Programming Basics

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.110.03024.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.PP.OL

Course coordinator

Iwona Cieślik

Course instructor

Iwona Cieślik

Period Semester 1	Form of verification of learning outcomes exam Form of teaching and hours lecture: 30 laboratory classes: 30	Number of ECTS credits 6.0

Educational goals for the course

C1	Developing basic programming skills in C and C++ languages.
C2	Developing basic skills in creating simple algorithms.

Learning outcomes for the course

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	the syntax of C and C++ languages and basic functions from the standard libraries of these languages	IAN_K1_W04, IAN_K1_W05	programming tasks, test
W2	methods of number representation in computer memory and properties of computer arithmetic	IAN_K1_W13	written exam
W3	basics of algorithms, basic data structures (arrays, lists, trees), their computer representations and operations performed on them, and basic techniques for algorithm construction and analysis	IAN_K1_W06, IAN_K1_W07, IAN_K1_W08	written exam, programming tasks, test
Skills – The student can:			
U1	program in C and C++ languages	IAN_K1_U03, IAN_K1_U05	programming tasks, test
U2	design and implement simple algorithms using basic data structures such as: arrays, strings, pointers, structures, objects, files, linked lists; uses these structures to describe simple problems presented in natural language	IAN_K1_U03, IAN_K1_U05, IAN_K1_U07	written exam, programming tasks, test
U3	use basic programming techniques such as function calls, recursion, backtracking; can choose the appropriate method	IAN_K1_U03, IAN_K1_U06	written exam, programming tasks, test
U4	write programs in a readable way and can analyze their code to locate errors	IAN_K1_U03, IAN_K1_U05	programming tasks, test
Social competences – The student is ready to:			
K1	formulate questions to better understand a given topic	IAN_K1_K01	graded credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	30
laboratory classes	30
independent solving of computer tasks	90
exam preparation	10
test preparation	5
exam participation	3

Total student workload	Number of hours 168	ECTS credits 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Programming in C language: a. basic and complex data types, b. operators, control instructions c. arrays, strings, pointers, functions d. input/output operations, working with files e. dynamic memory allocation f. complex data types g. dynamic data structures (lists, stacks, queues) h. basic functions from the standard library	W1, W3, U1, U2, U3, U4, K1
2.	Basics of object-oriented programming in C++ a. creating classes, public and private methods b. operator overloading c. streams, input/output operations d. dynamic memory allocation	W1, W3, U1, U2, U3, U4, K1
3.	Basics of algorithms a. Euclidean algorithm b. quadratic sorting algorithms c. binary search d. fast exponentiation e. reverse Polish notation f. positional systems and conversion algorithms g. recursion, backtracking algorithms, game tree h. dynamic data structures (lists, stacks, queues)	K1
4.	4. Theoretical foundations of programming a. Number representation in computers: positional systems, conversion algorithms, sign-magnitude and two's complement systems, fixed-point and floating-point representation, properties of computer arithmetic. b. Example of a digital machine - von Neumann Machine c. Algorithm correctness, invariants d. Basic concepts of computational complexity	W2, W3, K1

Extended information

Teaching methods:

multimedia lecture, discussion, problem solving, laboratory classes

Type of classes	Forms of credit	Course credit requirements
lecture	written exam	The student receives a final grade for the course based on points awarded during classes and points obtained during the written exam. The condition for receiving a positive final grade is obtaining class credit and accumulating a total of at least 60% of points.
laboratory classes	graded credit, programming tasks, test	The student receives a final grade for classes based on points awarded for systematically submitted programming tasks (mandatory and additional) and points obtained on the test. The condition for receiving class credit is submitting at least 70% of programming tasks, including all mandatory tasks, and accumulating a total of 60% of points.

Literature

Required

1. Lecture materials

Additional

1. B.W.Kerninghan, D.M.Ritchie, "The C Programming Language", Prentice Hall, 1988.
2. C.L.Tondo, S.E.Gimpel, "The C Answer Book: Solutions to the Exercises in 'The C Programming Language'", Prentice Hall, 1988.
3. J.Grębosz, Symfonia C++ Standard, Edition "2000" Publishing, Kraków 2008.
4. C and C++ language documentation.
5. J. Tomasiewicz, Zaprzyjaźnij się z algorytmami. Przewodnik dla początkujących i średnio zaawansowanych, PWN