COURSE DESCRIPTION CARD - SYLLABUS

Course name

Programming languages

Course

Field of study

Biomedical engineering

Area of study (specialization)

-

Level of study

First-cycle studies

Form of study

full-time

Year/Semester

1/2

Profile of study

general academic

Course offered in

English

Requirements

compulsory

Number of hours

Lecture

15

Laboratory classes

30

Other (e.g. online)

0

Tutorials

0

Projects/seminars

0

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

dr hab. inż. Maciej TABASZEWSKI

e-mail: Maciej.Tabaszewski@put.poznan.pl

tel. 61 665 23 90

Faculty of Mechanical Engeenering

ul. Piotrowo 3, 60-965 Poznań

tel.: 61 665 23 61

Responsible for the course/lecturer:

Prerequisites

Basic knowledge of logic and computer science

Course objective

Transfer of knowledge allowing procedural and object-oriented programming

Course-related learning outcomes

Knowledge

The student recognizes and knows the features of procedural, object-oriented and visual programming

The student knows the basic structures of selected programming languages

The student knows the concepts of classes, structures, objects, inheritance, polymorphism, encapsulation

Skills

The student can create dedicated software

Social competences

The student understands the role of computerization in the modern economy. Is able to participate creatively

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture: Test, 20 closed questions, passing the subject -50% of the maximum points

Laboratory: short tests, passing the subject- 50% of the maximum points

Programme content

Lecture: General principles of program construction. Compilers and interpreters. Programming in low and high level languages, overview and division of languages. Visual programming languages. Structural programming. Basics of programming in C / C ++. Variables, data types, pointers, operators, loops, conditional instruction, input and output functions. Object oriented programming. The concepts of encapsulation, classes, objects, inheritance, polymorphism. Basics of object-oriented programming in C ++. References, operator overloading, streams, exceptions, namespaces. Basics of Python language.

Lab: Structured programming in C / C ++, examples: data input and output, simple calculations, use of conditional instruction, selection instruction, loops, writing and reading a text, binary file, creating functions. Object-oriented programming in C ++, examples: creating classes, single-inheritance, operator overloading, using the STL library. Basics of Python language.

Teaching methods

Lecture: multimedia presentation with theory and examples.

Laboratory classes: practical exercises, problem solving

Bibliography

Basic

1. Liberty J., Rao S., Jones B, L, - C++ dla każdego, Helion, Gliwice 2011

2. Wróblewski P., Algorytmy, struktury danych i techniki programowania, Helion, Gliwice 2009

3. Sarbicki S., Python. Kurs dla nauczycieli i studentów, Helion, Gliwice 2019.

Additional

Sedgewick R., Algorytmy w C++, READ ME, Łódź 1999

Kliszewski M., Inżynieria oprogramowania obiektowego, WKT, Warszawa 1994

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	47	2,0
Student's own work (literature studies, preparation for lecture, for laboratory classes, preparation for tests)[1]	53	2,0

[1] delete or add other activities as appropriate