#### card of course

|  |  |
| --- | --- |
| Subject name | Programming paradigms |

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. Tomasz Giżewski |

2. General characteristics of the subject

|  |  |
| --- | --- |
| 2.1. Belonging to a subject group | Directional/Practical |
| 2.2. Number of ECTS | 2 |
| 2.3. Language of lectures | Polish |
| 2.4. Semesters in which the subject is taught | IV |
| 2.5.Criteria for selecting course participants | - |

1. Learning outcomes and course delivery
   1. Subject Objectives

|  |  |
| --- | --- |
| No. | Subject Objectives |
|
| C1 | Familiarization with declarative, imperative, logical, functional, procedural, structural, and object-oriented paradigms. |
| C2 | Practical application of the logical paradigm on the example of the Prolog language. |
| C3 | Practical application of the functional paradigm on the example of the Haskell language . |
| C4 | Practical application of the structural, procedural and object-oriented paradigms using the Python language as an example . |

* 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 | Familiar with a variety of programming paradigms, including declarative, imperative, logical, functional, procedural, structured, and object-oriented, and understands their application in various contexts. | INF\_W08 | X |  | X |  |
| W2 | Knows the basics of the Prolog, Haskell , and Python programming languages , understanding their distinctive features and relationships to the appropriate paradigms. | X |  | X |  |
| After passing the course, the student is **able** to: | | | | | | |
| U1 | Is able to analyze programming problems and select the appropriate programming paradigm depending on the requirements and specifics of the task. | INF\_U17  INF\_U19  INF\_U20 | X |  | X |  |
| U2 | Can solve logical programming problems using the Prolog language, applying logical rules and inference principles. | X |  | X |  |
| U3 | Can solve functional programming tasks in Haskell , using the mechanisms of expressions and functions. | X |  | X |  |
| U4 | Can create structured, procedural, and object-oriented programs in Python , using appropriate techniques and tools. | X |  | X |  |
| After completing the course, the student is ready to take part in **social competences.** | | | | | | |
| K1 | Understands the importance of choosing the appropriate programming paradigm in the context of solving real-world problems and the impact of this choice on the quality and effectiveness of solutions. | INF\_K02 | X |  | X |  |
| K2 | Is aware of the continuous development of programming technologies and the need to constantly expand knowledge and skills in order to adapt to new trends and requirements in the IT industry. | 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 ………………. | Other | **ECTS points** |
| **ST** |  |  |  | 30 |  |  |  |  |  | 2 |
| **NST** |  |  |  | 15 |  |  |  |  |  | 2 |

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: WORKSHOP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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. | Familiarization with declarative paradigms – analysis | W1, U1, K1 | X |  | X |  |
| 2. | Familiarization with imperative paradigms – analysis | W1, U1, K1 | X |  | X |  |
| 3. | Getting to know the logical paradigm -solving tasks | W1, U2, K1 | X |  | X |  |
| 4. | Familiarization with the functional paradigm - solving tasks | W1, U3, K1 | X |  | X |  |
| 5. | Getting to know the procedural paradigm – solving tasks | W1, U4, K1 | X |  | X |  |
| 6. | Getting to know the structural paradigm – solving tasks | W1, U4, K1 | X |  | X |  |
| 7. | Getting to know the object-oriented paradigm – solving tasks | W1, U4, K1 | X |  | X |  |
| 8. | Getting to know the basics of languages: Prolog, Haskell , Python – practical exercises | W2, U2, U3, U4, K2 | 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)

|  |  |  |  |
| --- | --- | --- | --- |
| Subject Effects | Teaching methods | Methods of verifying learning outcomes | Documentation methods |
| KNOWLEDGE | | | |
| W1-W2 | Informative lecture, case studies, problem solving | Closed test (50% of the final grade) and project (50% of the final grade)  Students complete a project that requires the use of various programming paradigms, such as:  - Creating a module in Prolog (logical).  - Implementation of functions in Haskell (functional).  - Writing a code fragment in Python (procedural, object-oriented). | Graded test sheet and project. |
| SKILLS | | | |
| U1- U4 | Informative lecture, case studies, problem solving | Closed test (50% of the final grade) and project (50% of the final grade)  Students complete a project that requires the use of various programming paradigms, such as:  - Creating a module in Prolog (logical).  - Implementation of functions in Haskell (functional).  - Writing a code fragment in Python (procedural, object-oriented). | Graded test sheet and project. |
| SOCIAL COMPETENCES | | | |
| K1-K2 | Informative lecture, case studies, problem solving | Closed test (50% of the final grade) and project (50% of the final grade)  Students complete a project that requires the use of various programming paradigms, such as:  - Creating a module in Prolog (logical).  - Implementation of functions in Haskell (functional).  - Writing a code fragment in Python (procedural, object-oriented). | Graded test sheet and project. |

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. Baka Benjamin, Python Data Structures and Algorithms. Improve application performance with graphs, stacks, and queues, Packt Publishing, Birmingham, 2017
2. Pierce, Benjamin C. Types and programming languages. London: MIT Press, 2002
3. Gorelick, Micha. Python: programuj szybko i wydajnie, Helion, 2015

**Supplementary**

1. Dokumentacja języka Prolog: <https://www.swi-prolog.org/pldoc/index.html>
2. Dokumentacja języka Haskell: <https://www.haskell.org/documentation/>
3. Dokumentacja języka Python: <https://docs.python.org/3/>

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** | **30** | **15** |
| Classes included in the study plan | 30 | 15 |
| **Student's own work** | **20** | **35** |
| Ongoing preparation for classes, preparation of project work/presentations/etc. | 10 | 20 |
| Preparation for passing classes | 10 | 15 |
| **TOTAL STUDENT HOURLY LOAD** | **50** | **50** |
| **Number of ECTS points** | **2** | **2** |

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