

MODULHANDBUCH

des Zertifikatsfernstudiengang Informatik (Weiterbildung)
im Fachbereich Informatik
Fach-Prüfungsordnung 2026

Inhaltsverzeichnis

Hinweise zu den Modulhandbucheinträgen	3
Android-Programmierung (AP).....	4
Automatentheorie, Formale Sprachen und Berechenbarkeit (AUFS)	5
Bildverarbeitung und Deep Learning (BDL)	7
C# und .NET (CN)	8
Datenbanksysteme (DBS)	9
Einführung in die Programmierung (PROG)	11
Embedded Systems (ES)	13
Fortgeschrittene Programmiertechniken (FOPT).....	15
Informatik in Produktion und Materialwirtschaft (IPM).....	17
Informatik und Gesellschaft (IUG).....	18
IT-Sicherheit (ITS)	19
Kommunikative Kompetenz (KOM).....	20
Projektmanagement (PROM)	22
Projektarbeit	23
Rechnernetze (RN).....	24
Rust in Aktion: Spieleprogrammierung und Parallelität (RUST).....	26
Software Engineering (SE)	27

Hinweise zu den Modulhandbucheinträgen

- Die Dauer aller Module beträgt jeweils ein Semester.
- Die zeitliche Lage aller Module ergibt sich aus dem unter <https://www.hochschule-trier.de/informatik/fernstudium/studium/masterfernstudium/studienverlauf-vollzeitteilzeit> dargestellten Studienverlauf in Voll- bzw. Teilzeit. Es wird empfohlen, die Module in der vorgeschlagenen Reihenfolge zu absolvieren. Die Reihenfolge ist jedoch nicht verpflichtend.
- Zur individuellen Vorbereitung kann der Besuch der im Feld „Empfohlene Voraussetzungen“ genannten Module dienen, ebenso wie die Angaben im Feld „Literatur“. Weitere Literaturangaben finden sich ggf. in den Vorlesungsunterlagen, die i.d.R. auf OpenOLAT (<https://olat.vcrp.de/dmz>) zur Verfügung gestellt werden.
- Falls in einem Modul eine Studienleistung zu erbringen ist, ist diese gemäß § 4 der allgemeinen Prüfungsordnung Voraussetzung für die Zulassung zur Prüfung. Die semesteraktuelle Form der Studienleistung wird zu Beginn der Lehrveranstaltung bekannt gegeben.
- Bei Angabe mehrerer alternativer Prüfungsformen für ein Modul wird die semesteraktuelle Prüfungsform zu Beginn der Lehrveranstaltung bekannt geben.
- Voraussetzung für die Vergabe von ECTS-Punkten ist eine Bewertung der Prüfungsleistung mit mindestens der Note „ausreichend“.
- Die Gewichtung eines Modulergebnisses zur Bildung der Gesamtnote ergibt sich gemäß § 14 der Fachprüfungsordnung aus dem Verhältnis der ECTS-Punkte für das Modul und der Gesamtanzahl der ECTS-Punkte.

Android-Programmierung (AP)

Fachgebiet	Praktische Informatik		
Inhalte	<ul style="list-style-type: none"> ▪ Mobilgeräte <ul style="list-style-type: none"> ▪ Einführung, ▪ Softwareüberblick ▪ Android: <ul style="list-style-type: none"> ▪ Basistechniken, ▪ grafische Benutzeroberflächen (GUI), ▪ persistente Datenspeicherung, ▪ Komponenten und Nebenläufigkeit, ▪ Kommunikation, ▪ Sensoren und ortsabhängige Dienste, ▪ Sicherheit und Datenschutz, ▪ Vertrieb und Kommerzialisierung 		
Lernergebnisse	Dieses Modul versetzt Sie am Beispiel der Plattform Android in die Lage, selbstständig Applikationen für Mobilgeräte zu spezifizieren und zu programmieren. Sie können zudem gegebene Applikationen analysieren und bewerten.		
Lehrform	<ul style="list-style-type: none"> ▪ Begleittext mit Einführung in die jeweilige Thematik ▪ Lehrmaterial mit Erklärvideos zu Quelltexten und vollständig ausführbaren Android-Programmen ▪ Übungsaufgaben ▪ Praktische Phase (4 Tage zzgl. Prüfungstag) 		
Empfohlene Voraussetzungen	Sichere Programmierkenntnisse in Java; praktische Erfahrungen mit einer Programmierumgebung (IDE) sind wünschenswert. Zudem müssen Sie englische Texte lesen können, um die Android-Online-Dokumentation zu verstehen. Wünschenswert, aber nicht zwingend erforderlich, sind weiterhin Grundkenntnisse in der nebenläufigen Programmierung (insbesondere Java-Threading), über Internet-Protokolle (insbesondere Socket-Schnittstelle) sowie relationale Datenbanken (insbesondere SQL).		
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Künneth, Thomas: Android 11: Das Praxisbuch für App-Entwickler, Rheinwerk Verlag.		
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input checked="" type="checkbox"/> Teilnahme an der Praktischen Phase		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	Prof. Dr. C. Vogt, Technische Hochschule Köln		
Modulverantwortliche(r)	Prof. Dr. C. Vogt, Technische Hochschule Köln		
Änderungsdatum	19.08.2022		

Automatentheorie, Formale Sprachen und Berechenbarkeit (AUFS)

Fachgebiet	Theoretische Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ Teil 1: Endliche Automaten und reguläre Sprachen <ul style="list-style-type: none"> ▪ Deterministische endliche Automaten ▪ Nichtdeterministische endliche Automaten ▪ Endliche Automaten mit ϵ-Übergängen ▪ Minimale endliche Automaten ▪ Anwendungen endlicher Automaten ▪ Reguläre Ausdrücke ▪ Typ-3-Grammatiken ▪ Eigenschaften regulärer Sprachen ▪ Endliche Maschinen ▪ Teil 2: Kontextfreie Sprachen und Kellerautomaten <ul style="list-style-type: none"> ▪ Kontextfreie Grammatiken ▪ Kellerautomaten ▪ Anwendungen kontextfreier Sprachen ▪ Teil 3: Berechenbarkeit und Komplexität <ul style="list-style-type: none"> ▪ Typ-1- und Typ-0-Sprachen ▪ Turingautomaten ▪ Berechenbarkeit ▪ Entscheidbarkeit ▪ Komplexität 	
Lernergebnisse	<p>Die Studierenden kennen die Definitionen der formalen Grammatiktypen sowie die zugehörigen äquivalenten Automatenmodelle. Sie können die Resultate zur Entscheidbarkeit der Sprachen in diesen Klassen erläutern. Sie können Abgrenzungen der Klassen innerhalb der Chomsky-Hierarchie und die Äquivalenzen zwischen Grammatiken und Automaten erklären sowie die Abschlusseigenschaften beweisen. Sie verstehen die Pumping-Lemmata und können diese auf gängige Beispiele anwenden. Sie kennen die Resultate zur Entscheidbarkeit von Sprachen, insbesondere wissen sie, dass unentscheidbare und nicht rekursiv aufzählbare Sprachen existieren, kennen Beispiele hierzu und können deren Nichtentscheidbarkeit bzw. deren nicht rekursive Aufzählbarkeit beweisen. Die Studenten können Berechenbarkeitsmodelle und deren Zusammenhänge erklären. Sie kennen die Definition der Zeit-Komplexität von deterministischen und nicht deterministischen Turingmaschinen. Die Studierenden verstehen die Bedeutung des P=NP-Problems, insbesondere dessen praktische Relevanz, und sie verstehen die Bedeutung der Klasse NPC der NP-vollständigen Probleme, insbesondere hinsichtlich der Beantwortung der P=NP-Frage. Sie können Methoden zum Beweis der Zugehörigkeit einer Sprache zur Klasse NPC erläutern.</p>	
Lehrform	<ul style="list-style-type: none"> ▪ Lehrbuch der Theoretischen Informatik ▪ Begleittext, der zum Studium des Lehrbuchs anleitet ▪ Übungsaufgaben ▪ Studienbegleitendes Online-Tutorium 	
Empfohlene Voraussetzungen	Mathematische Grundkenntnisse über die Begriffe „Menge“, „Relation“ und „Funktion“ in einem Umfang und einer Tiefe, wie sie das Propädeutikum Mathematik vermittelt. Fähigkeit zum abstrakten Denken.	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Vossen, Gottfried; Witt, Kurt-Ulrich: Grundkurs Theoretische Informatik - Eine anwendungsbezogene Einführung - Für Studierende in allen Informatik-Studiengängen, Springer, 2016 ▪ Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D.: Einführung in die Automatentheorie, formale Sprachen und Berechenbarkeit; 3. aktualisierte Auflage. Pearson Studium, 2018 [Reprint von 2011]; ISBN 3-86894-082-0 ▪ Schöning, Uwe: Theoretische Informatik – kurz gefasst; 5. Auflage. Heidelberg: Springer Spektrum, 2008; ISBN 978-3-8274-1824-1 ▪ Kozen, Dexter: Automata and Computability; New York: Springer, 1997 ▪ Sipser, Michael: Introduction to the Theory of Computation; Third Edition. Boston MA: Cengage Learning, 2013 ▪ Asteroth, Alexander; Baier, Christel: Theoretische Informatik: Eine Einführung in Berechenbarkeit, Komplexität und formale Sprachen mit 101 Beispielen; Pearson Studium, 2003; ISBN 978-3-8273-7033 	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input type="checkbox"/> Teilnahme an der Praktischen Phase	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) 	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF

