Technology Arts Sciences
TH Köln

Course Manual PI2

Practical Informatics 2

Version: 2 | Last Change: 10.09.2019 15:52 | Draft: 0 | Status: vom verantwortlichen Dozent freigegeben

− General information

Long name	Practical Informatics 2
Approving CModule	PI2_BaET
Responsible	Prof. Dr. Dieter Rosenthal Professor Fakultät IME
Valid from	summer semester 2021
Level	Bachelor
Semester in the year	summer semester
Duration	Semester
Hours in self-study	60
ECTS	5
Professors	Prof. Dr. Dieter Rosenthal Professor Fakultät IME Derichs
Requirements	basic knowledge in C
Language	German
Separate final exam	Yes

Literature

Elektronische Verweise auf ebooks und Online Tutorials

Final exam

Details	Written exam: Students shall prove that they can 1.) explain and apply fundamental terms, 2.) apply programming and more abstract concepts to solve application problems and 3.) assess the correctness of proposed solutions. Typical types of assignments are 1.) multiple choice questions, fill-in-the-blank texts, assessment of statements, 2.) solving given problems of limited size by programs and Nassi-Shneiderman diagrams and 3.) finding errors in given programs.
Minimum standard	At least 50% of the total number of points.
Exam Type	EN Klausur

Lecture / Exercises

Learning goals

Goal type	Description
Knowledge	Basic of object oriented programming
Knowledge	Structure of classes, creation of objects constructor, Overloading of methods, Initialization Lists reference vs. pointer
Knowledge	Inheritance
Knowledge	Polymorphism abstract methods and classes virtual methods
Knowledge	access mechanism private, protected, public friend
Knowledge	Keyword static usage in functions/methods vs. classes
Knowledge	templates methods classes
Skills	programming of classes and objects
Skills	programming of inherited classes and objects
Skills	Using polymorphism in inherited classes programming of abstract methods programming of virtual methods
Skills	Programming of attributes and methods in private, protected and public areas
Skills	programming of templates

Special requirements

basic knowledge in C

Accompanying material	lecture foils (electronic), free software development environments from the Web, example programs (in electronic form)
Separate exam	No

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Lecture	2

Exercises (whole course)	1
Exercises (shared course)	1
Tutorial (voluntary)	0

Practical training

Learning goals

Goal type	Description
Knowledge	programming of classes and objects
Knowledge	programming of inherited classes and objects
Knowledge	Using polymorphism in inherited classes programming of abstract methods programming of virtual methods
Knowledge	Programming of attributes and methods in private, protected and public areas
Skills	application of the aspects listed above to real-world scenarios in small teams

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Practical training	1
Tutorial (voluntary)	0

Special requirements

Accompanying material	example programs (in electronic form)
Separate exam	Yes

Separate exam

Exam Type	EN praxisnahes Szenario bearbeiten (z.B. im Praktikum)

Details	Students work in small teams. Each team completes multiple "rounds" with assigned appointments in the lab. In each round, programming assigments of an algorithmic and object-oriented nature are solved - firstly by a more abstract representation (e.g. description of an algorithm by a Nassi-Shneiderman diagram), secondly by an runnable implementation (e.g. C++ program). For the preparation of a laboratory appointment a "preparation sheet" has to be solved. The acquired knowledge will be tested at the beginning of the appointment (short written entrance test, interview with the supervisor). In case of failure, a follow-up appointment must be taken; in case of multiple failures, the student will be excluded from the lab. In case of success, a "laboratory work sheet" with further tasks will be worked on under supervision (and, if necessary, with assistance).
Minimum standard	Successful participation in all laboratory appointments, i.e. in particular independent solution (or with some assistance if necessary) of the programming assignments.