die im Rahmen des Projektes durchgeführt werden müssen. • führen eine Anforderungsanalyse durch und leiten hieraus die Software-Architektur sowie das Feindesign ab. • implementieren und testen die erstellte Anwendung. • bewerten im Team unterschiedliche Lösungsalternativen im Bereich Anforderungsanalyse, Design, Implementierung und Test. • führen regelmäßig Retrospektiven durch. • recherchieren und präsentieren ein ausgewähltes, vertieftes Thema aus dem Bereich Software Engineering.
Inhalte	LV1: • Studierende führen in einem Team ein Software-Entwicklungsprojekt durch. Hierbei erfahren sie, welche fachlichen und nicht-fachlichen Herausforderungen in einem Software-Entwicklungsprojekt zu meistern sind. • Die Umsetzung der Aufgabenstellung erfordert die selbstständige Einarbeitung in Entwicklungstools (IDEs, Buildtools, Versionsverwaltung, etc.) und, je nach konkretem Thema, die Analyse verschiedener Libraries oder Frameworks. LV 2: • Recherche und Präsentation eines ausgewählten Themas aus dem Bereich Software Engineering
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung für LV1: keine Bonusleistung für LV2: keine
Medienformen	Tafel, Beamer, Vorführung, Praktische Arbeiten
Literatur	Basisliteratur wie Software Engineering. Spezialliteratur je nach Fokusthema des Seminars. Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_28, Fachwissenschaftliches Wahlpflichtmodul

Modulbezeichnung	Fachwissenschaftliches Wahlpflichtmodul II
Kürzel	SD_28
Lehrveranstaltung(en)	Fachwissenschaftliches Wahlpflichtmodul II (SU/Ü/Pr)
Dozierende	Unterschiedliche
Verantwortliche	Beauftragter für die Studienplanung
Unterrichtssprache	Deutsch / Englisch (je nach gewähltem Wahlpflichtmodul)
Zuordnung zum Curriculum, Semester	Software Design, Sem. 7, WiSe
Arbeitsaufwand	Gesamtaufwand: 120 h (davon: Präsenz: 48h, Selbststudium: 72h (davon: 24h Vorbereitung, 28h Nachbereitung, 20h Prüfungsvorbereitung))
SWS / Lehrform	4 SWS (Seminaristischer Unterricht/Übung/Praktikum)
Kreditpunkte	4
Voraussetzungen	Je nach gewähltem Wahlpflichtmodul
Verwendbarkeit des Moduls	Wahlpflichtfächer sind i. d. R. in mehreren Studiengängen nutzbar. Detaillierte Beschreibungen finden sich im Modulhandbuch der Wahlpflichtfächer.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden erwerben fachspezifische Kenntnisse über die ausgewählten Fachgebiete bzw. weitere fachsprachliche Kenntnisse je nach der gewählten Sprache. Sie wenden die Kenntnisse in einfachen Anwendungen aus den fachspezifischen Gebieten an oder sind in der Lage, weitere sprachliche Fertigkeiten anzuwenden. Die Studierenden können technische Aufgabenstellungen der modulspezifischen Gebiete unter Berücksichtigung von fachübergreifenden Aspekten bearbeiten.
Inhalte	Die Inhalte werden in der Beschreibung der Wahlpflichtmodule angegeben.
Studien- / Prüfungsleistungen	Leistungsnachweis. Bonusleistung: Je nach gewähltem Wahlpflichtmodul
Medienformen	Tafel, Folien, Beamer (weitere je nach gewähltem Wahlpflichtmodul)
Literatur	Abhängig vom gewählten Wahlpflichtmodul
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_29, Praxissemester

