

Angewandte Informatik

Version vom 21.01.2021 für die Prüfungs- und Studienordnung vom 21.01.2021
Beschlissen im Fakultätsrat Technik und Informatik am 17.12.2020

Pflichtfächer

1. Semester
 - [Diskrete Mathematik](#)
 - [Grundlagen der Informatik](#)
 - [Programmiermethodik I](#)
 - [Programmiertechnik](#)
 - [Betriebswirtschaftslehre](#)
2. Semester
 - [Logik und Berechenbarkeit](#)
 - [Automatentheorie und Formale Sprachen](#)
 - [Datenbanken](#)
 - [Programmiermethodik II](#)
 - [Rechnerstrukturen und maschinennahe Programmierung](#)
3. Semester
 - [Graphentheoretische Konzepte und Algorithmen](#)
 - [Algorithmen und Datenstrukturen](#)
 - [Software Engineering I](#)
 - [Betriebssysteme](#)
 - [Data Science](#)
4. Semester
 - [Künstliche Intelligenz](#)
 - [Software Engineering II](#)
 - [Rechnernetze](#)
 - [Englisch](#)
 - [Ethik](#)
5. Semester
 - [Projekt](#)
 - [Seminar](#)
 - [Softwarearchitektur](#)
 - [Verteilte Systeme](#)
6. Semester
 - [IT-Sicherheit](#)
 - [Bachelorarbeit](#)

Wahlpflichtfächer

- [Wahlpflichtfach I, II und III](#)
- [Gesellschaftswissenschaften](#)

Prüfungsformen

Entsprechend § 14 APSO-INGI, jeweils in der geltenden Fassung, werden die Prüfungsformen für das anschließende Modulhandbuch wie folgt definiert:

1. Fallstudie (FS)

Die Fallstudie ist eine schriftliche Arbeit mit begründeter Lösung. In einer Fallstudie werden einzeln oder in Gruppen durch die Anwendung wissenschaftlicher Methoden und Erkenntnisse Praxisprobleme erfasst, analysiert und gelöst. Die Bearbeitung erfolgt veranstaltungsbegleitend. Die Bearbeitungszeit endet spätestens mit dem Ablauf der Lehrveranstaltung in dem jeweiligen Semester. Die Bearbeitungsdauer kann in den studiengangsspezifischen Prüfungs- und Studienordnungen näher geregelt werden.

2. Hausarbeit (H)

Eine Hausarbeit ist eine nicht unter Aufsicht anzufertigende schriftliche Ausarbeitung, durch die die oder der Studierende die selbstständige Bearbeitung eines gestellten Themas nachweist. Die Bearbeitungszeit einer Hausarbeit beläuft sich auf bis zu drei Monate. Handelt es sich bei der Hausarbeit um eine Prüfungsleistung, dann kann in der studiengangsspezifischen Prüfungs- und Studienordnung bestimmt werden, ob nach Abgabe der schriftlichen Ausarbeitung innerhalb einer Frist von in der Regel einem Monat ein Kolloquium zu halten ist. Die Dauer des Kolloquiums beträgt mindestens 15, höchstens 45 Minuten.

3. Klausur (K)

Eine Klausur ist eine unter Aufsicht anzufertigende schriftliche Arbeit, in der die Studierenden ohne Hilfsmittel oder unter Benutzung der zugelassenen Hilfsmittel die gestellten Aufgaben allein und selbstständig bearbeiten. Die Dauer einer Klausur beträgt mindestens 60, höchstens 240 Minuten.

4. Kolloquium (KO)

Ist bei einzelnen Prüfungsarten, der Bachelor- oder Masterarbeit ein Kolloquium vorgesehen, so handelt es sich dabei um ein Prüfungsgespräch, in dem die Studierenden in freier Rede darlegen müssen, dass sie den Prüfungsstoff beherrschen. Das Kolloquium ist ein Prüfungsgespräch von mindestens 15 und höchstens 45 Minuten Dauer, welches auch dazu dient, festzustellen, ob es sich bei der zu erbringenden Leistung um eine selbstständig erbrachte Leistung handelt. Kolloquien können als Einzelprüfung oder als Gruppenprüfung durchgeführt werden. Bei Gruppenprüfungen ist die Gruppengröße bei der Festlegung der Prüfungsdauer angemessen zu berücksichtigen.

5. Konstruktionsarbeit (KN)

Eine Konstruktionsarbeit ist eine schriftliche Arbeit, durch die anhand fachpraktischer Aufgaben die konstruktiven Fähigkeiten unter Beweis zu stellen sind. Die Bearbeitungszeit beträgt höchstens drei Monate.

6. Laborabschluss (LA)

Ein Laborabschluss ist erfolgreich erbracht, wenn die Studierenden die von der Prüferin oder dem Prüfer festgelegten experimentellen Arbeiten innerhalb des Semesters erfolgreich durchgeführt haben und ihre Kenntnisse durch versuchsbegleitende Kolloquien und/oder anhand von Protokollen und/oder durch schriftliche Aufgabenlösungen nachgewiesen haben. Die Dauer des Kolloquiums beträgt mindestens 15, höchstens 45 Minuten. Die schriftlichen Ausarbeitungen sind innerhalb einer von der Prüferin bzw. dem Prüfer festgesetzten Frist abzugeben. Diese Frist endet spätestens mit Ablauf des jeweiligen Semesters, in dem die zugeordnete Lehrveranstaltungsart durchgeführt wird.

7. Laborprüfung (LR)

Eine Laborprüfung besteht aus einem Laborabschluss und am Ende der Lehrveranstaltung aus einer abschließenden Überprüfung der Leistung. Bei dieser Überprüfung sollen die Studierenden eine experimentelle Aufgabe allein und selbstständig lösen. Die Dauer der Überprüfung beträgt mindestens 60, höchstens 240 Minuten.

8. Mündliche Prüfung (M)

Eine mündliche Prüfung ist ein Prüfungsgespräch, in dem die Studierenden darlegen müssen, dass sie den Prüfungsstoff beherrschen. Sie dauert in der Regel mindestens 15 und höchstens 45 Minuten. Mündliche Prüfungen können als Einzelprüfung oder als Gruppenprüfung durchgeführt werden. Eine mündliche Prüfung ist von einer oder einem Prüfenden und Beisitzenden nach § 13 Absatz 4 abzunehmen. Die mündliche Prüfung kann anstatt von einer Prüferin oder einem Prüfer auch von mindestens zwei Prüfenden abgenommen werden (Kollegialprüfung); dabei ist die oder der Studierende in den einzelnen Prüfungsfächern verantwortlich jeweils nur von einer Prüferin oder einem Prüfer zu prüfen. Die in der mündlichen Prüfung erbrachte Leistung wird sowohl bei einer Prüfung durch mehrere Prüfer, als auch bei einer Prüfung durch eine Prüferin oder einen Prüfer und eine Beisitzerin oder einen Beisitzer nur von der oder dem Prüfenden bewertet und benotet. Die verantwortliche Prüferin oder der verantwortliche Prüfer hört die anderen Prüferinnen oder Prüfer bzw. die Beisitzerin oder Beisitzer vor der Festsetzung der Note an. Die wesentlichen Gegenstände und Ergebnisse der mündlichen Prüfung sind in einem Protokoll festzuhalten. Es wird von den Prüfenden und der oder dem Beisitzenden unterzeichnet und bleibt bei den Prüfungsakten.

9. Projekt (Pj)

Ein Projekt ist eine zu bearbeitende fachübergreifende Aufgabe aus dem jeweiligen Berufsfeld des Studiengangs. Die Ergebnisse des Projektes sind zu dokumentieren. Die Bearbeitungszeit beträgt zwischen 6 bis 26 Wochen und wird mit einem Kolloquium abgeschlossen. In der jeweiligen studiengangsspezifischen Prüfungs- und Studienordnung können zusätzliche Bedingungen zu Form, Inhalt und Ziel des Projektes und eine andere Form des Abschlusses als durch ein Kolloquium festgelegt werden.

