

**SYLLABUS FOR 2023/2024 ENROLMENT
FORM OF STUDY: FULL-TIME PROGRAMME**

GENERAL INFORMATION

1. Course Basics of programming

2. Field of study Computer Science

3. Level of acquired education First-cycle programme

4. Number of ECTS credit 4

5. Number of hours per semester

semester	lecture	classes	laboratory/foreign language course	project/practical classes	self-study	internship
I	30		30			

6. Language of instruction: English

7. Lecturer mgr inż. Maciej Hawryluk, mgr inż. Jarosław Wetoszka

DETAILED INFORMATION

8. Preliminary requirements

Basic computer skills

9. Course objectives

C1 To introduce students to basic notions in the field of programming

C2 To teach students how to create algorithms in order to solve simple problems

C3 To teach students how to write programs

C4 To introduce students to the syntax of the C language

10. Field-specific learning outcomes in terms of knowledge, skills and social competences

A student who completed the course:	reference to field-specific learning outcomes
-------------------------------------	---

KNOWLEDGE

EU01 Knows and understands the terms: algorithm, computer program.	K_W07, K_W11
--	--------------

EU02 Knows and understands the terms: programming language, source code, compiler.	K_W07, K_W11
--	--------------

EU03 Knows and understands the terms: data type, constant, variable, identifier, operator.	K_W07, K_W11
--	--------------

EU04 Knows and understands the principle why the aesthetics of code and the choice of identifiers matter even though they do not affect the performance of a program.	K_W11
---	-------

SKILLS

EU05 Can create console programs that perform simple mathematical calculations based on user input.	K_U02, K_U15
---	--------------

EU06 Can select an appropriate data type for declarations of variables used in his/her program.	K_U02, K_U15
---	--------------

EU07 Can create programs, the course of which depends on decisions made by the user or other circumstances verified in the course of program operation.	K_U02, K_U15
---	--------------

EU08 Can use loop instructions in written programs to perform the same actions over and over again.	K_U02, K_U15
---	--------------

EU09 Can use arrays to store multiple data of the same type and operate on this data in programs.	K_U02, K_U15
---	--------------

EU10 Can use subroutines in a written program to decompose a complex algorithm.	K_U02, K_U15
EU11 Can use files to read and write data in programs.	K_U02, K_U15
SOCIAL COMPETENCES	
EU12 Is ready to take a responsible attitude towards the work of a programmer by understanding the consequences that may result from errors found in programs	K_K04
11. Course content	
Course delivery method – lectures/classes/laboratories/practical classes	
<p>Lectures:</p> <ol style="list-style-type: none"> 1) Basic concepts. The first program. Conditional statement. 2) Declarations of variables. Numerical data types. Loops. 3) Retrieving data from the keyboard. Subtitles and characters. Selection instructions. 4) Single and multi-dimensional arrays. 5) Subroutines. Local and global variables. 6) Subroutines. Subroutine parameters. 7) Reading and writing data to a file. 8) Defining custom types. Complex types. 9) Static and dynamic variables. 10) Errors in programs. <p>Laboratories:</p> <ol style="list-style-type: none"> 1) Introduction to the editor and compiler. First program - calculating the perimeter and area of a rectangle. 2) Conditional instruction - program to calculate volume and total area of selected solids. 3) Loops - a program to draw selected geometric figures in the console. 4) Selection instruction - a console calculator with four operations and a text menu. 5) Captions - a program that stores, displays, and combines captions. 6) One-dimensional arrays - a program to perform calculations on a sequence of numbers. 7) Two dimensional arrays - a program to perform calculations on arrays. 8) Subroutines - a program that displays decorative messages on the screen. 9) Subroutines - a program that performs simple calculations (e.g., multiplication). 10) Files - a program that reads and writes text to a file. 11) Files - a program that reads and writes numbers to a file. 12) Complex types - a phone book. 13) Pointers - the "Guess the Number" game. 	
12. Teachingtools and methods	
1. Lectures in the form of presentations	
2. Individual application development using Code::Blocks or CLion	
3. Microsoft Teams platform	
4. Office hours	
13. Assessment method (component, final)	
1) Each laboratory class begins with an "entrance test".	
2) Each program written by the student in the laboratory is evaluated.	
3) In the middle of the semester there is a lecture test.	
4) During the semester, students shall write a program independently at home, which shall be evaluated at the end of the semester.	
5) At the end of the semester there is a lecture test.	

