

Modultitel	Basismodul Grundlagen der Informatik I					
Pflichtmodul	Arbeitsaufwand		Leistungs- punkte	Studiensemester (empfohlen)	Häufigkeit des Angebots	Dauer (empfohlen)
	Kontakt- zeiten 60 h	Selbst- studium 120 h	6	1. Semester	jedes Wintersemester	1 Semester
	Summe 180 h					
Arbeitsaufwand/ Leistungspunkte	Lehrveranstaltungen		Kontaktzeiten	Selbststudium		
	Vorlesung Grundlagen der Programmierung I		30 h	60 h		
	Übung zu Grundlagen der Programmierung I		30 h	60 h		
	---		---	---		
Qualifikationsziele / Kompetenzen	<p><u>1.) Fachkompetenzen:</u> Die Studierenden</p> <ul style="list-style-type: none"> - können den Gegenstand der Wissenschaft Informatik beschreiben, die Entwicklung Informatik skizzieren und Teilgebiete der Informatik angeben. - können den Begriff des Algorithmus definieren, Merkmale von Algorithmen nennen, wesentliche Schritte der Algorithmenentwicklung beschreiben, den sprachlichen Aufbau von Algorithmen beschreiben und einfache Algorithmen in einer halbformalen Notation erstellen. - wissen um die Grenzen der Algorithmenisierung, können den Begriff Berechenbarkeit definieren, nicht-berechenbare Funktionen angeben, einfache Funktionen mithilfe von Diagonalisierung auf Berechenbarkeit überprüfen und die Aussage der Churchschen These erläutern. - können Algorithmen und Programme unterscheiden, einfache Algorithmen in Programme funktionaler und imperativer Notation umsetzen und dabei die Präzisierungsansätze erläutern. - können den Begriff des Modells und den Stellenwert der Modellbildung im informatischen Entwicklungsprozess erläutern, ferner unterschiedliche Modelltypen angeben. - können den Zweck einer Spezifikation erläutern und funktionale Spezifikationen zu einfachen Problemen angeben. - können den Begriff des Datentyps definieren, den Zweck von Datentypen im informatischen Modellbildungsprozess erläutern, elementare Datentypen mit zugrunde liegenden Mengen und Operationen angeben, Datentypkonstruktoren mit ihren mathematischen Konzepten beschreiben und wichtige Datenstrukturen in Programmiersprachen (z.B. Sequenz, Baum, File) definieren. - kennen die Grundprinzipien funktionaler Programmierung, darunter die Begriffe Rechenvorschrift, Funktional, Currying, Rekursion, Polymorphie und können kleinere funktionale Programme schreiben.. - können den Zusammenhang von Syntax und Semantik bei Programmiersprachen erklären, Sprachen mithilfe von Syntaxdiagrammen oder Grammatiken in Backus-Naur-Form definieren und das Prinzip von denotationalen Semantikdefinitionen erläutern. <p><u>2.) Methodenkompetenzen</u> Die Studierenden</p> <ul style="list-style-type: none"> - können zu vorgegebenen einfachen Problemstellungen alle Phasen der Programmentwicklung durchführen, also Spezifikation, Algorithmenentwicklung, Präzisierung von Daten und Abläufen, Programmierung in imperativer und funktionaler Darstellung. - können einfache funktionale Programme hinsichtlich ihrer Korrektheit mit formalen oder halbformalen Methoden beurteilen. - können vorgegebene Programme oder Algorithmen im Hinblick auf die berechnete Funktion analysieren. - verinnerlichen das Wesen informatischer Modellbildung und den fortwährenden Zwang zur Präzisierung aller Phasen und der jeweiligen Zwischenergebnisse. 					

	<p><u>3.) Handlungskompetenzen (gesellschaftsrelevante und strategische Kompetenzen)</u> <i>Die Studierenden</i></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"> • Einführung in die Informatik <ul style="list-style-type: none"> ○ Begriffsdefinition, Teilgebiete, Geschichte, Einordnung in den Wissenschaftskanon • Vom Problem zum Algorithmus <ul style="list-style-type: none"> ○ Naive Entwicklung eines Algorithmus, Merkmale von Algorithmen, Sprachmittel für die Darstellung von Algorithmen, Konstruktoren • Grenzen der Algorithmisierung <ul style="list-style-type: none"> ○ Abzählbarkeit, nicht-berechenbare Funktionen, Selbstanwendungs-/Halteproblem, Churchsche These • Vom Algorithmus zum Programm <ul style="list-style-type: none"> ○ Zwang zur Präzisierung von Algorithmen, Präzisierung von Daten und Anweisungen, Entwicklung einer (imperativen) Programmiersprache • Vom Programm zum Computer <ul style="list-style-type: none"> ○ Von-Neumann-Rechner, Maschinen- und Assemblersprachen, Ebenenmodell der Rechnerarchitektur, Übersetzer, Interpreter • Fundamentale Ideen der Informatik <ul style="list-style-type: none"> ○ Teilgebietsübergreifende typische Denkweisen und Methoden der Informatik • Informatische Modellbildung <ul style="list-style-type: none"> ○ Begriff des Modells, Modelltypen, Informatik als Wissenschaft ausführbarer Modelle • Funktionale Spezifikation <ul style="list-style-type: none"> ○ Präzisierung von Problemen • Präzisierung von Daten <ul style="list-style-type: none"> ○ elementare Datentypen, Konstruktoren, wichtige Datentypen (File, Baum) • Funktionale Programmierung <ul style="list-style-type: none"> ○ Funktionen als Objekte, Funktionale, Currying, Rechenvorschriften, Sprachelemente für Funktionen, Substitutionsregeln, Rekursion, Polymorphie • Grundlagen der Programmiersprachen <ul style="list-style-type: none"> ○ Syntaxdefinitionen mittels Syntaxdiagrammen, Backus-Naur-Form, Semantikdefinitionen mittels denotationaler Ansätze, Semantik rekursiver Funktionen, Fixpunkte
Schlüsselkompetenzen	<p>Techniken zur Literaturrecherche, Selbständige Erschließung wissenschaftlicher Literatur, Wissenschaftliche Denk- und Arbeitsweise, Anwendung mathematischer Methoden, Umgang mit Programmiersprachen, Präsentation wissenschaftlicher Sachverhalte im Umfang von 30 h</p>
Teilnahmevoraussetzungen	keine

Prüfungsleistungen	<i>Klausur (120-180 min)</i>
Leistungspunkte und Notenvergabe	<i>Klausurnote (100%)</i>
Verwendung des Moduls (in anderen Studiengängen)	<i>????????????????</i>
Modulbeauftragte/r	<i>Lehrstuhl für Didaktik der Informatik</i>
Bemerkungen	<i>keine</i>
Termin Modulprüfung	<i>in der Regel Anfang März</i>
2. Termin Modulprüfung	<i>in der Regel Ende März</i>
Termin Praktikum / Exkursion	<i>---</i>