10. Referat (R)

Ein Referat ist ein Vortrag über 15 bis 45 Minuten Dauer anhand einer selbst gefertigten schriftlichen Ausarbeitung. An das Referat schließt sich unter Führung einer Diskussionsleitung ein Gespräch an. Das Referat soll in freien Formulierungen gehalten werden. Die bei dem Vortrag vorgestellten Präsentationen bzw. Grafiken sind dem Prüfer in schriftlicher oder elektronischer Form zu übergeben. In der zusätzlichen schriftlichen Ausarbeitung, die dem Prüfer zu übergeben ist, sind die wichtigsten Ergebnisse zusammenzufassen.

11. Test (T)

Der Test ist eine schriftliche Arbeit, in dem die Studierenden nachweisen, dass sie Aufgaben zu einem klar umgrenzten Thema unter Klausurbedingungen bearbeiten können. Die Dauer eines Tests beträgt mindestens 15, höchstens 90 Minuten. In

studiengangsspezifischen Prüfungs- und Studienordnungen kann bestimmt werden, dass die Einzelergebnisse der Tests mit in die Bewertung der Klausuren einbezogen werden.

12. Übungstestat (ÜT)

Ein Übungstestat ist erfolgreich abgeschlossen, wenn die Studierenden die von der Prüferin oder dem Prüfer festgelegten theoretischen Aufgaben durch schriftliche Aufgabenlösungen erfolgreich erbracht sowie ihre Kenntnisse durch Kolloquien oder Referate nachgewiesen haben. Die Dauer des Kolloquiums beträgt mindestens 15, höchstens 45 Minuten. Die schriftlichen Ausarbeitungen sind innerhalb einer von der Prüferin bzw. dem Prüfer festgesetzten Frist abzugeben. Diese Frist endet spätestens mit Ablauf des jeweiligen Semesters, in dem die zugeordnete Lehrveranstaltungsart (Übung) durchgeführt wird.

13. Übungsabschluss (ÜA)

Für die erfolgreiche Ablegung eines Übungsabschlusses ist die kontinuierliche aktive Teilnahme der Studierenden erforderlich. Es kann die schriftliche Ausarbeitung oder eine sonstige Vorstellung einzelner Übungsaufgaben vorgesehen werden.

Modulbezeichnung	Diskrete Mathematik	Kürzel	DM / DMÜ
Lehrveranstaltung(en)	SeU: Diskrete Mathematik Praktikum: Diskrete Mathematik	Fachsemester	1
Arbeitsaufwand	54 Std. SeU, 18 Std. Übung, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Christoph Klauck	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	keine	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • können wichtige mathematische Strukturen sicher verwenden • beherrschen die logischen und algebraischen Grundlagen der theoretischen Informatik • können Definitionsprinzipien und Beweistechniken in unterschiedlichen Bereichen und an typischen Beispielen anwenden • können eine präzise und abstrakte Denkweise sowie die formale Denk- und Argumentationsweise in praxisorientierten Problemen anwenden 		
Inhalte	<ul style="list-style-type: none"> • Mathematische Grundlagen: Mengen, Relationen, Funktionen und deren Operatoren, Boolesche Algebra, Aussagenlogik • Mathematische Techniken: Grundlegende Beweisstrategien, Vollständige Induktion • Mathematische Strukturen: Lösung von linearen Gleichungssystemen, Vektoren • Vertiefung in eines oder mehrere der folgenden Themen: <ul style="list-style-type: none"> - Matrizen und Determinanten - Modulare Arithmetik (Zahlentheorie) - Kombinatorik - Diskrete Stochastik 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: selbstständiges Lösen von Übungsaufgaben</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben Voraussetzung (PVL): Übungstest (erfolgreiche Bearbeitung von Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • M. Aigner, G.M. Ziegler: Das Buch der Beweise • N. Dean: Diskrete Mathematik • P. Hartmann: Mathematik für Informatiker • C. Meinel, M. Mundhenk: Mathematische Grundlagen der Informatik • N. Nerode, R.A. Shore: Logic for applications • C. Stein, R.L. Drysdale, K. Bogart: Discrete Mathematics for Computer Scientists • Skripte der Dozenten 		

Modulbezeichnung	Grundlagen der Informatik	Kürzel	GI / GIÜ
Lehrveranstaltung(en)	SeU: Grundlagen der Informatik Praktikum: Grundlagen der Informatik	Fach-semester	1
Arbeitsaufwand	54 Std. SeU, 18 Std. Übung, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Martin Hübner	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	keine	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> verstehen die Problemstellungen der Informatik verstehen die entsprechenden Basis-Lösungskonzepte der Informatik und können diese anwenden können wichtige Informatik-Tools nutzen 		
Inhalte	<ul style="list-style-type: none"> Was ist „Informatik“? Computer-Hardwaregrundlagen Betriebssystem-Konzepte Grundkonzepte der Programmierung Internet-Technologien Grundlagen der IT-Sicherheit 		
Lehr- und Lernformen	SeU: Seminaristischer Unterricht, Tafel, Präsentation, Beispielaufgaben, Demos, freiwillige Übungsaufgaben, evtl. Tutorium Praktikum: Bearbeitung von Aufgaben mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben Voraussetzung (PVL): Übungstestat (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> Herold, Lurz, Wohlrab: Grundlagen der Informatik, Pearson Skripte der Dozenten 		

Modulbezeichnung	Programmiermethodik I	Kürzel	PM1
Lehrveranstaltung(en)	SeU: Programmiermethodik I	Fachsemester	1
Arbeitsaufwand	72 Std. SeU, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Birgit Wendholt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	keine	SWS	4
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • können die Basiskonzepte einer modernen Programmiersprache benennen und anwenden • können eine abstrakte Problembeschreibung algorithmisch umsetzen • können objektorientierte Konzepte zur Modellierung eines Problemraums benennen und anwenden 		
Inhalte	<ul style="list-style-type: none"> • Vom Problem zum Programm: Strukturiertes Vorgehen beim Programmieren • Typisierungskonzepte • dynamischer Umgang mit Typen • Darstellung und Analyse von Kontrollflüssen • funktionale Abstraktion, Datenabstraktion (ADT), Kontrollabstraktion (z.B. Iteratoren, Streams) • Zeichenketten • Polymorphie • Objekte und Klassen • Objektverträge • Objektinteraktion 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur</p> <p>Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben</p>		
Literatur	<ul style="list-style-type: none"> • Literaturhinweise werden je nach Programmiersprache und aktuellem Stand in der Vorlesung bzw. in den Vorlesungsunterlagen gegeben • Skripte der Dozenten 		

Modulbezeichnung	Programmiertechnik	Kürzel	PT / PTP
Lehrveranstaltung(en)	SeU: Programmiertechnik Praktikum: Programmiertechnik	Fach-semester	1
Arbeitsaufwand	36 Std. SeU, 36 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Birgit Wendholt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	keine	SWS	2+2
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • können eine moderne Entwicklungsumgebung bedienen (Editor, Debugger) • können einfache Programme in einer modernen Programmiersprache entwickeln • können fremden Quellcode analysieren und in eigene Programme integrieren • können Qualitätskriterien für lesbaren, wartbaren, wiederverwendbaren Quellcode nennen und diese beim Erstellen eigener Programme umsetzen • können Programme automatisiert testen 		
Inhalte	<ul style="list-style-type: none"> • Syntax einer modernen Programmiersprache: Variablen, primitive Datentypen, Arrays, Referenztypen, Sequenz, Selektion, Iteration, Klassen, Objekte, • Testabdeckung und Frameworks • Überführen eines Entwurfs in ein lauffähiges Programm • Umgang mit Daten • Ein-Ausgabe von Programmen 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur</p> <p>Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben</p> <p>Prüfungsvorleistung (PVL): Laborabschluss (erfolgreich durchgeführtes Praktikum)</p>		
Literatur	<ul style="list-style-type: none"> • Literaturhinweise werden je nach Programmiersprache und aktuellem Stand in der Vorlesung bzw. in den Vorlesungsunterlagen gegeben • Skripte der Dozenten 		

