**SYLLABUS**

**regarding the qualification cycle FROM 2023TO 2024**

1. Basic Course/Module Information

|  |  |
| --- | --- |
| Course/Module title | Object - Oriented Programming I |
| 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 | 1st Degree Engineering Studies |
| Profile | General Academic |
| Study mode | Full-time |
| Year and semester of studies | Year I, Semester II |
| Course type | Major engineering |
| Language of instruction | English |
| Coordinator | Wojciech Kozioł, PhD, Eng. |
| Course instructor | Wojciech Kozioł, PhD, 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** |
| 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)

pass with a grade

2. Prerequisites

|  |
| --- |
| Fundamentals of programming |

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

3.1. Course/Module objectives

|  |  |
| --- | --- |
| O1 | *To acquaint students with issues related to the object-oriented programming paradigm.* |
| O2 | *To teach students to think, design and solve problems using objects and relationships between objects.* |
| O3 | *To teach students to create simple programs in Java language.* |
| O4 | *To acquaint students with Java language and NetBeans environment.* |

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 | He knows basic programming constructs and data structures present in the Java language. | K\_W04, K\_W07 |
| LO\_02 | Has basic knowledge about the object oriented programming paradigm and its application; understands such concepts as: class, abstract class, interface, object, encapsulation, inheritance, polymorphism. | K\_W04, K\_W07 |
| LO\_03 | The student is able to accurately specify information technology problems and formulate solutions in the Java language, using the known object-oriented programming techniques. | K\_U11,K\_U12 |
| LO\_4 | The student is able to use basic programming constructs and data structures in Java languages. He understands their advantages and disadvantages and is able to select them properly taking into account the complexity, efficiency and quality of the created solution. | K\_U10, K\_U12 |
| LO\_05 | Is able to create simple applications in Java. | K\_U11 |

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

1. Lectures

|  |
| --- |
| Content outline |
| Java language genesis. Principle of Java technology. |
| General form of a Java program. Working in NetBeans environment. |
| Identifiers, types, variables, expressions, input-output operations and comments. |
| Program control flow and ways of its modification. Iteration and recursion in Java. |
| Text variables and string operations in Java. |
| Objects, classes, fields and methods, constructors in Java. |
| Hermetization of components (based on Java language). |
| Inheritance (based on Java). |
| Polymorphism (based on Java). |
| Abstract classes and interfaces (based on Java language). |
| Static class components: static fields, static methods, static initializers (based on Java language). |
| Exceptions in Java language. |
| Raw and generic types in Java. |
| Raw and generic collections in Java. |
| Streams and files in Java. |

1. Classes, tutorials/seminars, colloquia, laboratories, practical classes

|  |
| --- |
| Content outline |
| Introduction to NetBeans environment |
| Creating and running simple programs in NetBeans environment |
| Program control flow in Java - loops, recursions. Conditional expressions. |
| Operations on string variables in Java. |
| Creating classes and objects in Java. |
| Encapsulation in Java. |
| Inheritance in Java. |
| Polymorphism in Java. |
| Abstract classes and methods and interfaces in Java. |
| Members of static classes in Java. |
| Exception throwing and exception handling in Java. |
| Generalized types in Java. |
| Use of collections in Java. |
| Handling of streams in Java. |

3.4. Methods of Instruction

*Lecture: A lecture supported by a multimedia presentation*

*Laboratory classes: Create computer programs based on the content of the tasks in the lab handouts.*

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 | oral answer | lectures |
| LO-o2 | oral answer | lectures |
| LO-o3 | Colloquium | Laboratory classes |
| LO-o4 | Colloquium | Laboratory classes |
| LO-o5 | Colloquium | Laboratory classes |

4.2 Course assessment criteria

|  |
| --- |
| Lectures:  Passing oral answer with at least 50% correct answers.  Laboratory classes:  All colloquia must be passed with a positive mark.  A: The average of the grades earned on all colloquia and class activities yields a grade of A.  *B: The average of the grades earned on all colloquia and class activities yields a grade of B.*  *C: The average of the grades earned on all colloquia and class activities yields a grade of C.*  *D: The average of the grades earned on all colloquia and class activities yields a grade of D.*  *E: The average of the grades earned on all colloquia and class activities yields a grade of E.* |

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.) | 90 |
| Total number of hours | 125 |
| 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:   * Walter Savitch, *JAVA, An introduction to Computer Science & Programming*, Prentice Hall, 1999 * Cay S. Horstmann, Gary Cornell, *Core Java 2 Advanced Features,* Pearson Education (US), 2004 |
| Complementary literature:   * Bruce. Eckel, *Thinking in Java*, Prentice Hall, 2006. |

Approved by the Head of the Department or an authorised person