GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Informatics I, PG_00038090		
Field of study	Automation, Robotics and Control Systems		
Date of commencement of studies	October 2025	Academic year of realisation of subject	2025/2026
Education level	first-cycle studies	Subject group	
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	1	Language of instruction	Polish
Semester of study	2	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Faculty of Electrical and Control Engineering -> Wydziały Politechniki Gdańskiej		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Robert Smyk	
	Teachers		

Lesson types and methods of instruction

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	30.0	0.0	15.0	0.0	0.0	45

E-learning hours included: 0.0

Learning activity and number of study hours

Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
Number of study hours	45	10.0	45.0	100

Subject objectives

The general objective of the course is to introduce students to the fundamental concepts of programming and familiarize them with key techniques for data processing and software application development. The course aims to enhance programming skills through learning the basics of **C** and **Python**, as well as presenting the fundamental principles of object-oriented programming and graphical user interfaces (GUI).

Specific Objectives

1. Mastering the basic syntax and structures of **C** and **Python**.
2. Learning to use **arrays, functions, and dynamic memory allocation** in **C**.
3. Understanding the principles of **object-oriented programming** and utilizing **classes and objects** in Python.
4. Developing the ability to **process structured data** and manage files.
5. Implementing basic algorithms and data structures.
6. Creating applications with a **graphical user interface (GUI)**.
7. Understanding methods for **code organization** and modularity.
8. Enhancing **analytical and algorithmic thinking skills**.
9. Utilizing **debugging tools** and optimizing code.
10. Preparing for further studies in advanced programming techniques.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_W06] knows the structure of computers and microprocessors and the tasks of operating systems, has basic knowledge of the basics of computer software, drivers, microprocessor technology, design of simple algorithms and the operation of information networks	Can program the selected sorting or search algorithm.	[SW1] Assessment of factual knowledge
[K6_U04] has the ability to self-educate, among other things, in order to improve professional qualifications	Is able to solve programming tasks using loops and conditions.	[SU4] Assessment of ability to use methods and tools

<table>
<tr><td>Subject contents</td><td>

Fundamentals of Programming in C

- Syntax and structure of C programs.
- Data types, operators, and control statements.
- One-dimensional and multi-dimensional arrays.
- Functions and their applications.
- Dynamic memory allocation and resource management.

Fundamentals of Programming in Python

- Syntax and structure of Python programs.
- Data types, operators, and control statements.
- Lists, tuples, dictionaries, and sets.
- File handling and input/output operations.

Object-Oriented Programming in Python

- Classes, objects, methods, and attributes.
- Inheritance.
- Modules and packages in Python.

Structured Data Processing

- Operations on textual and binary data.
- Handling CSV, JSON, and XML files.
- Basic database operations (SQLite).

Graphical User Interface (GUI) Development

- Introduction to GUI libraries (Tkinter, PyQt, etc.).
- Designing and implementing application windows.
- Handling user events.

Algorithms and Data Structures

- Basic sorting and searching algorithms.
- Lists and arrays.
- Algorithm efficiency.

Code Management and Debugging

- Fundamentals of version control systems.
- Code debugging and error handling.
- Code optimization and best programming practices.

</td></tr>
<tr><td>Prerequisites and co-requisites</td><td></td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>Homeworks</td><td>60.0%</td><td>12.5%</td></tr>
<tr><td>Assessments - theory</td><td>60.0%</td><td>37.5%</td></tr>
<tr><td>Preparation checks</td><td>60.0%</td><td>12.5%</td></tr>
<tr><td>Assessment - practical</td><td>60.0%</td><td>37.5%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>1. B. Kernighan, D. Ritchie, Język C, WNT 1988.
2. Niklaus Wirth, Algorytmy + struktury danych = programy, WNT 1989.
3. William Stallings, Computer Organization And Architecture. Designing for performance. 8th-edition.</td></tr>
<tr><td>Supplementary literature</td><td>none</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>Change the representation of numbers using the decimal, binary, hexadecimal and octal systems
Enumerate the tasks of the operating system
Explain the difference between recursive and iterative way of programming
Describe the rules of algorithm complexity analysis
Present the working of selected sorting algorithms
Present the approaches to programming-in-the-large and the differences between these approaches
Creating programs in C language, to perform give tasks and employ known programming techniques:
- numerical programs
- simple computer game
- string processing</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

Document generated electronically. Does not require a seal or signature.