Modulbezeichnung	Betriebswirtschaftslehre	Kürzel	BWL / BWLÜ
Lehrveranstaltung(en)	SeU: Betriebswirtschaftslehre Praktikum: Betriebswirtschaftslehre	Fachsemester	1
Arbeitsaufwand	54 Std. SeU, 18 Std. Übung, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Martin Schultz	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	keine	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> • können organisatorische und finanzielle Strukturen von Unternehmen erläutern, • können Aspekte der Aufbau- und Ablauforganisation von Unternehmen modellieren, • können Investitionsentscheidungen nach betriebswirtschaftlichen Kriterien treffen, • können Aufbau und Funktionsweise integrierter Informationssysteme erläutern und integrierte Informationssysteme anhand von Fallstudien anwenden, • können Methoden zum Management von Informationssystemen in Unternehmen beschreiben und anwenden 		
Inhalte	<p>Grundlagen</p> <ul style="list-style-type: none"> • Das Unternehmen als System • Rechtsformen und Aufbauorganisation • Ablauforganisation und Geschäftsprozessmanagement <p>Unternehmensführung</p> <ul style="list-style-type: none"> • Grundlagen der Finanzbuchhaltung (Buchführung und Jahresabschluss) • Kosten- und Leistungsrechnung • Finanzierung und Investitionsrechnung <p>Informationsmanagement in Unternehmen</p> <ul style="list-style-type: none"> • BWL und Informationstechnologie • Integrierte Informationssysteme (ERP-Systeme) • IT-Management 		
Lehr- und Lernformen	SeU: Seminaristischer Unterricht, Tafelarbeit, Multimedia-Präsentationen, freiwillige Übungsaufgaben Praktikum: Selbstständiges Bearbeiten von Übungsaufgaben		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur</p> <p>Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben.</p> <p>Voraussetzung (PVL): Übungstestat (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • G. Wöhe: Einführung in die allgemeine BWL, Verlag Franz Vahlen • S. Schmolke, M. Deitermann: Industrielles Rechnungswesen IKR, Winklers Verlag • K. Laudon, J. Laudon: Wirtschaftsinformatik • H. Hansen, J. Mendling, G. Neumann: Wirtschaftsinformatik • A. Gadatsch: Grundkurs Geschäftsprozessmanagement • H. Krcmar: Informationsmanagement • Skripte der Dozenten 		

Modulbezeichnung	Logik und Berechenbarkeit	Kürzel	LB / LBP
Lehrveranstaltung(en)	SeU: Logik und Berechenbarkeit Praktikum: Logik und Berechenbarkeit	Fach-semester	2
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Christoph Klauck	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Grundlagen der Informatik, Diskrete Mathematik	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> • können formale Modellierungskonzepte auf Basis von Kalkülen und Grammatiken anwenden • können eigenständig präzise Beschreibungen für praxisorientierte Probleme erstellen • besitzen die Fähigkeit, Berechenbarkeitsgrenzen zu erkennen und praktische Probleme zu klassifizieren 		
Inhalte	<ul style="list-style-type: none"> • Einführung in die Aussagenlogik: logische Systeme, wohlgeformte Ausdrücke, Wahrheitswerte/-tafeln, Belegungsfunktion, Normalformen, Literale und Klauseln • Grundkonzepte der Semantik: Allgemeingültigkeit, Unerfüllbarkeit, Erfüllbarkeit und Folgerung • Beweistheorie und Ableitungssysteme für die Aussagenlogik: Schlussregeln, modus ponens, Resolutionskalkül und Tableauekalkül in der Aussagenlogik • Einführung in die Prädikatenlogik erster Stufe: Charakterisierung wohlgeformter Ausdrücke, Quantoren und Quantorenskopus • Entscheidbarkeit und Berechenbarkeit: primitiv rekursive und allgemein-rekursive Funktionen, Register- und Turing-Maschine als formale Algorithmusdefinition • Vertiefung in eines oder mehrere der folgenden Themen: <ul style="list-style-type: none"> - Halteproblem und andere unentscheidbare Fragen - Landausche Notation, - Konkrete Komplexitätstheorie für einfache Algorithmen, auch für einfache randomisierte Algorithmen 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> • C. Baier, A. Asteroth: Theoretische Informatik; Einführung in Berechenbarkeit, Komplexität und formale Sprachen • J.E. Hopcroft, R. Motwani, J.D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie • M. Kreuzer, S. Kühling: Logik für Informatiker • E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kann, D.B. Shmoys: The Traveling Salesman Problem • R. Socher: Theoretische Grundlagen der Informatik • G. Vossen, K.-U. Witt: Grundkurs Theoretische Informatik • Eigene Skripte der Dozenten 		

Modulbezeichnung	Automatentheorie und Formale Sprachen	Kürzel	AF / AFÜ
Lehrveranstaltung(en)	SeU: Automatentheorie und Formale Sprachen Praktikum: Automatentheorie und Formale Sprachen	Fachsemester	2
Arbeitsaufwand	54 Std. SeU, 18 Std. Übung, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Padberg	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Mathematische Grundlagen, Kenntnisse von formalen Beweisen, Prädikatenlogik, speziell Induktion	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden können Zusammenhänge zwischen Automatenmodellen, regulären Sprachen und Grammatiken herstellen und Modelle ineinander überführen und können formale Spezifikationen auf Problemstellungen der realen Welt anwenden. Hierfür verfügen Sie über die folgenden theoretischen und praktischen Fähigkeiten, sie</p> <ul style="list-style-type: none"> • sind in der Lage, die grundlegenden Begriffe und Formalismen eigenständig anzuwenden • verstehen formale Modelle in Form von endlichen Automaten, regulären Ausdrücken und Grammatiken • verstehen, warum viele der bekannten Sprach- und Automatenfamilien die gleichen Eigenschaften haben • kennen die Beziehungen zwischen diesen Eigenschaften • können die dazugehörigen formalen Beweise erläutern und selber durchführen • können formale Modelle in Form von Automaten, regulären Ausdrücken und Grammatiken erstellen 		
Inhalte	<ul style="list-style-type: none"> • Automaten: Die Grundlagen und Methoden • Endliche Automaten • Reguläre Ausdrücke und Sprachen • Eigenschaften regulärer Sprachen • Kontextfreie Grammatiken und Sprachen • Eigenschaften kontextfreier Sprachen • Vertiefung in eine der folgenden Richtungen: Kellerautomaten, zeitbehaftete Automaten, Modellierung mit formalen Methoden 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: selbstständiges Lösen der Übungsaufgaben in Gruppen</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • J.E. Hopcroft, R. Motwani; J.D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Addison-Wesley • C. Baier, A. Asteroth: Theoretische Informatik • R. Socher: Theoretische Grundlagen der Informatik • The Theory of Timed I/O Automata, (Synthesis Lectures on Distributed Computing Theory) von Dilsun Kaynar, Nancy Lynch, Roberto Segala und Frits Vaandrager von Morgan & Claypool Publishers • Skripte der Dozenten 		

Modulbezeichnung	Datenbanken	Kürzel	DB / DBP
Lehrveranstaltung(en)	SeU: Datenbanken Praktikum: Datenbanken	Fachsemester	2
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Thomas Clemen	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiertechnik, Diskrete Mathematik	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden:</p> <ul style="list-style-type: none"> • kennen die Einsatzgebiete und Grenzen von relationalen Datenbanksystemen, • beherrschen den Prozess des Datenbankentwurfs, • kennen die theoretischen Grundlagen von relationalen Datenbanksystemen, • können einfache Datenbankanwendungen entwickeln, • beherrschen die relationale Anfragesprache SQL im Rahmen des Standards und • verstehen aktuelle gesellschaftliche Herausforderungen. 		
Inhalte	<ul style="list-style-type: none"> • Grundkonzepte relationaler Datenbanksysteme • Datenmodelle und -modellierung • Datenbankentwurf • Relationale Anfragesprachen • Relationale Entwurfstheorie • Transaktionsverwaltung • Mehrbenutzersynchronisation • Datenintegrität • Implementierung und Befüllung von Datenbanksystemen • NoSQL Datenbanksysteme 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • A. Kemper, A. Eickler: Datenbanksysteme – Eine Einführung, Oldenbourg • R.A. Elmasri, S.B. Navathe: Grundlagen von Datenbanksystemen, Pearson • Skripte der Dozenten 		