	• Zertifikatsfernstudiengang Informatik (Weiterbildung)		
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	10 Stunden	290 Stunden
Herausgeber / Autor	Prof. Dr. K.-U. Witt, Hochschule Bonn-Rhein Sieg, St. Augustin		
Modulverantwortliche(r)	Prof. Dr. K.-U. Witt, Hochschule Bonn-Rhein Sieg, St. Augustin		
Änderungsdatum	15.06.2026		

<h1>Bildverarbeitung und Deep Learning (BDL)</h1>		
Fachgebiet	Angewandte Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ Vorkurs Python <ul style="list-style-type: none"> ▪ Grundlagen ▪ Datenanalyse und Visualisierung ▪ Algorithmische Bildverarbeitung <ul style="list-style-type: none"> ▪ Repräsentation von Bildern, Farbräume ▪ Verfahren der Bildvorverarbeitung ▪ Filterung im Orts- und Spektralbereich ▪ Texturanalyse ▪ Segmentierungsverfahren ▪ Merkmalsextraktion und Klassifikation ▪ Deep Learning <ul style="list-style-type: none"> ▪ Theoretische Grundlagen Neuronaler Netze ▪ Vollständig verknüpfte Netze ▪ Convolutional Neural Networks, Rekurrente Netze ▪ Architekturen zur Bildklassifikation ▪ Semantische Segmentierung 	
Lernergebnisse	<p>Dieses Modul versetzt Sie in die Lage, moderne Bildverarbeitungsalgorithmen in Python und aktuelle Architekturen Neuronaler Netze mittels TensorFlow zu entwickeln. Im Bereich der Bildverarbeitung beherrschen Sie die Methoden der gesamten Verarbeitungskette, ausgehend von der Bildvorverarbeitung zur Bildverbesserung, über Segmentierungsverfahren bis hin zur Merkmalsextraktion für die Klassifikation von Bildinhalten. Sie besitzen fundierte theoretische Kenntnisse von Bildverarbeitungsverfahren und können diese mittels Python in die Praxis überführen. In dem sich gegenwärtig sehr dynamisch entwickelnden Bereich des Deep Learnings beherrschen Sie die theoretischen Grundlagen des Funktionsprinzips Neuronaler Netze. Sie kennen wichtige Netzarchitekturen und sind in der Lage diese mittels TensorFlow zu implementieren, zu trainieren und zu evaluieren.</p>	
Lehrform	<ul style="list-style-type: none"> ▪ Vorkurs Python (Der Vorkurs wird 3-4 Wochen vor Lehrveranstaltungsbeginn zur Verfügung gestellt.) ▪ 8 Kurseinheiten ▪ Lehrbuch, Jupyter Notebooks, Lehrvideos ▪ Übungsaufgaben ▪ Regelmäßige Webmeetings für fachliche Fragen 	
Empfohlene Voraussetzungen	<p>Voraussetzung für das Modul sind elementare Kenntnisse der objektorientierten Programmierung sowie grundlegende Python-Kenntnisse. Das für das Modul erforderliche Wissen zur Programmiersprache Python wird in einem Vorkurs vor Beginn des Semesters vermittelt. Eine vertiefte Einarbeitung in die Python-Programmierung ist nicht erforderlich.</p> <p>Für das Modul sind grundlegende Mathematikkenntnisse in linearer Algebra und Analysis erforderlich. Die weiteren für das Modul relevanten Mathematikkenntnisse werden innerhalb des Moduls vermittelt (Statistik, komplexe Zahlen, Faltung, Fourier-Transformation etc.).</p>	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Burger, Wilhelm; Burge, Mark, James: Digitale Bildverarbeitung	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input type="checkbox"/> Teilnahme an der Praktischen Phase	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	10 Stunden
		Selbststudium
		290 Stunden
Herausgeber / Autor	Prof. Dr. Jörg Lohscheller, Hochschule Trier	
Modulverantwortliche(r)	Prof. Dr. Jörg Lohscheller, Hochschule Trier	
Änderungsdatum	22.06.2026	

C# und .NET (CN)		
Fachgebiet	Praktische Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ C# und .NET <ul style="list-style-type: none"> ▪ Einführung in .NET und Grundlagen der Sprache C# ▪ Vererbung, Polymorphie und Interfaces ▪ Delegaten, Ereignisse und Lambda-Ausdrücke ▪ Auflistungsklassen [Collections] und generische Datentypen [Generics] ▪ Weitere C#-Sprachfeatures und Language Integrated Query (LINQ) ▪ Dateien, Streams und Serialisierung ▪ Multithreading und die Task Parallel Library (TPL) ▪ Windows Presentation Foundation (WPF) <ul style="list-style-type: none"> ▪ Layoutcontainer und Steuerelemente ▪ Dependency Properties und Datenbindung ▪ Ereignisse und Commands ▪ Das MVVM-Pattern 	
Lernergebnisse	<p>Dieses Modul versetzt Sie in die Lage, moderne Konsolen- und Desktop-Anwendungen mit C# zu entwickeln. Sie beherrschen sämtliche Sprachgrundlagen von C#, die Konzepte der objektorientierten Programmierung und wichtige Programmier Techniken (Serialisierung, Multithreading, TPL, LINQ usw.). Sie kennen grundlegende .NET-Klassen und beherrschen die Entwicklung und das Debugging mit Visual Studio. Sie besitzen fundierte Kenntnisse in der Entwicklung von grafischen Benutzungsschnittstellen mit der Windows Presentation Foundation (WPF): Sie kennen die wichtigsten Layout-Panels und Controls und können diese gezielt auswählen und verwenden. Sie können Ressourcen, Stile, Trigger und Vorlagen definieren und in einer WPF-Anwendung verwenden. Sie beherrschen den Umgang mit Ereignissen und Befehlen. Sie können Datenbindungen realisieren und eigene Wertkonverter, Filter und Ansichten implementieren. Sie kennen wichtige Entwurfsmuster für interaktive Benutzeroberflächen und können diese implementieren.</p>	
Lehrform	<ul style="list-style-type: none"> ▪ Lehrbuch mit Begleittext ▪ Studienbegleitendes Tutorium ▪ Praktische Phase (4 Tage zzgl. Prüfungstag) 	
Empfohlene Voraussetzungen	<p>In diesem Modul werden viele Bereiche der Informatik angerissen und nur kurz im speziellen Kontext von .NET und C# dargestellt, z.B. Multithreading, Konzepte der objektorientierten Programmierung, Auflistungsklassen und generische Datentypen. Die Module des Fernstudiums Informatik, insbesondere die Module PROG und FOPT, bereiten nur teilweise auf die Anforderungen vor. Sollten Ihnen einzelne Vorkenntnisse fehlen, setzen wir die Bereitschaft voraus, sich das nötige Wissen mit Hilfe des Lehrbuchs anzueignen.</p>	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Kühnel, Andreas; Deitelhoff, Fabian (2025): C# mit.NET. Das umfassende Handbuch. Rheinwerk Verlag [Rheinwerk computing].	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input checked="" type="checkbox"/> Teilnahme an der Praktischen Phase	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	50 Stunden
		Selbststudium
		250 Stunden
Herausgeber / Autor	Prof. Dr. Martin Rumpler, Hochschule Trier, Umwelt-Campus Birkenfeld	
Modulverantwortliche(r)	Prof. Dr. Martin Rumpler, Hochschule Trier, Umwelt-Campus Birkenfeld	
Änderungsdatum	15.06.2026	

Datenbanksysteme (DBS)

