**Nazwa przedmiotu:**

Programming Fundamentals

**Koordynator przedmiotu:**

Mariusz Kaleta

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Computer Science

**Grupa przedmiotów:**

Technical Courses

**Kod przedmiotu:**

EPFU

**Semestr nominalny:**

1 / rok ak. 2015/2016

**Liczba punktów ECTS:**

6

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

140: lecture 30, laboratory 15, tutorials 15, homeworks 40, preparation to classes 40

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

3

**Język prowadzenia zajęć:**

angielski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

3

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 15h |
| Laboratorium: | 15h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

English language

**Limit liczby studentów:**

60

**Cel przedmiotu:**

Students learn the fundamentals of structured programming.
Students should acquire skills of writing and debugging programs in C. They should become familiar with project development C from small C programs to multimodule applications.

**Treści kształcenia:**

The lecture presents the fundamentals of structured programming in ANSI C language and makes students familiar with syntax elements necessary to design console applications. The course include general structure of C programs, types and variables, arithmetic, conditional statements and loops, advanced types, usage of pointers. Standard libraries and header files allow to use libraries and undertake more complicated projects. Elementary knowledge on memory management and I/O operations helps to create efficient programs. Concluding part of material is dedicated to software architecture design and testing strategies.
Lectures
1. Introduction (2h): General information about programming problems. C among other programming languages.
2. Basic and formatted input/output (2h): The basic data types, declaring variables, function scanf() and printf, character strings, #define directive.
3. Arithmetic (2h): Operators, expressions, and statements.
4. Flow control: loops (2h): for, while, do while keywords, operators to construct expressions to control loops, arrays (a first look), functions returning values
5. Flow control: branching and jumps (2h): keywords if, else, switch, continue, break, case, default, goto, functions getchar() and putchar(), nesting conditional statements, character analysis functions library ctype.h.
6. Validation of character input/output (2h): getchar() and putchar() in details, buffers, terminating the keyboard input, redirection and files, input validation.
7. Functions (2h): Review, ANSI C prototyping, recursion, the & operator, altering variables in the calling function, a first approach to pointers.
8. Arrays and pointers (2h): Arrays and multidimensional arrays, pointers, pointer operations, protecting array contents, VLAs, compound literals.
9. Character strings (2h): Strings representation and string functions, creating and using strings, using command line arguments
10. Storage classes, linkage, and memory management (2h): Keywords auto, extern, static, register, const, volatile, restricted; functions rand(), srand(), time(), malloc(), calloc(), free(). Designing more complex programs.
11. File input/output (2h): Text modes and binary modes and formats. Buffered and non-buffered input/output. Sequential and random access.
12. Structures and various data forms (2h): Keywords struct, union, typedef; accessing the members of structures, nesting structures.
13. Operations on bits (2h): bitwise operators and bit fields.
14. The C preprocesors, macros (2h): Preprocessor directives, function-like macros, conditional compilation, inline functions, C libraries, ADTs.
15. SLC, Testing and summary (2h): Comments on Software Life Cycle, Testing strategies for software development, summary
The laboratory consists of five simple tasks
1. Basic communication with the console.
2. Variables, linear statements.
3. Conditional statements and loops.
4. Arrays and records.
5. Pointers.

**Metody oceny:**

Tests (30 pts), marks on laboratory exercises (30 pts), activity on tutorials (30 pts), lecture activity and extra points for other activities (10 pts).
Table of grades (local scale is given in brackets)
A (5) 91-110 pts.
B+ (4,5) 81-90 pts.
B (4) 71-80 pts.
C+ (3,5) 61-70 pts.
C (3) 51-60 pts.
D (2) 1-50 pts.
The minimal sum of points to pass the subject positively is equal to 51 pts.
There is possibility to undertake special challenges like outstanding projects under decision of the lecturer if the equivalent minimal qualificatiion has been proven. This proposal is designed for students pretending to the highest grades.

**Egzamin:**

nie

**Literatura:**

1. Michael A. Vine. C Programming for the Absolute Beginner. Course Technology CENGAGE learning.
2. Brian Kernighan, Dennis Ritchie: The C Programming Language. Also known as K&R — The original book on C. 2nd, Prentice Hall 1988; ISBN 0-13-110362-8. ANSI C.
3. ISO/IEC 9899.
4. Samuel P. Harbison, Guy L. Steele: C: A Reference Manual. 5th, Prentice Hall 2002; ISBN 0-13-089592-X.
5. Derek M. Jones: The New C Standard: A Cultural and Economic Commentary, Addison-Wesley, ISBN 0-201-70917-1, online material
6. Robert Sedgewick: Algorithms in C, Addison-Wesley, ISBN 0-201-31452-5 (Part 1–4) and ISBN 0-201-31663-3 (Part 5)
7. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery: Numerical Recipes in C (The Art of Scientific Computing), ISBN 0-521-43108-5
8. S. Prata: C Primer Plus, Fifth Edition, Sams, Nov. 23, 2004

**Witryna www przedmiotu:**

http://www.ise.pw.edu.pl/~antekg/epfu.html

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt EPFU\_W01:**

Student knows syntax of C programming

Weryfikacja:

tests, questions on tutorials

**Powiązane efekty kierunkowe:** K\_W07

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07

**Efekt EPFU\_W02:**

Student knows basic library routines and can use them

Weryfikacja:

tests, evaluation on laboratories and tutorials

**Powiązane efekty kierunkowe:** K\_W07, K\_W11

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07, T1A\_W03, T1A\_W04, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt EPFU\_U01:**

Student is able to develop a computer program

Weryfikacja:

laboratory exercises

**Powiązane efekty kierunkowe:** K\_U12

**Powiązane efekty obszarowe:** T1A\_U15, T1A\_U16

**Efekt EPFU\_U02:**

Student can transform a simple numerical problem to the form of computer program

Weryfikacja:

laboratory, tutorial

**Powiązane efekty kierunkowe:** K\_U03

**Powiązane efekty obszarowe:** T1A\_U03

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt EPFU\_K01:**

Student well places the programing skills in software life cycle

Weryfikacja:

test

**Powiązane efekty kierunkowe:** K\_K04, K\_K05

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04, T1A\_K06