**SYLLABUS**

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

1. Basic Course/Module Information

|  |  |
| --- | --- |
| Course/Module title | Object - Oriented Programming 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  | 1st Degree Engineering Studies |
| Profile | General Academic |
| Study mode | Full-time |
| Year and semester of studies | Year II, Semester III |
| 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**  |
| 3 |  |  |  | 45 |  |  |  |  | 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

|  |
| --- |
| Fundamentals of programming, Object - Oriented Programming I  |

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

3.1. Course/Module objectives

|  |  |
| --- | --- |
| O1 | *To teach students to develop simple programs in C# language.* |
| O2 | *To acquaint students with the C# language and the Visual Studio environment.* |
| O3 | *To acquaint students with the issues of creating graphical user interfaces and accessing relational database in Java.* |
| O4 | *The student acquires the ability to develop applications with a graphical user interface in Java and C#.* |
| O5 | *Acquire skills to develop applications to access relational databases in Java.* |

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 and C# languages. | K\_W04, K\_W07 |
| LO\_02 | Has a basic understanding of how to create graphical user interfaces and interfaces to connect to relational databases and how to use them in Java. | K\_W04, K\_W07 |
| LO\_03 | The student is able to accurately specify information technology problems and formulate solutions in the Java and C# languages, 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 C# 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 and C#. | K\_U11 |
| LO\_06 | The student is able to use standard libraries and interfaces of Java and C# languages to create software in the object oriented paradigm, including libraries to create graphical user interfaces and to connect to relational databases. | K\_U14 |

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

1. Lectures

|  |
| --- |
| **Content outline** |
| Graphical user interface - SWING library. |
| Graphical user interface - JavaFX library. |
| Connecting to relational databases using JDBC. |
| Object-relational mapping in Java using Hibernate. |
| Introduction to JSP technology. |
| Introduction to .NET platform and C# language. Comparison of .NET and Java technologies. Using Microsoft Visual Studio .NET. |
| Basic data types, expressions, instructions, single and multi-dimensional arrays, pointers in C# language. |
| Object-oriented programming in C#: creating classes and objects, fields and methods, encapsulation, inheritance, abstraction, polymorphism. |
| Exception handling in the C# language. |
| Operations on streams in C#. |
| Collections in C# language. |
| Creating graphical user interface using standard .NET components. |
| Handling mouse and keyboard events, 2D motion graphics programming in C#. |

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

|  |
| --- |
| **Content outline**  |
| Creating GUI in Java using Swing library. |
| Creating GUI in Java language using JavaFX library. |
| Handling relational databases in Java using JDBC interface. |
| Handling relational databases using Hibernate framework. |
| Creating simple applications in JSP technology. |
| Introduction to Visual Studio environment. |
| Data types, conditional expressions and control instructions, pointers in C# language. |
| Creating classes, objects and structures in C# language. |
| Hermetization, inheritance and composition in C#. |
| Polymorphism and abstraction (abstract classes and interfaces) in the C# language. |
| Exception handling in the C# language. |
| Operations on streams in the C# language. |
| Collection handling in the C# language. |
| GUI creation and event handling in the C# language. |

3.4. Methods of Instruction

*Lecture: A lecture supported by a multimedia presentation*

*Laboratory classes: Creation of computer programs based on the content of tasks included in laboratory handouts, practical project in the form of software implemented mainly as homework largely independent but consulted.*

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 EXAM | lectures |
| LO-o2 | ORAL EXAM | lectures |
| LO-o3 | ORAL EXAM | lectures |
| LO-o4 | PROJECT, COLLOQUIUM | laboratory classes |
| LO-o5 | PROJECT, COLLOQUIUM | laboratory classes |
| LO-o6 | PROJECT, COLLOQUIUM | laboratory classes |

4.2 Course assessment criteria

|  |
| --- |
| Lectures: Passing grade on the exam. The exam applies to knowledge of Object-Oriented Programming I and Object-Oriented Programming II. - The student receives a grade of A from the exam if answering at least 90% of the exam questions on the exam. - The student receives a grade of B from the exam if he/she answers at least 80% of the exam questions on the. - The student receives a grade of C from the exam if he/she answers at least 70% of the exam questions on the. - The student receives a grade of D from the exam if he/she answers at least 60% of the exam questions on.- The student receives a grade of E on the exam if they answer at least 50% of the exam questions on.Laboratory classes:Passing grade on the project in Java and all C# colloquia.A: The average of the grades earned on the project and the colloquia yields a grade of A.*B: The average of the grades earned on the project and the colloquia yields a grade of B.**C: The average of the grades earned on the project and the colloquia yields a grade of C.**D: The average of the grades earned on the project and the colloquia yields a grade of D.**E: The average of the grades earned on the project and the colloquia 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 | 45 |
| Other contact hours involving the teacher (consultation hours, examinations) | 5 |
| Non-contact hours - student's own work (preparation for classes or examinations, projects, etc.) | 75 |
| 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:* Paul Fischer, *Introduction to Graphical User Interfaces with Java Swing*, Addison-Wesley (November 26, 2004)
* Lawrence PremKumar, Praveen Mohan, *Beginning JavaFX 1st ed. Edition*, Apress, 2010
* Kishori Sharan, *Java APIs, Extensions and Libraries: With JavaFX, JDBC, jmod, jlink, Networking, and the Process API 2nd Edition*, Apress, 2018
* Christian Bauer, Gavin King, and Gary Gregory, *Java Persistence with Hibernate, Second Edition*, Manning, 2015
* Adam Seebeck, *C# Fundamentals - C# 10 and .NET 6 using Visual Studio 2022: Course in a book*, unQbd, 2021
* Huw Collingbourne, *The Little Book Of C# Programming: Learn To Program C-Sharp For Beginners*, Bitwise Books, 2019
 |
| Complementary literature: * Bruce Eckel, *Thinking in Java*, Prentice Hall, 2006.
* https://hibernate.org/
 |

Approved by the Head of the Department or an authorised person