



## Subject card

Subject name and code	Programming Languages, PG_00047657						
Field of study	Informatics						
Date of commencement of studies	October 2026	Academic year of realisation of subject			2027/2028		
Education level	first-cycle 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	2	Language of instruction			Polish		
Semester of study	3	ECTS credits			3.0		
Learning profile	general academic profile	Assessment form			assessment		
Conducting unit	Department of Algorithms and Systems Modelling -> Faculty of Electronics Telecommunications and Informatics -> Faculties of Gdańsk University of Technology						
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. inż. Piotr Mironowicz					
	Teachers	dr Magdalena Godlewska mgr inż. Tomasz Goluch dr hab. inż. Piotr Mironowicz dr hab. inż. Jan Daciuk prof. dr hab. inż. Bogdan Wiszniewski					
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	0.0	15.0	0.0	30
	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	30	3.0		42.0	75	
Subject objectives	Student learned popular programming paradigms and learned to use them.						

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_U03] can design, according to required specifications, and make a simple device, facility, system or carry out a process, specific to the field of study, using suitable methods, techniques, tools and materials, following engineering standards and norms, applying technologies specific to the field of study and experience gained in the professional engineering environment	The student is able to code a solution to a specified problem using the properties of the selected programming paradigm. He recognizes which of the modeling approaches will be most appropriate for the given problem.	[SU1] Assessment of task fulfilment
	[K6_W04] knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	The student is familiar with all important programming paradigms and their importance for the principles of software development. He understands the relationship between high-level languages and the specificity of microprocessors.	[SW1] Assessment of factual knowledge
	[K6_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study	Students are able to apply appropriate programming paradigms (imperative, object-oriented, functional, logical) depending on the project requirements and problem characteristics. They learn to select appropriate programming languages and tools for software implementation, both in the context of user applications and embedded systems. Through the implementation of team projects, they develop the ability to practically use the acquired knowledge in real programming scenarios.	[SU1] Assessment of task fulfilment [SU4] Assessment of ability to use methods and tools
	[K6_W10] knows and understands, to an advanced extent, the parameters, functions, and methods of analysis, design, and optimization of electronic circuits and systems, the definitions of error and measurement uncertainty, measurement methods, including time, frequency, and phase measurements, the properties of converters, and methods of digital signal processing, as well as the basic processes occurring in the life cycle of technical devices, objects, and systems, and methods of supporting processes and functions, specific to the field of study	Students understand and understand the programming paradigms used in the design of IT and technical systems, particularly imperative (structural and object-oriented) and declarative (functional and logical) approaches, along with methods of abstraction, parameterization, and modularization. Students can analyze and design programming solutions using selected languages (e.g., Ada, Smalltalk, Haskell, Prolog), applying appropriate problem representation and algorithm optimization methods in the context of technical system modeling.	[SW3] Assessment of knowledge contained in written work and projects
Subject contents	Course content – lecture 1. Procedural programming. 2. Linear syntax. FORTRAN. 3. Activation records and subroutines. 4. Recursive procedure call. 5. Block syntax. Control flow abstraction. 6. Binding of the names with objects. Range bonds. 7. The parameters of the procedure call. 8. Activation records for languages with recursion. 9. Static and dynamic calls. ALGOL. PASCAL. 10. Restrictions of block languages. 11. Abstraction of data and access protection. 12. Modularization. Modula-2. ADA83, ADA95. 13. Exceptions. Exception handling models. 14. Concurrent procedures. Rendezvous. 15. Object-oriented programming. Objects, classes, hierarchies. 16. Dynamic types. Polymorphism. Smalltalk. C++. 17. Recursive interpreted commands. 18. Symbolic transformation. Tail recursion. 19. LISP. Atoms and lists. 20. Functional programming. Haskell, XSL. 21. Reduction, filtering and casting. 22. Lambda calculus. 23. Memory management in LISP systems. 24. Programming in logic. Prolog.		
Prerequisites and co-requisites			

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Project	50.0%	60.0%
	Tests	50.0%	40.0%
Recommended reading	Basic literature	S. Manganò: XSLT receptury, wyd.2, Helion 2007Cincom Smalltalk Downloads, <a href="http://www.cincomsmalltalk.com/SAXON">http://www.cincomsmalltalk.com/SAXON</a> - The XSLT and XQuery Processor, <a href="http://saxon.sourceforge.net/">http://saxon.sourceforge.net/</a> W.F. Clocksin, W.F., Mellish, C.S.: Prolog Programowanie. Helion 2003Ada Programming, <a href="http://en.wikibooks.org/wiki/AdaSWI-Prolog_downloads">http://en.wikibooks.org/wiki/AdaSWI-Prolog_downloads</a> , <a href="http://www.swi-prolog.org/download.html">www.swi-prolog.org/download.html</a> ADA Core, the GNAT Pro Company, <a href="http://www.adacore.com/home">http://www.adacore.com/home</a> , <a href="https://libre.adacore.com/">https://libre.adacore.com/</a> D. S. Touretzky: Common Lisp: A Gentle Introduction to Symbolic Computation, <a href="http://www.cs.cmu.edu/~dst/LispBook/">http://www.cs.cmu.edu/~dst/LispBook/</a> Z. Huzar, Z. Fryźlewicz, I. Dubielewicz, B. Hnatk: Ada 95, Helion 1998Polski serwis języka Smalltalk, <a href="http://www.objectspace.net/">http://www.objectspace.net/</a>	
	Supplementary literature	<a href="http://en.wikipedia.org/wiki/Programming_paradigm">http://en.wikipedia.org/wiki/Programming_paradigm</a>	
	eResources addresses		
Example issues/ example questions/ tasks being completed			
Practical activities within the subject	Not applicable		

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