Modulbezeichnung	Programmiermethodik II	Kürzel	PM2 / PMP2
Lehrveranstaltung(en)	SeU: Programmiermethodik II Praktikum: Programmiermethodik II	Fachsemester	2
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Birgit Wendholt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	keine (erfolgreiches Absolvieren von PM1/PT1 wird empfohlen)	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> • beherrschen fortgeschrittene Programmier Techniken und können diese in kleinen Beispielprogrammen in einer aktuellen Programmiersprache einsetzen • können technische Basis-Hilfsmittel für die teamorientierte SW-Entwicklung anwenden (z.B. Versionsverwaltung) 		
Inhalte	<ul style="list-style-type: none"> • Generische Template-Klassen (Generics, Reflexion) • Nebenläufige bzw. asynchrone Programmierung (Threads) • Vertiefungen: Typ- vs. Implementierungshierarchie • Entwurfsmuster (Singleton, Template, Factory, Subject-Observer, MVC etc.) • Entwurf gemäß Vertrag (Bedingungen, Invarianten) • vertiefte Teststrategien • GUI-Programmierung • Reguläre Ausdrücke zur Verarbeitung von großen Textdateien • Grundlagen teamorientierter SW-Entwicklung 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben Prüfungsvorleistung (PVL): Laborabschluss (erfolgreich durchgeführtes Praktikum)		
Literatur	<ul style="list-style-type: none"> • Literaturhinweise werden je nach Programmiersprache und aktuellem Stand in der Vorlesung bzw. in den Vorlesungsunterlagen gegeben • Skripte der Dozenten 		

Modulbezeichnung	Rechnerstrukturen und Maschinennahe Programmierung	Kürzel	RMP / RMPP
Lehrveranstaltung(en)	SeU: Rechnerstrukturen und Maschinennahe Programmierung Praktikum: Rechnerstrukturen und Maschinennahe Programmierung	Fachsemester	2
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Tim Tiedemann	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Grundlagen der Informatik	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden: <ul style="list-style-type: none"> • können maschinennahe Datentypen und Programmierparadigmen verstehen und anwenden, • können grundlegende Rechnerarchitekturkonzepte sowie die Instruction-set-Architektur eines aktuellen Prozessors erklären • können den Zusammenhang zwischen einer maschinennahen Hochsprache (z.B. ANSI-C) und der unterlagerten Maschinensprache beschreiben • können Programme auf niedrigem Abstraktionslevel strukturieren, modularisieren und entwickeln • können einfache Assembler-Programme für einen ausgewählten Prozessor erstellen • können in einer maschinennahen Hochsprache (ANSI-C) einfache Anwendungen entwickeln • können Entwicklungswerkzeuge für die maschinennahe Programmierung verwenden 		
Inhalte	<ul style="list-style-type: none"> • Darstellung von Daten im Computer • Rechnerarchitekturgrundlagen auf Ebene der Instruction-set-Architektur: Rechner-Grundkomponenten (Speicher, ALU usw.), von Neumann- und Harvardarchitektur, aktuelle Rechnerarchitekturkonzepte und Bussysteme • maschinennahe Programmierung auf Basis des Maschinenbefehlssatzes eines aktuellen Prozessors • Konzepte einer hardwarenahen höheren Programmiersprache, zum Beispiel ANSI-C • Projektverwaltung, Modultechnik, Bibliotheken • Interfaces zur Verzahnung von Hochsprachen und Assembler • Abbildung von Daten und Kontrollstrukturen prozeduraler Hochsprachen in maschinennahe Implementierungen • Weitere Themen nach Aktualität 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> • Grundlagenbücher: s. Bibliothek, Signatur Dat 001 • Andrew S. Tanenbaum, James Goodman: Computerarchitektur, Pearson Studium • Joachim Groll, Ulrich Bröckl, Manfred Dausmann: C als erste Programmiersprache, Teubner Verlag • B.W. Kernighan, D.M. Ritchie: Programmieren in C, Hanser Verlag • W. Hohl: ARM Assembly Language: Fundamentals and Techniques • Skripte der Dozenten, bereitgestellte Zusatzmaterialien (Befehlsreferenz) 		

Modulbezeichnung	Graphentheoretische Konzepte und Algorithmen<	Kürzel	GKA / GKAP
Lehrveranstaltung(en)	SeU: Graphentheoretische Konzepte und Algorithmen Praktikum: Graphentheoretische Konzepte und Algorithmen	Fach-semester	3
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Christoph Klauck	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Grundlagen der Informatik, Diskrete Mathematik, Logik und Berechenbarkeit, Automaten und formale Sprachen	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> • können in praxisorientierten Problemen erfolgreich graphentheoretische Modellierungsparadigmen und Formalismen anwenden • beherrschen die grundlegenden Konzepte, Formalismen und Notationen sowie die wichtigsten Algorithmen • können eigenständig praxisorientierte Probleme mit graphentheoretischen Methoden Modellieren und Lösen • besitzen die Fähigkeit zum eigenständigen Modellieren, Analysieren und einem Redesign von nebenläufigen Prozessen mittels Petri-Netzen 		
Inhalte	<ul style="list-style-type: none"> • Graphentheoretische Grundbegriffe, Wege, Kreise, Zusammenhang • Bäume, Wälder • Suchstrategien, Kürzeste Wege, Flüsse und Strömungen • Matchings, Routing, Planare Graphen • Grundlegende Eigenschaften von Petri-Netzen • Vertiefung in eines oder mehrere der folgenden Themen: <ul style="list-style-type: none"> - Färbungen und Überdeckungen - Graphtransformationen - Berechenbarkeit, Erreichbarkeit und Erzeugbarkeit von Petri-Netzen 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> • R. Diestel: Graph Theory • H. Ehrig, K. Ehrig, U. Prange, G. Taentzer: Fundamentals of Algebraic Graph Transformation • H. Ehrig, W. Reisig, G. Rozenberg: Petri Net Technology for Communication-Based Systems • J. Hromkovic, M. Nagl, B. Westfechtel: Graph-Theoretic Concepts in Computer Science • S.O. Krumke, H. Noltemeier: Graphentheoretische Konzepte und Algorithmen • M. Nietzsche: Graphen für Einsteiger: Rund um das Haus vom Nikolaus • V. Turau: Algorithmische Graphentheorie • Skripte der Dozenten 		

Modulbezeichnung	Algorithmen und Datenstrukturen	Kürzel	AD / ADP
Lehrveranstaltung(en)	SeU: Algorithmen und Datenstrukturen Praktikum: Algorithmen und Datenstrukturen	Fachsemester	3
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Köhler-Bußmeier	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiermethodik II	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> • bestehende Algorithmen und Datenstrukturen zur Lösung von Berechnungsaufgaben einsetzen, • indem Sie anhand von Anforderungen eigene Algorithmen entwerfen, implementieren und analysieren, • um in SW-Entwicklungskontexten Aufgaben mittels geeigneter Algorithmen und Datenstrukturen lösen zu können. 		
Inhalte	<ul style="list-style-type: none"> • Prinzipien der Algorithmenanalyse • Designprinzipien von Algorithmen (z.B. Divide and Conquer, dynamic programming) • Komplexität (Komplexitätsklassen, NP-Vollständigkeit) • Datenstrukturen (z.B. Liste, Stack, Queue, Tree) • Suchen (z.B. Suchbäume, Hashing) • Sortieren • Weitere Themen nach Aktualität 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms, McGraw-Hill • Sedgewick, R., Wayne, K.: Algorithms in Java, Addison-Wesley • Skripte der Dozenten 		

