

## PROGRAMMING BASICS

Faculty of Computer Science			
Study programme:	Computer Science		Degree level: Engineer's degree full-time programme
Specialization	---		Diploma path: 2026/2027W - 2026/2027S
Module name:	Programming Basics ( Podstawy programowania )		
Module type:	obligatory	Semester: 1	ECTS:6 Module ID:FCS-00031
No. of hrs in semester:	Lecture (L) - 30 Classes(C) - 0 Specialization workshop (SW) - 45 Project (P) - 0 Laboratory classes (LC) - 0 Seminar (S) - 0		
Prerequisites	-		
Aims and 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.		
Forms of teaching activities::	lecture, specialization workshop,	Assessment:	Evaluation must be relevant to the intended learning outcomes:  Lecture - written examination; Specialistic Workshop - tests of theoretical knowledge, evaluation of selected programs in the classroom, a larger program / project implemented outside the classroom.
Module content:	<p>Lecture: 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.</p> <p>Specialization Workshop: Program Structure in C. Sample Program. Output-Output Statement. Assignment Statement. Conditional Statement. Coding Branching Programs. Coding Iteration. Using One-Dimensional Arrays. Using Functions. Using Recursive Functions. Implementing Sorting Algorithms. File Processing. Processing Multidimensional Arrays. Implementing File Processing. Using Complex Types. Implementing List-Structure Processing Operations.</p>		
Teaching methods:	subject exercises, programming, lecture problem,		
Learning outcomes			
Symbol	Specify min. 4, max. 8 learning outcomes in the following order: knowledge – skills – competence. Each learning outcome must be verifiable	Reference to the programme learning outcomes of education	
LO1	knows methodologies, techniques and tools of structured programming along with examples and answers questions in this area	INF1_W04 INF1_W05 INF1_W11 INF1_W12	
LO2	is able to apply the principles of design and structured programming in the created programs and their fragments	INF1_U04 INF1_U05 INF1_U11	
LO3	can test a structural program and, if errors are detected, can diagnose them	INF1_U06	
LO4	can formulate an algorithm using a structured programming language to develop a computer program	INF1_U04 INF1_U05 INF1_U07 INF1_U11	
LO5	is ready to take into account sustainable development policies when selecting efficient algorithms	H1_K03	
No. of learning outcome	Methods of assessing the learning outcome	Type of teaching activities (if more than one) during which the outcome is assessed	
LO1	written assessment at the last lecture	L	
LO2	Evaluation of programs and simple project	Sw	
LO3	Evaluation of programs and simple project	Sw	
LO4	Evaluation of programs	Sw	
LO5	Evaluation of simple project	Sw	
Student's workload (in hours)	1 - Attendance in lectures	2 x 15h	30
	2 - Attendance in a specialization workshop	3 x 15h	45
	3 - Attendance on the consultations	---	4
	4 - Preparation to the examination	---	20
	5 - Preparation for ongoing practical classes	---	32
	6 - Preparing the project (including its presentation)	2 x 9h + 1h	19
		<b>TOTAL:</b>	<b>150</b>
Quantitative indicators	Student's workload - activities that require direct teacher participation: (1)+(2)+(3)	79	<b>ECTS</b>
			3.2

	Student's workload connected with practical classes (6)+(2)+(5)	96	3.8
Basic references:	<ol style="list-style-type: none"> <li>1. B. W. Kernighan, D. M. Ritchie, The C Programming Language, Second Edition, Prentice Hall Software Series</li> <li>2. S. Prata, C Primer Plus, 6th Edition, Pearson Education, 2013</li> <li>3. R. Reese, Understanding and Using C Pointers, O'Reilly Media, 2013</li> <li>4. R. C. Seacord, Effective C: An Introduction to Professional C Programming, Random House LLC US, 2020</li> <li>5. <a href="https://en.cppreference.com/w/c">https://en.cppreference.com/w/c</a></li> </ol>		
Further reading	<ol style="list-style-type: none"> <li>1. Wirth N., Systematic Programming: An Introduction.</li> <li>2. Stephen Prata, C Primer Plus.</li> <li>3. Herbert Schildt, C/C++ Programmer's Reference.</li> <li>4. Dev-C++ (<a href="http://www.bloodshed.net/devcpp.html">http://www.bloodshed.net/devcpp.html</a>).</li> <li>5. Tondo C.L., Gimpel S. E., The C answer book.</li> </ol>		
Unit:	Software Department	Lecturer/ instructor	dr inż. Dorota Duda
Date of issuing the programme:	25th March 2026	Author of the programme:	dr inż. Dorota Duda

L - lecture, C - classes, LC - laboratory classes, P-project, SW - specialization workshop, S - seminar