POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name

Low level programming in C

Course

Field of study	Year/Semester
Computing	1/2
Area of study (specialization)	Profile of study
-	general academic
Level of study	Course offered in
First-cycle studies	english
Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
15	15	0
Tutorials	Projects/seminars	
0	0	

Number of credit points

3

Lecturers

Responsible for the course/lecturer:	Responsible for the course/lecturer:
Marcin Radom, Ph.D.	

Prerequisites

The student should have a basic knowledge about basic computer programming and be familiar with the basic terminology (memory, bytes, operating systems, etc.) Student should possess skills in solving basic problems and in acquiring knowledge from specific sources. He or she should understand the necessity of constant extending of programming knowledge.

Course objective

1. To provide knowledge to students about computer programming on the basis of ANSI C programming language on the beginner and intermediate level.

2. Develop students’ skills in solving basic algorithmic problems and the skills concerning the division of complex problems into the elementary steps that can be programmed in a given language.

Course-related learning outcomes

Knowledge

Upon completion of the course the student:

1. Has the general knowledge about algorithms, programming languages and programming techniques and a specific knowledge in selected areas of programming.

2. Has basic knowledge about life cycles of computer systems both hardware and software ones, and specifically about the processes existing in such systems.

3. Knows basic techniques, methods and tools used in the process of solving computer science problems, mainly the engineering ones.

Skills

Upon completion of the course the student:

1. Can – according to the given specifications – model and develop a computer systems (in broad sense), choosing programming language suitable to the given task and using proper methods, techniques and tools.

2. Can formulate algorithms and implement them using at least one from the current popular tools.

Social competences

Upon completion of the course the student:

1. Understands the necessity of learning new skills and rising her or his qualifications because of constantly changing and evolving computer tools and programming languages.

2. Understand the significance of knowledge in solving engineering problems, also knows and understands the causes of defective computer systems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment

a) verification of assumed learning objectives related to lectures:

- answers to the questions related to the discussed material

b) verification of assumed learning objectives related to laboratory classes:

- verification of completed exercises and the discussion concerning the results

Total assessment

a) verification of assumed learning objectives related to lectures:

• evaluation of knowledge by test in a form of 10 questions worth of 10 points in total. The points for each question depend on its difficulty level. Positive grade is obtained by acquiring at least 5 points.

•

b) verification of assumed learning objectives related to laboratory classes:

• constant evaluation of students skills and knowledge based on the results of solving practical problems given during the laboratories

• a test taking place on the last laboratory

• evaluation of the quality of the programming project

Programme content

The main task of the module is to teach student the C programming language. Module consists of full lecture of the C language with many examples. Apart from the lecture are the laboratories, during which student solve task and techniques described in the lectures.

The first lecture is the introduction to the C programming language, with small historical part, description of simple programs and basic orders of C language. Next lecture describe basic and advanced C orders like loops, condition clauses with numerous examples. The third lecture present preprocessor orders, declaration and definitions of global variables, data structures and functions. The fourth lecture tells about modularization, arrays, different types of functions and procedures and typical errors occurring during the creation of the algorithm / program. The fifth lecture tells about variables visibility, location of variables in the code and memory allocation. The last lecture tells about basic dynamic structures.

During the laboratories students learn about programming environment and start writing simple and more advanced programs. In the last laboratory a test evaluating students knowledge will take place.

Teaching methods

1. lecture: presentations with numerous examples of basic and advanced C programs

2. laboratories: exercises, solving task, practical exercises

Bibliography

Basic

1. Język ANSI C, B.W. Kernighan, D.M. Ritchie, WNT, Warszawa, 1998.

2. Symfonia C++, J. Grębosz, Oficyna Kallimach, Kraków, 2001

3. Programowanie w języku C++, J. Kniat , Nakom, Poznań, 2002.

Additional

1. http://pl.wikibooks.org/wiki/C - free ebook on the GNU GPL license.

Breakdown of average student's workload

	Hours	ECTS
Total workload	72	3,0
Classes requiring direct contact with the teacher	36	1,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	36	1,5

[1] delete or add other activities as appropriate