Modulbezeichnung	Software Engineering I	Kürzel	SE1 / SEP1
Lehrveranstaltung(en)	SeU: Software Engineering I Praktikum: Software Engineering I	Fachsemester	3
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Stefan Sarstedt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiermethodik II, Datenbanken	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> verstehen den Unterschied zwischen sequenziellem und iterativem Vorgehen in Projekten verstehen verschiedene Ansätze zur Anforderungsfindung und Spezifikation können Verfahren zu Anforderungsfindung und Spezifikation anwenden können einfache Softwareentwürfe erstellen, die grundlegenden Qualitätskriterien entsprechen können qualitativ hochwertigen Code schreiben können qualitativ hochwertige Tests auf unterschiedlichen Teststufen formulieren können Zusammenarbeit im Team organisieren und verfolgen können die Aktivitäten auf Projekte in unbekanntem Anwendungsfeldern übertragen 		
Inhalte	<ul style="list-style-type: none"> Projektaktivitäten: Analyse, Spezifikation, Entwurf, Implementierung, Test Vorgehen: Wasserfall vs. Iterativ/agil, Kanban, einfache Planungs- und Steuerungsaufgaben Analyse: Erarbeiten von Anforderungen, Kommunikation mit dem Kunden, Stakeholderanalyse, User Storys, Qualitätskriterien für Anforderungen Spezifikation: Wireframes, Fachliche Datenmodelle, Fachliche Datentypen, Use Cases, Geschäftsprozesse, ausgewählte Modellierungssprachen Entwurf/Architektur: Entwurfsprinzipien, Komponenten, Schnittstellen, Architekturstile, Referenzarchitekturen, Dependency Injection, Design Patterns, Antipatterns Implementierung: Codequalität, Clean Code Testen: Teststufen, Test-Pyramide, Unit Tests, Komponententests, Integrationstests, Akzeptanztests, Test-Doubles, einfache Code Coverage Tools: Versionierung, Automatisierung, Build-Tools 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben</p> <p>Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur</p> <p>Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben.</p> <p>Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> Mike Cohn, User Stories Applied, Addison Wesley E. Evans: Domain-Driven Design, Addison Wesley L. Koskela: Effective Unit Testing, Manning C. Larmann: Applying UML and Patterns, Pearson Ludewig, Lichten: Software Engineering: Grundlagen, Menschen, Prozesse, Techniken, dpunkt.verlag Robert C. Martin: Clean Code, Prentice Hall 		

Modulbezeichnung	Software Engineering I	Kürzel	SE1 / SEP1
	<ul style="list-style-type: none">• S. & J. Robertson: Mastering the Requirements Process: Getting Requirements Right, Addison Wesley• J. Siederleben: Moderne Software-Architektur, dpunkt.verlag• Skripte der Dozenten		

Modulbezeichnung	Betriebssysteme	Kürzel	BS / BSP
Lehrveranstaltung(en)	SeU: Betriebssysteme Praktikum: Betriebssysteme	Fachsemester	3
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Martin Hübner	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiermethodik II, Rechnerstrukturen und Maschinennahe Programmierung	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> verstehen die Architektur, die Konzepte und die Funktionsweise moderner Betriebssysteme sowie des Zusammenspiels von Hard- und Software, verstehen die Konzepte zur Implementierung systemnaher Software können das Verhalten von Computersystemen analysieren und beschreiben können Grundkonzepte der nebenläufigen Programmierung anwenden 		
Inhalte	<ul style="list-style-type: none"> Architekturen und Betriebsarten Prozess- und Thread-Konzept, Scheduling Synchronisation, Interprozesskommunikation, Deadlocks Hauptspeicherverwaltung, Virtueller Speicher Verwaltung externer Geräte Dateisysteme Schutzmechanismen, Sicherheitsaspekte Exemplarische Betrachtung aktueller Betriebssysteme 		
Lehr- und Lernformen	SeU: Seminaristischer Unterricht, Tafelarbeit, Multimedia-Präsentationen, Beispiele und Demos Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> Andrew S. Tanenbaum, Modern Operating Systems, Pearson Studium Verlag Skripte der Dozenten 		

Modulbezeichnung	Data Science	Kürzel	DS / DSP
Lehrveranstaltung(en)	SeU: Data Science Praktikum: Data Science	Fachsemester	3
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Marina Tropmann-Frick	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiermethodik II, Diskrete Mathematik	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • verstehen wesentliche mathematische Grundlagen aus den Bereichen der Stochastik, Analysis und linearen Algebra • können Verfahren zu Datensammlung, -vorbereitung und -visualisierung anwenden • können Hypothesen formulieren und statistische Analysen auf Daten durchführen • können ausgewählte Methoden aus den Bereichen Data Mining, Maschinelles Lernen und Künstliche Intelligenz auf Daten aus unterschiedlichen Domänen anwenden • können Anwendungen für Datenanalysen programmieren und Ergebnisse der Analysen interpretieren • kennen die Konzepte von Data Governance, Datenethik und Privacy 		
Inhalte	<ul style="list-style-type: none"> • Ausgewählte Themen der Wahrscheinlichkeitstheorie • Ausgewählte Themen der deskriptiven und schließenden Statistik • Korrelationsanalyse • Data Mining • Visual Analytics / explorative Datenanalyse • Clustering • Lineare und logistische Regression • Klassifikation • Ausgewählte Themen aus Machine Learning und Künstlicher Intelligenz 		
Lehr- und Lernformen	<p>SeU: Seminaristischer Unterricht, Tafelarbeit, Multimedia-Präsentationen, Beispiele und Demos Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • Steven Skiena, Data Science Design Manual, Springer Verlag • Henk Tijms, Understanding Probability, Cambridge University Press • Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis und Florian Mansmann, „Visual Analytics“ • J. Han, M. Kamber, Data Mining. Concepts and Techniques • Deep Learning. Das umfassende Handbuch: Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze. Von Ian Goodfellow, Yoshua Bengio, et al. • Eigene Skripte der Dozenten 		

Modulbezeichnung	Künstliche Intelligenz	Kürzel	KI / KIP
Lehrveranstaltung(en)	SeU: Künstliche Intelligenz Praktikum: Künstliche Intelligenz	Fach-semester	4
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Michael Neitzke	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Diskrete Mathematik, Grundlagen der Informatik, Programmiermethodik I, Programmiermethodik II, Logik und Berechenbarkeit, Automatentheorie und Formale Sprachen, Graphentheorie, Algorithmen und Datenstrukturen, Data Science, Software-Engineering 1	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> überprüfen Fragestellungen bezüglich der Methoden und Konzepte der Künstlichen Intelligenz und aus den Anwendungsfeldern der Künstlichen Intelligenz unter Zuhilfenahme einer softwaretechnischen Umsetzung, <p>um</p> <ul style="list-style-type: none"> Grenzen und Möglichkeiten der jeweiligen Methoden und Konzepte einschätzen zu können, indem Sie sich in verschiedene Methoden und Konzepte der Künstlichen Intelligenz einarbeiten, im Sinne des forschenden Lernens neben vorgegebenen Fragestellungen auch eigene Fragestellungen formulieren, dafür relevante Literatur suchen und sich damit auseinandersetzen Experimente zur Klärung der Fragestellungen für eine softwaretechnische Umsetzung konzipieren, die erforderliche softwaretechnische Umsetzung entwickeln, die geplanten Experimente durchführen, die Ergebnisse und Erkenntnisse bezüglich der Fragestellungen für Außenstehende verständlich aufbereiten, die Eigenschaften, Grenzen und Möglichkeiten der eingesetzten Methoden und Konzepte diskutieren oder erläutern. 		
Inhalte	<ul style="list-style-type: none"> Logikbasierte Repräsentationen und Schlussfolgerungsverfahren Zustandsraumsuche: Uninformierte, informierte, adversariale, Constraint-basierte, stochastische (u.a. genetische) Suche Maschinelles Lernen: Überwachtes, unüberwachtes, verstärkendes Lernen Philosophische und ethische Aspekte Ausgewählte Anwendungsfelder wie z.B. Planungsprobleme, 2-Personen-Spiele, Regressionsprobleme, Klassifikationsprobleme, Verarbeitung von sequentiellen Daten (natürlichsprachliche, Videos, Musik) 		
Lehr- und Lernformen	<p>SeU: Seminaristischer Unterricht, Tafelarbeit, Multimedia-Präsentationen, Beispiele und Demos</p> <p>Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur</p> <p>Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben.</p> <p>Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		

Modulbezeichnung	Künstliche Intelligenz	Kürzel	KI / KIP
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none">• Russell, Norvig: „Artificial Intelligence: A Modern Approach“, Addison Wesley• Goodfellow, Bengio, Courville: „Deep Learning“ The MIT Press• Skripte der Dozenten		

