#### card of course

|  |  |
| --- | --- |
| Subject name | * + - 1. **Programming**
 |

**1. The placement of the subject in the study system**

|  |  |
| --- | --- |
| 1.1. Field of study | **Computer science** |
| 1.2. Form and path of study | **Full-time/Part-time** |
| 1.3. Level of education | **First-cycle studies** |
| 1.4. Study profile | **Practical** |

|  |  |
| --- | --- |
| 1.5. Specialty | **-** |
| 1.6. Subject Coordinator | **dr inż. Kamil Żyła** |

**2. General characteristics of the subject**

|  |  |
| --- | --- |
| 2.1. Belonging to a subject group | **Directional/Practical** |
| 2.2. Number of ECTS | **4** |
| 2.3. Language of lectures | **English** |
| 2.4. Semesters in which the subject is taught | **III** |
| 2.5.Criteria for selecting course participants | **-** |

1. **learning outcomes and course delivery**
	1. **Subject Objectives**

|  |  |
| --- | --- |
| **No.** | **Subject Objectives** |
|
| C1 | To provide the student with knowledge about programming constructs that can be used to acquire, analyze and process data. |
| C2 | To provide students with practical knowledge and skills in algorithms and programming in Python and C++. |
| C3 | Acquiring the ability to use programming paradigms to create computer programs, especially the object-oriented paradigm. |
| C4 | Development of abstract thinking and programming thinking. |

* 1. **Subject-specific learning outcomes, divided into knowledge , skills and competences , with reference to the directional learning outcomes**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description of subject learning outcomes** | **Reference to directional effects****learning (symbols)** | **Method of implementation (mark "X")** |
| **ST** | **NST** |
| **Classes at the University** | **Activities on the platform** | **Classes at the University** | **Activities on the platform** |
| After passing the course, the student knows and understands **the knowledge** |
| W1 | syntax and semantics of C++ and Python in terms of object orientation, collections, file usage, persisting data in a computer system, exceptions, program-user interaction, as well as the structural paradigm mechanisms present in these languages | INF\_W08INF\_W20 | X |  |  | X |
| W2 | programming constructs for data structures in C++ and Python and how to use them in practice | X |  |  | X |
| W3 | the essence and process of developing an algorithm that solves a programming problem | X |  |  | X |
| After passing the course, the student is **able** to: |
| U1 | construct programs in the language Python and C++ using the structural paradigm mechanisms present in these languages | INF\_U15 INF\_U17 INF\_U19 | X |  | X |  |
| U2 | program object-oriented in Python and C++, using object-oriented features | X |  | X |  |
| U3 | use libraries and read technical documentation to develop Python and C++ programs that perform data operations and interact with the user | X |  | X |  |
| After completing the course, the student is ready to take part in **social competences.** |
| K1 | dividing the programming problem into tasks (stages) and then defining the priorities of these tasks before starting to implement them | INF\_K04 | X |  | X |  |

**3.3. Forms of teaching and their number of hours - Full-time studies (ST), Part-time studies (NST)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Path** | **Lecture** | **Exercises** | **Design** | **Workshop** | **Laboratory** | **Seminar** | **Lecturer** | **Classes conducted using distance learning methods and techniques in the form of a lecture** | **Other** | **ECTS points** |
| **ST** |  |  |  |  | 40 |  |  | 30 |  | 4 |
| **NST** |  |  |  |  | 15 |  |  | 10 |  | 4 |

**3.4. Content of education** (separately for each form of classes: (W, ĆW, PROJ, WAR, LAB, LEK, OTHER). It should be marked (X) how the given content will be implemented (classes at the university or classes on the e-learning platform conducted using distance learning methods and techniques)

**TYPE OF CLASS: LECTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Content of the course** | **Reference to subject-specific learning outcomes** | **Method of implementation (mark "X")** |
| **ST** | **NST** |
| **Classes at the University** | **Activities on the platform** | **Classes at the University** | **Activities on the platform** |
| **1.** | Introduction to programming. Familiarization with programming environments. Discussion of the course completion process. | **W3** | **X** |  |  | **X** |
| **2.** | Structural mechanisms present in C++ and Python . Data types, functions, operators, order of operations, control statements, library imports. | **W1** | **X** |  |  | **X** |
| **3.** | Arrays and data structures in C++ and Python . Passing arrays and structures to functions. The concept of a pointer to a memory area. | **W1, W2** | **X** |  |  | **X** |
| **4.** | Classes and objects in Python and C++. Defining class attributes and methods. Creating class objects. | **W1, W2** | **X** |  |  | **X** |
| **5.** | Collections of objects in Python and C++. Design and implementation of algorithms that operate on collections of objects. | **W1, W2, W3** | **X** |  |  | **X** |
| **6.** | String operations in Python and C++. | **W1, W2** | **X** |  |  | **X** |
| **7.** | Data persistence programs in Python and C++. | **W1, W2, W3** | **X** |  |  | **X** |
| **8.** | Interaction between a Python or C++ program and its user. | **W1, W2, W3** | **X** |  |  | **X** |
| **9.** | Summary of classes and discussion of grades. |  | **X** |  |  | **X** |

