POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Informatics [N1EiT1>INF1]

Course

Field of study	Year/Semester
Electronics and Telecommunications	1/1
Area of study (specialization)	Profile of study
–	general academic
Level of study	Course offered in
first-cycle	Polish
Form of study	Requirements
part-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
25	20	0
Tutorials	Projects/seminars	
0	0	

Number of credit points

6,00

Coordinators	Lecturers
prof. dr hab. inż. Grzegorz Danilewicz grzegorz.danilewicz@put.poznan.pl	prof. dr hab. inż. Grzegorz Danilewicz grzegorz.danilewicz@put.poznan.pl

Prerequisites

(S)he knows the basics of algebra and mathematical logic, knows different counting systems, knows the applications of computers, and recognizes the need to use computers in engineering work. (S)he should also have the ability to obtain information from the indicated sources and be ready to take cooperation with the team.

Course objective

To familiarize students with the basics of computer programming using the C language.

Course-related learning outcomes

Knowledge:
1. Knows the principles of designing computer programs, has knowledge in the field of computer science and knows
C software language syntax.
2. Has basic knowledge of algorithms (sorting, table searching), data types and
complex structures (arrays).

Skills:
1. Can programmatically implement basic computational algorithms using the language
C programming
2. Can use the C medium level programming language.
3. Can write and run programs to solve basic technical problems
related to telecommunications.

Social competence:
1. Knows the limitations of his own knowledge and skills, understands the need for further training in
in the field of programming in the C language.
2. Is aware of the need for a professional approach to solving problems
and taking responsibility for the technical solutions they propose.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The knowledge acquired during the lecture is verified in an oral and/or written exam.
The written exam consists of many questions (not less than 7) of various nature, checking the knowledge of
basic programmer's tools (block diagrams, pseudocode, etc.) and the mathematical basics of computer
functioning. Passing threshold: 50% of points. Used grading scale.
The oral exam consists of answers to at least 3 questions. The lecturer asks the questions. Passing
threshold: 50% of points.
<= 50% 2.0
51%-60% 3.0
61%-70% 3.5
71%-80% 4.0
81%-90% 4.5
91%-100% 5.0
The skills acquired during laboratory classes are verified when writing a credit program on a topic assigned
by the teacher. The program verifies the knowledge of the construction of a programming language, the
ability to use simple algorithms, the ability to build a program using a structured programming language.
The correctness of writing computer programs is assessed on a scale from 2 (insufficient – negative grade)
to 5.

Programme content

The module program covers the following topics:
1. basic concepts of computer science
2. basic concepts of computer design
3. basic concepts of computer programming
4. basics of programming in C language

Course topics

Lectures:1. The concept of computer science
2. The basics of computer construction and use
3. The concept of information, units of information
4. The basics of algebra, including Boolean algebra
5. Numbering systems including binary, octal, and hexadecimal
6. Representation of fixed and floating point numbers in computer memory
7. Basics of algorithmic and engineering software
8. The basics of the C language
9. The process of compiling programs
10. Data types
11. Complex structures
12. Functions
13. Pointers and operations on pointers
14. Complex structures on the example of arrays
Laboratories: Practice of the C language by writing programs using simple and complex types, division of
the program structure into functions, arithmetic operations, pointer operations, the relationship between

pointer arithmetic and array handling, the use of library functions on the example of input-output operations and mathematical operations.

Teaching methods

Lecture with the use of a board/projector.
Laboratory exercises: practical exercises, performing tasks given by the teacher.

Bibliography

Basic
Programowanie w C, Wikibooks (dostępne on-line)
Paweł Mikołajczak, Język C – podstawy programowania, UMCS, Lublin, 2011 (dostępne on-line)
Complementary
Brian W. Kernighan, Dennis M. Ritchie, Język ANSI C. Programowanie. Wydanie II, Helion
Greg Perry, Dean Miller, Język C. Programowanie dla początkujących. Wydanie III, Helion
Zed A. Shaw, Programowanie w C. Sprytne podejście do trudnych zagadnień, których wolałbyś unikać (takich jak język C), Helion
Robert C. Martin, Czysty kod. Podręcznik dobrego programisty, Helion

Breakdown of average student's workload

	Hours	ECTS
Total workload	130	6,00
Classes requiring direct contact with the teacher	55	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	75	4,00