# Programming Methods

Course description

## Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : obligatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.120.03269.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.MP.OL

Course coordinator

Maciej Ślusarek

Course instructors

Maciej Ślusarek, Iwona Cieślik

| | | |
|---|---|---|
| Period Semester 2 | Form of verification of learning outcomes<br>exam<br>Form of instruction and hours<br>lecture: 30 laboratory exercises: 30 | Number of ECTS credits 6.0 |

## Learning outcomes for the course

| Code | Outcomes in the field of | Field-specific learning outcomes | Verification methods |
|---|---|---|---|

| Code | Outcomes in the field of | Field-specific learning outcomes | Verification methods |
|---|---|---|---|
| Knowledge – The student knows and understands: | | | |
| W1 | knows basic data structures (trees, graphs, hash tables) and methods of their programming implementation | IAN_K1_W04, IAN_K1_W06 | written exam, credit |
| W2 | knows selected techniques of algorithm construction | IAN_K1_W06 | written exam, credit |
| W3 | knows basic techniques of sorting and searching data | IAN_K1_W06 | written exam, credit |
| Skills – The student can: | | | |
| U1 | uses basic data structures to describe simple problems presented in natural language | IAN_K1_U01, IAN_K1_U07, IAN_K1_U08 | written exam, credit |
| U2 | designs and implements algorithms using basic data structures and selected programming techniques | IAN_K1_U03, IAN_K1_U05, IAN_K1_U06, IAN_K1_U07, IAN_K1_U08 | written exam, credit |
| U3 | can write a program in a readable way, test it at a basic level, look for errors and optimize it | IAN_K1_U17 | written exam, credit |
| U4 | can propose a solution for a simple algorithmic problem by choosing the appropriate method for its solution | IAN_K1_U11, IAN_K1_U21 | written exam, credit |
| U5 | can orally and in writing present the development of a solution to a simple problem | IAN_K1_U21 | written exam, credit |
| Social competences – The student is ready to: | | | |
| K1 | can precisely formulate questions that serve to deepen or supplement their own understanding of a given topic | IAN_K1_K01 | written exam, credit |

## ECTS credits balance

| Student activity form | Average number of hours* devoted to completed activity types |
|---|---|
| lecture | 30 |
| laboratory exercises | 30 |
| independent solving of computer tasks | 60 |
| preparation for classes | 30 |
| preparation for exam | 30 |

| | | |
|---|---|---|
| Total student workload | Number of hours 180 | ECTS credits 6.0 |

* hour (lesson) means 45 minutes

## Program content

| No. | Program content | Learning outcomes for the course |
|---|---|---|
| 1. | 1. Computational complexity of algorithms – definition, notation, comparison of complexity functions. The concept of abstract data type (ADT). Basic data structures: list, stack, queue, priority queue, dictionary; array, pointer, cursor implementation. Examples: binary search, topological sorting, memory management. 2. Tree structures: trees, binary trees, representations, elementary and advanced traversal algorithms, binary search tree. 3. Amortized complexity, dynamic arrays, hashing (basics). 4. Graphs: representation, BFS and DFS traversal, connected components, cycles, topological sorting with DFS, Euler cycle. 5. Recursion, conversion to iteration, examples: DFS with stack, backtracking. 6. Divide and conquer method, fast multiplication, merge sort, master theorem (simplified version). 7. Quicksort, variants (Hoare, Lomuto), non-recursive version. 8. Binary heap, heapsort, order statistics, counting tree. 9. Radix sort, lower bound on sorting complexity. 10. Dynamic programming – introduction. 11. Greedy algorithms – introduction. | W1, W2, W3, U1, U2, U3, U4, U5, K1 |

## Extended information

### Teaching methods:

conventional lecture, laboratory exercises

| Type of classes | Credit forms | Course credit conditions |
|---|---|---|

| Type of classes | Credit forms | Course credit conditions |
|---|---|---|
| lecture | written exam | Positive grade from the exam. Admission to the exam under the condition of a positive grade from the laboratory. The final grade is the average of the laboratory grade and the exam. |
| laboratory exercises | credit | Laboratory credit based on credit programs, homework assignments, and tests. |

## Prerequisites and additional requirements

Programming Basics

## Literature

**Required**

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, wydanie III, PWN, 2012

**Additional**

1. L.Banachowski, K.Diks, W.Rytter, Algorytmy i struktury danych, PWN, 2018
2. D. Knuth, Sztuka programowania, tom 1 i 3, WNT, 2002
3. A.V.Aho, J.E.Hopcroft, J.D.Ullman, Projektowanie i analiza algorytmów, PWN 1985, Helion 2003.