Fachgebiet	Praktische Informatik
Inhalte	<ul style="list-style-type: none"> ▪ Grundlagen <ul style="list-style-type: none"> ▪ Systemarchitektur ▪ Schichtenmodell ▪ Datenbankentwurf <ul style="list-style-type: none"> ▪ Entity-Relationship-Modell ▪ Relationales Modell <ul style="list-style-type: none"> ▪ Definition des relationalen Modells ▪ relationale Algebra ▪ Transformation des ER-Modells ins relationale Modell ▪ SQL <ul style="list-style-type: none"> ▪ Datendefinition ▪ Datenmanipulation ▪ Anfragen ▪ Relationale Entwurfstheorie <ul style="list-style-type: none"> ▪ Anomalien ▪ funktionale Abhängigkeiten ▪ Normalformen ▪ Transaktionen <ul style="list-style-type: none"> ▪ ACID-Eigenschaften ▪ Mehrbenutzersynchronisation ▪ Recovery ▪ Physische Speicherorganisation und Anfrageoptimierung <ul style="list-style-type: none"> ▪ Speicherverwaltung ▪ Indizes ▪ Anwendungsentwicklung mit Datenbanken <ul style="list-style-type: none"> ▪ Programmierschnittstellen ▪ Architekturmuster ▪ OLAP und Data Warehouse <ul style="list-style-type: none"> ▪ Dimensionale Modellierung nach Kimball ▪ Analytische Anfragen in SQL ▪ Semistrukturierte Daten <ul style="list-style-type: none"> ▪ JSON ▪ XML ▪ NoSQL-Datenbanksysteme <ul style="list-style-type: none"> ▪ Kategorien von NoSQL-DBS ▪ Key-Value-Stores ▪ dokumentenorientierte DBS
Lernergebnisse	<p>Nach erfolgreicher Bearbeitung des Moduls können die Studierenden</p> <ul style="list-style-type: none"> ▪ Konzepte unterschiedlicher Datenbanksysteme verstehen ▪ Datenmodelle im Entity-Relationship-Modell erstellen und ins relationale Modell überführen ▪ Daten in SQL definieren, manipulieren und anfragen ▪ relationale Datenmodelle analysieren und anhand der Normalformen optimieren ▪ dimensionale Modelle für Data Warehouses erstellen und anfragen ▪ Anwendungsfälle für semistrukturierte Daten erkennen und semistrukturierte Daten modellieren ▪ Einsatzzwecke für verschiedene NoSQL-Datenbanksysteme erkennen und geeignete Systeme auswählen
Lehrform	<ul style="list-style-type: none"> ▪ Lehrbuch und Begleittext ▪ 4 Lehrhefte ▪ Übungsaufgaben ▪ studienbegleitendes Online-Tutorium
Empfohlene Voraussetzungen	Gute mathematische, insbesondere algebraische, Grundkenntnisse in einem Umfang und einer Tiefe, wie sie das Propädeutikum Mathematik vermittelt. Fähigkeit zum abstrakten Denken. Grundkenntnisse in einer Programmiersprache (z.B. C, C++, Java).
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Elmasri, Ramez A.; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen. Pearson Studium, 2009; ISBN 978-3-86894-012-1 ▪ Kemper, Alfons; Eickler, André: Datenbanksysteme: eine Einführung; 10. Aufl. De Gruyter, 2015; ISBN 3-11-044375-9-6 ▪ Kimball, Ralph; Ross, Margy: The Data Warehouse Toolkit; 3. Aufl. Wiley, 2013. ISBN 978-1118530801. ▪ Redmond, Eric; Wilson, Jim R.: Sieben Wochen, sieben Datenbanken. Moderne Datenbanken und die NoSQL-Bewegung; O'Reilly, 2012. ISBN 978-3-86899-791-0
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input checked="" type="checkbox"/> Teilnahme an der Praktischen Phase
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur

	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	Prof. Dr. Christoph Schmitz		
Modulverantwortliche(r)	Prof. Dr. Christoph Schmitz		
Änderungsdatum	15.06.2026		

<h1>Einführung in die Programmierung (PROG)</h1>	
Fachgebiet	Praktische Informatik
Inhalte	<ul style="list-style-type: none"> ▪ Einführung in Java <ul style="list-style-type: none"> ▪ Grundelemente von Java ▪ Operatoren und Ausdrücke ▪ Einführung in die Objektorientierte Programmierung ▪ Klassen und Objekte ▪ Anweisungen ▪ Zeichenketten und Felder ▪ Ausnahmebehandlung ▪ Module ▪ Vererbung ▪ Datenstrukturen und Algorithmen <ul style="list-style-type: none"> ▪ Einführung: formale, algorithmische und implementierungstechnische Beschreibungsebene ▪ Effizienzbetrachtungen: O-, Ω- und Θ-Notation [Landau'sche Symbole] ▪ Fortgeschrittene Datenstrukturen: einfach und doppelt verkettete Listen, binäre (Such-)Bäume, Heaps, AVL-Bäume, Hashtabellen ▪ Einfache und fortgeschrittene Sortierverfahren (Bubble, Insertion, Selection, Merge, Quick, Heap, Bucket Sort) ▪ Java Collections Framework
Lernergebnisse	<p>Datenstrukturen und darauf arbeitende Algorithmen sind die Grundlage für die Entwicklung von Programmen. Bei der Spezifikation und Erstellung von effizienten Algorithmen spielt die geeignete Auswahl der zugrundeliegenden Datenstrukturen eine essentielle Rolle. Von daher sind Datenstrukturen und Algorithmen untrennbar ineinandergreifende Themen, die von jedem Informatiker und jeder Informatikerin beherrscht werden sollten, um effiziente Programme zu realisieren. Das Modul vermittelt nicht nur die theoretischen Konzepte zum Thema, sondern auch, wie die verschiedenen Datenstrukturen und Algorithmen in der Programmiersprache Java implementiert werden können. Dazu ist es notwendig, zunächst die grundlegenden Programmierkenntnisse in Java zu erlernen. Neben den allgemeinen Prinzipien der objektorientierten Programmierung werden sowohl die Basiselemente der Sprache als auch wichtige Themen wie Vererbung, Sichtbarkeit, Ausnahmebehandlung eingeführt. Das Modul führt Teilnehmer, die bisher keine oder nur wenig Berührung mit Programmierung hatten, in alle genannten Konzepte ein und befähigt sie dazu, nicht nur effektive, sondern auch effiziente Programme zu grundlegenden Fragestellungen der Informationsverarbeitung (Suchen, Sortieren) zu erstellen.</p> <p>In der praktischen Phase werden die erworbenen Kompetenzen anhand themenbezogener Programmieraufgaben vertieft, um damit die Basis für das erfolgreiche Absolvieren weiterführender Module, insbesondere „Fortgeschrittene Programmierertechniken“, zu schaffen.</p>
Lehrform	<ul style="list-style-type: none"> ▪ Webbasierter Online-Kurs ▪ Übungsaufgaben ▪ Studienbegleitendes Tutorium ▪ Praktische Phase (4 Tage zzgl. Prüfungstag)
Empfohlene Voraussetzungen	Mathematische Grundkenntnisse, wie sie das Propädeutikum Mathematik vermittelt, sowie eine mathematisch-logische Begabung. Programmierkenntnisse sind nicht erforderlich.
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Einführung in Java <ul style="list-style-type: none"> ▪ Krüger, Guido; Hansen, Heiko: Java-Programmierung. Das Handbuch zu Java 8. O'Reilly-Verlag ▪ Ratz, Dietmar; Schulmeister-Zimolong, Dennis; Seese, Detlef: Grundkurs Programmieren in Java. Hanser-Verlag ▪ Schiedermeier, Reinhard: Programmieren mit Java. Pearson Studium ▪ Ullenboom, Christian: Java ist auch eine Insel. Einführung, Ausbildung, Praxis. Rheinwerk-Verlag ▪ Datenstrukturen und Algorithmen <ul style="list-style-type: none"> ▪ Güting, Ralf H.; Dieker, Stefan: Datenstrukturen und Algorithmen ▪ Ottmann, Thomas; Widmayer, Peter: Algorithmen und Datenstrukturen ▪ Cormen, Thomas H.: Algorithmen – Eine Einführung. Oldenbourg-Verlag ▪ Sattler, Kai-Uwe; Saake, Gunter: Algorithmen und Datenstrukturen – Eine Einführung mit Java. dpunkt.verlag
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Tests <input checked="" type="checkbox"/> Bearbeitung von Programmieraufgaben <input checked="" type="checkbox"/> Teilnahme an der Praktischen Phase
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) <div style="text-align: right;"> <input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF </div>

Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	Prof. Dr. Andreas Künkler, Prof. Dr. Andreas Lux, Hochschule Trier		
Modulverantwortliche(r)	Prof. Dr. Georg Schneider, Prof. Dr. Andreas Lux, Hochschule Trier		
Änderungsdatum	15.06.2026		

