Bialystok University of Technology									
Field of study	**Computer Science**							**Degree level and programme type**	**Engineer's degree full-time programme**
Specialization/ diploma path	**---**							**Study profile**	academic
Course name	**Programming Basics**							**Course code**	**FCS-00031**
								Course type	obligatory
Forms and number of hours of tuition	L	C	LC	P	SW	FW	S	**Semester**	3
	30	30			30			**No. of ECTS credits**	**6**
Entry requirements									
Course objectives	To familiarize students with the basic techniques of structured programming and implementation of these techniques in C / C ++. Teaching the practical application of these techniques in simple programs written in C / C ++. Learning design, implement, track performance of the software and documentation of larger programs using structured programming. Education practice the use of literature and documentation programming language. Education habit of systematically acquiring knowledge of the lecture.								
Course content	Algorithm. Programming: algorithmization, coding. Flowcharts. Systems and programming aids. Structured programming. Symbols basic, basic data structures: constants, standard scalar types. Operators. Expressions. Defining variables, basic instructions. Structural instructions. Iterations. Defining types. Arrays. Scalar types, flow of control. Functions. Passing parameters. Side effects. Recursion. The structure of the program, structural types: structure, strings. Files. Formatting output. Pointers. Dynamic variables. Memory management. Dynamic data structures. Lists.								
Teaching methods	lecture problem, programming, subject exercises,								
Assessment method	Lecture - written exam (only after the positive grade from the Specialistic Workshop and Classes); Specialistic Workshop - tests of theoretical knowledge, evaluation of selected programs in the classroom, a larger program / project implemented outside the classroom; Classes - test.								

Symbol of learning outcome	**Learning outcomes**	**Reference to the learning outcomes for the field of study**
LO1	familiar with the methodology, techniques and tools for structured programming	K_W04
LO2	knows and applies the principles of structured programming design of simple applications	K_W06 K_U04
LO3	is able to test a structural program, in case of detecting errors, is able to diagnose them	K_U04
LO4	able to formulate an algorithm using a programming language to develop a computer program	K_W04 K_U06

Symbol of learning outcome	**Methods of assessing the learning outcomes**	**Type of tuition during which the outcome is assessed**
LO1	exam, quiz	L
LO2	evaluation of programs and simple project	L, C
LO3	evaluation of programs	
LO4	exam, quiz	L, Sw, C

Student workload (in hours)	**No. of hours**
Calculation	
1 - Attendance in lectures -	30
2 - Attendance in Classes and Specialistic workshops -	60
3 - Preparing to Classes and Sw -	13
4 - Attendance on the consultations -	5
5 - Preparation to the examination -	14
6 - Preparation to Classes -	13
7 - Preparing the project (including its presentation) -	15
TOTAL:	**150**

Quantitative indicators	**HOURS**	**No. of ECTS credits**
Student workload – activities that require direct teacher participation	95 (4)+(2)+(1)	3.8
Student workload – practical activities	101 (7)+(3)+(2)+(6)	4.0

Basic references	1. Kernighan B.W., Richtie D.M.: The C Programming Language. 2. Code::Blocks (http://www.codeblocks.org/).
Supplementary references	1. Wirth N., Systematic Programming: An Introduction. 2. Stephen Prata, C Primer Plus. 3. Herbert Schildt, C/C++ Programmer's Reference. 4. Dev-C++ (http://www.bloodshed.net/devcpp.html). 5. Tondo C.L., Gimpel S. E., The C answer book.
Organisational unit conducting the course	Software Department
Date of issuing the programme	Feb. 17, 2022
Author of the programme	dr inż. Dorota Duda

L – lecture, C – classes, LC – laboratory classes, P – project, SW – specialization workshop, FW – field work,S – seminar