



COURSE DESCRIPTION

Course code	full-time studies	
	part-time-studies	
Course name	Inżynieria Oprogramowania 1	
Course name in English	Software Engineering 1	
Valid from academic year	2022/23	

PLACEMENT IN THE TEACHING PROGRAM

Field of study	Computer Science
Level of education	1st degree
Studies profile	General
Form and method of teaching classes	Full-time and part-time studies
Specialization	All specializations
Organizational unit responsible for the course	Department of Information Systems
Course coordinator	Arkadiusz Chrobot, PhD
Approved by	Dean of the Faculty of Electrical Engineering, Automatic Control and Computer Science Roman Deniziak, KUT prof., DSc, PhD

GENERAL CHARACTERISTIC OF THE COURSE

Course affiliation	Major Subject	
Course status	Mandatory	
Language	English	
Semester	full-time studies	5th Semester
	part-time-studies	5th Semester
Requirements	Object-Oriented Programming 1, Object-Oriented Programming 2	
Exam (YES/NO)	YES	
ECTS	5	

Course form		lecture	classes	laboratory	project	other
Hours per semester	full-time studies	30		30		
	part-time-studies	18		18		

LEARNING RESULTS

Category	Result Symbol	Learning Results	References to the field of study results
Knowledge	W01	The student knows the issues related to working in a team that develops software.	INF1_W20
	W02	The student knows the issues related to software design and implementation.	INF1_W20
	W03	The student knows the issues related to software verification and validation.	INF1_W20
	W04	The student knows the syntax of UML.	INF1_W20
	W05	The student knows the GoF Design Patterns: creational, structural and behavioral.	INF1_W20
Skills	U01	The student is able to read UML diagrams and use them to model software.	INF1_U20
	U02	The student can apply GoF software design patterns in practise.	INF1_U20
	U03	The student can work together with others on developing software and can take different responsibilities in a team.	INF1_U20
Social competence	K01	The student can objectively assess her or his involvement in a project and knows her or his responsibilities in the team.	INF1_K2
	K02	The student can objectively evaluate her or his ideas and ideas of other members of a team.	INF1_K2
	K03	The student can recognize the requirements of people outside the IT industry, in the context of developed software functionality.	INF1_K1

COURSE CONTENT

Course Form	Content
lecture	<ol style="list-style-type: none"> 1. Introduction to Software Engineering. 2. Unified Modeling Language (structural, behavioral and interaction diagrams). 3. Agile Software Development. 4. Requirements Engineering. 5. Software Architecture. 6. The GoF Design Patterns (Creational, Structural and Behavioral). 7. Software Testing.
laboratory	<ol style="list-style-type: none"> 1. Use Case Modelling. 2. Software Design and Modelling - creational pattern. 3. Software Design and Modelling - structural pattern. 4. Software Design and Modelling - behavioral pattern. 5. Implementation Modelling.

LEARNING RESULTS VERIFICATION METHODS

Result Symbol	Learning results verification methods					
	Oral Exam	Written Exam	Midterm	Project	Report	Other
W01		X				
W02		X				
W03		X				
W04		X				
W05		X				

U01				X		X
U02				X		X
U03				X		X
K01				X		X
K02				X		X
K03				X		X

ASSESSMENT FORMS AND CRITERIA

Course Form	Assessment Form	Assessment Criteria
lecture	exam	The student should obtain at least 50% of points at the exam.
laboratory	passing grade	The student should obtain at least 50% of points by completing assignments at the laboratory classes.

STUDENT'S VOLUME OF WORK

ECTS Balance												
No.	Activity Type	Student Involvement										Unit
		full-time studies					part-time-studies					
		Lec	C	Lab	P	S	Lec	C	Lab	P	S	
1.	Participation in classes according to the schedule	30		30			18		18			h
2.	Other (consultations, exams)	2		2			2		2			h
3.	Total with the direct assist of an academic teacher	64					40					h
4.	Number of ECTS, that the student obtains with the direct assist of an academic teacher	2,56					1,6					ECTS
5.	Hours of unassisted student work	61					85					h
6.	Number of ECTS that student obtains working unassisted	2,44					3,4					ECTS
7.	Practical classes volume of work	30					18					h
8.	Number of ECTS obtained by student at practical classes	1,2					0,72					ECTS
9.	Total student's volume of work expressed in hours	125					125					h
10.	ECTS	5										ECTS

BIBLIOGRAPHY

1. Ian Sommerville, "Software Engineering", Pearson Education Limited, Harlow, 2016
2. Gamma Erich, Helm Richard, Johnson Ralph, Vlissides John, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, Boston, 1995
3. Robert C. Martin, "Clean Architecture: A Craftsman's Guide to Software Structure and Design", Pearson Education, London, 2018
4. Robert C. Martin, "Clean Code: A Handbook of Agile Software Craftsmanship", Pearson Education, London, 2009