**TYPE OF CLASS: LABORATORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Content of the course** | **Reference to subject-specific learning outcomes** | **Method of implementation (mark "X")** |
| **ST** | **NST** |
| **Classes at the University** | **Activities on the platform** | **Classes at the University** | **Activities on the platform** |
| **1.** | Introduction to the course topics – languages, tools, method of obtaining credit. | **U1-U3, K1** | **X** |  | **X** |  |
| **2.** | Writing programs that use the structural mechanisms present in C++ and Python , including functions, operators, order of operations, control statements, and library imports. | **U1, U3, K1** | **X** |  | **X** |  |
| **3.** | Writing programs using arrays and data structures in C++ and Python . | **U1, U3, K1** | **X** |  | **X** |  |
| **4.** | Writing programs that use user classes and their objects in Python and C++. | **U1-U3, K1** | **X** |  | **X** |  |
| **5.** | Writing programs that operate on collections of objects in Python and C++. Practical aspects of algorithmization and code refactoring . | **U1-U3, K1** | **X** |  | **X** |  |
| **6.** | Writing string programs in Python and C++. | **U1-U3, K1** | **X** |  | **X** |  |
| **7.** | Writing data persistence programs in Python and C++. | **U1-U3, K1** | **X** |  | **X** |  |
| **8.** | Implementing the interaction between a Python or C++ program and its user. | **U1-U3, K1** | **X** |  | **X** |  |
| **9.** | Summary of classes and discussion of grades. |  | **X** |  | **X** |  |

**3.5. Methods of verifying learning outcomes** (indication and description of methods of conducting classes and verification of achievement of learning outcomes and method of documentation)

Verification methods:

Lecture – written paper – 100% of final grade.

Laboratory – colloquium – practical tasks to be completed within a specified time – 100% of the final grade.

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject Effects** | **Teaching methods** | **Methods of verifying learning outcomes** | **Documentation methods** |
| **KNOWLEDGE** |
| **W1-W3** | Multimedia presentation, discussion | A written work containing content related to the topics included in the syllabus | Archived written work |
| **SKILLS** |
| **U1-U3** | Practical classes using computer stations | Colloquium - practical tasks related to the topics included in the syllabus | Archived solutions to practical tasks |
| **SOCIAL COMPETENCES** |
| **K1** | Practical classes using computer stations | Colloquium - practical tasks related to the topics included in the syllabus | Archived solutions to practical tasks |

**3.6. Assessment criteria for the achieved learning outcomes**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Learning effect** | **For a grade of 3 or " zal ."****the student knows and understands/is able to/is ready to** | **For a grade of 3.5, the student knows and understands/is able to/is ready to** | **For a grade of 4, the student knows and understands/is able to/is ready to** | **For a grade of 4.5, the student knows and understands/is able to/is ready to** | **For a grade of 5, the student knows and understands/is able to/is ready to** |
| W | 51-60% of knowledge indicated in learning outcomes | 61-70% of knowledge indicated in learning outcomes | 71-80% of knowledge indicated in learning outcomes | 81-90% of knowledge indicated in learning outcomes | 91-100% of knowledge indicated in learning outcomes |
| U | 51-60% of skills indicated in learning outcomes | 61-70% of skills indicated in learning outcomes | 71-80% of skills indicated in learning outcomes | 81-90% of skills indicated in learning outcomes | 91-100% of skills indicated in learning outcomes |
| K | 51-60% of skills indicated in learning outcomes | 61-70% of skills indicated in learning outcomes | 71-80% of skills indicated in learning outcomes | 81-90% of skills indicated in learning outcomes | 91-100% of skills indicated in learning outcomes |

**3.7. Literature**

**Basic**

1. Sebesta Robert W, Concepts of programming languages, Pearson Education, Harlow, 2016
2. Baka Benjamin, Python Data Structures and Algorithms. Improve application performance with graphs, stacks, and queues, Packt Publishing Birmingham, 2017
3. Luciano Ramalho, "Zaawansowany Python, wyd. 2. Przejrzyste, zwięzłe i efektywne programowanie", Promise, 2022
4. Jerzy Grębosz, "Opus magnum C++. Programowanie w języku C++. Wydanie III poprawione", Helion, 2024

**Supplementary**

1. Scott Michael Lee, Programming language pragmatics, Morgan Kaufmann Publishers, San Francisco, 2015
2. Pierce Benjamin C., Types and programming languages, MIT Press, London, 2002
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, „Wprowadzenie do algorytmów”. Wyd. 2. Wydawnictwo Naukowe PWN, 2024.

**4. student workload - ECTS points balance**

|  |  |
| --- | --- |
| **Types of student activity** | **Student Load** |
| **ST** | **NST** |
| **Classes requiring direct contact between the student and the academic teacher at the university premises** | **70** | **25** |
| Classes included in the study plan | 70 | 25 |
| **Student's own work** | **30** | **75** |
| Ongoing preparation for classes, preparation of project work/presentations/etc. | 15 | 45 |
| Preparation for passing classes | 15 | 30 |
| **TOTAL STUDENT HOURLY LOAD** | **100** | **100** |
| **Number of ECTS points** | **4** | **4** |

|  |  |
| --- | --- |
| Last change date | 30/09/2024 |
| The changes were introduced | INF Education Quality Team |
| The changes were approved | Arkadiusz Gwarda, M.A. |