Modulbezeichnung	Praxissemester
Kürzel	SD_29
Lehrveranstaltung(en)	Praxissemester
Dozierende	-
Verantwortliche	Prof. Dr.-Ing. Konrad Doll
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 5, WiSe
Arbeitsaufwand	Gesamtaufwand: 720 h
SWS / Lehrform	0 SWS, Praxissemester
Kreditpunkte	24
Voraussetzungen	70 ECTS
Verwendbarkeit des Moduls	Inhalte des Moduls sind auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden... • kennen die betriebliche Arbeitswelt sowie typische Tätigkeiten für eine Software Designerin/Informatikerin bzw. einen Software Designer/ Informatiker. • erwerben Kenntnisse über technische, organisatorische und betriebswirtschaftliche Zusammenhänge in Unternehmen. • sind in der Lage, in Unternehmen adäquat aufzutreten, sich in (Projekt-)Teams einzufügen und mit anderen Mitarbeitern fachlich zusammenzuarbeiten. • sind befähigt, die Unternehmensvorgaben zu verstehen und ihre eigene Tätigkeit daran auszurichten. • arbeiten effektiv und verantwortlich in Unternehmen im Team und mit Partnern auf unterschiedlichen Ebenen zusammen. • übertragen Methoden und Erkenntnisse ihres bisherigen Studiums auf Problemstellungen in die Praxis.
Inhalte	• Praxissemester im Unternehmen. • Inhalt ist abhängig vom Unternehmen und der konkreten Aufgabenstellung.
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Praktische Arbeiten
Literatur	Abhängig vom Unternehmen und der konkreten Aufgabenstellung
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme am betrieblichen Praktikum ist nach Absprache mit der/dem Dozierenden möglich (individuelle Gefährdungsbeurteilung am jeweiligen Arbeitsplatz)

Modul: SD_30, Praxisbegleitendes Vertiefungsmodul

Modulbezeichnung	Praxisbegleitendes Vertiefungsmodul
Kürzel	SD_30
Lehrveranstaltung(en)	Praxisbegleitendes Vertiefungsmodul
Dozierende	Prof. Dr. Fabian Fürst
Verantwortliche	Prof. Dr. Barbara Sprick
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 5, WiSe
Arbeitsaufwand	Gesamtaufwand: 90 h (davon: Präsenz: 30h, Selbststudium: 60h (davon: 20h Vorbereitung, 40h Nachbereitung))
SWS / Lehrform	2 SWS (Seminaristischer Unterricht)
Kreditpunkte	3
Voraussetzungen	keine
Verwendbarkeit des Moduls	Inhalte des Moduls sind auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • geben Fachbegriffe des Arbeitsumfeldes für zukünftiger Software Designer wieder. • erkennen und benennen Stake-Holder in berufspraktischen Arbeitsbereichen. • benennen grundlegende Rahmenbedingungen, Verordnungen und Gesetze im Arbeitsumfeld. • reflektieren Prozesse im Arbeitsumfeld. • ordnen Handlungsempfehlungen und Maßgaben Arbeitsprozessen zu. • identifizieren Herausforderungen im berufspraktischen Umfeld. • entwerfen Maßnahmen zu einem ganzheitlichen Arbeitsumfeld.
Inhalte	• Arbeitsplatz • Arbeitsplatzgestaltung und Umfeld • Arbeitsprozesse • Gesundheitsschutz und Prävention • Qualitätsanforderungen und Normen • Prozessbeschreibungen
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. keine
Medienformen	Tafel, Beamer, Vorführung, Praktische Arbeiten, Präsentation
Literatur	• Martin Klaffke (Hrsg.): Arbeitsplatz der Zukunft, Springer • Rositta Beck: Büro-Effizienz: In 10 Tagen zum schlank organisierten Büro Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_31, Praxisseminar

Modulbezeichnung	Praxisseminar
Kürzel	SD_31
Lehrveranstaltung(en)	Praxisseminar
Dozierende	Prof. Dr. Barbara Sprick
Verantwortliche	Prof. Dr. Barbara Sprick
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 5, WiSe
Arbeitsaufwand	Gesamtaufwand: 90 h (davon: Präsenz: 30h, Selbststudium: 60h (davon: 24h Vorbereitung, 24h Nachbereitung, 12h Prüfungsvorbereitung))
SWS / Lehrform	2 SWS (Seminaristischer Unterricht)
Kreditpunkte	3
Voraussetzungen	Praxissemester
Verwendbarkeit des Moduls	Inhalte des Moduls sind auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden ... • reflektieren, bewerten ihre Tätigkeit in der Unternehmenspraxis und analysieren Stärken und Schwächen und leiten daraus Verbesserungen für das eigene Verhalten ab.
Inhalte	• Aufgaben im Praxissemester
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung. Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung, Praktische Arbeiten
Literatur	-
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_32, Seminar zur Bachelorarbeit

