**SYLLABUS**

**regarding the qualification cycle FROM 2022TO 2023**

1. Basic Course/Module Information

|  |  |
| --- | --- |
| Course/Module title | Introduction to Programming |
| 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 Mathematics* |
| Field of study | Mathematics |
| Qualification level | First degree |
| Profile | *Academic* |
| Study mode | *Full-time* |
| Year and semester of studies | *1 Year, 2 semester* |
| Course type | *Basic* |
| Language of instruction | English |
| Coordinator | Ewa Rak, PhD |
| Course instructor | Ewa Rak, PhD |

\* - 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** |
| 2 |  | 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 without a grade)

Exam

2. Prerequisites

|  |
| --- |
| Basic information in the field of computer building and operation |

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

3.1. Course/Module objectives

|  |  |
| --- | --- |
| O1 | Preparation for solving problems (including math) with the use of computer science and using computers during math classes. |
| O2 | To familiarize students with the problems of algorithmic description of problem solving - searching, noticing and constructing algorithms. |
| O3 | To familiarize students with the basic methods of algorithm design, writing, running and testing programs. Analysis of the calculation results. |

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 | the student knows the basics of computational and programming techniques that can be used to support the work of a mathematician and understands the limitations that arise when using them; | K\_W05 |
| LO\_02 | is able to recognize and specify a problem that can be solved algorithmically, searches for the above-mentioned problems based on practical issues; | K\_U14 |
| LO\_o3 | for a given problem is able to compose and analyze an algorithm compliant with the specification and write it in an appropriate programming language, and then test the computer program written by himself and make the necessary; | K\_ U14 |
| LO\_04 | formulates questions to understand the problem under study and expresses his / her own opinions on theoretical and practical issues in mathematics related to a given problem; | K\_K01 |
| LO\_05 | is ready to present a critical attitude towards the received content, is aware of errors that may appear when arranging algorithms and writing in the programming languages used; | K\_K02 |
| LO\_06 | is ready to recognize the importance of knowledge of arranging algorithms for solving practical problems. | K\_K03 |

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

|  |
| --- |
| Content outline |
| **Preliminary information**  The concept of computer science, information and its processing. Computer science fields. |
| **Algorithms**  The concept of an algorithm. Interpretation of variables in the algorithm, writing of algorithms. Block diagrams. Algorithm testing. Classification of simple algorithms. |
| **Computers**  Functional diagram of a computer, memory organization, arithmometer, control system, internal language. Computer arithmetic. |
| **Algorithmic languages**  The concept of language. Combinatorial grammars. Programming languages. |
| **Program structure on the example of JAVA language**  Relationship between the program and the algorithm. Constants and variables. Basic instructions. Standard functions and expression programming. Domain and set of values of standard functions. Arithmetic and logical operations. Grammar and semantics of expressions. |
| **Compound instructions**  Conditional instructions. Various types of loops. Array Type, File Type. Operations on variables of the above-mentioned types.  **Sub-programs**  The idea of a sub-program, functions. Data exchange between the program and the sub-program. |

3.4. Methods of Instruction

e.g.

*Lecture: a problem-solving lecture/a lecture supported by a multimedia presentation/ distance learning*

*Classes: text analysis and discussion/project work (research project, implementation project, practical project)/ group work (problem solving, case study, discussion)/didactic games/ distance learning*

*Laboratory classes: designing and conducting experiments*

Classes: working in groups and individual - task solving and proving theorems.

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 | Test, observation during classes | class |
| LO\_02 | Test, Project, observation during classes | class |
| LO\_03 | exam, observation during classes | class |
| LO\_o4 | OBSERVATION DURING CLASSES | class |
| LO\_05 | OBSERVATION DURING CLASSES | class |
| LO\_06 | OBSERVATION DURING CLASSES | class |

4.2 Course assessment criteria

|  |
| --- |
| students are Assessed regularly solving tasks writing.  The examination of students' knowledge in an oral form.  Grading score:  3.0 for 50 - 60%, 3.5 for 61 - 70 %, 4.0 for 71 – 80%, 4.5 for 81 – 90%; 5.0 for 91 – 100 % |

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 |
| Non-contact hours - student's own work (preparation for classes or examinations, projects, etc.) | 70 |
| Total number of hours | 100 |
| Total number of ECTS credits | 5 ECTS |

\* 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: **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein**: Introduction to Algorithms, third edition, MIT Press, 2009.Robert Lafore: Data Structures and Algorithms in Java, second edition, SAMS, 2003.  1. Narsimha Karumanchi: Data Structures and Algorithms Made Easy, LIGHTNING SOURCE INC, 2011. |
| Complementary literature:   1. Michael T. Goodrich, Roberto Tamassia: Data Structures and Algorithms in Java, fourth edition, John Wiley & Sons Inc, 2001.   <https://enos.itcollege.ee/~japoia/algorithms/GT/Data%20Structures%20and%20Algorithms%20in%20Java%20Fourth%20Edition.pdf> |

Approved by the Head of the Department or an authorised person