Technology
Arts Sciences
TH Köln

Course Manual PI1

Practical Informatics 1

Version: 2 | Last Change: 01.08.2019 16:59 | Draft: 0 | Status: vom verantwortlichen Dozent freigegeben

− General information

Long name	Practical Informatics 1
Approving CModule	PI1_BaTIN
Responsible	Prof. Dr. Cartsten Vogt Professor Fakultät IME
Valid from	winter semester 2020/21
Level	Bachelor
Semester in the year	winter semester
Duration	Semester
Hours in self-study	60
ECTS	5
Professors	Prof. Dr. Cartsten Vogt Professor Fakultät IME
Requirements	none
Language	German
Separate final exam	Yes

Literature

siehe http://www.nt.fh-koeln.de/vogt/dv/dv_lit.pdf

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	algorithms characteristics of algorithms description of algorithms
Knowledge	digital computers bits/bytes structure of the hard- and software architecture
Knowledge	basic concepts of programming high-level programming languages vs. machine languages compilation vs. interpretation procedural vs. object-oriented languages: C vs. Java
Knowledge	basic concepts of variables
Knowledge	scalar data types in Java (and C) numbers value ranges representation of constants operations characters coding standards: ASCII, Unicode operations character strings boolean values representation of constants operations
Knowledge	control structures in Java (und C) abstract representation Nassi-Shneiderman diagrams flow charts blocks conditional statements if if-else switch-case loops pre-test loops for while post-test loops: do-while

Special requirements

none

Accompanying material	lecture foils (electronic), free software development environments from the Web, example programs (in electronic form), links to relevant Web pages, recommendations for further reading
Separate exam	No

Knowledge	static methods in Java method definition header with parameters and return type body with return statement method call parameter passing: call by value vs. call by reference overloading storage classes
Knowledge	arrays in Java storage organisation: references indexing and loops multi-dimensional arrays
Knowledge	objects and classes in Java object-oriented programming: motivation and fundamental concepts encapsulation objects with members and methods classes constructors access control class members and methods
Skills	writing algorithms to solve given problems (in natural language and in graphical form - Nassi-Shneiderman diagrams, flow charts)
Skills	programming with elementary operations in a higher programming language
Skills	programming with control structures
Skills	programming with methods
Skills	programming with structured data, esp. arrays
Skills	programming with fundemental concepts of object-oriented programming (classes and objects)

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 elementary operations on scalar variables
Knowledge	programming with control structures (including the design of Nassi-Shneiderman diagrams or flow charts)
Knowledge	programming with methods
Knowledge	programming with structured data, esp. arrays
Skills	working with a software development environment
Skills	finding and correcting errors in programs
Skills	designing algorithms and implementing them in a higher language
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

none

Accompanying material	example programs (in electronic form), free software development environments from the Web
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. Java 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.