Modulbezeichnung	Software Engineering II	Kürzel	SE2 / SEP2
Lehrveranstaltung(en)	SeU: Software Engineering II Praktikum: Software Engineering II	Fachsemester	4
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Stefan Sarstedt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiermethodik II, Datenbanken, Software Engineering I	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • können wichtige Begriffe im Kontext von Softwareprojekten darstellen • können unterschiedliche Vorgehens- und Prozessmodelle und deren Einsatzbereiche erläutern • können ausgewählte Vorgehens- und Prozessmodelle anwenden • können typische Aufgaben eines Projektleiters benennen • können typische Tätigkeiten in verschiedenen Projektphasen (u.a. Planung und Kontrolle) beschreiben • können Aspekte des Qualitäts-, Risiko- und Konfigurationsmanagement darstellen • können ausgewählte Qualitätssicherungsmaßnahmen anwenden und Softwarequalität bewerten • können Software-Ergonomie als Bestandteil des Entwicklungsprojektes berücksichtigen 		
Inhalte	<ul style="list-style-type: none"> • Begriffe im Kontext eines Softwareprojekts: Projekt, Projektphase, u.a. • Vorgehens- und Prozessmodelle in der Software-Entwicklung • Kostenschätzung • Projektplanung • Projektcontrolling • Risikomanagement • Qualitätssicherung • Konfigurationsmanagement • Software-Ergonomie 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • Tom DeMarco: Bärenango: Mit Risikomanagement Projekte zum Erfolg führen. Hanser • Tom DeMarco: Vom Mythos des Mann-Monats. Essays zum Software-Engineering. Mitp-Verlag • Dirk W. Hoffmann. Software-Qualität. Springer Vieweg • Ian Sommerville. Software Engineering. Pearson Education • Jochen Ludewig und Horst Lichter. Software Engineering. Grundlagen, Menschen, Prozesse, Techniken. dpunkt Verlag • Skripte der Dozenten 		

Modulbezeichnung	Rechnernetze	Kürzel	RN/RNP
Lehrveranstaltung(en)	SeU: Rechnernetze Praktikum: Rechnernetze	Semester	4
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Martin Hübner	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiermethodik II, Betriebssysteme	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> verstehen die Konzepte und die Funktionsweise von Rechnernetzen können grundlegende Konzepte zur Performanceanalyse von Rechnernetzen anwenden können auf der Socket-Schnittstelle basierende Client- / Server-Anwendungen erstellen können Methoden und Werkzeuge für die Konfiguration und Administration von Rechnernetzen anwenden 		
Inhalte	<ul style="list-style-type: none"> Architektur des Internet Grundkonzepte der Datenübertragung und Performanceanalyse Wichtige Anwendungsschichtprotokolle Socket-Programmierung Protokolle und Dienste der Netzwerk- und Transportschicht, insbesondere die TCP/IP-Protokollsuite Protokolle der Sicherungsschicht Grundlagen der LAN-Technologien Sicherheit in Netzwerken 		
Lehr- und Lernformen	SeU: Seminaristischer Unterricht, Tafelarbeit, Multimedia-Präsentationen, Beispiele und Demos Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> James F. Kurose, Keith W. Ross: Computernetze – Der Top-Down-Ansatz, Pearson Studium Andrew S. Tanenbaum, David J. Wetherall: Computernetzwerke, Pearson Studium Skripte der Dozenten 		

Modulbezeichnung	Englisch	Kürzel	ENG
Lehrveranstaltung(en)	S: Englisch	Semester	4
Arbeitsaufwand	36 Std. S, 54 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Stefan Sarstedt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	3
Voraussetzungen	siehe Zulassungsordnung zum Studiengang	SWS	2
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	englisch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden können in der englischen Sprache</p> <ul style="list-style-type: none"> • fachliche Diskussionen führen • fachliche Präsentationen durchführen, sowie • fachliche Texte verfassen. 		
Inhalte	<ul style="list-style-type: none"> • spezielles Vokabular des Kontextes • Besonderheiten des Stils und Strukturen des jeweiligen Kontextes • Besonderheiten bei der Präsentation und Textorganisation des jeweiligen Kontextes 		
Lehr- und Lernformen	S: Seminar, praktische Übungsaufgaben		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Studienleistung: Klausur, mündliche Prüfung, Referat, Hausarbeit oder Übungsabschluss</p> <p>Regelhafte Prüfungsform: nicht benotete Klausur</p> <p>Alternative Prüfungsformen: nicht benotete mündliche Prüfung oder nicht benotetes Referat oder nicht benotete Hausarbeit oder nicht benoteter Übungsabschluss</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben.</p>		
Literatur	<ul style="list-style-type: none"> • Skripte und Empfehlungen der Dozenten 		

Modulbezeichnung	Ethik	Kürzel	ETH
Lehrveranstaltung(en)	S: Ethik	Semester	4
Arbeitsaufwand	36 Std. S, 54 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Jens von Pilgrim	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	3
Voraussetzungen	-	SWS	2
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • kennen die Grundlagen von Ethik und Berufsethik • entwickeln eine Wahrnehmungsfähigkeit für ethikrelevante Problemsituationen • können Fallbeispiele gegebener Problemsituationen unter Verwendung verschiedener Ethikkonzepte analysieren • üben Strategien zur Problemlösung von Fallbeispielen ein 		
Inhalte	<ul style="list-style-type: none"> • Begriffskunde Informatik, Ethik, Berufsethik • Technikfolgeabschätzung • Ethikkonzepte • spezielle Problemstellungen in der Informatik, bspw. in den Bereichen Datenschutz, Sicherheit, maschinelles Lernen 		
Lehr- und Lernformen	S: Seminar, praktische Übungsaufgaben		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: nicht benotete Klausur</p> <p>Alternative Prüfungsformen: nicht benotete mündliche Prüfung oder nicht benotetes Referat oder nicht benotete Hausarbeit oder nicht benoteter Übungsabschluss</p> <p>Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben.</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • Datenethikkommission der Bundesregierung: Gutachten der Datenethikkommission • Grunwald, A.: Technikfolgenabschätzung; Berlin • Grunwald, A., Simonidis-Puschmann, M.: Handbuch Technikethik. Springer-Verlag. • Kuhlen, Rainer: Informationsethik. Umgang mit Wissen und Informationen in elektrischen Räumen. • Misselhorn, K.: Grundfragen der Maschinenethik. Reclam. • Spiekermann, S.: Ethical IT Innovation: A Value-Based System Design Approach. Taylor & Francis Group. • Weber-Wulff, Debora; Class, Christina; Coy, Wolfgang; Kurz, Constanze ; Zellhöfer, David: Gewissensbisse: Ethische Probleme der Informatik. Biometrie - Datenschutz - geistiges Eigentum. • Skripte und Empfehlungen der Dozenten 		

Modulbezeichnung	Projekt	Kürzel	PRO
Lehrveranstaltung(en)	Pj: Projekt	Semester	5
Arbeitsaufwand	270 Std. Pj	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Olaf Zukunft	Turnus	semesterweise
Art des Moduls	Wahlpflichtmodul	CP	9
Voraussetzungen	je nach Projekt	SWS	6
Verwendbarkeit	für die Studiengänge „Angewandte Informatik“, „Informatik Technischer Systeme“, „Wirtschaftsinformatik“, „European Computer Science“	Sprache	deutsch oder englisch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> haben die Fähigkeit zur Lösung informatikspezifischer Probleme unter Berücksichtigung begrenzter Ressourcen (Zeit, Mitarbeiter, Werkzeuge etc.), zur Spezifikation von Anforderungen, zur Modellierung von Systemen, zur Zielsetzung und Planung von Projekten, zur Sicherung der Qualität, zur Vor- und Nachkalkulation des Zeitaufwandes und zur verständlichen Dokumentation. haben die Fähigkeit zur Teamarbeit mit Entwicklern und (wenn möglich) Anwendern, speziell: zur Präsentation von Arbeitsergebnissen, zur Leitung und Moderation von Besprechungen, zur Lösung von Konflikten und zur Beurteilung von Arbeitsergebnissen. 		
Inhalte	<p>Die Studierenden wählen die Projekte aus einem Pool, der für die Informatik-Bachelorstudiengänge des Department Informatik gemeinsam angeboten wird. Die Beschreibung des aktuellen Angebots ist auf den Informatikseiten zu finden.</p> <ul style="list-style-type: none"> Die in den Projekten direkt benötigten spezifischen Kenntnisse – sowohl aus dem anwendungs- und berufsbezogenen als auch aus dem informatischen und mathematischen Bereich – werden in Blockveranstaltungen vermittelt bzw. erarbeitet. Soweit im Rahmen des jeweiligen Projekts machbar, sollte den Studierenden Gelegenheit gegeben werden, die Ermittlung fachlicher Anforderungen in Interviews mit „echten Kunden“ zu trainieren. Im Bereich Dokumentation sollte zur Schärfung des Problembewusstseins die Dokumentation von Vorgänger- oder Zuarbeitergruppen genutzt werden. Regelmäßige Projektsitzungen geben den Studierenden die Möglichkeit, die die oben genannten Fähigkeiten zur Teamarbeit durch Einübung zu erwerben. Dabei wird insbesondere die Qualitätssicherung durch Präsentation von Ergebnissen aus Analyse, Entwurf und Implementierung trainiert. 		
Lehr- und Lernformen	Pj		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Unbenotetes Projekt		
Literatur	<ul style="list-style-type: none"> abhängig vom Projektthema 		

