GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Programming Techniques, PG_00047806		
Field of study	Informatics		
Date of commencement of studies	October 2020	Academic year of realisation of subject	2020/2021
Education level	first-cycle studies	Subject group	Optional subject group
Mode of study	Part-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	general academic profile	Assessment form	exam
Conducting unit	Department of Microelectronic Systems -> Faculty of Electronics, Telecommunications and Informatics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Maciej Kokot	
	Teachers	dr inż. Maciej Kokot dr hab. inż. Zbigniew Łubniewski	

Lesson types and methods of instruction							
	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						
	Adresy na platformie eNauczanie:						

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	5.0	90.0	125

Subject objectives	Getting familiar with programming techniques. Acquisition of skills for readable and reliable programming. Knowledge of programming support tools - compiler, linker, debugger. Understanding the principles of structured programming and object-oriented, polymorphism, encapsulation, inheritance.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_U41] can produce, test or evaluate software using modern programming platforms, tools, languages and paradigms of different levels, as well as use software packages supporting scientific and research processes as well as business decision-making processes and teamwork	The student is able to plan the course of work needed to produce the software. Can appropriately select development nvironment. He can produce software and test it	[SU4] Assessment of ability to use methods and tools [SU1] Assessment of task fulfilment
	[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	Student knows the difference between structured programming and object-oriented, understand the concept of polymorphism and inheritance. He can program structurally and object-oriented.	[SU4] Assessment of ability to use methods and tools [SU1] Assessment of task fulfilment
	[K6_U08] while identifying and formulating specifications of engineering tasks related to the field of study and solving these tasks, can:n- apply analytical, simulation and experimental methods,n- notice their systemic and non-technical aspects,n- make a preliminary economic assessment of suggested solutions and engineering work n	The student is able to plan the course of work needed to produce the software. Can appropriately select development nvironment. He can produce software and test it	[SU2] Assessment of ability to analyse information [SU1] Assessment of task fulfilment
	[K6_W42] Knows and understands, to an advanced extent, architecture, design principles and methods of hardware and software support for local and distributed information systems, including computing systems, databases, computer networks and information applications, as well as the principles of human cooperation with computers and computer-aided teamwork	Student knows the various programming models. He knows the evolution of the models and programming languages	[SW1] Assessment of factual knowledge
	[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	Student knows the difference between structured programming and object-oriented, understand the concept of polymorphism and inheritance. He can program structurally and object-oriented. The student is able to plan the course of work needed to produce the software. Can appropriately select development nvironment. He can produce software and test it	[SU4] Assessment of ability to use methods and tools [SU2] Assessment of ability to analyse information [SU1] Assessment of task fulfilment

Subject contents	ntroduction. Programs properties. Programming style rules. Code readability. Structural programming. Non-structural exceptions Code programming techniques. Modular programming. Graphical representation of code. Programming tools – compiler, debugger, linker. Header files. Make and its applications. Makefile and its rules. Advanced usage of make. Hierarchical programming as object oriented concept. Inheritance, encapsulation, class members access. Object oriented programming. Polymorphism. Constructors and destructors. Generic programming. Structural versus object-oriented programming.
Prerequisites and co-requisites	

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Project	50.0%	50.0%
Written exam	66.0%	50.0%

Recommended reading	Basic literature	1. D. van Tassel: Praktyka programowania. WNT, Warszawa. 2. Kernighan, Ritchie: Język C. WNT, Warszawa. 3. Stallman, McGrath, Smith: GNU Make. www.gnu.org/software/make/manual/
	Supplementary literature	1. C++ bez cholesterolu. www.intercon.pl/~sektor/cbx/cbx.html
	eResources addresses	
Example issues/ example questions/ tasks being completed		

Work placement	Not applicable