STUDY MODULE DESCRIPTION FORM

Name of the module/subject	Code
An introduction to programming	**1010334511010334957**

Field of study	Profile of study (general academic, practical)	Year /Semester
Information Engineering	**(brak)**	**1 / 1**

Elective path/specialty	Subject offered in:	Course (compulsory, elective)
-	**Polish**	**obligatory**

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

No. of hours				No. of credits
Lecture: **16** Classes: **-** Laboratory: **16** Project/seminars: **-**				**5**

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

Education areas and fields of science and art	ECTS distribution (number and %**)**
technical sciences	**5 100%**

Responsible for subject / lecturer:

dr Jerzy Bartoszek
email: jerzy.bartoszek@put.poznan.pl
tel. 61 665-3713, 61 665-2378
Wydział Elektryczny
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	has basic knowledge from the high school [PRK 4]
2	**Skills**	can carry out tasks corresponding to the high school program [PRK 4]
3	**Social competencies**	has social skills resulting from the high school [PRK 4]

Assumptions and objectives of the course:

Basic programming styles and programming concepts with examples of programs in C++/C

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

Knowledge:

1. knows and understands the advanced knowledge in the field of basic programming constructs, algorithms implementation, paradigms and programming styles, methods for verifying the correctness of programs, formal languages, compilers and platforms - [[K1_W05 (P6S_WG)]]

Skills:

1. can use programming environments and platforms to write, execute and test simple coded programs in the imperative, object and declarative programming languages, use analytical, simulation and experimental methods for this purpose   - [[K1_U10 (P6S_UW)]]

2. has skills to construct algorithms using basic algorithmic techniques, analyze their complexity and evaluate them   - [[K1_U09 (P6S_UW)]]

Social competencies:

1. is ready to critically evaluate his knowledge in the field of computer science and recognize the importance of knowledge in solving cognitive and practical problems in the area of computer science - [[K1_K01 (P6S-KK)]]

Assessment methods of study outcomes

Lectures: written tests, pass criterion of 50% points.

Laboratory: exercises tests and laboratory reports.

Course description

Lectures:

Introduction: the structure of simple programs, selected data types, arithmetical and logical operators, expressions, assignments, conditionals, loops, simple I/O statements, namespaces. An introduction to functions. Dynamic and static arrays. References. Structures and operator overloading. Text and binary files. Header files. Dynamic data structures. Selected elements of C.

Course update 2017:

Pointers and dynamic memory allocation: RAII, smart pointers, make_unique, make_shared. More about functions and their parameters: function overloading, passing arguments, templates, lambdas.

Laboratory:

An introduction: main, int, std::string, arithmetic operators, if/else, cin/cout, debugger. Simple types and Loops. SVN. Funtions. Dynamic and static arrays. References: std::vector, std::array, for_each, auto. Structures. Text and binary files: std::fstream, reinterpret_cast. Header files. Namespaces. Function and operator overloading.

Pointers and dynamic memory allocation: RAII, smart pointers, make_unique, make_shared. Lambdas. Tamplates. How to read C programs?: printf, scanf, malloc, free, static and dynamic arrays.

Teaching methods:

lectures - with multimedia presentation, additional topics included in Moodle course

laboratory - with multimedia presentation, additional topics included in Moodle course, used tools enable students to perform tasks at home

Basic bibliography:

1. Grębosz J., Symfonia C++ standard, Programowanie w języku C++ orientowane obiektowo, T.1 i 2

2. Stroustrup B., Programming - Principles and Practice Using C++

3. http://en.cppreference.com/w/

4. https://isocpp.org/faq

5. https://msdn.microsoft.com/en-us/library/3bstk3k5.aspx

6. http://www.cplusplus.com/

Additional bibliography:

1. Banachowski L., Kreczmar A., Rytter W., Analysis of Algorithms and Data Structures, Addison Wesley, 1991

Result of average student's workload

Activity	Time (working hours)
1. participation in lectures	16
2. participations in labs.	16
3. exam, consultation	8
4. preparation for labs., reports	48
5. preparation for tests and exam	40

Student's workload

Source of workload	hours	ECTS
Total workload	128	5
Contact hours	40	2
Practical activities	75	3