Modulbezeichnung	Seminar	Kürzel	SEM
Lehrveranstaltung(en)	S: Seminar	Semester	5
Arbeitsaufwand	90 Std. S	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Bettina Buth	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	3
Voraussetzungen	-	SWS	2
Verwendbarkeit	für die Studiengänge „Angewandte Informatik“, „Informatik Technischer Systeme“, „Wirtschaftsinformatik“, „European Computer Science“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • sind in der Lage, ein Thema selbständig zu erarbeiten • beherrschen die Methodik zur sachgerechten Literaturrecherche • sind in der Lage, einen eigenen Standpunkt herauszuarbeiten • können überzeugend argumentieren • können eine professionelle Präsentation erstellen • sind in der Lage, eine themenzentrierte konstruktive Diskussion zu führen 		
Inhalte	<ul style="list-style-type: none"> • Es werden Themen zu einem gegebenen Themengebiet oder nach eigener Wahl und Absprache mit dem Betreuer bearbeitet. • Die Vorbereitung erfolgt unter individueller Betreuung durch eine(n) der veranstaltenden Professorinnen oder Professoren. • Dabei wird auf methodische inhaltliche Arbeit ebenso geachtet wie auf eine gute didaktische Aufbereitung und eine professionelle Präsentation. Dazu gehört auch ein Probevortrag vor der betreuenden Professorin bzw. dem betreuenden Professor. 		
Lehr- und Lernformen	S: Kritik an Inhalt und Form durch Betreuer und Studierende		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Unbenotetes Referat		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • Martin Hartmann, Rüdiger Funk, Horst Nietmann: Präsentieren; Beltz • Josef W. Seifert: Visualisieren, Präsentieren, Moderieren; Gabal • Christian W. Dawson: Computerprojekte im Klartext; Pearson Studium • Fachliteratur abhängig vom Thema 		

Modulbezeichnung	Softwarearchitektur	Kürzel	SWA / SWAP
Lehrveranstaltung(en)	SeU: Softwarearchitektur Praktikum: Softwarearchitektur	Fachsemester	5
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Stefan Sarstedt	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiertechnik, Programmiermethodik II, Datenbanken, Software Engineering I, Software Engineering II, Rechnernetze	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>Die Studierenden</p> <ul style="list-style-type: none"> • verstehen wichtige Begriffe im Kontext der Software-Architektur • können typische Aufgaben eines Software-Architekten benennen • kennen Heuristiken zur Gestaltung von Architekturen, Muster und Architekturstile und können diese anwenden • können Einflüsse von Randbedingungen und Risiken auf die Architekturgestaltung ableiten • können unterschiedliche Architektursichten anhand von Fallbeispielen erarbeiten • besitzen Kenntnisse in ausgewählten technischen Konzepten und können diese anwenden • können die Qualität einer Architektur bewerten 		
Inhalte	<ul style="list-style-type: none"> • Begriffe im Kontext Software-Architektur • Aufgaben von Software-Architekten • Vorgehen bei der Architekturentwicklung • Sichten und Modelle • Ausgewählte Diagrammart der UML • Heuristiken und Muster • Architekturstile • Ausgewählte technische Konzepte • Qualität von Architekturen 		
Lehr- und Lernformen	<p>SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch</p>		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	<p>Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)</p>		
Literatur	<p>Jeweils in der aktuellen Auflage:</p> <ul style="list-style-type: none"> • Eric Evans: Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley Longman • Craig Larman: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall • Robert C. Martin: Agile Software Development, Principles, Patterns, and Practices. Prentice Hall International • Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall International 		

Modulbezeichnung	Softwarearchitektur	Kürzel	SWA / SWAP
	<ul style="list-style-type: none">• Johannes Siedersleben. Moderne Software-Architektur. Umsichtig planen, robust bauen mit Quasar. dpunkt, Heidelberg• Gernot Starke. Effektive Software-Architekturen. Ein praktischer Leitfaden. Carl Hanser Verlag GmbH & Co. KG• Skripte der Dozenten		

Modulbezeichnung	Verteilte Systeme	Kürzel	VS / VSP
Lehrveranstaltung(en)	SeU: Verteilte Systeme Praktikum: Verteilte Systeme	Fachsemester	5
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Christoph Klauck	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Betriebssysteme, Rechnernetze, Software Engineering I, Software Engineering II, Algorithmen und Datenstrukturen	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> • beherrschen die Grundlagen verteilter Systeme; • besitzen die Fähigkeit die Möglichkeiten, Grenzen und Risiken verteilter Systeme zu bewerten • beherrschen und verstehen einschlägige Algorithmen zur Realisierung verteilter Anwendungsszenarien und können diese auf praxisorientierte Probleme übertragen • können eine Systeminfrastruktur eines verteilten Systems entwerfen und realisieren; • können eine Middleware eines verteilten Systems entwerfen und realisieren; • können ein Konzept für replizierte Daten entwerfen und realisieren; 		
Inhalte	<ul style="list-style-type: none"> • Eine Einführung im Sinne einer Beschreibung der charakteristischen Eigenschaften verteilter Systeme • Interprozesskommunikation zwischen verteilten Prozessen und einschlägige Programmiermodelle • Zeit, Koordination und Übereinstimmung • Wahlen, Wechselseitiger Ausschluss • Replikation • Vertiefung in eines oder mehrere der folgenden Themen: <ul style="list-style-type: none"> - Namensdienste und exemplarische Anwendungen - Verteilte Transaktionen - Verteilte Dateisysteme - Fehlertoleranz 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> • G. Bengel, C. Baun, M. Kunze, K.-U. Stucky: Masterkurs Parallele und Verteilte Systeme • Oliver Haase: Kommunikation in verteilten Anwendungen • G. Coulouris, J. Dollimore, T. Kindberg. Distributed Systems: Concepts and Design. • A.S. Tanenbaum, M.v. Stehen. Distributed Systems: Principles and Paradigms • Gerard Tel: Introduction to Distributed Algorithms • Nancy A. Lynch. Distributed Algorithms • Peter Mandl: Masterkurs Verteilte betriebliche Informationssysteme • A. Schill, T. Springer: Verteilte Systeme 		

Modulbezeichnung	Verteilte Systeme	Kürzel	VS / VSP
	<ul style="list-style-type: none">• Skripte der Dozenten		

Modulbezeichnung	IT-Sicherheit	Kürzel	ITS / ITSP
Lehrveranstaltung(en)	SeU: IT-Sicherheit Praktikum: IT-Sicherheit	Fach-semester	6
Arbeitsaufwand	54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Martin Hübner, Prof. Dr. Klaus-Peter Kossakowski	Turnus	semesterweise
Art des Moduls	Pflichtmodul	CP	6
Voraussetzungen	Empfohlen: Programmiermethodik I, Programmiertechnik, Programmiermethodik II, Betriebssysteme, Rechnernetze	SWS	3+1
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> verstehen Konzepte und Methoden zur Konstruktion von sicheren verteilten Systemen und können diese praktisch anwenden verstehen Sicherheitsmodelle und Sicherheitseigenschaften von Betriebssystemen und können diese zum sicheren Betrieb von Anwendungen einsetzen können Angriffstechniken in Netzwerken sowie den gezielten Einsatz von Abwehrmaßnahmen beurteilen 		
Inhalte	<ul style="list-style-type: none"> Einführung in das Security Engineering Angriffstechniken Grundlagen der Kryptographie Public Key Infrastrukturen (PKI) Authentifikations- und Autorisationsmodelle und –verfahren Sicherheitsprotokolle in Kommunikationsnetzen Weitere Themen nach Aktualität 		
Lehr- und Lernformen	SeU: Tafelarbeit, Rechnerpräsentation, freiwillige Übungsaufgaben Praktikum: Bearbeitung von Aufgaben in Gruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	Jeweils in der aktuellen Auflage: <ul style="list-style-type: none"> Claudia Eckert: IT-Sicherheit, Oldenbourg Verlag Klaus Schmech: Kryptografie, dpunkt-Verlag Skripte der Dozenten 		

