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 2026	Academic year of realisation of subject	2026/2027
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 -> Faculties of Gdańsk University of Technology		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Robert Smyk	
	Teachers		

Lesson types

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 course aims to introduce students to data processing and the principles of creating computer applications. Students will learn data structures, arrays and linked lists, as well as methods for efficient information processing, including algorithms for organizing and retrieving data.

The course also introduces the principles of data representation and organization in information systems, including data storage in structured XML and JSON formats used for information exchange between systems. Basic principles of graphical user interface (GUI) design are discussed.

Two programming languages are used during the course. C allows students to familiarize themselves with the general principles of computer system operation and data representation, while Python introduces the fundamentals of object-oriented approach and the creation of data processing applications. The course prepares students for further solving more complex problems using computer tools.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[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
[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

Subject contents

Course content – lecture

Introduction to Programming in C
Introduction to the syntax and structure of programs written in the C language. Overview of basic data types, operators, and control statements. Use of one-dimensional and multidimensional arrays and functions in program development. Introduction to dynamic memory allocation and basic resource management.

Introduction to Programming in Python
Basic syntax and program structure in Python. Overview of data types, operators, and control statements. Use of common data structures such as lists, tuples, dictionaries, and sets. File handling and basic input/ output operations.

Object-Oriented Programming in Python
Fundamental concepts of object-oriented programming, including classes, objects, attributes, and methods. Principles of inheritance and code organization using modules and packages.

Structured Data Processing
Operations on text and binary data. Processing structured data stored in formats such as CSV, JSON, and XML. Introduction to basic database operations using SQLite.

Graphical User Interface (GUI) Development
Introduction to libraries for building graphical user interfaces (e.g., Tkinter, PyQt). Designing application windows and implementing user interaction through event handling.

Algorithms and Data Structures
Overview of fundamental data structures, including arrays and lists. Introduction to basic sorting and searching algorithms and basic principles of algorithm efficiency.

Code Management and Debugging
Introduction to version control systems. Techniques for debugging and handling program errors. Basic principles of code optimization and good programming practices.

Course content – laboratory
Laboratory classes focus on the practical application of fundamental programming concepts and data processing techniques. Students learn how to construct basic program structures and implement simple computational procedures.

The course covers the **basic elements of program code**, including the structure of programs and the creation and use of programmer-defined functions. Students practice **passing data to functions**, including passing arguments by value as well as **passing data using pointers and references**. Special attention is given to techniques for **maintaining program state using variables passed by reference**.

During the laboratory sessions, students implement and test **basic data processing operations**, including selected **data searching methods** and simple algorithms. The course also includes **simulation of selected algorithms directly in program code**, allowing students to understand their operation and efficiency.

An important part of the laboratory work is the **presentation of data generated by computations**, both in textual form and through **basic graphical visualization**, enabling interpretation of results and better understanding of program behavior.

Prerequisites and co-requisites

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Assessment of work in class	60.0%	30.0%
Course project	60.0%	40.0%
Short tests and lecture quizzes	60.0%	30.0%

Recommended reading

Basic literature	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.
Supplementary literature	none
eResources addresses	

Example issues/ example questions/ tasks being completed	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
Practical activites within the subject	Not applicable

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