

Modul Inf-Einf-B Einführung in die Informatik <i>Introduction to Computer Science</i>	9 ECTS / 270 h
(seit WS24/25) Modulverantwortliche/r: Prof. Dr. Dominik Herrmann	
<p>Inhalte:</p> <p>In diesem Modul erhalten die Studierenden eine umfassende Einführung in die Informatik. Dazu werden gängige Prinzipien der Programmierung und Techniken zur Problemlösung, sowohl mit als auch ohne Code, vermittelt. Dies befähigt die Studierenden, sich eigenständig in Programmiersprachen einzuarbeiten und komplexe Problemstellungen zu bearbeiten.</p> <p>Nach einer Einführung essenzieller Konzepte wie Variablen, Funktionen, Bedingungen und Schleifen machen sich die Studierenden mit gängigen Linux-Kommandozeilenprogrammen und Dateisystemkonzepten vertraut, die eine text- und dateibasierte Datenverarbeitung mittels Shell-Skripten ermöglichen. Dies bildet die Basis für die Einführung in die systemnahe Programmiersprache C. Dabei werden imperative und prozedurale Programmierung sowie dynamische Speicherverwaltung, stapelbasierte Programmausführung und für die Programmierung relevante Mechanismen der Datenrepräsentation vermittelt (bspw. Overflows, Unicode, Escape-Sequenzen). Im weiteren Verlauf wird die Programmiersprache Python eingeführt, anhand derer Konzepte moderner Programmiersprachen sowie objektorientierter und funktionaler Programmierparadigmen erörtert werden. Parallel dazu werden Arbeitstechniken zur Erstellung nachvollziehbarer und sicherer Programme vermittelt, etwa Debugging und Quellcodedokumentation, und verbreitete Fehlertypen aufgezeigt.</p> <p>Das Modul bietet zudem einen ersten Einblick in Algorithmen, abstrakte Datentypen und gängige Datenstrukturen. Die Studierenden implementieren iterative und rekursive Algorithmen zur Lösung grundlegender Probleme wie Sortieren und Suchen und modellieren Problemlösungen mit passenden Datenstrukturen wie Listen, Bäumen, Tries und Hash-Tabellen. Ergänzend erhalten sie Einblicke in die für die Anwendungsprogrammierung relevanten Aspekte von Rechnernetzen (TCP/IP). Abschließend erhalten die Studierenden einen Einblick in die Paradigmen, die bei der Entwicklung einfacher Webanwendungen mit HTML, Python und JavaScript zum Einsatz kommen.</p> <p>Die Inhalte werden theoretisch fundiert; der Schwerpunkt des Moduls liegt jedoch auf der Entwicklung praktischer Problemlösungskompetenzen durch Übungsaufgaben, die ein intensives Selbststudium erfordern. Das Erlernete wird durch ein semesterbegleitendes Programmierprojekt, das am Ende des Semesters präsentiert wird, angewandt und gefestigt.</p>	
<p>Lernziele/Kompetenzen:</p> <p>Nach Abschluss des Moduls sind die Studierenden in der Lage, prozedural sowie mit objektorientierten und funktionalen Konzepten zu programmieren und dabei die im Modul vermittelten grundlegenden Programmierkonzepte und -techniken anzuwenden. Darüber hinaus können die Studierenden Informationen angemessen darstellen und verarbeiten, Probleme analysieren, Muster in Problemstellungen erkennen und komplexe Problemstellungen in kleinere, handhabbare Teile zerlegen. Sie verstehen die verschiedenen Abstraktionsebenen eines Programms und können zwischen Entwurfs- und Implementierungsdetails unterscheiden.</p> <p>Die Studierenden sind auch in der Lage, Korrektheit, Design und Stil von Code zu beurteilen und neue Programmiersprachen selbständig zu erlernen. Sie können eine vorhandene Lösung testen, Fehler und ausgewählte Sicherheitsprobleme identifizieren und Spezialfälle in der Problemstellung erkennen. Darüber hinaus können sie Situationen erkennen, in denen ein Trade-off zwischen Laufzeit und Speicherplatz besteht. Mit den Grundlagen von Rechnernetzen und Webanwendungen sind sie vertraut. Schließlich</p>	

können sie ihre Programmierkenntnisse anwenden, um ein selbst gewähltes Problem zu lösen und ihre Lösung zu präsentieren.

Sonstige Informationen:

Der Arbeitsaufwand von 270 Stunden verteilt sich ausgehend von einem 15 Arbeitswochen dauernden Semester in etwa wie folgt:

30 Std. Vorlesungsteilnahme in Präsenz

30 Std. Übungsteilnahme in Präsenz

60 Std. Bearbeiten von wöchentlichen Übungsaufgaben, d.h. ca. 4 Std./Woche

90 Std. Vor- und Nachbereitung von Vorlesung und Übung, d.h. ca. 6 Std./Woche

40 Std. Programmierprojekt

20 Std. Vorbereitung auf und Zeit für die Abschlussklausur

Bei diesen Angaben handelt es sich um Empfehlungen; es besteht weder in Vorlesung noch Übung Anwesenheitspflicht. Der Gesamtaufwand für das Modul ist aber nur einzuhalten, wenn die o.g. Empfehlung in etwa eingehalten wird.

Zulassungsvoraussetzung für die Belegung des Moduls:

keine

Empfohlene Vorkenntnisse:

keine

Besondere

Bestehensvoraussetzungen:

keine

Angebotshäufigkeit: WS, jährlich

Empfohlenes Fachsemester:
ab dem 1.

Minimale Dauer des Moduls:
Semester

Lehrveranstaltungen

1. Vorlesung

Lehrformen: Vorlesung

Sprache: Deutsch

Angebotshäufigkeit: WS, jährlich

4,00 SWS

Literatur:

Für diesen Kurs werden keine Bücher benötigt oder empfohlen. Die unten aufgeführten Bücher könnten jedoch von Interesse sein.

- Hacker's Delight, Zweite Ausgabe, Henry S. Warren Jr., Pearson Education, 2013
- How Computers Work, Zehnte Ausgabe, Ron White, Que Publishing, 2014
- Programmieren in C, Vierte Ausgabe, Stephen G. Kochan, Pearson Education, 2015

2. Übung

Lehrformen: Übung

Sprache: Deutsch

Angebotshäufigkeit: WS, jährlich

2,00 SWS

Inhalte:

In der Übung werden die wichtigsten Konzepte der Vorlesung an theoretischen und praktischen Beispielen veranschaulicht und durch die Besprechung von typischen Aufgaben zum jeweiligen Thema vertieft.

Prüfung

schriftliche Prüfung (E-Prüfung) / Prüfungsdauer: 180 Minuten

Beschreibung:

Die maximale Punktzahl, die in der Prüfung erreicht werden kann, beträgt 100. Teilnehmer, die eine semesterbegleitende Studienleistung erbringen können bis zu 10 Bonuspunkte erreichen. Details zu Anforderungen und Abgabefristen werden in der ersten Vorlesung bekannt gegeben. Wenn die in der Prüfung erreichte Punktzahl ausreicht, um damit allein die Prüfung zu bestehen (in der Regel ist dies der Fall, wenn mindestens 50 von 100 Punkten erreicht wurden), werden die Bonuspunkte zu den in der Prüfung erreichten Punkten addiert. Die Note 1,0 kann auch ohne die Bonuspunkte erreicht werden.