



### COURSE DESCRIPTION

Course code	full-time studies	
	part-time-studies	
Course name	<b>Inżynieria Oprogramowania 2</b>	
Course name in English	<b>Software Engineering 2</b>	
Valid from academic year	<b>2022/23</b>	

### PLACEMENT IN THE TEACHING PROGRAM

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

### GENERAL CHARACTERISTIC OF THE COURSE

Course affiliation	<b>Major Subject</b>	
Course status	<b>Mandatory</b>	
Language	<b>English</b>	
Semester	full-time studies	<b>6<sup>th</sup> Semester</b>
	part-time-studies	<b>6<sup>th</sup> Semester</b>
Requirements	<b>Software Engineering 1</b>	
Exam (YES/NO)	<b>NO</b>	
ECTS	<b>1</b>	

Course form		lecture	classes	laboratory	project	other
Hours per semester	full-time studies				<b>15</b>	
	part-time-studies				<b>9</b>	

## 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
project	Students work in a multi-person team in order to develop relatively complex software with the use of contemporary software engineering methods. The problem that the software should solve may be related to company or institution management (like a hotel, a student's dormitory or a transport company). Completing the assignment requires the students to perform typical for software development tasks, like requirements elicitation, software designing, implementation and testing. Each student should learn the internal organization of a software development team and the responsibilities of its members.

## 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		
U02				X		
U03				X		

K01				X		
K02				X		
K03				X		

### ASSESSMENT FORMS AND CRITERIA

Course Form	Assessment Form	Assessment Criteria
project	passing grade	The student should obtain at least 50% of points by completing an assignment that consists of designing and developing software, writing a documentation and defending the project.

### 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				15						9		h
2.	Other (consultations, exams)				1					1		h	
3.	<b>Total with the direct assist of an academic teacher</b>	<b>16</b>					<b>10</b>					h	
4.	<b>Number of ECTS, that the student obtains with the direct assist of an academic teacher</b>	<b>0,64</b>					<b>0,4</b>					ECTS	
5.	<b>Hours of unassisted student work</b>	<b>9</b>					<b>15</b>					h	
6.	<b>Number of ECTS that student obtains working unassisted</b>	<b>0,36</b>					<b>0,6</b>					ECTS	
7.	<b>Practical classes volume of work</b>	<b>15</b>					<b>9</b>					h	
8.	<b>Number of ECTS obtained by student at practical classes</b>	<b>0,6</b>					<b>0,36</b>					ECTS	
9.	<b>Total student's volume of work expressed in hours</b>	<b>25</b>					<b>25</b>					h	
10.	<b>ECTS</b>	<b>1</b>										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