University of Gdańsk

Subject card

Subject name and code	Introduction to programming, PG_00136370		
Field of study	Information Science and Econometrics		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2024/2025
Education level	undergraduate studies	Subject group	Obligatory subject group in the field of study
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	5.0
Learning profile	academic	Assessment form	
Conducting unit	Katedra Informatyki Ekonomicznej -> Faculty of Management		
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. Jerzy Auksztol	
	Teachers		

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	15.0	0.0	45.0	0.0	0.0	60

E-learning hours included: 0.0

Additional information:

Problem-based lecture. Laboratory

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	60	15.0	50.0	125

Subject objectives

Mastering computer programming skills, with particular emphasis on the basic constructs of the C# and Python programming languages and related data analysis techniques.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[IiEL3_W07] The student has advanced knowledge of ethical norms applicable to business, good practices for its conduct and legal regulations for the protection of intellectual property. Has knowledge of the risks and responsibilities associated with the informatization of business processes, knows the principles of netiquette.	Analyzes business rules in the context of their implementation in computer programs constructed in C# and Python languages in accordance with the principles of business ethics, good practices and legal standards regarding intellectual property.	[SW4] test/exam - oral or written [SW5] implementation of a problem task
	[IiEL3_K01] The student understands the need to constantly supplement and deepen acquired knowledge; is open to new ideas and teaching methods.	She/he uses new theoretical and practical solutions supporting the creation of computer programs, such as new data analysis algorithms and, for practical purposes, language models.	[SK4] test/exam - oral or written [SK5] implementation of a problem task
	[IiEL3_K02] The student can communicate with the public on computer science and econometrics topics in and outside the workplace, communicate his knowledge and share his skills through various media.	Is able to present the algorithmic and business challenges he/she has undertaken in a broad discussion forum bringing together many stakeholders, while simultaneously discussing ways of solving them using computer programs.	[SK4] test/exam - oral or written [SK5] implementation of a problem task
	[IiEL3_U02] The student is able to acquire basic information about economic processes and phenomena by direct observation, planned experiment or database queries, and collect and process it using modern information technology tools.	Solves complex algorithmic and business problems in C# and Python. Designs computer programs that solve business problems in accordance with the object-oriented approach. Designs computer programs using database technology.	[SU4] test/exam - oral or written [SU5] implementation of a problem task
	[IiEL3_W04] The student has advanced knowledge of mathematical, statistical, econometric and computer methods of obtaining, processing and analysing data depicting the functioning and development of the national economy and its elements, as well as the phenomena and processes occurring in their environment.	Distinguishes between object-oriented and structured programming principles. Evaluates which programming paradigm is appropriate for solving a given algorithmic and/or business problem. Implements mathematical, statistical, and econometric methods in programs written in C# and Python.	[SW4] test/exam - oral or written [SW5] implementation of a problem task
	[IiEL3_U04] The student can plan, design and program information systems to support the functioning of economic entities.	Writes programs in C# and Python based on reported expectations of business entities. Criticizes computer programs in terms of the effectiveness of the algorithms and data structures used in them.	[SU4] test/exam - oral or written [SU5] implementation of a problem task
	[IiEL3_K03] The student can work in, co-create and manage a team; adapts his/her behavior and conduct to his/her role in it; is ready to take responsibility for the team and bear the consequences; understands the necessity of regularity and consistency in action; is open to other team members and critical of himself.	Is able to work in a self-organizing and mutually learning team of programmers who share acquired knowledge in an open manner while focusing on solving given algorithmic and business tasks.	[SK4] test/exam - oral or written [SK5] implementation of a problem task
	[IiEL3_U03] The student is able to identify institutional-legal and social constraints on the functioning of economic structures and institutions and reflect them in modelling, forecasting and optimization.	Analyzes user requirements and translates them into the functionality of a computer program.	[SU4] test/exam - oral or written [SU5] implementation of a problem task