<h1>Embedded Systems (ES)</h1>	
Fachgebiet	Praktische Informatik
Inhalte	<ul style="list-style-type: none"> ▪ Digitaltechnik und Rechnerarchitektur ▪ Grundlagen von Betriebssystemen ▪ Programmieren in C und Einführung in die Systemprogrammierung unter Linux ▪ Embedded Linux ▪ Embedded Sensors ▪ Kommunikationsverfahren für Sensoren, auch unter Einbeziehung von IoT-Plattformen
Lernergebnisse	<p>Dieses Modul ist eine praxisnahe Einführung in Entwurf, Aufbau, Konfiguration und Inbetriebnahme von Eingebetteten Systemen mit dem Ziel, solche Systeme systematisch, effizient und zukunftssicher in den folgenden Bereichen zu erstellen: Telekommunikation; Haushalt- und Unterhaltungselektronik; Industrielle Steuerungs- und Automatisierungstechnik; Bürokommunikation</p> <p>Die Teilnehmenden werden dazu in die Lage versetzt, die Aufgaben eines Embedded Systems zu definieren, Probleme bei der Realisierung zu identifizieren und zu untersuchen, Lösungsideen zu produzieren und Lösungswege zu gestalten.</p> <ul style="list-style-type: none"> ▪ Nach der Bearbeitung der Kurseinheit Digitaltechnik und Rechnerarchitektur können die Teilnehmenden Schaltfunktionen und Schaltwerke entwerfen. Mikroprozessoren sind die Kernelemente von Eingebetteten Systemen; ihr Aufbau und ihre Arbeitsweise werden in dem Abschnitt Rechnerarchitektur vorgestellt. Nach der Bearbeitung der Kurseinheit kennen die Teilnehmenden Aufbau und die Realisierung von Rechnerbaugruppen und die Grundlagen der Assemblerprogrammierung. Am Beispiel eines modernen Mikrocontrollers wird ein Überblick über die Eigenschaften von Mikrocontrollern vermittelt. ▪ Der heutigen Erwartungshaltung an die Funktionalität eines Eingebetteten Systems genügen fast nur noch Softwaresysteme, die im Kern auf einem Betriebssystem beruhen. In der Kurseinheit Betriebssysteme erlernen die Teilnehmenden die zentralen Konzepte und Mechanismen, die einem modernen Betriebssystem zugrunde liegen. Sie wissen nach der Durcharbeitung, wie diese Betriebssystem-Software implementiert ist und können aus eigenen Programmen heraus auf die Dienste eines Betriebssystems (Linux) zugreifen. ▪ Zugriffe auf Hardwarekomponenten erfolgen in den Treibern des Betriebssystems des eingebetteten Systems. Nach der Bearbeitung der Kurseinheit Systemprogrammierung wissen die Teilnehmenden, wie die Aufgabenteilung zwischen Betriebssystemkern und einem Treiber am Beispiel von Linux aussieht. Sie können zu realisierende Aufgaben so strukturieren, dass sie systematisch in Treibern implementiert werden können. Die Studierenden können Treiber für Hardwarekomponenten testen und auf korrekte Funktionalität überprüfen. ▪ In der Kurseinheit Embedded Linux lernen die Teilnehmenden, wie die Anforderungen an ein zukunftssicheres Embedded System effizient und kostengünstig umgesetzt werden können. Sie kennen die anwendungsnahe Funktionen des Linux-Kernels, häufig vorkommende Schnittstellen zur Kommunikation mit anderen Geräten und bewährte Dateisysteme. Aus dem Pool freier Software kennen sie wesentliche Teile und können diese für ihre Applikation nutzen, ohne eigenen Entwicklungsaufwand leisten zu müssen. ▪ Eingebettete Systeme leisten wertvolle Dienste in vielen Bereichen: In der Haushalts- und Unterhaltungselektronik (Smart Home) übernehmen sie Aufgaben wie Geräte- und Beleuchtungssteuerung und nicht zuletzt in der Absicherung gegen unbefugtes Eindringen in den Wohnbereich. In der industriellen Anwendung, z.B. der Steuerungs- und Automatisierungssysteme (Smart Industry) und Datenerfassung (Smart City, Smart Energy, Smart Farming) stehen Steuerungsaufgaben aber noch viel mehr die Erfassung von physikalischen Daten von mittels Sensoren. Als Oberbegriff aller dieser Smarten Technologien wird Internet of Things (IoT) verwendet. Der Beitrag, den Eingebettete Systeme hier leisten können, wird in zwei Kurseinheiten behandelt: 1.) In der Kurseinheit Embedded Sensors stehen physikalische und elektrische Gesichtspunkte der Informationsgewinnung im Vordergrund. 2.) Die Kurseinheit Kommunikationsverfahren für Sensoren, auch unter Einbeziehung von IoT-Plattformen geht auf die informationstechnischen Aspekte ein, die sich durch Einsatz von über Internet verbundenen Systemkomponenten ergeben. ▪ In der Kurseinheit Embedded Sensors werden die Teilnehmenden eingeführt in die (sensornahe) Messwerterfassung und -verarbeitung. Typische Sensoren (Kraftmessung, Temperatur, Helligkeit, Vibration/Schwingungen) werden vorgestellt und ihre Integration in das Eingebettete System exemplarisch beschrieben. An mehreren Beispielen lernen die Studierenden, wie typische Schnittstellen zwischen dem Embedded System und den Sensoren aussehen, wie Sensoren mit Energie aus der Umwelt versorgt werden können und wie Sensorsignale aufbereitet werden. ▪ In der Kurseinheit Kommunikationsverfahren für Sensoren, auch unter Einbeziehung von IoT-Plattformen wird die Signalverarbeitungskette vom Sensor über das Gateway bis zum Server dargestellt und besprochen. Die Verbindung zwischen Sensor und Gateway geschieht typischerweise über Funk. Besonderer Wert wird dabei auf Kenntnisse von energiesparenden Funkverfahren (Low Power WAN, am Beispiel von LoRa und Sigfox) gelegt, da sehr oft die Sensoren mobil oder sogar (energie)autark betrieben werden. Server arbeiten typischerweise nicht isoliert, sondern innerhalb einer IoT-Plattform oder einer IoT-Cloud. Die Studierenden lernen deshalb den Umgang mit einer IoT-Cloud: Sie erfassen Daten, transportieren sie in die Cloud, visualisieren und analysieren sie dort. Dabei wird auch der Umgang mit dem Nachrichtenprotokoll (MQTT) geübt, das sich besonders bei der Datenübertragung über Strecken mit geringer Bandbreite und/oder IoT-Geräten mit hoher Latenzzeit bewährt hat.
Lehrform	<ul style="list-style-type: none"> ▪ 6 Lehrhefte mit Übungsaufgaben ▪ Lehrbuch ▪ Software ▪ Praktische Phase (4 Tage zzgl. Prüfungstag)

Empfohlene Voraussetzungen	Grundkenntnisse in einer Programmiersprache		
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen		
	<input checked="" type="checkbox"/> Literatur: Mandl, Peter: Grundkurs Betriebssysteme 5. Auflage 2020 Zusätzliche, einführende oder begleitende Literatur kann den Literaturempfehlungen in den einzelnen Lehrheften entnommen werden.		
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben		
	<input checked="" type="checkbox"/> Teilnahme an der Praktischen Phase		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	Prof. Dr. Helmut Bollenbacher, Hochschule Koblenz		
Modulverantwortliche(r)	Prof. Dr. Helmut Bollenbacher, Hochschule Koblenz		
Änderungsdatum	15.06.2026		

<h2 style="margin: 0;">Fortgeschrittene Programmiertechniken (FOPT)</h2>			
Fachgebiet	Praktische Informatik		
Inhalte	<ul style="list-style-type: none"> ▪ Parallele Programmierung 1 + 2: Grundlegende und fortgeschrittene Synchronisationskonzepte in Java ▪ Grafische Benutzeroberflächen 1 + 2: Grundlagen und fortgeschrittene Programmierung grafischer Benutzeroberflächen in Java ▪ Programmierung verteilter Anwendungen 1 + 2: Eigenständige und webbasierte Client-Server-Anwendungen in Java 		
Lernergebnisse	<p>Nach Absolvierung dieses Moduls besitzen die Studierenden vertiefte und erweiterte Programmierkompetenzen in Java vor allem in drei Bereichen: parallele Programmierung mit Java Threads, grafische Benutzeroberflächen mit JavaFX und verteilte Anwendungen mit Sockets, RMI, Servlets und Java Server Faces. Alle drei Bereiche stellen wesentliche und zentrale Lehrinhalte im Rahmen einer Informatik-Ausbildung dar und haben eine hohe Relevanz für die berufliche Praxis.</p> <p>Im ersten Teil über parallele Programmierung werden die Studierenden in die Lage versetzt zu erläutern, welche grundsätzlichen Probleme der Einsatz von Parallelität birgt. Die Studierenden können diese Probleme durch die Anwendung unterschiedliche Synchronisationsprimitive anhand zahlreicher konkreter Beispiele lösen und diese Lösungskompetenz auch auf neue Aufgabenstellungen übertragen. Der zweite Teil über die Programmierung grafischer Benutzeroberflächen befähigt die Studierenden nicht nur, Programme mit einer grafischen Benutzeroberfläche zu entwickeln, sondern vor allem auch, Entwurfs- und Architekturmuster zielgerichtet und Gewinn bringend einzusetzen. Sie werden dadurch vom "Programmieren im Kleinen" zum "Programmieren im etwas Größeren" geführt. Sie können Programme, die aus einer größeren Anzahl an Klassen und Schnittstellen besteht, anhand klarer Prinzipien strukturieren. Durch den dritten Teil des Moduls erlangen die Studierenden die Kompetenz zur Entwicklung verteilter Java-Anwendungen. Sie können sowohl eigenständige Anwendungen programmieren, die über Sockets oder Java RMI kommunizieren, als auch webbasierte Anwendungen, die auf Servlets und Java Server Faces basieren.</p> <p>Die drei Themengebiete sind aber nicht nur isoliert jeweils für sich genommen bedeutsam, sie hängen in vielfältiger Weise miteinander zusammen: So können die Studierenden bei der Entwicklung von Programmen mit grafischer Benutzeroberfläche erläutern, warum der Einsatz von Parallelität notwendig ist, falls von den Benutzerinnen und Benutzern des Programms länger andauernde Aktionen angestoßen werden, und die dabei erlernten Prinzipien in praktischen Beispielen anwenden. Die Studierenden können für verteilte Anwendungen die Client-Seite mit einer grafische Benutzeroberfläche ausstatten und die Server-Seite mit mehreren parallelen Threads programmieren. Außerdem können die Studierenden das Architekturmuster MVP, das sie aus Anwendungen mit einer grafischen Benutzeroberfläche bereits kennen, auf die Strukturierung verteilter Anwendungen übertragen.</p>		
Lehrform	<ul style="list-style-type: none"> ▪ 6 Kurseinheiten mit Übungsaufgaben ▪ Lehrbuch ▪ Studienbegleitendes Tutorium ▪ Praktische Phase (4 Tage zzgl. Prüfungstag) 		
Empfohlene Voraussetzungen	<p>Für das Studium des Moduls sind Kenntnisse in objektorientierter Programmierung mit Java sowie über Datenstrukturen und Algorithmen unabdingbar. Sie sollten mindestens das Modul „Einführung in die Programmierung“ absolviert oder gleichwertige Kenntnisse anderweitig erworben haben.</p> <p>Zur Auffrischung von Grundlagenwissen dient der Vorkurs „Fortgeschrittene Programmiertechniken“. In diesem Vorkurs werden Themenbereiche aufgegriffen, mit denen sich Programmierneulinge erfahrungsgemäß schwertun, und zahlreiche Programmieraufgaben zur Verbesserung der Programmierpraxis zur Verfügung stellt.</p>		
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Rainer Oechsle: Parallele und verteilte Anwendungen in Java. 6., überarbeitete und erweiterte Auflage, 06/2022		
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Programmieraufgaben <input checked="" type="checkbox"/> Teilnahme an der Praktischen Phase		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	Prof. Dr. Rainer Oechsle, Hochschule Trier		
Modulverantwortliche(r)	Prof. Dr. Rainer Oechsle, Hochschule Trier		

