STUDY MODULE DESCRIPTION FORM

Name of the module/subject		Code
Software engineering		**1010331551010330109**
Field of study: **Information Engineering**	Profile of study (general academic, practical): **(brak)**	Year /Semester: **3 / 5**
Elective path/specialty: **-**	Subject offered in: **Polish**	Course (compulsory, elective): **obligatory**
Cycle of study: **First-cycle studies**	Form of study (full-time,part-time): **full-time**	
No. of hours: Lecture: **15** Classes: **-** Laboratory: **15** Project/seminars: **-**		No. of credits: **3**
Status of the course in the study program (Basic, major, other): **(brak)**	(university-wide, from another field): **(brak)**	
Education areas and fields of science and art: **technical sciences**; **Technical sciences**		ECTS distribution (number and %): **3 100%**; **3 100%**

Responsible for subject / lecturer:

dr inż. Andrzej Sikorski
email: andrzej.sikorski
tel. 6653958
Faculty of Electrical Engineering
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Basic knowledge learnt at high school. Student has theoretical and partially practical knowledge concerning: programming constructions, implementation of algorithms, programming styles, verification of software correctness, formal languages, compilers, and platforms.
2	**Skills**	Student is able to find information from professional literature, databases and other sources; he/she can also integrate and correctly interpret the gained information and then to conclude and formulate his/her own opinions.
3	**Social competencies**	Student is aware of an importance of non-technical aspects and then consequences of software engineer's activities; he/she understands is/her responsibility for his/her decisions.

Assumptions and objectives of the course:

The aim of the two-semester course of software engineering is to present an engineering approach to software development. The motivation of Software Engineering approach.

introduced in 2017: the main focus is on asynchronous tasks available on .net platform.

Topics in use case and class diagrams.

The knowledge closely related to laboratory classes i.e. asynchronous design pattern and techniques available in UML modeling.

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. Student has basic knowledge of some Object Oriented Laguage (e.g. C++, java, javascript) - [K_W12]
2. Student is knowledgeable with the state of art and modern trends in software engineering and computing. - [K_W19]

Skills:

1. Student is able to formulate requirements, to build an object model, and assess a simple information system, its functions, and - [K_U16]
2. Student is able to prepare and present a short presentation about his/her own engineering solution. - [K_U04]

Social competencies:

1. Student has a broaded awareness of an importance of non-technical aspects and then consequences of software engineer - [K_K02]
2. Student is aware of his/her responsibility for the work done. He/she points out his/her readiness to work in team and to be responsible for results of tasks realized in team. - [K_K04]

Assessment methods of study outcomes

The content of lectures presented in the first semester of the software engineering course is a subject of an exam after the second semester of this course. After the first semester student's work is assessed on a base of his/her activity in classess and results of a test.

Student's work in laboratories is assessed on the base of partial marks given for each UML diagram and other artefact (requirements document).

Course description

Lectures. Field of software engineering.

(intorduced in 2017)

Asynchronous programming for .NET platform considered as a fundamental design pattern for the modern software architecture and an alternative for multithreading.

Coexistence of multithreading and asynchronous objects.

Reverse engineering in the context of facade/wrapper design pattern.

Object Factory design pattern.

OOP programming in javascript based on the low level functions.

Prototype based inheritance.

Laboratories. exercises in asynchronous design patterns. programming and modelling exercises in class and use case diagrams.

Basic bibliography:

Additional bibliography:

Result of average student's workload

Activity	Time (working hours)
1. Participation in lectures	15
2. Participation in labs	15
3. Constuction of an object model, preparation to pass a test after the first part of software engineering course	30
4. Consultation, test	10

Student's workload

Source of workload	hours	ECTS
Total workload	70	3
Contact hours	20	1
Practical activities	50	1