

Modul: Programmieren II

Niveau	Bachelor	Stundenplankürzel	Prog II
Modulname englisch	Programming II		
Modulverantwortliche	Nane Kratzke, Andreas Schäfer		
Fachbereich	Elektrotechnik und Informatik		
Studiengang	Informatik/Softwaretechnik, Bachelor		
Verpflichtungsgrad	Pflicht	ECTS-Leistungspunkte	6
Fachsemester	2	Semesterwochenstunden	5
Dauer in Semestern	1	Arbeitsaufwand in Stunden	180
Angebotshäufigkeit	SoSe	Präsenzstunden	75
Lehrsprache	Deutsch	Selbststudiumsstunden	105

Der folgende Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

Prüfungsleistung		Prüfsprache	
Dauer PL in Minuten		Bewertungssystem PL	
Lernergebnisse			
Teilnahmevoraussetzungen			

Der vorige Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

Berücksichtigung von Gender- und Diversity-Aspekten	<ul style="list-style-type: none"> ✘ Verwendung geschlechtergerechter Sprache (THL-Standard) ✘ Zielgruppengerechte Anpassung der didaktischen Methoden ✘ Sichtbarmachen von Vielfalt im Fach (Forscherinnen, Kulturen etc.)
Verwendbarkeit	
Bemerkungen	Die Berücksichtigung von Gender- und Diversity-Aspekten erfolgt nicht nach formalen Vorgaben, sondern wird im Rahmen des allgemein üblichen und gesellschaftlich akzeptierten Miteinanders praktiziert.

Lehrveranstaltung: Programmieren II (Vorlesung)

(zu Modul: Programmieren II)

Lehrveranstaltungsart	Vorlesung	Lernform	Präsenz
LV-Name englisch	Programming II		
Anwesenheitspflicht	nein	ECTS-Leistungspunkte	3
Teilnahmebeschränkung		Semesterwochenstunden	3
Gruppengröße	200	Arbeitsaufwand in Stunden	90
Lehrsprache	Deutsch	Präsenzstunden	45
Studienleistung		Selbststudiumsstunden	45
Dauer SL in Minuten		Bewertungssystem SL	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Prüfungsleistung	Klausur	Prüfsprache	Deutsch
Dauer PL in Minuten	120	Bewertungssystem PL	Drittelnoten

Lernergebnisse	<p>Die Studierenden können objektorientierte Konzepte wie Klassen, Schnittstellen, abstrakte Klassen problembezogen und zielgerichtet in Java einsetzen.</p> <p>Die Studierenden können aus gegebenen UML-Klassendiagrammen sicher Java-Quelltext ableiten und umgekehrt.</p> <p>Die Studierenden berücksichtigen bei der Entwicklung einfacher Software in Java objektorientierte Entwurfsprinzipien und Architekturpattern wie MVC.</p> <p>Die Studierenden können problembezogen und zielgerichtet generische Datentypen und eigenständig generische Methoden in Java definieren, nutzen und einsetzen.</p> <p>Die Studierenden können in Java die Konzepte Unit Testing und Code Coverage einsetzen, um Testfälle zu entwickeln, die eine vorgegebene Codeabdeckung erreichen.</p> <p>Die Studierenden sind sich des Leser/Schreiber-Problems und des Erzeuger/Verbraucher-Problems im concurrent Programming bewusst und kennen Java-Sprachmittel wie synchronized (Monitor) und wait()/notify() (Thread Kommunikation) um diese zu lösen.</p> <p>Die Studierenden haben am Bsp. der Programmiersprache C erfahren wie sich generelle Konzepte imperativer/prozeduraler Programmiersprachen auf andere Programmiersprachen übertragen lassen (hier: Transfer von JAVA auf C).</p>
-----------------------	--

Teilnahmevoraussetzungen	Es empfiehlt sich das Modul Programmieren I besucht und bestanden zu haben.
---------------------------------	---

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine Lehrveranstaltungsspezifische Prüfung gibt.

Lehrinhalte	<p>Vorlesung JAVA:</p> <ul style="list-style-type: none"> • Objektorientiertes Programmieren • Polymorphie und Verträge • Code Coverage und Unit Testing • Generische Datentypen und eigenständig generische Methoden • Objektorientierter Entwurf und Entwurfsprinzipien • Architekturpattern am Bsp. Model View Controller und grafische Oberflächen • Concurrent Programming <p>Vorlesung C:</p> <ul style="list-style-type: none"> • Struktur eines C-Programms • Variablen, Datentypen, Abarbeitung von Ausdrücken • Anweisungen und Kontrollstrukturen • Einsatz von Unterprogrammen • Ein- und Ausgabe von Daten • Dateiverarbeitung (Datenströme, Text- und Binärdateien, sequenzieller und wahlfreier Dateizugriff, Standardfunktionen für den Dateizugriff) • Programmierung mit Zeigern (Zeigertypisierung, Zeigerarithmetik, dynamische Datenelemente, Heap-Verwaltung, dynamische rekursive Datenstrukturen, verkettete Listen) • Modulare Programmierung (Präprozessor, bedingte Compilierung, Macros, Include-Dateien, Projekte in C)
Literatur	<p>Dietmar Ratz, Jens Scheffler, Detlef Seese, Jan Wiesenberger, „Grundkurs Programmieren in JAVA“, Hanser Verlag</p> <p>Cornelia Heinisch, Frank Müller-Hofmann, Joachim Goll, „JAVA als erste Programmiersprache“, Vieweg-Teubner</p> <p>Kernighan/Ritchie, „Programmieren in C“, Hanser Verlag, München</p>
Bemerkungen	Die Prüfungsleistung der Vorlesung deckt den Java-Anteil des Moduls ab.

Lehrveranstaltung: Programmieren II (Praktikum)

(zu Modul: Programmieren II)

Lehrveranstaltungsart	Praktikum	Lernform	Präsenz
LV-Name englisch	Programming II (Practical course)		
Anwesenheitspflicht	ja	ECTS-Leistungspunkte	3
Teilnahmebeschränkung		Semesterwochenstunden	2
Gruppengröße	18	Arbeitsaufwand in Stunden	90
Lehrsprache	Deutsch	Präsenzstunden	30
Studienleistung	(Flexibel)	Selbststudiumsstunden	60
Dauer SL in Minuten		Bewertungssystem SL	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Prüfungsleistung	Projektarbeit	Prüfsprache	Deutsch
Dauer PL in Minuten		Bewertungssystem PL	Drittelnoten
Lernergebnisse	Siehe Modulebene		
Teilnahmevoraussetzungen	Siehe Modulebene		

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Lehrinhalte	Siehe Vorlesung		
Literatur	Siehe Vorlesung		
Bemerkungen	Die Prüfungsleistung des Praktikums deckt den C-Anteil des Moduls ab.		