GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Systems Software Design, PG_00048287		
Field of study	Electronics and Telecommunications		
Date of commencement of studies	February 2024	Academic year of realisation of subject	2023/2024
Education level	second-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	1	Language of instruction	Polish
Semester of study	1	ECTS credits	3.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Department of Teleinformation Networks -> Faculty of Electronics, Telecommunications and Informatics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Arkadiusz Harasimiuk	
	Teachers	dr inż. Marcin Narloch dr hab. inż. Marek Wójcikowski mgr inż. Tobiasz Dryjański dr inż. Wojciech Siwicki dr inż. Arkadiusz Harasimiuk dr inż. Bartłomiej Dec dr inż. Arkadiusz Szewczyk dr inż. Jan Schmidt	

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						

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	6.0	39.0	75

Subject objectives	Student describes software design and development for complex IT systems based on Linux (POSIX) and Windows operation systems platforms with the aid of C/C++, C#, Java programming languages.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K7_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, making assessment and critical analysis of the prepared software as well as a synthesis and creative interpretation of information presented with it	Student knows the basics of agile methods of software development. Student knows the tools that enable group work during software development. Student is able to configure and use coding tools, debug tools and software optimization tools.	[SU4] Assessment of ability to use methods and tools
	[K7_U07] can apply advanced methods of process and function support, specific to the field of study	Student knows and understands tools that support software development processes, software debug tools and optimize tools	[SU3] Assessment of ability to use knowledge gained from the subject
	[K7_W08] Knows and understands, to an increased extent, the fundamental dilemmas of modern civilisation, the main development trends of scientific disciplines relevant to the field of education.	Student knows and understands teamwork processes, communications and reporting methods, and distributed work	[SW1] Assessment of factual knowledge
	[K7_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	Sstudent describes design and development of software for complex systems in the field of TI based on operating system platforms, Linux (POSIX) and Windows (.NET) using the programming languages: C / C ++, C #, Java.	[SW1] Assessment of factual knowledge
Subject contents	Design and development of complex IT (Information Technology) systems. Decomposition of processing in complex IT systems. System specification and project documentation in IT systems. Tools for workgroup cooperation, source code revision control, automatic generation of software documentation and bug tracking systems. Software runtime configurations. Tools for software debugging, profiling (performance analysis) and quality control. Role of the processes in IT systems. Process management in operation systems. Threads and thread synchronization. Interproces Communications mechanizms. Access to system services and resources (timers and counters, files and I-O devices, memory and storage management, dynamic and shared librares). Network and inter platform communication in complex IT systems (network sockets, , RPC/RMI, middleware platforms). Application of software libraries and open source projects in system software development.		
Prerequisites and co-requisites	No requirements		

<table>
<tr><td>Assessment methods and criteria</td><td colspan="3"><table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>Project realised during semester</td><td>50.0%</td><td>50.0%</td></tr>
<tr><td>Midterm colloquium.</td><td>50.0%</td><td>50.0%</td></tr>
</table></td></tr>
<tr><td rowspan="3">Recommended reading</td><td>Basic literature</td><td colspan="2">Material prepared by the lecturer in the form of xeroxcopy.</td></tr>
<tr><td>Supplementary literature</td><td colspan="2">Love R., Linux System Programming, O’Reilly, 2013.

Kerrisk M., The Linux Programing Interfacce. A Linux and UNIX® System Programming Handbook, No Starch Press, 2010.

Hart J., Windows System Programming, Addison-Wesley, 2010.</td></tr>
<tr><td>eResources addresses</td><td colspan="2">Adresy na platformie eNauczanie:
Projektowanie Oprogramowania Systemów 2024 - Moodle ID: 36603
https://enauczanie.pg.edu.pl/moodle/course/view.php?id=36603</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td colspan="3"></td></tr>
<tr><td>Work placement</td><td colspan="3">Not applicable</td></tr>
</table>