

Modultitel	<i>Basismodul Grundlagen der Informatik II</i>					
<i>Pflichtmodul</i>	Arbeitsaufwand		Leistungs- punkte	Studiensemester (empfohlen)	Häufigkeit des Angebots	Dauer (empfohlen)
	<i>Kontakt- zeiten</i> 60 h	<i>Selbst- studium</i> 120 h				
	<i>Summe 180 h</i>		6	1. Semester	jedes Sommer- semester	1 Semester
Arbeitsaufwand/ Leistungspunkte	Lehrveranstaltungen		Kontaktzeiten	Selbststudium		
	<i>Vorlesung Grundlagen der Programmierung II</i>		30 h	60 h		
	<i>Übung zu Grundlagen der Programmierung II</i>		30 h	60 h		
	---		---	---		

<p>Qualifikations- ziele / Kompetenzen</p>	<p><u>1.) Fachkompetenzen:</u> Die Studierenden</p> <ul style="list-style-type: none"> - kennen die Merkmale der drei Programmierstile und können den Zweck unterschiedlicher Programmierstile erklären. - können angeben, wie Datentypen und Datenstrukturen im Von-Neumann-Rechnermodell implementiert werden. - können die Zeit- und Platzkomplexität eines Algorithmus/Programms definieren, die verschiedenen Abstraktionsschritte hin zum O-Kalkül erläutern und mit dem O-Kalkül rechnen. - können effiziente Algorithmen zum Multiplizieren und Matrixmultiplizieren beschreiben, ihre Laufzeit abschätzen und mit den geläufigen Schulalgorithmen vergleichen. - kennen verschiedene Entwurfparadigmen, nennen Beispiele für Algorithmen, die die Entwurfparadigmen verwenden, und skizzieren die Laufzeiten von Algorithmen, die die Entwurfparadigmen verwenden. - beschreiben die Arbeitsweise und Effizienz von Standard-Algorithmen auf Folgen, Bäumen, Graphen und Punktmengen, u.a. zum Suchen und Sortieren auf Folgen, zum Durchlaufen, zum Suchen, Einfügen, Löschen auf allgemeinen und ausgeglichenen Suchbäumen, Suchen kürzester Wege und minimaler Spannbäume auf Graphen, Suchen kürzester Abstände und Bilden konvexer Hüllen auf Punktmengen. - können Effizienzmaße auf Parallelrechnersystemen erläutern und zwei effiziente parallele Algorithmen beschreiben. <p><u>2.) Methodenkompetenzen</u> Die Studierenden</p> <ul style="list-style-type: none"> - können die Qualität von Algorithmen anhand ihrer Effizienz beurteilen und verschiedene Algorithmen miteinander vergleichen. - können für typische Probleme abhängig von gegebenen Rahmenbedingungen geeignete effiziente Algorithmen vorschlagen und ihren Vorschlag begründen. - sind in der Lage, vorgestellte Algorithmen mit einer Programmiersprache ihrer Wahl zu implementieren. - erwerben ein Gefühl für die Komplexität eines Problems und die Zugänglichkeit für eine akzeptable praktische Lösung. <p><u>3.) Handlungskompetenzen (gesellschaftsrelevante und strategische Kompetenzen)</u> Die Studierenden</p> <ul style="list-style-type: none"> - können bearbeitete Aufgaben präsentieren und gegen kritische Einwände verteidigen. - trauen sich, auch erfolglose Lösungsansätze zu präsentieren, und sind in der Lage, präzise zu beschreiben, an welcher Stelle welche scheinbar unüberwindbaren Hindernisse bestehen. - bedienen sich zur Lösung von Problemen ausgewählter Literatur in verständiger Weise. - können zu Themen der Vorlesung präzise Fragen stellen.
--	--

Inhalte	<ul style="list-style-type: none"> • <i>Programmierstile</i> <ul style="list-style-type: none"> ○ <i>Klassifikation von Programmiersprachen (imperativ/funktional/prädikativ)</i> • <i>Abstrakte Datentypen</i> • <i>Implementierung von Datentypen</i> • <i>Qualität von Programmen</i> <ul style="list-style-type: none"> ○ <i>Korrektheit und Komplexität</i> • <i>Algorithmen auf Zahlen</i> <ul style="list-style-type: none"> ○ <i>Multiplizieren, Matrizen multiplizieren</i> • <i>Entwurfsparadigmen für Algorithmen</i> <ul style="list-style-type: none"> ○ <i>Divide-and-Conquer</i> ○ <i>Backtracking</i> ○ <i>Greedy-Methode</i> • <i>Algorithmen auf Folgen</i> <ul style="list-style-type: none"> ○ <i>Durchlaufen, Einfügen, Entfernen, Verknüpfen, Spiegeln, Suchen von Elementen und Teilfolgen, Sortieren</i> • <i>Algorithmen auf Bäumen</i> <ul style="list-style-type: none"> ○ <i>Durchlaufen, Einfügen, Entfernen, Suchen von Elementen, Vergleichen, Optimieren</i> • <i>Algorithmen auf Graphen</i> <ul style="list-style-type: none"> ○ <i>Durchlaufen, Suchen von best. Teilstrukturen (Wege, Spannbäume)</i> • <i>Algorithmen auf Punktmengen</i> <ul style="list-style-type: none"> ○ <i>Suchen, Ermitteln ausgewählter Informationen (Distanzen, Clusterbildung)</i> • <i>Parallele Algorithmen</i> <ul style="list-style-type: none"> ○ <i>u.a. Sortieren</i>
Schlüsselkompetenzen	<p><i>Techniken zur Literaturrecherche, Selbständige Erschließung wissenschaftlicher Literatur, Wissenschaftliche Denk- und Arbeitsweise, Anwendung mathematischer Methoden, Umgang mit Programmiersprachen, Präsentation wissenschaftlicher Sachverhalte</i></p> <p><i>im Umfang von 30 h</i></p>
Teilnahmevoraussetzungen	<i>keine</i>
Prüfungsleistungen	<i>Klausur (120-180 min)</i>
Leistungspunkte und Notenvergabe	<i>Klausurnote (100%)</i>
Verwendung des Moduls (in anderen Studiengängen)	

Modul- beauftragte/r	<i>Lehrstuhl für Didaktik der Informatik</i>
Bemerkungen	<i>keine</i>
Termin Modulprüfung	<i>in der Regel Anfang September</i>
2. Termin Modulprüfung	<i>in der Regel Ende September</i>
Termin Praktikum / Exkursion	---