# COURSE DESCRIPTION CARD - SYLLABUS

Course name
Computer programming and the C language [S1ETI2>PiJC]

## Course

Field of study
Education in Technology and Informatics

Year/Semester
1/2

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

## Number of hours

Lecture
15

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

## Number of credit points

3,00

## Coordinators

## Lecturers

## Prerequisites

Basic knowledge of computer science and mathematics. The ability to write computer programs in any highlevel programming language at the basic level, algorithmizing tasks, and logical and abstract thinking. Understanding the need to develop computer programs to increase productivity, computations, visualization of results, and presentation of the data collected in databases.

## Course objective

The main purpose of the subject is to familiarize the student with the basic principles of the C/C++ programming languages, gaining the skills of self-development of applications using the basic rules of these languages and algorithmizing tasks.

## Course-related learning outcomes

Knowledge:
w01 in-depth knowledge of the concepts and computer science issues necessary to understand programming in high-level languages, including C/C++.
knowledge and understanding of the elements of the C/C++ programming language that are necessary to create extensive computer programs within the scope specific to the field of study.
knowledge that is necessary to develop algorithms and implement them in the C/C++ programming language on its own.

Skills:
ability to create computer programs using the c/c++ programming language.
ability to use the acquired skills in creating algorithms and effective implementation of computational tasks and engineering problems.
self-learning skills in learning advanced c/c++ and other high-level languages.
ability to obtain information from literature, databases and other available sources of knowledge and ability to work individually and in a team.

Social competences:
responsibility for given tasks, e.g. an individual programming project.
understanding of the need for continuous learning in order to improve professional competences, acquiring up-to-date knowledge of programming languages and computer science (e.g., by reading computer journals, participating in postgraduate courses and studies).
ability to work on the assigned task independently and in the team taking on different roles in it.
understanding of the importance of algorithm development in the process of creating efficient computer programs.

## Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Written exam verifying a knowledge and a proper understanding of the field of study. The final grade depends on the percentage of points:
3 50.1%-70.0%
4 70.1%-90.0%
5 from 90.1%
Project tasks required in the form of computer programs and evaluated at the end of laboratory classes and/or during exams. The final grade depends on the percentage of points:
3 50.1%-70.0%
4 70.1%-90.0%
5 from 90.1%

## Programme content

The subjects of the module includes issues related to the basics of programming in C and C++ languages. The curriculum content assumes a discussion of data types and variables, operators, pointers, tables and structures, expressions and instructions of the C/C++ language, file input/output operations and functions.

## Course topics

Lectures:
1. Basic concepts and issues:
- Programming languages (low- and high-level languages and their characteristics).
- History of the C/C++ programming languages and their advantages.
- C++ standard.
- Introduction to integrated development environment Visual Studio.
- Numeral systems - binary and hexadecimal numeral systems.
- Standards for representing numbers, letters and special characters.
2. Data types and variables:
- Definition and division into simple (scalar) and structural data types.
- Characteristics of simple data types - integer, floating-point, logical, character.
3. C++ lexical units (keywords, identifiers, literals, constants).
4. Operators:
- Arithmetic operators (additive, multiplicative, incremental, decremental).
- Logical operators (conjunctions, alternatives, negation).
- Assignment and comparison operators.
- Conditional operator.
5. Pointers, tables and structures:
- Pointer types and variables - declaration, initiation and dereference.

- void data types.
- Reference data types and variables.
- Static arrays - declaration, initiation, access to array elements.
- Structures - declaration and access to structure elements.
- Dynamic arrays.
6. Expressions and instructions of the C/C++ language.
7. Pointer arithmetic - a relationship between static arrays and pointers.
8. File input/output operations.
9. Functions of the C/C++ language.
Laboratories:
1. Introduction to programming in the C/C++ languages. Data types, variables, constants, lexical units, operators.
2. One- and two-dimensional static arrays. C/C++ instructions:
- Conditional instructions. Arithmetic operations. Boolean expressions and relationships.
- Standard mathematical functions.
- Iterative instructions.
- Control instructions.
- One-dimensional static arrays. Generating pseudorandom numbers.
- Switch instruction. Operations on text files.
- Two-dimensional static arrays.
3. Structures. Static and dynamic arrays of structures.
4. Functions of the C/C++ language:
- One-dimensional dynamic arrays and functions.
- Two-dimensional dynamic arrays and functions.

## Teaching methods

Lectures:
Multimedia presentation illustrated with example programs written in the C/C++ language.
Laboratories:
Writing of computer programs in the C/C++ language.

## Bibliography

Basic:
1. H. M. Deitel, P. J. Deitel, Arkana C++ Programowanie, Wydawnictwo RM, Warszawa 1998.
2. S. Prata, Szkoła Programowania. Język C++, Wydawnictwo Helion, Gliwice 2006.
3. A. Zalewski, Programowanie w językach C i C++ z wykorzystaniem pakietu Borland C++, Wydawnictwo Nakom, Poznań 1996.
4. J. Grębosz, Symfonia C++. Programowanie w języku C++ orientowane obiektowo, Tom 1,2,3, Oficyna Kallimach, Kraków 1999.

Additional:
1. D. E. Knuth, Sztuka programowania. Tom1 Algorytmy podstawowe, Wydawnictwa Naukowo-Techniczne, Warszawa 2002.
2. N. Wirth, Algorytmy + struktury danych = programy, Wydawnictwa Naukowo-Techniczne, Warszawa 2004.

## Breakdown of average student's workload

| | Hours | ECTS |
|---|---|---|
| Total workload | 75 | 3,00 |
| Classes requiring direct contact with the teacher | 45 | 2,00 |
| Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation) | 30 | 1,00 |