University of Gdańsk

Subject card

Subject name and code	Advanced Programming Languages, PG_00203607		
Field of study	Informatics		
Date of commencement of studies	October 2026	Academic year of realisation of subject	2026/2027
Education level	Master'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	1	ECTS credits	5.0
Learning profile	academic	Assessment form	exam
Conducting unit	Institute of Informatics -> Faculty of Mathematics, Physics and Informatics -> Rector		
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. Wiesław Pawłowski	
	Teachers		

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	30.0	0.0	30.0	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	0.0	65.0	125

Subject objectives

The purpose of the course is to familiarize the student with the advanced mechanisms found in modern programming languages and their correct and effective use.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFMU2_U05] can apply known algorithms in specific situations, can effectively select the type of algorithm depending on the problem at hand		
[INFMU2_W03] has in-depth knowledge of programming paradigms and architectures and advanced programming constructs; knows current trends in programming languages		
[INFMU2_U03] can design and analyze for correctness and computational complexity, and build algorithms using advanced programming techniques	knows how to solve programming problems using learned programming methods, tools and paradigms	[SU4] test/exam - oral or written

Subject contents

- Application development using hybrid programming methods based on the object-functional approach.
- Creating parallel and distributed systems based on the actor model.

<table>
<tr><td>Prerequisites and co-requisites</td><td>• Familiarity with basic programming concepts and constructs of object-oriented languages, such as methods, classes, inheritance.
• Familiarity with the Java virtual machine environment (JRE/JDK) and related tools.
• Ability to proficiently navigate Windows and Linux operating system environments.</td></tr>
<tr><td>Assessment methods and criteria</td><td><table><tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr><tr><td>oral exam</td><td>51.0%</td><td>20.0%</td></tr><tr><td>programming colloquia</td><td>51.0%</td><td>80.0%</td></tr></table></td></tr>
<tr><td rowspan="3">Recommended reading</td><td>Basic literature</td><td>• M. Odersky, L. Spoon, B. Venners, F. Sommers, Programming in Scala, Fifth Edition, Artima Press, 2021.
• F. Lopez-Sancho, Akka in Action, Second Edition, Manning 2023</td></tr>
<tr><td>Supplementary literature</td><td>• M. Pilquist, R. Bjarnason, P. Chiusano, Functional Programming in Scala, Second Edition, Manning 2023</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td></td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

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