

# Principles of programming - opis przedmiotu

| Informacje ogólne   |                                                                   |
|---------------------|-------------------------------------------------------------------|
| Nazwa przedmiotu    | Principles of programming                                         |
| Kod przedmiotu      | 11.3-WE-INFP-PrinofProgr-Er                                       |
| Wydział             | <a href="#">Wydział Informatyki, Elektrotechniki i Automatyki</a> |
| Kierunek            | Informatyka                                                       |
| Profil              | ogólnoakademicki                                                  |
| Rodzaj studiów      | Program Erasmus                                                   |
| Semestr rozpoczęcia | semestr zimowy 2017/2018                                          |

| Informacje o przedmiocie        |                                                                          |
|---------------------------------|--------------------------------------------------------------------------|
| Semestr                         | 1                                                                        |
| Liczba punktów ECTS do zdobycia | 5                                                                        |
| Typ przedmiotu                  | obowiązkowy                                                              |
| Język nauczania                 | angielski                                                                |
| Sylabus opracował               | <ul style="list-style-type: none"><li>dr inż. Elżbieta Kawecka</li></ul> |

| Formy zajęć  |                                         |                                        |                                            |                                           |                     |
|--------------|-----------------------------------------|----------------------------------------|--------------------------------------------|-------------------------------------------|---------------------|
| Forma zajęć  | Liczba godzin w semestrze (stacjonarne) | Liczba godzin w tygodniu (stacjonarne) | Liczba godzin w semestrze (niestacjonarne) | Liczba godzin w tygodniu (niestacjonarne) | Forma zaliczenia    |
| Wykład       | 30                                      | 2                                      | -                                          | -                                         | Zaliczenie na ocenę |
| Laboratorium | 30                                      | 2                                      | -                                          | -                                         | Zaliczenie na ocenę |

## Cel przedmiotu

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.

## Wymagania wstępne

-

## Zakres tematyczny

Computer system structure: Operating memory, CPU, I/O devices, idea of the interrupt, dual model Computer system architecture and resources. Operating system. Program design. Programming languages. The data and its representation. Algorithm visualisation. Algorithmic languages. Program performance analysis.

C programming. Program structure, commands syntax. Constants, variables, data types. Operators, expressions and basic instructions of C.

Basic operations on variables. Arithmetical operators, hierarchy. Data input and output. Printout formatting with printf function. Flag, field width, precision, formatting character. Character conversion. ASCII table.

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

Expressions and operators. Functions: structure, arguments, result, prototype, declaration, calling out. Communication with other elements. Use of functions. Recurrence functions.

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

Tables: declaration, usage, examples. String as a table of characters. Name of a table as a pointer. Tables of tables: declaration, usage, examples.

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

Disk file. Definition, structure, buffering. Directory, path. File operations: creating a stream, file opening, reading/writing, closing.

## Metody kształcenia

Lecture, laboratory exercises.

## Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

| Opis efektu                                                                                 | Symbole efektów | Metody weryfikacji                                                           | Forma zajęć                                                                 |
|---------------------------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Can realize a programming project individually, if necessary with additional self-studying. |                 | <ul style="list-style-type: none"><li>kolokwium</li><li>sprawdzian</li></ul> | <ul style="list-style-type: none"><li>Wykład</li><li>Laboratorium</li></ul> |

| Opis efektu                                                                                             | Symbole efektów | Metody weryfikacji                                                                  | Forma zajęć                                                                        |
|---------------------------------------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Knows and can solve examples of software tasks working individually or in a team                        |                 | <ul style="list-style-type: none"> <li>• kolokwium</li> <li>• sprawdzian</li> </ul> | <ul style="list-style-type: none"> <li>• Wykład</li> <li>• Laboratorium</li> </ul> |
| Knows and can practically apply principles of C language software design and analyze an example program |                 | <ul style="list-style-type: none"> <li>• kolokwium</li> <li>• sprawdzian</li> </ul> | <ul style="list-style-type: none"> <li>• Wykład</li> <li>• Laboratorium</li> </ul> |

## Warunki zaliczenia

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%

## Literatura podstawowa

1. Brad Jones. Teach Yourself C in 21 Days, Macmillan Computer Publishing, <http://lib.daemon.am/Books/C/>
2. Mike Banahan, Declan Brady and Mark Doran, The C Book, Addison Wesley, 1991, available free on-line: [http://publications.gbdirect.co.uk/c\\_book/](http://publications.gbdirect.co.uk/c_book/)
3. K. N. King, C Programming: A Modern Approach, 2008
4. Silberschatz A., Galvin P.B., Gagne G.: Operating system concepts. Seventh Edition, Wiley, 2005.

## Literatura uzupełniająca

### Uwagi

Zmodyfikowane przez dr inż. Elżbieta Kawecka (ostatnia modyfikacja: 04-05-2017 22:53)

Wygenerowano automatycznie z systemu SylabUZ