Course Syllabus

Course from study programme for the cycle: 2023/2024

I. General Information

Course name	Introduction to computer science
Programme	Informatics
Level of studies (BA, BSc, MA, MSc, long-cycle MA)	BA
Form of studies (full-time, part-time)	Full-time studies
Discipline	Informatics
Language of instruction	English

Course coordinator	Dorota Pylak, PhD

Type of class *(use only the types mentioned below)*	Number of teaching hours	Semester	ECTS Points
lecture	30	I	7
tutorial			
classes			
laboratory classes	45	I	
workshops			
seminar			
introductory seminar			
foreign language classes			
practical placement			
field work			
diploma laboratory			
translation classes			
study visit			

Course pre-requisites	Basic computer skills. Searching for information on the Internet.

II. Course Objectives

Familiarize the students with the basics of structural programming in C++.
Presentation of the basic control statements.
Familiarize the students with the GitHub system

III. Course learning outcomes with reference to programme learning outcomes

Symbol	Description of course learning outcome	Reference to programme learning outcome
	KNOWLEDGE	
W_01	The student formulates the scheme of the number conversion between different numerical systems. He knows how to construct the block diagram for a given problem.	K_W01, K_W06, K_W03
W_02	The student knows the syntax of C ++: the conditional statement and the loop instructions. He can design different elements of the application.	K_W01, K_W06
W_03	The student is able to present the syntax of the function, he knows the methods of passing parameters to the functions and knows how to present examples of their use.	K_W01, K_W06,K_W03
W_04	The student can present the definition of the array and the basic functions operating on the arrays.	K_W01, K_W03, K_W06
W_05	The student knows how to define a simple class: its fields, constructors and methods	K_W01, K_W03, K_W06
W_06	The student knows the basics of using the GitHub version management system. Distinguishes between Git and GitHub	K_W04
	SKILLS	
U_01	The student knows how to convert numbers between the different numerical systems.	K_U04, K_U06, K_U08
U_02	The student can write a program which solves the given problem. He can test the solution and rule out possible errors in his reasoning.	K_U07, K_U08, K_U11
U_03	The student can use variables of the different simple types, conditional statements, loops and arrays. He can improve the program by finding more efficient solution.	K_U02, K_U08, K_U11, K_U17
U_04	The student can create a function, select appropriate parameters and determine the result of the function	K_U02, K_U04, K_U11
U_05	The student is able to define a simple class, write a program operating on simple classes and using previously created functions	K_U02, K_U04, K_U11
U_06	The student is able to use the enumeration type	K_U02, K_U04, K_U11
U_07	The student is able to use the GitHub version control system, add new documents and commit changes	K_U04, K_U01
	SOCIAL COMPETENCIES	
K_01	The student is able to express his opinion and formulate a solution to the given problem. He is open to the new solutions. It cares about the readability of the application.	K_K01, K_K02
K_02	The student solves the given problems individually and while working in a group.	K_K02

IV. Course Content

Numerical systems.

Block diagrams- basics.
GitHub version control system.
Simple variable types.
Data loading.
Conditional statement if.
Switch statement. Enum.
For, while and do…while loops.
Functions. Syntax and the use of a function, returning a result by the function, passing arguments to the function by value, and by reference.
Arrays and operations on arrays.
Classes and an introduction to object-oriented programming. Class definition, member functions, constructors, destructors.

V. Didactic methods used and forms of assessment of learning outcomes

Symbol	Didactic methods *(choose from the list)*	Forms of assessment *(choose from the list)*	Documentation type *(choose from the list)*
KNOWLEDGE			
W_01	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test
W_02	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test
W_03	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test
W_04	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test
W_05	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test
W_06	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test
SKILLS			
U_01	Practical classes Design thinking	Exam/Written test	Examination card / written test
U_02	Practical classes Design thinking	Exam/Written test	Examination card / written test
U_03	Practical classes Design thinking	Exam/Written test	Examination card / written test
U_04	Practical classes Design thinking	Exam/Written test	Examination card / written test
U_05	Practical classes Design thinking	Exam/Written test	Examination card / written test
U_06	Practical classes Design thinking	Exam/Written test	Examination card / written test
U_07	Practical classes Design thinking	Exam/Written test	Examination card / written test
SOCIAL COMPETENCIES			
K_01	Discussion, PBL (Problem-Based Learning)	Exam/Written test	Examination card / written test
K_02	Discussion, PBL (Problem-Based Learning)	Exam/Written test	Examination card / written test

VI. Grading criteria, weighting factors.....

To pass a course, the student has to attend a classes and has to pass the tests and the final exam.

- passing classes - colloquia (numerical systems, conditional statements, loops and functions) - 90% of the final grade, student's activity and work during classes - 10% of the final grade.

- written exam - for people who have passed the classes. Detailed conditions of exemption are given to students with each course edition.

Detailed assessment rules are given to the students with each edition of the course.

VII. Student workload

Form of activity	Number of hours
Number of contact hours (with the teacher)	**135**
Number of hours of individual student work	**75**

VIII. Literature

Basic literature
S. Prata, C++ Primer Plus. 5th Edition, Pearson Education , 2011 www.cplusplus.com Jerzy Grębosz, Opus magnum C++11, Helion, 2017 B. Stroustrup, The C++ Programming Language. Addison-Wesley Longman, Amsterdam, 2014 S. B. Lippman, J. Lajoie, C++ Primer, Addison-Wesley Longman, Amsterdam 2012.
Additional literature
N. Dale, Ch. Weems, M. Headington, Programming in C++, 2nd ed., Jones and Bartlett Publishers, Sudbury 2000. N. Wirth, Algorithms + Data Structures = Programs, Prentice-Hall 1976