Änderungsdatum	15.06.2026
-----------------------	------------

<h1>Informatik in Produktion und Materialwirtschaft (IPM)</h1>		
Fachgebiet	Angewandte Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ Einführung, Grundlagen ▪ Artikel und Materialien ▪ Beschaffung ▪ Stücklisten ▪ Arbeitsplätze und Arbeitspläne ▪ Fertigungsaufträge ▪ Einführung in die Programmierung mit ABAP® ▪ Programmierung von Menüfolgen - DYNPROS® ▪ Objektorientiertes ABAP® 	
Lernergebnisse	<p>In den Bereichen Logistik, Produktion und Materialwirtschaft werden zunehmend Stellen geschaffen, die neben Kenntnissen von Betriebsabläufen und spezifischen Anwendungen auch klassisches Informatikwissen erfordern. Anwender- und Entwicklersicht müssen hier zusammenkommen. Das Modul vermittelt daher Kenntnisse aus beiden Bereichen. Die Studierenden</p> <ul style="list-style-type: none"> ▪ überblicken die Gebiete Materialwirtschaft und Produktion mit ERP-Systemen, ▪ kennen die typischen Aufgaben und Prozesse bei Beschaffung und Produktion, ▪ können SAP® S4/HANA® mit den Modulen MM und PP praktisch anwenden, ▪ können sich in neue Bereiche einarbeiten, ▪ beherrschen die Einschätzung der Möglichkeiten des ERP-Systems durch gezieltes Reengineering ▪ beherrschen die Erweiterung von Funktionalitäten durch spezifische Programmierung, z.B. zur Erstellung von Schnittstellen zu anderen Systemen oder zur Integration eigener betriebspezifischer Prozesse. ▪ beherrschen den Zugriff auf das Datenbankmanagementsystem und den Umgang wie die Abfrage, das Speichern und die Verarbeitung von Daten. 	
Lehrform	<ul style="list-style-type: none"> ▪ eLectures mit Begleittexten und Übungsaufgaben ▪ Regelmäßige Webmeetings für fachliche Fragen ▪ Praktische Übungen (mit Zugang zum SAP-System) ▪ Lehrbuch ▪ Ausarbeitung einer Hausarbeit mit Präsentation 	
Empfohlene Voraussetzungen	Vorkenntnisse aus den Bereichen Datenbanksysteme und Programmierung sind hilfreich, aber nicht zwingend notwendig.	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Hans-Peter Wiendahl: Betriebsorganisation für Ingenieure. Carl Hanser Verlag	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von zwei kleineren Praxisprojekten	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input checked="" type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	130 Stunden
		Selbststudium
		170 Stunden
Herausgeber / Autor	Prof. Dr. F. N. Rudolph, Hochschule Trier	
Modulverantwortliche(r)	Prof. Dr. F. N. Rudolph, Hochschule Trier	
Änderungsdatum	15.06.2026	

<h1>Informatik und Gesellschaft (IUG)</h1>							
Fachgebiet	Angewandte Informatik						
Inhalte	<ul style="list-style-type: none"> ▪ Datenschutz / Überwachung ▪ Digitale Medien ▪ Desinformation und Manipulation ▪ Zukunft der Arbeit ▪ Cyberangriffe ▪ Automatische Entscheidungen ▪ Generative KI ▪ Superintelligenz ▪ Frühwarn- und Entscheidungssysteme ▪ Autonome Waffen ▪ Revolution der Kriegsführung durch KI ▪ Existenzielle Risiken und Maßnahmen zur Risikoreduzierung 						
Lernergebnisse	<p>Die Studierenden haben in diesem Modul mögliche Gefahren und Auswirkungen bestimmter Entwicklungen der Informatik kennengelernt. Für Absolventinnen und Absolventen sind die Auswirkungen von Informatik-Anwendungen aus verschiedenen Gesichtspunkten relevant: Sie können selbst an Entwicklungen aktiv beteiligt sein, müssen eventuell Systeme bewerten, Anwender schulen und beraten oder sind einfach nur selbst Nutzer. Bezogen auf jede dieser möglichen Situationen haben sie ein Bewusstsein für das Abschätzen möglicher Auswirkungen erlangt, um bezogen auf eventuelle negative Folgen verantwortlich handeln zu können.</p> <p>Die Studierenden haben fundierte Kenntnisse von Datenschutz und Überwachung sowie Risiken von digitalen Medien, einschließlich Desinformation und Manipulation erlangt und Möglichkeiten für problematische Schlussfolgerungen aus Überwachungsdaten kennengelernt. Sie haben Erkenntnisse gewonnen zur Lösung von Erkennungsaufgaben in unsicherem Kontext, die Grundlage für automatische Entscheidungen sein können. Sie kennen die Gefahren, die von militärischen computergestützten Frühwarn- und Entscheidungssystemen ausgehen, insbesondere wenn aufgrund immer kleinerer Entscheidungszeiträume immer mehr Techniken der Künstlichen Intelligenz eingesetzt werden müssen. Sie kennen die Gefahren von Cyberangriffen und Cyberkriegskapazitäten sowie von autonomen Waffensystemen, die in der Lage sind, vollautomatisch Entscheidungen zu treffen. Sie kennen mögliche Folgen für die Arbeitswelt durch eine zunehmende Automatisierung sowie den Gefahren, die von intelligenten Systemen auf menschlichem Niveau ausgehen könnten.</p>						
Lehrform	<ul style="list-style-type: none"> ▪ Lehrmaterial ▪ Übungsaufgaben 						
Empfohlene Voraussetzungen	Keine speziellen Vorkenntnisse nötig.						
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Karl Hans Bläsius: Künstliche Intelligenz und Krieg, Springer, 2026, ISBN 978-3-662-72525-2						
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben						
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)						
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) <div style="text-align: right;"><input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF</div>						
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig						
Arbeitsaufwand	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">ECTS-Punkte</th> <th style="width: 25%;">Kontaktzeit</th> <th style="width: 50%;">Selbststudium</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">10</td> <td style="text-align: center;">2 Stunden</td> <td style="text-align: center;">298 Stunden</td> </tr> </tbody> </table>	ECTS-Punkte	Kontaktzeit	Selbststudium	10	2 Stunden	298 Stunden
ECTS-Punkte	Kontaktzeit	Selbststudium					
10	2 Stunden	298 Stunden					
Herausgeber / Autor	Prof. Dr. Karl Hans Bläsius, Hochschule Trier						
Modulverantwortliche(r)	Prof. Dr. Karl Hans Bläsius, Hochschule Trier						
Änderungsdatum	15.06.2026						

