STUDY MODULE DESCRIPTION FORM

Name of the module/subject	Code
Low level programming in C	**1010511321010500190**

Field of study	Profile of study (general academic, practical)	Year /Semester
Computing	**general academic**	**1 / 2**
Elective path/specialty	Subject offered in:	Course (compulsory, elective)
-	**English**	**elective**

Cycle of study:	Form of study (full-time,part-time)
First-cycle studies	**full-time**

No. of hours	No. of credits
Lecture: **15** Classes: **-** Laboratory: **15** Project/seminars: **-**	**3**

Status of the course in the study program (Basic, major, other)	(university-wide, from another field)
major	**from field**

Education areas and fields of science and art	ECTS distribution (number and %)
technical sciences	**3 100%**
Technical sciences	**3 100%**

Responsible for subject / lecturer:

dr inż. Marcin Radom
email: marcin.radom@put.poznan.pl
tel. (061)665 30 29
Faculty of Computing
ul. Piotrowo 3 60-965 Poznań

Responsible for subject / lecturer:

dr inż. Marcin Radom
email: marcin.radom@put.poznan.pl
tel. (061)665 30 29
Faculty of Computing
ul. Piotrowo 3 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	The student should have a basic knowledge about basic computer programming and be familiar with the basic terminology (memory, bytes, operating systems, etc.)
2	**Skills**	Student should possess skills in solving basic problems and in acquiring knowledge from specific sources.
3	**Social competencies**	Student should understand the necessity of constant extending of programming knowledge

Assumptions and objectives of the course:

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

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. The student has a structured and theoretically founded general knowledge in the field of key issues in computer science, and detailed knowledge in the field of selected issues of this discipline of science (i.e., programming languages and programming techniques) - [K_W4]

2. The student has basic knowledge about the life cycle of IT systems, both hardware and software, and in particular about the key processes taking place in them. - [K_W6]

3. The student knows the basic techniques, methods and tools used in the process of solving IT problems, mainly of an engineering nature, in the field of key IT problems. - [K_W7]

Skills:

1. The student can - according to a given specification - design (i.e., create a model of a reality fragment) and implement a device or broadly understood IT system, selecting programming language suitable for a given programming task and using appropriate methods, techniques and tools. - [K_U10]

2. The student has the ability to formulate algorithms and implement them using at least one of the popular tools. - [K_U11]

Social competencies:

1. The student understands that in the field of IT the knowledge and skills quickly become obsolete. - [K_K1]
2. The student is aware of the importance of knowledge in solving engineering problems and knows examples and understands the reasons for malfunctioning IT systems that led to serious financial and social losses or to serious health conditions or even to death. - [K_K2]

Assessment methods of study outcomes

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

Course description

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

Basic bibliography:

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 bibliography:

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

Result of average student's workload

Activity	Time (working hours)
1. Participating in laboratories	15
2. Preparing for the laboratories	5
3. Finishing laboratory exercises	5
4. Consulting issues related to the subject of the course; especially related to the laboratory classes and projects	4
5. Writing programs, their execution and verification (in addition to the laboratory classes)	6
6. Preparing for laboratory tests	6
7. Participating in lectures	15
8. Studying literature / learning aids	10
9. Preparing to and participating in lecture knowledge verification (small exam)	5

Student's workload

Source of workload	hours	ECTS
Total workload	72	3
Contact hours	34	2
Practical activities	28	1