GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Scalabale Object-Oriented Systems Design, PG_00047967		
Field of study	Informatics		
Date of commencement of studies	October 2021	Academic year of realisation of subject	2024/2025
Education level	first-cycle studies	Subject group	Optional subject group 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	4	Language of instruction	Polish
Semester of study	7	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	exam
Conducting unit	Department of Software Engineering -> Faculty of Electronics, Telecommunications and Informatics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr Adam Przybyłek	
	Teachers	dr Adam Przybyłek	

Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	15.0	15.0	0.0	45
	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	45	4.0	51.0	100

Subject objectives	The aim of this course is to introduce students to the principles of object-orientation and design patterns. Each design pattern allows programmers to implement some aspect of system functionality vary independently of other aspects, thereby making a system more robust to a particular kind of change. Moreover, this course discusses aspect-oriented programming as an approach to deal with crosscutting concerns and as a tool for supporting non-invasive evolution of software systems. In addition, the course covers two emergent technologies to process Big Data, i.e. MapReduce and Storm.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[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	A student is able to: design for change by using design patterns.	[SW1] Assessment of factual knowledge
[K6_U43] can analyse date and formulate, apply and assess appropriate formal models and algorithms for solving problems in the field of information systems and applications	A student is able to analyse Big Data using MapReduce.	[SU2] Assessment of ability to analyse information
[K6_W05] Knows and understands, to an advanced extent, methods of supporting processes and functions, specific to the field of study	A student is able to: develop modular software systems according to OO principles, use unit testing to demonstrate program correctness.	[SW1] Assessment of factual knowledge
[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	A student is able to use MapReduce for batch processing of Big Data.	[SU1] Assessment of task fulfilment
[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	A student is able to: leverage AspectJ to implement crosscutting concerns and absorb unanticipated changes that occur due to evolution of business requirements.	[SU1] Assessment of task fulfilment

Subject contents

1. Principles of Object-Orientation – 2h
2. Object-Oriented Analysis and Design – 1h
3. Gang of Four Design Patterns – 3h
4. Prototype-Based Object-Oriented Programming – 1h
5. Aspect-Oriented Programming – 3h
6. Test-Driven Development – 1h
7. Distributed-Computing Architectures – 2h
8. Cloud Computing – 2h

Prerequisites and co-requisites

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Projects	50.0%	50.0%
Labs	50.0%	15.0%
Final exam	50.0%	35.0%

Recommended reading

Basic literature	1. Booch et al.: Object-Oriented Analysis and Design, with Applications. Addison-Wesley, 2007 2. Tegarden et al.: Systems Analysis and Design with UML. Wiley, 2012 3. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston, MA, 1995 4. Maciaszek: Requirements Analysis and Systems Design. Addison-Wesley, 2007 5. Schach: Object-Oriented & Classical Software Engineering. McGraw Hill, New York, 2007
Supplementary literature	1. Fowler: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, 2004 2. Booch et al.: The Unified Modeling Language User Guide. Addison-Wesley, 2005 3. Martin & Odell: Podstawy metod obiektowych. WNT, 1997
eResources addresses	Adresy na platformie eNauczanie:

Example issues/ example questions/ tasks being completed	
Work placement	Not applicable