University of Gdańsk

Subject card

Subject name and code	Programming Essentials, PG_00178695						
Field of study	Informatics and Econometrics						
Date of commencement of studies	October 2026		Academic year of realisation of subject	2026/2027			
Education level	Bachelor's studies		Subject group	Obligatory subject group in the field of study Subject group related to scientific research 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	credit			
Conducting unit	Department of Business Informatics -> Faculty of Management -> Rector						
Name and surname of lecturer (lecturers)	Subject supervisor		dr hab. Jerzy Auksztol, prof. UG				
	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						
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		4.0	61.0	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_W06] Knows and understands, to an advanced degree, the processes and methods of creating, developing, and providing appropriate conditions for using informatics or statistics tools, particularly those that improve human and organizational functioning	The student knows the principles of object-oriented programming in contrast to structured programming. The student possesses the knowledge required to evaluate which programming paradigm is appropriate for solving a given algorithmic and/or business problem. The student knows mathematical, statistical, and econometric methods applied in programs written in C# and Python.	[SW4] test/exam - oral or written
[IiEL3_U12] Can design and implement IT systems to enhance business operations and effectively utilize modern ICT technologies for management and business communication	The student is able to write programs in C# and Python based on user requirements. The student is able to critically evaluate computer programs in terms of the efficiency of the algorithms and data structures applied.	[SU2] presentation/project/paper/ report [SU4] test/exam - oral or written
[IiEL3_U02] Can select or construct econometrics, informatics or statistics tools and apply them to describe and solve economic and social problems	The student is able to solve complex algorithmic and business problems using C# and Python. The student is able to design computer programs that solve business problems using an object-oriented approach. The student is able to design computer programs utilizing database technologies.	[SU2] presentation/project/paper/ report

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.

<table>
<tr><td>Assessment methods and criteria</td><td colspan="2"><table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>Colloquium</td><td>51.0%</td><td>30.0%</td></tr>
<tr><td>Lecture knowledge test</td><td>51.0%</td><td>40.0%</td></tr>
<tr><td>Semester project</td><td>51.0%</td><td>30.0%</td></tr>
</table></td></tr>
<tr><td rowspan="3">Recommended reading</td><td>Basic literature</td><td>• 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.</td></tr>
<tr><td>Supplementary literature</td><td>Matthes E., Python. Instrukcje dla programisty (ang. Python. Programmer's Manual), Wydawnictwo Helion, Gliwice 2020.</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td colspan="2">• 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.</td></tr>
<tr><td>Work placement</td><td colspan="2">Not applicable</td></tr>
</table>

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