

Lehrveranstaltungshandbuch PI1

Praktische Informatik 1

Version: 2 | Letzte Änderung: 10.09.2019 15:49 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

– Allgemeine Informationen

Langname Praktische Informatik 1

**Anerkennende
LModule** PI1 BaET

Verantwortlich Prof. Dr. Dieter
Rosenthal
Professor Fakultät IME

Gültig ab Wintersemester
2020/21

Niveau Bachelor

Semester im Jahr Wintersemester

Dauer Semester

**Stunden im
Selbststudium** 60

ECTS 5

Dozenten Prof. Dr. Dieter
Rosenthal
Professor Fakultät IME
Derichs

Voraussetzungen keine

Unterrichtssprache deutsch

**separate
Abschlussprüfung** Ja

Literatur

Elektronische Verweise auf ebook und Online
Tutorials

Abschlussprüfung

Details

Schriftliche Klausur:
Die Studierenden sollen nachweisen, dass sie 1.) mit grundlegenden Begrifflichkeiten sicher umgehen, 2.) programmiersprachliche und abstraktere Konstrukte zur Lösung von Anwendungsproblemen einsetzen und 3.) programmiersprachliche Lösungsvorschläge auf Korrektheit prüfen können. Typische Aufgabenformen zu 1.) sind Multiple-Choice-Fragen, Lückentexte, Bewertung von Aussagen hinsichtlich ihrer Korrektheit, zu 2.) Lösung kleinerer umgangssprachlich formulierter Probleme durch Struktogramme und Programmstücke und zu 3.) das Finden von Fehlern in vorgegebenen Programmstücken.

Mindeststandard

Mindestens 50% der möglichen Gesamtpunktzahl.

Prüfungstyp

Klausur

– Vorlesung / Übungen

Lernziele

Zieltyp	Beschreibung
Kenntnisse	Algorithmen Eigenschaften Darstellungsmöglichkeiten
Kenntnisse	Digitalrechner Bit/Byte-Begriff Grobaufbau der Soft- und Hardware-Architektur
Kenntnisse	Grundbegriffe der Programmierung höhere Programmiersprachen vs. Maschinensprachen Übersetzung vs. Interpretation prozedurale vs. objektorientierte Sprachen: C vs. C++
Kenntnisse	Variablen-Begriff
Kenntnisse	einfache Datentypen in C Zahlen Wertebereiche Konstantendarstellung Operationen Zeichen Codierungen: ASCII, Unicode Operationen Zeichenketten boolesche Werte Konstantendarstellung Operationen
Kenntnisse	Kontrollstrukturen in C abstrakte Darstellungsformen Struktogramme Programmablaufpläne Blöcke bedingte Anweisungen if if-else Mehrfachverzweigungen: switch- case Schleifen kopfgesteuerte Schleifen for while fußgesteuerte Schleifen: do-while
Kenntnisse	Arrays in C Indizierung und Schleifen mehrdimensionale Arrays

Besondere Voraussetzungen

keine

Begleitmaterial

elektronische
Vortragsfolien zur
Vorlesung, frei
verfügbare
Softwareentwicklungsumgebungen
aus dem Internet

Separate Prüfung

Nein

Kenntnisse	Funktionen: Aufbau Parameterübergabe (Call by value, Call by reference)
------------	--

Kenntnisse	Speicherorganisation: Pointer-Begriff Dynamischer Speicher
------------	--

Kenntnisse	Strukturen in C: Aufbau Anwendung (statisch/dynamisch)
------------	--

Fertigkeiten	Erstellung von Algorithmen zur Lösung vorgegebener Probleme Beschreibung in natürlicher Sprache Beschreibung in grafischer Form (Struktogramme und/oder Programmablaufpläne)
--------------	--

Fertigkeiten	Programmierung elementarer Operationen in einer höheren Sprache auf Zahlen auf Zeichen und Zeichenketten auf booleschen Werten
--------------	---

Fertigkeiten	Programmierung mit Kontrollstrukturen
--------------	--

Fertigkeiten	Programmierung mit Funktionen
--------------	-------------------------------

Fertigkeiten	Programmierung mit strukturierten Datentypen, insbes. Arrays und Strukturen
--------------	---

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	1
Tutorium (freiwillig)	0

– Praktikum

Lernziele

Zieltyp	Beschreibung
Kenntnisse	Programmierung elementarer Operationen auf einfachen Datentypen
Kenntnisse	Programmierung mit Kontrollstrukturen (mit vorheriger Erstellung entsprechender Struktogramme und/oder Programmablaufpläne)
Kenntnisse	Programmierung mit strukturierten Datentypen, insbes. Arrays
Fertigkeiten	Umgang mit einer Softwareentwicklungsumgebung
Fertigkeiten	Fehlersuche und -beseitigung in Programmen
Fertigkeiten	Erstellung von Algorithmen und Umsetzung in Programme
Fertigkeiten	Anwendung der unter "Kenntnisse" und "Fertigkeiten" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	1
Tutorium (freiwillig)	0

Besondere Voraussetzungen

keine

Begleitmaterial elektronische Sammlung von Beispielprogrammen, frei verfügbare Softwareentwicklungsumgebungen

Separate Prüfung Ja

Separate Prüfung

Prüfungstyp praxisnahes Szenario bearbeiten (z.B. im Praktikum)

Details

Die Studierenden schließen sich zu Kleingruppen zusammen. Jede Kleingruppe absolviert mehrere "Praktikumsrunden" mit zugewiesenen Laborterminen. In jeder Runde werden Programmieraufgaben algorithmischer und objektorientierter Art gelöst - erstens durch eine abstraktere Darstellung (z.B. Beschreibung eines Algorithmus durch ein Struktogramm), zweitens durch eine programmtechnische Umsetzung (z.B. C-Programm). Zur Vorbereitung eines Labortermins muss ein "Vorbereitungsblatt" praktisch gelöst werden. Die dabei erworbenen Kenntnisse werden zu Beginn des Termins geprüft (kurzer schriftlicher Eingangstest, persönliches Gespräch mit dem Betreuer). Wird diese Prüfung nicht bestanden, so muss ein Folgetermin wahrgenommen werden; im Wiederholungsfall führt dies zum Nichtbestehen des Praktikums. Im Erfolgsfall wird ein "Laborarbeitsblatt" mit weiteren Aufgaben unter Aufsicht (und ggf. mit Hilfestellung) bearbeitet.

Mindeststandard

Erfolgreiche Teilnahme an allen Laborterminen, d.h. insbesondere selbstständige (ggf. mit Hilfestellung) Lösung der Programmieraufgaben.