Subject contents

A. Lecture topics

- discussion of formal languages in relation to syntax, semantics and pragmatics; classification of programming languages according to the paradigm of declarative, functional, logical and object-oriented programming.
- algorithms and their formalization - the concept of an algorithm; techniques for its description on the example of algorithms for linear equations, quadratic equations and the greatest common divisor,
- numerical systems in computers: binary, octal, decimal and hexadecimal systems; representation of integers and floating-point numbers,
- components of declarative programming: data types (basic, arrays, records, files, enumerated and pointer types), variables, operators, expressions, loop and selection instructions; division into modules and subroutines (methods), the body of the subroutine and the method call instruction, passing parameters by value and reference; visibility and lifetime of variables,
- components of object-oriented programming in C# and Python: classes, objects, attributes, methods, constructors, destructors, inheritance, encapsulation, abstract classes, interfaces
- data structures: doubly linked list; stack; queue; hash table; binary tree,
- search and sorting algorithms bubble sort, selection sort, merge sort and quick sort; linear and binary search,
- programming techniques: recursion (recursion) on the example of factorial, Fibonacci sequence and Euclid's algorithm; dynamic programming; divide and conquer technique,
- selected applications: big data, web scraping, machine learning and text mining.

B. Laboratory topics

- discussion of formal languages - discussion of the programming environment; writing and running the first program; presentation of the concepts of syntax, semantics and pragmatics,
- algorithms and their formalization writing algorithms for linear and quadratic equations using one of the description techniques,
- numerical systems and elements of declarative programming implementation of an algorithm for linear and quadratic equations; development of an algorithm and writing a program that performs the mutual exchange of binary, octal, decimal and hexadecimal representations,
- data structures development of basic data structures such as: list, binary tree, stack, queue,
- search and sorting algorithms implementation of selected sorting and search algorithms,
- computational complexity and efficiency comparison of the computational complexity of previously written sorting algorithms,
- selected programming techniques development of a factorial and Fibonacci sequence algorithm using recursive and non-recursive techniques, indicating the advantages and disadvantages of each of them,
- implementation of selected tasks implementing web scraping, machine learning and text mining.

Prerequisites and co-requisites

Basic knowledge of high school level mathematics and the ability to use a desktop computer operating system.

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Semester project	60.0%	30.0%
Colloquium 2	60.0%	15.0%
Colloquium 1	60.0%	15.0%
Exam	60.0%	40.0%

Recommended reading

Basic literature	• Informatyka ekonomiczna. (ang. Business Informatics), pod red. S. Wrycza., J. Maślankowski, Wydawnictwo Naukowe PWN, Warszawa 2019, rozdziały: 7, 8. • Boduch A., Wstęp do programowania w języku C#, Helion, Warszawa 2006. • Lutz M., Ascher D., Python. Wprowadzenie (ang. Python. Introduction), Wydawnictwo Helion, Gliwice 2011. • Müller A. C., Guido S., Machine learning, Python i data science. Wprowadzenie (ang. Machine learning, Python and data science. Introduction), Wydawnictwo Helion, Gliwice 2021.
Supplementary literature	Matthes E., Python. Instrukcje dla programisty (ang. Python. Programmer's Manual), Wydawnictwo Helion, Gliwice 2020.
eResources addresses	Adresy na platformie eNauczanie:

Example issues/ example questions/ tasks being completed	• Preparation of computer programs explaining the functioning of basic C# and Python language constructs. • Preparation of computer programs solving the same algorithmic problem in languages of many paradigms, e.g. structural, object-oriented and functional. • Preparation of computer programs for linear and quadratic equation algorithms, • Preparation of a computer program implementing the mutual exchange of binary, octal, decimal and hexadecimal representations. • Preparation of computer programs using basic data structures such as: list, binary tree, stack, queue, • Preparation of computer programs using selected search and sorting algorithms, • Preparation of computer programs assessing the complexity and computational efficiency of selected sorting algorithms. • Preparation of computer programs implementing factorial and Fibonacci sequence algorithms using recursive and non-recursive techniques, e.g. iterative, indicating the advantages and disadvantages of each of them. • Implementation of selected tasks implementing web scraping, machine learning and text mining.
Work placement	Not applicable

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