SYLLABUS OF A MODULE

Polish name of a module	Wprowadzenie do algorytmów i programowania
English name of a module	Introduction to Algorithms & Programming
ISCED classification - Code	0613
ISCED classification - Field of study	*Sztuczna Inteligencja i Data Science*
Languages of instruction	*English*
Level of qualification:	*2*
Number of ECTS credit points	*5*
Examination:	*EW*
Available in semester:	*S*

Number of hours per semester:

Lecture	Exercises	Laboratory	Seminar	E-learning	Project
30	15	30	0	0	0

MODULE DESCRIPTION

MODULE OBJECTIVES

O1. Obtaining knowledge in the basic methods of programming using object-oriented programming languages.

O2. Familiar with programming tools, environment, optimization techniques, methods of adaptation of codes to computing platforms

O3. Obtaining knowledge in the area of developing and implementing selected algorithms

O4. Acquisition by students practical skills to work independently and in a team, develop reports, analyze the results, etc.

PRELIMINARY REQUIREMENTS FOR KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. Knowledge of mathematics.
2. Basics of computer skills.

3. Rational and logical thinking.
4. Ability to perform mathematical operations to solve given tasks.
5. Ability to use various sources of information including manuals and technical documentation.
6. Ability to work independently and in a group.
7. Ability to correctly interpret and present their own actions

LEARNING OUTCOMES

LO 1. able to use an object-oriented programming language

LO 2 – able to develop and implement a given algorithm

LO 3 – able to solve a mathematical problem by developing an appropriate algorithm

LO 4 - able to use programming tools including compilers, debuggers, and profilers,

LO 5 - able to work independently and in a team, develop and analyze reports

MODULE CONTENT

Type of classes – Lecture	**Number of hours**
Lect. 1 Introduction to C++ Programming	2
Lect. 2 Built-in Data Types	2
Lect. 3 Common Strings Operations	2
Lect. 4 Loops, Nested Loops, and Functions	2
Lect. 5 Reference, Parameters, and Pointers	2
Lect. 6 Arrays, and Dynamic Memory Management	2
Lect. 7 File Streams	2
Lect. 8 Structs and Classes	2
Lect. 9 Class Implementation	2
Lect. 10 Functions and Classes Templates	2
Lect. 11 Introduction to Algorithms	2
Lect. 12 Complexity of the Algorithms	2
Lect. 13 Presentation of Selected Algorithms	2
Lect. 14 Performance analysis for Selected Algorithms	2

Lect. 15 Techniques of Optimizations	2
Type of classes – Tutorial	**Number of hours**
Ex1 - Introduction to Programming and Algorithms	1
Ex2 - Bits, Data Types, and Operations	1
Ex3 - Internal Representation of Fixed Point Data Types	1
Ex4 - Internal Representation of Floating Point Data Types	1
Ex5 - Logical Operations	1
Ex6 - Bits Operations	1
Ex7 - Reference, Parameters, and Pointers	1
Ex8 - Multi Dimensional Computation	1
Ex9 - Multi Dimensional Computation	1
Ex10 - Memory Management	1
Ex11 - Computation Management	1
Ex12 - Complexity of the Algorithms	1
Ex13 - Theoretical Performance Models	1
Ex14-15 - Performance analysis for Selected Algorithms	2
Type of classes – Laboratory	**Number of hours**
Lab. 1 Introduction to Compilers, Coding, and Programs Execution	2
Lab. 2 Using Built-in Data Types	2
Lab. 3 Application of Common Strings Operations	2
Lab. 4 Loops, Nested Loops, and Functions	2
Lab. 5 Reference, Parameters, and Pointers	2
Lab. 6 Arrays, and Dynamic Memory Management	2
Lab. 7 File Streams	2
Lab. 8 Structs and Classes	2
Lab. 9 Class Implementation	2
Lab. 10 Functions and Classes Templates	2
Lab. 11 Implementation of Selected Algorithms	2
Lab. 12 Implementation of Selected Algorithms	2
Lab. 13 Implementation of Selected Algorithms	2
Lab. 14 Techniques of Computation Optimizations	2

Lab. 15 Techniques of Memory Optimizations	2
Sum	**45**

TEACHING TOOLS

1. Instructions for laboratories
2. Wide range of algorithm and programming tools
3. Workplaces for students equipped with workstations

WAYS OF ASSESSMENT (F – FORMATIVE, S – SUMMATIVE

F1. Assessment of preparation for laboratory
S1. Assessment of the ability to solve the problems posed and the manner of presentation obtained results - pass mark *

*) in order to receive a credit for the module, the student is obliged to attain a passing grade in all laboratory classes as well as in achievement tests.

STUDENT'S WORKLOAD

L.p.	**Forms of activity**	**Average number of hours required for realization of activity**
1.	**Contact hours with teacher**	
1.1	Lectures	30
1.2	Tutorials	15
1.3	Laboratory	30
1.4	Seminar	0
1.5	Project	0
1.6	Examination teacher during their duty hours	0
	Total number of contact hours with teacher:	75
2.	**Student's individual work**	
2.1	Preparation for tutorials and tests	10
2.2	Preparation for laboratory exercises, writing reports on laboratories	13
2.3	Preparation of project	0
2.4	Preparation for final lecture assessment	0

2.5	Preparation for examination	13
2.6	Individual study of literature	13
	Total number of hours of student's individual work:	49
	Overall student's workload:	125
Overall number of ECTS credits for the module		5 ECTS
Number of ECTS points that student receives in classes requiring teacher's supervision:		3.04 ECTS
Number of **ECTS** credits acquired during practical classes including laboratory exercises and projects:		1.8 ECTS

BASIC AND SUPPLEMENTARY RESOURCE MATERIALS

1. Bruce Eckel, "Thinking in C++: Introduction to Standard C++", Prentice Hall, 2008
2. Bruce Eckel, Thinking In C++: Practical Programming, Prentice Hall, 2009
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, The Mit Press, 2009

MODULE COORDINATOR (NAME, SURNAME, E-MAIL ADDRESS)

Dr hab. inż. Krzysztof Rojek, prof. PCz, krojek@icis.pcz.pl