Modulbezeichnung	Seminar zur Bachelorarbeit
Kürzel	SD_32
Lehrveranstaltung(en)	Seminar
Dozierende	verschiedene
Verantwortliche	Prof. Dr.-Ing. Konrad Doll
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 7, WiSe
Arbeitsaufwand	Gesamtaufwand: 60 h
SWS / Lehrform	2 SWS, Besprechungen mit der Betreuerin/dem Betreuer, Präsentation der Ergebnisse
Kreditpunkte	4
Voraussetzungen	SD_33 Bachelorarbeit
Verwendbarkeit des Moduls	Inhalte des Moduls sind auf den Studiengang Software Design abgestimmt. Das Modul wird ausschließlich für den Studiengang Software Design genutzt.
Modulziele/Angestrebte Lernergebnisse	Die Studierenden … • präsentieren, erläutern und verteidigen ihre Ergebnisse in nachvollziehbarer und aktivierender Art und Weise. • reflektieren die eigene Arbeit kritisch. • treten in einen fachlichen Diskurs ein, vertreten Argumente für die eigene Position und diskutieren mit Fachleuten über Themen des Software Design / der Informatik ergebnisorientiert.
Inhalte	• Darstellung der Ergebnisse als Referat/Präsentation mit anschließender Diskussion
Studien- / Prüfungsleistungen	Siehe Studien- und Prüfungsordnung.
	Bonusleistung: keine
Medienformen	Tafel, Beamer, Vorführung
Literatur	Alle Bücher jeweils in der aktuellen Auflage
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich

Modul: SD_33, Bachelorarbeit

Modulbezeichnung	Bachelorarbeit
Kürzel	SD_33
Lehrveranstaltung(en)	-
Dozierende	verschiedene
Verantwortliche	Prof. Dr.-Ing. Konrad Doll
Unterrichtssprache	Deutsch
Zuordnung zum Curriculum, Semester	Software Design, Sem. 7, WiSe
Arbeitsaufwand	Gesamtaufwand: 360 h
SWS / Lehrform	0 SWS
Kreditpunkte	12
Voraussetzungen	Freischaltung durch Studienbüro
Verwendbarkeit des Moduls	Das Modul bündelt die wissenschaftlichen, methodischen und praktischen Kompetenzen, die im Studiengang Software Design erworben wurden.
Modulziele/Angestrebte Lernergebnisse	Im Rahmen der Bachelorarbeit sollen die Studierenden ihre im Rahmen des Studiums erworbenen Fach-, Methoden-, Sozial- und Selbst-Kompetenzen auf ein praxisrelevantes Problem aus dem Studiengang Software Design selbstständig und auf wissenschaftlicher Grundlage anwenden. Die Studierenden ... • bearbeiten ein Problem selbstständig aus dem Bereich des Software Designs (Informatik) innerhalb einer vorgegebenen Frist nach wissenschaftlichen Methoden unter Anleitung und unter Anwendung des einschlägigen Theorie-und Methodenwissens. • zeigen für die Problemlösung notwendiges Selbst-, Zeit-und Projektmanagement. • dokumentieren die Grundlagen, eingesetzte Methoden und Ergebnisse gemäß wissenschaftlicher Standards. • vertiefen ihre Problemlösungskompetenz. Ihnen gelingt der zielgerichtete Transfer des Theorie- und Methodenwissens des Software Designs in Anwendungsbereiche der Praxis. • analysieren kritisch die Prozesse und Herausforderungen in der Informatik. • beleuchten Probleme und Lösungsansätze aus interdisziplinärer Perspektive. • denken vernetzt und streben nach praxistauglichen und nachhaltigen Lösungen.
Inhalte	Der Studierenden verfassen eine schriftliche Arbeit, welche in der Regel die Ergebnisse einer theoretischen oder experimentellen Untersuchung oder einer praktischen Entwicklungsaufgabe präsentiert. Die Bearbeitungsschritte werden individuell in Abhängigkeit vom jeweiligen Thema festgelegt. Mögliche Schritte können dabei sein: • Einarbeitung in die Thematik und in den aktuellen Forschungsstand/Stand der Technik • Erarbeitung/Auswahl der Methoden und Techniken zur Problemlösung und Entwicklung eines Lösungskonzeptes • Einsatz/Durchführung der Methode und Realisierung des eigenen Konzeptes • Auswertung, Validierung und Bewertung der Ergebnisse
Studien- / Prüfungsleistungen	Bachelorarbeit Bonusleistung: keine
Medienformen	-
Literatur	-
Gefährdungsbeurteilung für schwangere oder stillende Studierende	Teilnahme ist möglich