**SYLLABUS**

**regarding the qualification cycle from 2024 to 2025**

1. Basic Course/Module Information

|  |  |
| --- | --- |
| Course/Module title | Algorithms and Data Structures II |
| Course/Module code \* |  |
| Faculty (name of the unit offering the field of study) | College of Natural Sciences |
| Name of the unit running the course | Institute of Computer Science |
| Field of study | Computer Science & Computer Science and Econometrics |
| Qualification level  | First degree |
| Profile | Academic |
| Study mode | Full-time |
| Year and semester of studies | Year II, semester III |
| Course type | Major |
| Language of instruction | English |
| Coordinator | Jan Bazan, PhD, DSc |
| Course instructor | Dawid Kosior, MSc, Eng. |

\* - as agreed at the faculty

1.1.Learning format – number of hours and ECTS credits

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Semester(n0.) | Lectures | Classes | Colloquia | Lab classes | Seminars | Practical classes | Internships | others | **ECTS credits**  |
| 1 |  |  |  | 30 |  |  |  |  | 5 |

1.2. Course delivery methods

- conducted in a traditional way

- involving distance education methods and techniques

1.3. Course/Module assessment (exam, pass with a grade)

- pass with a grade

2. Prerequisites

|  |
| --- |
| Discrete Mathematics, Programming Basics |

3. Objectives, Learning Outcomes, Course Content, and Instructional Methods

3.1. Course/Module objectives

|  |  |
| --- | --- |
| O1 | THIS IS AN INTERMEDIATE LEVEL COMPUTER SCIENCE COURSE. IT IS ABOUT THE DESIGN AND ANALYSIS OF ALGORITHMS FOR COMPUTATIONAL PROBLEMS, AND HOW TO THINK CLEARLY ABOUT ANALYSING CORRECTNESS AND RUNNING TIME. |

3.2. Course/Module Learning Outcomes (to be completed by the coordinator)

|  |  |  |
| --- | --- | --- |
| Learning Outcome | The description of the learning outcome defined for the course/module | Relation to the degree programme outcomes |
| LO\_01 | Student knows the main methods of constructing algorithms  |  |
| LO\_02 | Student knows the method of determining the computational complexity of algorithms |  |
| LO\_03 | Student knows the basic data structures |  |
| LO\_04 | Student knows basic algorithms for data structures processing. |  |
| LO\_05 | Student is able to determine the computational complexity of a simple iterative algorithm and recursive |  |
| LO\_06 | Student is able to use in practice of the selected method of constructing algorithms |  |

**3.3. Course content (to be completed by the coordinator)**

1. Lectures

|  |
| --- |
| **Content outline** |
| Computational complexity (time and memory) of algorithms and its determination. |
| Algorithm types. Exact methods of algorithm construction Approximation methods of algorithm construction. |
| Abstract and concrete data structures. |
| Sorting and searching algorithms.  |
| Basic graph algorithms. |

1. Classes, laboratories, seminars, practical classes

|  |
| --- |
| **Content outline**  |
| Determining the computational complexity of algorithms. |
| Constructing exact and approximation algorithms. |
| Constructing data structures. |
| Examples of sorting and searching algorithms.  |
| Examples of graph algorithms. |

3.4. Methods of Instruction

Lecture: a lecture supported by a multimedia presentation

Laboratory: solving exercises

4. Assessment techniques and criteria

4.1 Methods of evaluating learning outcomes

|  |  |  |
| --- | --- | --- |
| Learning outcome | Methods of assessment of learning outcomes (e.g. test, oral exam, written exam, project, report, observation during classes) | Learning format (lectures, classes,…) |
| LO-01 | written test | Classes |
| LO-02 | written test | Classes |
| LO-03 | written test | Classes |
| LO-04 | written test | Classes |
| LO-05 | Exercises to solve with using a computer | Classes |
| LO-o6 | Exercises to solve with using a computer | Classes |

4.2 Course assessment criteria

|  |
| --- |
| GRADING SYSTEM : 2, 3, 3.5, 4, 4.5, 5 |

5. Total student workload needed to achieve the intended learning outcomes

– number of hours and ECTS credits

|  |  |
| --- | --- |
| Activity | Number of hours |
| Scheduled course contact hours | 30 |
| Other contact hours involving the teacher (consultation hours, examinations) | 5 |
| Non-contact hours - student's own work (preparation for classes or examinations, projects, etc.) | 95 |
| Total number of hours | 130 |
| Total number of ECTS credits | 5 |

\* One ECTS point corresponds to 25-30 hours of total student workload

6. Internships related to the course/module

|  |  |
| --- | --- |
| Number of hours | - |
| Internship regulations and procedures | - |

7. Instructional materials

|  |
| --- |
| Compulsory literature: * Anany V. Levitin, Introduction to the Design and Analysis of Algorithms, Addison Wesley
* T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, MIT Press
* Walter Savitch, JAVA, An introduction to Computer Science & Programming, Prentice Hall
 |

Approved by the Head of the Department or an authorised person