14. Student workload	
Form of activity	Number of hours
1. Classes with direct participation of the teacher and office hours	70
2. Student workload	30
	sum 100
	NUMBER OF ECTS CREDITS 4
15. Reference books	
Primary:	
1) Stephen Prata, Język C. Szkoła programowania. Wydanie VI, Helion, 2016.	
2) Marek Tłuczek, Programowanie w języku C. Ćwiczenia praktyczne. Wydanie II, Helion, 2011.	
3) Społeczność Wikibooks, Programowanie w C, Wikibooks, 2010. https://pl.wikibooks.org/wiki/C	
4) Greg Perry, Dean Miller, Język C. Programowanie dla początkujących. Wydanie III, Helion, 2016.	
Secondary:	
1) L. Banachowski, K. Diks, W. Rytter, Algorytmy i struktury danych, Wydawnictwo Naukowe PWN, 2022.	
2) Maciej M. Sysło, Algorytmy, Helion, 2016.	
3) Peter Prinz, Tony Crawford, C in a Nutshell: The Definitive Reference 2nd Edition. O'Reilly Media, 2015.	
4) Brian W. Kernighan, Dennis M. Ritchie, Język ANSI C. Programowanie. Wydanie II. Helion, 2010.	
5) N. Wirth, Algorytmy + Struktury Danych = Programy, WNT, 2004.	
16. Assessment form - details	
Conditions for obtaining course credit: the course ends with a pass/fail.	
<u>Laboratory credit</u>	
Each laboratory class begins with a 5-minute entrance test containing two questions to check the student's preparation for the class. The test may be marked as 0, ½ or 1 point:	
0 - the student comes to class unprepared - the exercise is not completed	
½ - the test is passed	
1 - the test is passed, the grade for the exercise is increased by half a grade	
After the entry test each student writes a programme according to the specifications provided by the teacher. The programme is graded on a scale from 2 to 5 (some programmes are graded from 2 to 6). Students who have not managed to finish the programme in class can finish it at home and hand it in at the next class (for each week of delay, the grade is lowered by half a grade).	
In addition, in the second half of the semester, students write the program at home, completing the assigned project - it is evaluated at the end of the semester on a scale from 2 to 6.	
The final mark for the laboratory is given on the basis of the average of the partial marks, with the reservation that failure to complete more than 3 exercises results in a mark of ndst, and taking into account that the weight of the mark for the project is 3 times higher than the weight of the mark for the regular programme.	
Average required for individual grades:	
< 2.50 - 2.0 (fail)	
2.50 - 3.24 - 3.0 (sufficient)	
3.25 - 3.74 - 3.5 (satisfactory)	
3.75 - 4.24 - 4.0 (good)	
4.25 - 4.74 - 4.5 (very good)	
> 4.74 - 5.0 (excellent)	
<u>Lecture credit</u>	
During the semester students write two tests, which evaluate their knowledge and skills. The duration of each test is 30 minutes. Most questions are open. Points obtained from both tests are added up and on this basis a grade is given at the end of the semester.	
The percentage of points obtained is required for individual grades:	
0% - 50% - 2.0 (fail)	

50% - 59% - 3.0 (sufficient)

60% - 69% - 3.5 (satisfactory)

70% - 79% - 4.0 (good)

80% - 89% - 4.5 (very good)

90% - 100% - 5.0 (excellent)

17. Other details concerning the course

1. Direct information about the issues of classes and a program content is provided by the teacher during classes and during office hours.

2. Classes will be held at AB in Biała Podlaska

3. Classes will be held in accordance with the current schedule

4. Office hours will be held in accordance with the applicable schedule