

Course Manual IP

Programming Project

Version: 2 | Last Change: 30.09.2019 13:05 | Draft: 0 | Status: vom verantwortlichen Dozent freigegeben

– General information

Long name Programming Project

Approving CModule [IP_BaET](#)

Responsible Prof. Dr. Stefan Kreiser
Professor Fakultät IME

Valid from winter semester
2020/21

Level Bachelor

Semester in the year winter semester

Duration Semester

Hours in self-study 72

ECTS 3

Professors Prof. Dr. Stefan Kreiser
Professor Fakultät IME
Al Ghouz

Requirements Basic programming
knowledge in C

Language German, English if
necessary

Separate final exam No

Literature

Reg. Rechenzentrum der Uni Hannover: Die
Programmiersprache C (Campus-IT FH Köln)

S. Kochan: Programming in C (Pearson)

P. Prinz, T. Crawford: C in a Nutshell (O'Reilly)

R.Lischner: C++ in a Nutshell (O'Reilly)

T. DeMarco: Structured Analysis and System
Specification (Prentice Hall PTR)

M. Dausmann et. al.: C als erste
Programmiersprache (Vieweg, Teubner)

J. Wolf: C von A bis Z, Das umfassende Handbuch
(Openbook, Rheinwerk Computing)

– Lecture / Exercises

Learning goals

Goal type	Description
-----------	-------------

Skills

Students show that they can develop executable programs in the procedural programming language C, which solve algorithmically solvable project tasks of medium complexity comprehensibly and completely. The students are able to:

1. Extract algorithms and data structures from a textual task description and develop and systematically test a program code in C solving the task by making use of a sound functional decomposition, the extracted and any possibly given algorithms and data structures as well as the given programming guidelines.
2. Explain, document, justify and modify the function of the software and the structure of the program code.
3. Firmly use an integrated development environment to build executable programs.

The project tasks to be processed are assigned to one of three distinct pools (low, medium and high complexity tasks / workload) depending on the presumed workload to solve the tasks. Each pool requires students to complete one or more project tasks by a given deadline. In order to complete a project task, students are allowed to form project groups of a maximum of three persons each. For each pool, students have to deliver all assigned executables and source codes and then explain, justify and, if necessary, modify one of their programs in a technical discussion / interview.

Special literature

Tutorials zur Installation der Virtuellen Maschine und zur Nutzung der Entwicklungsumgebung

Special requirements

personal notebook computer

Accompanying material

- digital collection of project tasks (eLearning platform Ilias)
- virtual linux machine to develop software programs in C including graphical structogram editor and integrated development system (Eclipse/C)

Separate exam

Yes

Separate exam

Exam Type	undefined
-----------	-----------

Expenditure classroom teaching

Type	Attendance (h/Wk.)
------	--------------------

Project	1
<hr/>	
Tutorial (voluntary)	0

Details

Per pool students verify that they are able to develop solutions / software programs for any of the given project tasks by working in a small project team in compliance with specified programming guidelines and to complete the solutions on schedule.

By documenting the program output from tests with predefined and additional own test criteria, they prove that their programs fulfill the required function. In the technical interview, the students prove that they have developed the programs themselves. For this purpose, they explain and justify the functionality of essential sections of the program code (e.g. conditional expressions, alternatives, iterations, function calls, data types and structures) based on test runs, if necessary by making use of additional or extended test criteria, and are able to make at least one meaningful modification and/or extension to their program code and show or prove the functionality of the modified program.

Minimum standard

- All necessary elements of the solution are available, i.e. the elements marked as "generally required" for any solution and the elements that are obviously required as part of the specific solution.
- The programming guidelines are met.
- Students can comprehensibly explain the implemented algorithm in its essential parts. They are able to explain and comprehensibly justify the use of such essential functions, control and data structures on the basis of function calls, condition expressions and memory management.
- Students are able to discuss, justify and prove that the programmed function fulfills the requirements given by the task in at least one essential section of the program code by making reasonable use of test runs.
- Students are able to purposively modify the program code to an algorithmically relevant extent based on additional requirements or specifications arising from the technical discussion, i.e. they are able to make significant changes to the program code and justify the functionality of the modified program.