Modulbezeichnung	Bachelorarbeit	Kürzel	BA / BAK
Lehrveranstaltung(en)	Bachelorarbeit	Semester	6
Arbeitsaufwand	360 Std. Bachelorarbeit, 90 Std. Kolloquiumsvorbereitung und -durchführung	Dauer	6 Monate
Modulverantwortliche(r)	Prof. Dr. Olaf Zukunft	Turnus	regelmäßig
Art des Moduls	Pflichtmodul	CP	12 + 3
Voraussetzungen	Die Bachelorarbeit kann angemeldet werden, wenn alle bis auf drei Module erfolgreich abgelegt worden sind. Der Umfang der noch nicht erfolgreich abgelegten Module darf 18 Leistungspunkte nicht übersteigen.	SWS	-
Verwendbarkeit	für den Studiengang „Angewandte Informatik“	Sprache	deutsch oder englisch
Zu erwerbende Kompetenzen / Lernergebnisse	<p>In der Bachelorarbeit zeigen die Studierenden, dass sie in der Lage sind, ein Problem aus den wissenschaftlichen, anwendungsorientierten oder beruflichen Tätigkeitsfeldern dieses Studiengangs selbständig unter Anwendung wissenschaftlicher Methoden und Erkenntnisse zu bearbeiten und dabei in die fächerübergreifenden Zusammenhänge einzuordnen.</p> <p>Folgende Kompetenzen werden erworben:</p> <ul style="list-style-type: none"> • Kompetenz, sich in das Thema einzuarbeiten, es einzuordnen, einzugrenzen, kritisch zu bewerten und weiter zu entwickeln • Kompetenz, das Thema anschaulich und formal angemessen in einem bestimmten Umfang schriftlich darzustellen • Kompetenz, die wesentlichen Ergebnisse der Arbeit fachgerecht und anschaulich in einem Vortrag einer vorgegebenen Dauer zu präsentieren • Kompetenz, aktiv zu fachlichen Diskussionen beizutragen 		
Inhalte	<ul style="list-style-type: none"> • Die Bachelorarbeit ist eine theoretische, empirische und/oder experimentelle Abschlussarbeit mit schriftlicher Ausarbeitung, die individuell durchgeführt wird. • Typischerweise umfasst sie auch eine Entwurfs- und Implementierungsleistung des Studierenden, die allein aber nicht ausreichend ist. 		
Lehr- und Lernformen	-		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Benotete Bachelorarbeit und benotetes Kolloquium (KO)		
Literatur	<ul style="list-style-type: none"> • abhängig vom Thema 		

Modulbezeichnung	WP-Kürzel-Kurztitel (diese Beschreibung gilt für alle drei Wahlpflichtmodule WP I, WP II und WP III)	Kürzel	WP / WPP
Lehrveranstaltung(en)	SeU: WP-Kürzel-Kurztitel Praktikum: WP-Kürzel-Kurztitel	Fachsemester	4-6
Arbeitsaufwand	36 Std. SeU, 36 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium (bei 2+2) oder 54 Std. SeU, 18 Std. Praktikum, 108 Std. Eigenarbeit/Selbststudium (bei 3+1)	Dauer	ein Semester
Modulverantwortliche(r)	Anbieter/-in des Moduls	Turnus	nach Angebot und Wahl
Art des Moduls	Wahlpflichtmodul	CP	6
Voraussetzungen	Falls notwendig angeben: „Empfohlen: <empfohlene Module>“	SWS	2+2 oder 3+1
Verwendbarkeit	für die Studiengänge „Angewandte Informatik“, „Informatik Technischer Systeme“, „Wirtschaftsinformatik“ und „European Computer Science“	Sprache	deutsch oder englisch
Zu erwerbende Kompetenzen / Lernergebnisse	Je nach Wahlpflichtmodul. <ul style="list-style-type: none"> Die Beschreibungen der aktuellen Angebote sind auf den Informatikseiten zu finden. 		
Inhalte	Die Studierenden wählen die Wahlpflichtmodule aus einem Pool, der für die Informatik-Bachelorstudiengänge des Department Informatik gemeinsam angeboten wird. Die Beschreibung des aktuellen Angebots ist auf den Informatikseiten zu finden. <ul style="list-style-type: none"> ... 		
Lehr- und Lernformen	SeU: Seminaristischer Unterricht, Tafelarbeit, Multimedia-Präsentationen, Beispiele und Demos Praktikum: Bearbeitung von Aufgaben in Zweiergruppen mit abschließendem Abnahmegespräch		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: benotete Klausur Alternative Prüfungsformen: benotete mündliche Prüfung oder benotetes Referat oder benotete Hausarbeit Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben. Voraussetzung (PVL): Laborabschluss (erfolgreiche Bearbeitung der Aufgaben)		
Literatur	<ul style="list-style-type: none"> abhängig vom Wahlpflichtmodul 		

Modulbezeichnung	Gesellschaftswissenschaften	Kürzel	GW
Lehrveranstaltung(en)	SeU: GW-Kürzel-Kurztitel	Fach-semester	6
Arbeitsaufwand	36 Std. SeU, 54 Std. Eigenarbeit/Selbststudium	Dauer	ein Semester
Modulverantwortliche(r)	Prof. Dr. Julia Padberg	Turnus	nach Angebot und Wahl
Art des Moduls	Wahlpflichtmodul	CP	3
Voraussetzungen	Falls notwendig angeben: „Empfohlen: <empfohlene Module>“	SWS	2
Verwendbarkeit	für die Studiengänge „Angewandte Informatik“, „Informatik Technischer Systeme“, „Wirtschaftsinformatik“ und „European Computer Science“	Sprache	deutsch oder englisch
Zu erwerbende Kompetenzen / Lernergebnisse	Die Studierenden sind in der Lage <ul style="list-style-type: none"> aktuelle Fragestellungen gesellschaftlicher Natur im Umfeld der Informatik zu debattieren und sich weiterführende Literatur beschaffen und interpretieren die entsprechenden Methoden die das wissenschaftliche Arbeiten, die berufliche Praxis oder die persönliche Entwicklung betreffen, anwenden und erfolgreich nutzen. 		
Inhalte	Die Studierenden wählen die Gesellschaftswissenschaftenmodule aus einem Pool, der für die Informatik-Bachelorstudiengänge des Department Informatik gemeinsam angeboten wird. Die Beschreibung des aktuellen Angebots ist auf den Informatikseiten zu finden. <ul style="list-style-type: none"> Die Studierenden wählen das gesellschaftswissenschaftliche Fach aus einem Pool, der für die Informatik-Bachelorstudiengänge des Department Informatik gemeinsam angeboten wird. Die gesellschaftswissenschaftlichen Fächer dürfen sich nicht mit den Inhalten der Pflichtmodule überschneiden. Themengebiete mit gesellschaftswissenschaftlicher Relevanz und einem Bezug zur Informatik können Inhalte sein. Darüber hinaus können GW-Fächer Themen, die das wissenschaftliche Arbeiten, die berufliche Praxis oder die persönliche Entwicklung betreffen, beinhalten. Die entsprechenden Methoden und Kompetenzen sollen didaktisch aufbereitet und professionell präsentiert werden. 		
Lehr- und Lernformen	SeU: Seminaristischer Unterricht, praktische Übungsaufgaben		
Voraussetzungen für die Vergabe von Leistungspunkten (Studien- und Prüfungsleistungen)	Regelhafte Prüfungsform: unbenotete Klausur Alternative Prüfungsformen: nicht benotete mündliche Prüfung oder nicht benotetes Referat oder nicht benotete Hausarbeit oder nicht benoteter Übungsabschluss Bei mehr als einer möglichen Prüfungsform im Modul wird die zu erbringende Prüfungsform von dem verantwortlichen Lehrenden zu Beginn der Lehrveranstaltung bekannt gegeben.		
Literatur	<ul style="list-style-type: none"> abhängig vom Gesellschaftswissenschaftenmodul 		