Principles of programming - course description

General information

Course name	Principles of programming
Course ID	11.3-WE-ELEKTP-PrinProgr-Er
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Electrical Engineering
Education profile	academic
Level of studies	First-cycle Erasmus programme
Beginning semester	winter term 2022/2023

Course information

Semester	3
ECTS credits to win	5
Course type	obligatory
Teaching language	english
Author of syllabus	• dr hab. inż. Paweł Majdzik, prof. UZ

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Credit with grade
Laboratory	30	2	-	-	Credit with grade

Aim of the course

To provide basic knowledge about computer system architecture and programming.

To provide basic knowledge about C program structure and design.

To give basic skills in using C commands and functions to solve programming problems.

Prerequisites

-

Scope

Computer system structure. Operating system. Program structure and design.

Programming languages. Algorithmic languages. C programming. Program structure, commands syntax, identifiers, types, constans, declarations of data.

Arithmetic Operations, Relational and Logical Operations, Bitwise Operators, Assignment Operators, Type Conversions

Instructions: expressional instruction, empty instruction, grouping instruction.

Control instructions: if-else , for loop, switch, while loop. Printout formatting with printf function. Flag, field width, precision, formatting character.

Complex instructions, expressional instruction, grouping instruction. Control instructions: if-else, switch. Loops: do, while, for.

Functions: prototypes, declaration, definition, benefits for functions, arguments, result, calling out, use of functions, recurrence functions.

Scopes of names: local scope, external scope, modular programming.

Pointers: pointers syntax, declaration, operators, using the address and the pointed value. Use of pointers to communicate with other elements.

Arrays : declaration, usage, pointers and arrays, strings.

Data structures. Features, operation. Arrays of structures. Fields. Unions.

Dynamic memory: standard memory allocation functions, dynamic memory management

Data structures: lists, stacks, binary trees, circular buffers.

Teaching methods

Lecture, laboratory exercises.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Can realize a programming project individually, if necessary with additional self-studying.		• a quiz • an evaluation test	• Lecture • Laboratory
Knows and can practically apply principles of C language software design and analyze an example program		• a quiz • an evaluation test	• Lecture • Laboratory

Outcome description	Outcome symbols	Methods of verification	The class form
Knows and can solve examples of software tasks working individually or in a team		• a quiz • an evaluation test	• Lecture • Laboratory

Assignment conditions

Lecture – the passing condition is to obtain a positive mark from the final test.

Laboratory – the passing condition is to obtain positive marks from all laboratory exercises to be planned during the semester.

Calculation of the final grade: lecture 50% + laboratory 50%

Recommended reading

1. Język ANSI C. Programowanie. Wydanie II, Brian W. Kernighan, Dennis M. Ritchie, Wydawnictwo Helion, Gliwice 2010
2. Summit S. Programowanie w języku C, Helion, 2003,
3. Kisilewicz J. Język C w środowisku Borland C++, Wydanie IV, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2003
4. Stephen Prata, Języ.k C. Szkoła programowania, Robomatic, Wrocław 2001

Further reading

Notes

Modified by dr hab. inż. Paweł Szcześniak, prof. UZ (last modification: 06-04-2022 22:42)