<h2 style="margin: 0;">IT-Sicherheit (ITS)</h2>		
Fachgebiet	Angewandte Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ Systemsicherheit ▪ Sicherheitsmanagement ▪ Angewandte Kryptologie ▪ Netzsicherheit ▪ Chipkarten und Anwendungen ▪ Sicherheitsinfrastrukturen 	
Lernergebnisse	<p>Ziel des Moduls ist es, relevante Teilaspekte der IT-Sicherheit (Sicherheit in informationstechnischen Systemen) derart zu behandeln, dass sie praxisrelevant eingesetzt werden können. Zudem werden Grundkenntnisse vermittelt, die dazu dienen, IT-Sicherheit als interdisziplinäre Aufgabe im Kontext sicherheitsrelevanter Anwendungen zu identifizieren.</p> <p>Dazu werden zunächst allgemeine Aspekte sicherheitsrelevanter Systeme betrachtet und deren Bezug zu konkreten Anwendungen aufgezeigt. Neben technischen Aspekten werden insbesondere Fragestellungen im Kontext des Managements informationstechnischer Systeme (Sicherheitsmanagement) betrachtet. Kryptographische Basismechanismen, Netzwerksicherheit, Security Token und insbesondere Chipkarten und relevante Anwendungen und Sicherheitsinfrastrukturen bilden den Kern des Moduls, indem sowohl die technischen Grundlagen als auch die anwendungsrelevanten Bezüge aufgezeigt werden.</p>	
Lehrform	<ul style="list-style-type: none"> ▪ 6 Lehrhefte mit Übungsaufgaben ▪ Praktische Phase (22 UE verteilt auf zwei Abende und zwei Tage zzgl. 3 Fragestunden und Prüfungstag) 	
Empfohlene Voraussetzungen	Theoretische (auch elementare mathematische) und praktische Grundkenntnisse der Informatik. Fähigkeit, komplexe Strukturen zu erkennen (und zu analysieren), identische Sachverhalte in unterschiedlichen Kontexten zu identifizieren und gesamtheitlich zu betrachten. Zudem werden grundlegende Kenntnisse im Bereich der Netzwerkarchitektur und der verwendeten Protokolle (etwa TCP/IP) vorausgesetzt.	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Eckert, Claudia: IT-Sicherheit – Konzepte, Verfahren, Protokolle; 11. Auflage, De Gruyter Studium, 2023. ISBN 978-3-11-055158-7 ▪ Menezes, Alfred J.; VanOorschot, Paul C.; Vanstone, Scott A.: Handbook of Applied Cryptography; 5. print. Boca Raton u.a.: CRC Press, 2001. ISBN 0-8493-8523-7 ▪ Bishop, Matt: Computer Security – Art and Science; 2. ed. Pearson Education, 2018. ISBN 978-0-3217-1233-2 	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input type="checkbox"/> Ausarbeitung eines Fachvortrags <input checked="" type="checkbox"/> Teilnahme an zwei praktischen Phasen	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	50 Stunden
		Selbststudium
		250 Stunden
Herausgeber / Autor	Assoc. Prof. Dr. P. Schartner, em. Univ.-Prof. Dr. P. Horster, Universität Klagenfurt	
Modulverantwortliche(r)	Assoc. Prof. Dr. P. Schartner, Universität Klagenfurt	
Änderungsdatum	15.06.2026	

Kommunikative Kompetenz (KOM)

Fachgebiet	Fachübergreifende Schlüsselqualifikation
Inhalte	<p><i>Inhalte der Lehrhefte</i></p> <ul style="list-style-type: none"> ▪ Zwischenmenschliche Kommunikation: Kommunikative Kompetenz <ul style="list-style-type: none"> ▪ Kommunizieren heute: ein modernes Anforderungsprofil ▪ „Kommunikative Kompetenz“: Begriffsgeschichte ▪ Eine Rahmentheorie Kommunikativer Kompetenz ▪ Anlässe zur Förderung Kommunikativer Kompetenz ▪ Kommunikative Kompetenz durch: fundiertes Wissen, Selbstreflexion, Erweiterung des Handlungsspielraums ▪ Rhetorik und Körpersprache <ul style="list-style-type: none"> ▪ Grundlagen: Rhetorik; Körpersprache – Sprache des Körpers ▪ Drei Säulen der Rhetorik: Ethos; Pathos; Logos ▪ Körpersprache und Rhetorik im beruflichen Alltag: Gespräche führen; Reden halten; Argumentieren; Verhandeln ▪ Kommunikation in Teams <ul style="list-style-type: none"> ▪ Person und Rolle: Rollen in Arbeitsgruppen, Rollentausch, Rollenflexibilität und Rollendistanz ▪ Gruppenstrukturen: Gruppen und Teams; Gruppenbildung; In Großgruppen ins Gespräch kommen ▪ Gruppenprozesse ▪ Gesprächsprozesse in Teams <p><i>Vertiefende Inhalte Praktische Phase</i></p> <ul style="list-style-type: none"> ▪ Präsentationstechniken <ul style="list-style-type: none"> ▪ Grundlagen ▪ Präsentationsvorbereitung ▪ Medienpsychologische Aspekte des Präsentierens ▪ Präsentationsmedien und -technik ▪ Techniken des Visualisierens ▪ Computergestützte Präsentationen ▪ Präsentationsdurchführung ▪ Grundlagen zwischenmenschlicher Kommunikation <ul style="list-style-type: none"> ▪ Vier-Ebenen-Modell / 5-Ohren-Modell ▪ Soziale Bedürfnisse ▪ Soziale Wahrnehmung und Bewertung ▪ Anforderungen an kommunikative Kompetenz ▪ Kommunikation und Kommunikationsformate am Arbeitsplatz <ul style="list-style-type: none"> ▪ Kommunikative Aspekte der Arbeitskultur ▪ Beziehungsebenen professioneller Kommunikation ▪ Kommunikation im Team ▪ Arbeitsbesprechung ▪ Verkaufsgespräch ▪ Führung ▪ Mediale Kommunikationsformate ▪ Konfliktbewältigung
Lernergebnisse	<p>Hohe Fachkompetenz ist keine Garantie für Erfolg im Beruf. Sie brauchen kommunikative Kompetenz, um berufliche Anforderungen zu meistern und die eigene berufliche Attraktivität zu steigern.</p> <p>Der erste Schwerpunkt dieses Moduls liegt darauf, das Wissen der Fernstudierenden über das Thema Kommunikation, v.a. Kommunikation am Arbeitsplatz, zu erweitern und zu vertiefen und praktisch relevante Kenntnisse zu vermitteln. Ein fundiertes Wissen garantiert zwar noch keine erfolgreiche Kommunikation (es erklärt aber u.a., warum dies so ist), aber es schafft eine wesentliche Grundlage dafür.</p> <p>Zum Zweiten zielt das Modul darauf ab, die Reflexionsfähigkeit über eigene kommunikative Erfahrungen zu fördern. Dies soll dazu befähigen, das eigene Kommunikationsverhalten souverän und klug einschätzen und auf dieser Grundlage ebenso handeln zu können. Dabei geht es darum, die persönliche spezielle Situation am eigenen Arbeitsplatz in den Blick zu nehmen.</p> <p>Beide Aspekte sind Gegenstand der praktischen Phasen, die zwischen Impulsen der Wissensvermittlung und Raum zur Erfahrungsreflexion wechseln.</p> <p>Die Teilnehmer</p> <ul style="list-style-type: none"> ▪ kennen grundlegende Prozesse und Methoden der Kommunikation (theoretisches Wissen). ▪ können verbale als auch non-verbale Kommunikation fachspezifisch interpretieren und kontrollieren (Transferleistung, praktisches Wissen). ▪ erwerben Kommunikations- und Führungskompetenz durch die praktische Anwendung der erlernten Techniken zu Monolog und Dialog (praktisches Wissen). ▪ sind in der Lage, überzeugend zu argumentieren und entscheidende Gespräche zu leiten (methodisches Wissen).
Lehrform	<ul style="list-style-type: none"> ▪ Theorieteil mit Übungsaufgaben (3 Lehrhefte) ▪ Praktische Phase (5 Tage verteilt auf zwei Blockseminare zzgl. Abschlussprüfung) ▪ Mündliche Präsentation und Argumentation im Dialog
Empfohlene Voraussetzungen	Keine speziellen Vorkenntnisse nötig.
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen

	<input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Hoyos, Carl Graf; Frey, Dieter (Hrsg.): Arbeits- und Organisationspsychologie, Beltz Psychologie Verlags Union, 1999, ISBN 3621274324 ▪ Hoberg, Gerrit: Vor Gruppen bestehen. Besprechungen? Workshops? Präsentationen, Klett, 1994, ISBN 9-934122-17-5 ▪ Schulz von Thun, Friedemann: Miteinander reden, Bd. 1, Störungen und Erklärungen, Rowohlt, 1981, ISBN 3499174898 ▪ Weisbach, Christian-Rainer: Professionelle Gesprächsführung. München: DTV, 2003, ISBN 978-3-423-50947-3 ▪ Wesselhöft, Martina: Individuelle Effektivität, in Dresler, M. (Hrsg.) Sprache und Kommunikation Mind Akademie 2005, Books on demand, 2006, ISBN: 3-8334-5380-X <p>Zusätzliche, einführende oder begleitende Literatur kann den Literaturempfehlungen in den einzelnen Lehrheften entnommen werden.</p>		
Studienleistung	<input type="checkbox"/> Ausarbeitung eines Fachvortrags in schriftlicher Form als Hausarbeit und als Bildschirmpräsentation		
	<input checked="" type="checkbox"/> Präsentation und Argumentation im Dialog		
	<input checked="" type="checkbox"/> Teilnahme an Praktischer Phase		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF	<input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	<ul style="list-style-type: none"> ▪ Lehrheft „Zwischenmenschliche Kommunikation: Kommunikative Kompetenz“: Prof. Dr. Werner Pfab, Professor i.R. für Theorie und Praxis sozialer Kommunikation an der Hochschule Fulda ▪ Lehrheft „Rhetorik und Körpersprache“: Dr. phil. Hubert Habig, Regisseur und Autor; Prof. Dr. Rüdiger Nagel, Seniorprofessor im Fachbereich Wirtschaft der Hochschule Mainz ▪ Lehrheft „Kommunikation in Teams“: Prof. Dr. Bernd Schwandt, Professor für Praktische Kommunikation an der FH Erfurt 		
Modulverantwortliche(r)	Dr. rer. nat. Martina Wesselhöft Diplom-Biologin, Diplom-Wirtschaftswissenschaftlerin, Psychologin (M.Sc.) Kommunikationstrainerin		
Änderungsdatum	15.06.2026		

<h1 style="margin: 0;">Projektmanagement (PROM)</h1>		
Fachgebiet	Praktische Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ 1. Projekte als Problemlösungsprozesse ▪ 2. Gründung, Organisation und Strukturierung von Projekten ▪ 3. Projektplanung ▪ 4. Projektsteuerung ▪ 5. Management von Software-Projekten ▪ 6. Werkzeuge im Projektmanagement ▪ Praxisprojekt 	
Lernergebnisse	<p>Die Teilnehmer werden dazu in die Lage versetzt,</p> <ul style="list-style-type: none"> ▪ Vorhaben mit Projektcharakter zu erkennen, zu klassifizieren und zu analysieren [1.1], ▪ die einem Projekt zugrundeliegenden Probleme zu identifizieren und zu untersuchen, Lösungsideen zu produzieren und Lösungswege zu gestalten [1.2], ▪ die vollständigen Anforderungen an ein Projekt sowie dessen Inhalt und Umfang zu erfassen und zu dokumentieren [2.1], ▪ den personellen Aufbau, den grundsätzlichen Ablauf der Teilprojekte und den Austausch der Informationen in einem Projekt zu organisieren [2.2], ▪ das angestrebte Projektergebnis und die dazu notwendigen Arbeiten zu strukturieren [2.3], ▪ den erforderlichen Aufwand zur Durchführung eines Projektes methodisch zu schätzen [3.1], ▪ den Projektablauf unter Berücksichtigung der Abhängigkeiten, der Ressourcenbegrenzungen, der Qualitätsziele und der Risiken detailliert zu planen [3.2] ▪ den Projektfortschritt systematisch zu erfassen und zu analysieren [4.1], ▪ auf Abweichungen des tatsächlichen vom geplanten Fortschritt mit passenden steuernden Eingriffen zu reagieren [4.2], ▪ den Nutzen verschiedener Vorgehensmodelle in konkreten Software-Projekten zu beurteilen und das passende Modell auszuwählen [5.1], ▪ agile Vorgehensmodelle in einem Software-Projekt einzusetzen [5.2], ▪ den Reifegrad einer Organisation für verschiedene Vorgehensmodelle zu ermitteln [5.3], ▪ alle erlernten Arbeitsschritte, die zur Planung und Steuerung von Projekten benötigt werden, werden anhand eines selbst gewählten Praxisprojekts vertieft [6.1]. 	
Lehrform	<ul style="list-style-type: none"> ▪ 6 Kurseinheiten mit Übungsaufgaben ▪ Lehrbuch ▪ Ausarbeitung eines Praxisprojekts ▪ Projektpräsentation 	
Empfohlene Voraussetzungen	Keine speziellen Vorkenntnisse nötig.	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Jakoby, Walter: Projektmanagement für Ingenieure: 6. Auflage, 2025, Verlag: Springer Vieweg; ISBN 978-3-658-47468-3 ▪ Jakoby, Walter: Intensivtraining Projektmanagement; Übungsbuch; 4. Auflage, 2025, Verlag: Springer Vieweg; ISBN 978-3-658-47300-6 	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input checked="" type="checkbox"/> Erarbeitung eines Praxisprojekts	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	30 Stunden
		Selbststudium
		270 Stunden
Herausgeber / Autor	Prof. Dr. Walter Jakoby, Hochschule Trier	
Modulverantwortliche(r)	Prof. Dr. Walter Jakoby, Hochschule Trier	
Änderungsdatum	15.06.2026	

Projektarbeit			
Inhalte	Das Projektstudium umfasst die selbstständige Bearbeitung einer anspruchsvollen praxisorientierten Aufgabenstellung unter wissenschaftlicher Betreuung. Die Komplexität der Aufgabenstellung orientiert sich an den Anforderungen der beruflichen Praxis von Masterabsolventinnen und Masterabsolventen. Die Studierenden wenden dabei methodische Vorgehensweisen und geeignete Arbeitstechniken an und übertragen informatische Konzepte auf konkrete Anwendungsgebiete. Die Ergebnisse werden dokumentiert, reflektiert und in Form einer wissenschaftlichen Ausarbeitung unter Beachtung der Grundsätze wissenschaftlichen Arbeitens präsentiert.		
Lernergebnisse	Die Studierenden sind in der Lage, sich innerhalb einer vorgegebenen Frist selbstständig in ein Anwendungsgebiet der Informatik einzuarbeiten und ein praktisches Problem zu analysieren und zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen sowie ein kritisches Verständnis zentraler Theorien, Methoden und Werkzeuge der Informatik. Sie wenden die im Studium erworbenen Kenntnisse, Fähigkeiten und Methoden auf konkrete Fragestellungen an und entwickeln darauf basierend fundierte und dem Stand der Technik entsprechende Lösungsansätze. Darüber hinaus beherrschen sie grundlegende Methoden der Projektplanung, -steuerung und -kontrolle und können komplexe Aufgabenstellungen strukturieren und bearbeiten. Sie sind in der Lage, technische und organisatorische Schnittstellenprobleme sowie Herausforderungen der Zusammenarbeit in Teams angemessen zu bewältigen. Die Studierenden dokumentieren ihre Arbeitsergebnisse unter Anwendung der Grundsätze wissenschaftlichen Arbeitens und präsentieren diese in angemessener Form.		
Lehrform	Projekt		
Empfohlene Voraussetzungen	Abhängig von der Aufgabenstellung; wird vom Betreuer festgelegt		
Vorbereitung	<input type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	20 Stunden	280 Stunden
Modulverantwortliche(r)	Studiengangsleitung des Master-Fernstudiengang Informatik (Aufbaustudium)		
Änderungsdatum	15.06.2026		

<h1>Rechnernetze (RN)</h1>		
Fachgebiet	Praktische Informatik	
Inhalte	<ul style="list-style-type: none"> ▪ TCP/IP: Transport und Vermittlung im Internet ▪ Lokale Netze ▪ Fest- und Mobilfunknetze im Weitverkehrsbereich ▪ Internet-Anwendungen I: Verschiedene Anwendungen ▪ Internet-Anwendungen II: Webtechnologien ▪ Internet-Anwendungen III: Peer-To-Peer-Systeme und Multimedia-Anwendungen 	
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ Schichtenmodelle anhand der im Internet eingesetzten TCP/IP-Protokollhierarchie erläutern, ▪ das Vermittlungsprotokoll IP und die dabei benutzten Adressen verstehen sowie die Planung von lokalen Netzen mit IP-Netzadressierung durchführen, ▪ die Probleme und die Lösungsansätze für die Realisierung von IP-Verkehr über unterschiedliche Arten von Netzwerken benennen und bewerten, ▪ die Arbeitsweise und Eigenschaften von Transportprotokollen verstehen und erklären, ▪ Konzepte zum Aufbau lokaler Netze beschreiben und beurteilen, die im Bereich der Ethernet-Netze etablierten Standards und Technologien benennen und einordnen sowie die Problemstellungen und Lösungen für Ethernet-basierte Hochgeschwindigkeitsnetze überblicken, ▪ wichtige Vermittlungstechniken, Weitverkehrsfestnetze und Mobilfunknetze beschreiben, ▪ exemplarische Netzunterstützungs- und Managementdienste wie DNS und LDAP, SNMP einsetzen und in Ihrer Bedeutung für die Netzkommunikation bewerten, ▪ Wegewahlprotokolle klassifizieren und die wichtigsten in der Praxis eingesetzten Protokolle verstehen, ▪ Konzepte der „klassischen“ Anwendungen wie elektronische Post, Fernbenutzung von Rechnern, Dateitransfer und verteilte Dateisysteme erläutern, ▪ Ideen und Prinzipien des Cloud Computing einschätzen und bewerten, ▪ für mögliche Bedrohungen aus dem Internet sensibel sein und die Schutzmöglichkeiten, die Firewalls, Verschlüsselungstechniken und digitale Unterschriften bieten, überblicken und beurteilen, ▪ die Funktionsprinzipien des WWW mit dem Einsatz von Hypertexten verstehen, die Bedeutung des HTTP-Protokolls und des Adressierungsschemas URI beschreiben und verstehen, Konzepte für die Gestaltung von Webseiten mit XHTML, CSS und JavaScript realisieren, Webseiten dynamisch durch die Erstellung von serverseitiger Software erzeugen, ▪ Peer-to-Peer-Systeme gegenüber Client-Server-Systemen abgrenzen und die wesentlichen Merkmale von Peer-to-Peer-Systemen erläutern und auf konkrete Beispiele anwenden, ▪ die Problematik der Multimedia-Kommunikation im Internet darstellen und ihre wesentlichen Bestandteile erklären. 	
Lehrform	<ul style="list-style-type: none"> ▪ 6 Lehrhefte mit Übungsaufgaben ▪ Praktische Phase (4 Tage zzgl. Prüfungstag) 	
Empfohlene Voraussetzungen	Keine speziellen Vorkenntnisse nötig.	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: <ul style="list-style-type: none"> ▪ Kurose, James F.; Ross, Keith W.: Computernetzwerke – der Top-Down-Ansatz; 8. aktualis. Auflage; Pearson Studium, 2018; ISBN 3-86894-237-8 ▪ Tanenbaum, Andrew S.; Wetherall, David J.: Computernetzwerke; 5. aktualis. Auflage. Pearson Studium, 2012; ISBN 3-86894-137-1 	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input checked="" type="checkbox"/> Teilnahme an der praktischen Phase	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	50 Stunden
		Selbststudium
		250 Stunden
Herausgeber / Autor	Prof. Dr. Klaus Lang, Technische Hochschule Bingen	
Modulverantwortliche(r)	Prof. Dr. Klaus Lang, Technische Hochschule Bingen	

Änderungsdatum	15.06.2026
-----------------------	------------

<h2>Rust in Aktion: Spieleprogrammierung und Parallelität (RUST)</h2>			
Fachgebiet	Praktische Informatik		
Inhalte	<ul style="list-style-type: none"> ▪ Grundlagen der Rust-Programmierung: Cargo, Datentypen, Kontrollstrukturen, Funktionen ▪ Ownership, Borrowing, Lifetimes und idiomatic Rust-Programmierung (Camouflage Programming) ▪ Datenstrukturen, Enums, Pattern Matching, Generics, Traits, Option und Result ▪ Funktionale Programmierung: Closures, Iteratoren und itertools ▪ Grafikprogrammierung mit Macroquad: Sprites, Sound, Input, Camera2D, egui ▪ Smart Pointer (Box, Rc, RefCell) und Trait Objects ▪ Parallele Programmierung: Threading, Channels, Arc/Mutex, Rayon, Gather/Modify-Pattern ▪ Spieleprogrammierung: Game Loop, Kollisionserkennung, A*-Pathfinding, Asteroids-Projekt ▪ Optional: Tokio/Axum Webserver, Testen/Dokumentieren, Cargo Deepdive, unsafe/FFI/Macros 		
Lernergebnisse	<p>Dieses Modul versetzt Studierende in die Lage, sichere und performante Anwendungen in der Systemprogrammiersprache Rust zu entwickeln. Sie verstehen das Ownership- und Borrowing-Modell von Rust und können es gezielt einsetzen, um Speicherfehler und Data Races bereits zur Kompilierzeit auszuschließen. Studierende beherrschen fortgeschrittene Sprachkonzepte wie Generics, Traits, Closures und Iteratoren und können diese idiomatisch anwenden. Sie sind in der Lage, ein eigenständiges Spielprojekt (Asteroids-Klon) mit der Macroquad-Bibliothek zu entwickeln, einschließlich Game Loop, Kollisionserkennung und A*-Pathfinding und können rechenintensive Spiellogik mit Threads und der Rayon-Bibliothek parallelisieren sowie dabei das Gather/Modify-Pattern anwenden. Am Ende des Moduls präsentieren die Studierenden einen spielbaren Prototypen („First Playable“) und verfügen damit über fundierte Kenntnisse in moderner Systemprogrammierung sowie eine vorführbare Demo für ihr Portfolio.</p>		
Lehrform	<ul style="list-style-type: none"> ▪ Screencasts mit begleitendem Source Code ▪ Projektarbeit mit zwei Meilensteinen (Asteroids-Spielprojekt) ▪ Beratungstermine (Einzel- oder Sammeltermine) ▪ Meilensteinpräsentationen im Plenum 		
Empfohlene Voraussetzungen	<p>Grundkenntnisse in einer Programmiersprache (idealerweise C, C++ oder Java) werden empfohlen. Der Kurs richtet sich an Studierende des Aufbaustudiums, die nicht zwingend einen informatischen Erststudiengang absolviert haben müssen. Für Studierende ohne C/C++-Hintergrund ist ein optionaler Exkurs zum Speichermodell (Stack, Heap, Cache) im Kurs enthalten. Ebenso werden optionale Screencasts zu grundlegenden Konzepten der Spieleprogrammierung angeboten. Voraussetzung sind hier Mathematikkenntnisse insbesondere lineare Algebra auf Abiturniveau.</p>		
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Klabnik, Steve; Nichols, Carol: The Rust Programming Language, 3rd edition, 2025, NoStarch Press. Steenter, Alen: Denken in Rust, 2025. Gjengset, John: Rust for Rustaceans, 2022, NoStarch Press.		
Studienleistung	<input type="checkbox"/> Bearbeitung von Einsendeaufgaben <input type="checkbox"/> Teilnahme an der praktischen Phase		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	10	50 Stunden	250 Stunden
Herausgeber / Autor	Prof. Dr. Christoph Lürig, Hochschule Trier		
Modulverantwortliche(r)	Prof. Dr. Christoph Lürig, Hochschule Trier		
Änderungsdatum	15.06.2026		

<h1>Software Engineering (SE)</h1>		
Fachgebiet	Praktische Informatik	
Inhalte	Lehrinhalte sind: <ul style="list-style-type: none"> ▪ Grundlagen der Softwaretechnik und des Requirements Engineering ▪ Objektorientierte Analyse und Entwurf ▪ Systemmodellierung ▪ Qualitätssicherung ▪ Projektmanagement In einer praktischen Phase werden die erworbenen Kompetenzen anhand verschiedener, themenbezogener Aufgabenstellungen vertieft.	
Lernergebnisse	Die Studierenden <ul style="list-style-type: none"> • können den kompletten Entwicklungszyklus einer Softwareanwendung vom Requirements Engineering bis zum Betrieb erläutern und die Aufgaben der einzelnen Schritte beschreiben; • können theoretische Konzepte und Begriffe aus dem Software Engineering erläutern, auf konkrete Fälle anwenden und zur sachgerechten Beschreibung und Analyse der Fälle nutzen; • können methodische Vorgehensweise zur Erhebung und Dokumentation von Anforderungen, zur objektorientierten Analyse, zu Entwurf und Modellierung von objektorientierter Software und Software-Architekturen sowie zur Qualitätssicherung von Software benennen und beschreiben; • können in einem gegebenen, realitätsnahen und komplexen Szenario passende Vorgehensweisen für das Requirements Engineering, die Analyse, den Entwurf und die Qualitätssicherung auswählen und anwenden; • können Ergebnisse aus dem Entwicklungszyklus sachgerecht dokumentieren, präsentieren, diskutieren und bzgl. ihrer Vollständigkeit und Qualität beurteilen. 	
Lehrform	<ul style="list-style-type: none"> ▪ 5 Lehrhefte mit Übungsaufgaben ▪ Praktische Phase (4 Tage zzgl. Prüfungstag) 	
Empfohlene Voraussetzungen	Kenntnisse und Erfahrung in objektorientierter Programmierung – vorzugsweise in Java.	
Vorbereitung	<input checked="" type="checkbox"/> Angaben in den Vorlesungsunterlagen <input checked="" type="checkbox"/> Literatur: Balzert, Heide: Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2; Nachdruck der 2. Auflage von 2004; Spektrum Akademischer Verlag; ISBN 978-3-8274-2903-2	
Studienleistung	<input checked="" type="checkbox"/> Bearbeitung von Einsendeaufgaben <input checked="" type="checkbox"/> Teilnahme an der praktischen Phase	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	<ul style="list-style-type: none"> • Master-Fernstudiengang Informatik (Aufbaustudium) • Zertifikatsfernstudiengang Informatik (Weiterbildung) 	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	10	50 Stunden
		Selbststudium
		250 Stunden
Herausgeber / Autor	Prof. Dr. Axel Buhl, Prof. Dr. Gero Wedemann, Hochschule Stralsund	
Modulverantwortliche(r)	Prof. Dr. Michael Striewe, Hochschule Trier	
Änderungsdatum	15.06.2026	