Course name : **Software engineering**		
Type of study: **Computer science**	Type of study: **Full-time**	Course code: **C5_19**
Course characteristics: **Mandatory within the additional content**	Level: **FIrst (B.Sc.)**	Year: III Semester: V
Type of classes: **lectures, laboratories**	Hours per week: **2 lect (exam), 2 lab**	ECTS points: **5 ECTS**

COURSE GUIDE

AIMS

A1. To acquaint students with the course of the software development process, starting with the strategic phase, by setting requirements on the user side, until the final phase: testing and maintenance.

A2. The acquisition by the the students practical skills in the software design.

PREREQUISITES

1 Knowledge of the mathematics and programming fundamentals.

2 Knowledge of the most popular programming paradigms: procedural and object-oriented.

3 Class programming skills.

4 Knowledge of the types of relationships between classes.

5 The ability to apply appropriate algorithms and data structures to solve tasks.

6 The ability to use various sources of information including instructions and technical documentation.

7 The ability individual work and work in the group.

8 The ability to properly present their own actions.

LEARNING OUTCOMES

EE 1 – has basic theoretical knowledge of the software design.

EE 2 – knows the trends in the field of design and development of the computer systems

EE 3 – is able to use the process of requirements engineering to the identification of the software requirements

EE 4 – is able to provide the software project in the form of documentation and can read this project

EE 5 – is able to use the selected software engineering techniques in the process of the software design with using the appropriate design patterns

EE 6 – knows the basic concepts of verification, validation and to automate testing of the software

EE 7 – is able to write unit tests

EE 8 – knows the methods of the software life cycle management and the basics of programming project management

CONTENT

Lectures	**Hours**
Lect. 1 - Software design models	**2**
Lect. 2 - The requirements engineering process	**2**
Lect. 3 - Introduction to UML	**2**
Lect. 4 - UML Structural diagrams	**2**
Lect. 5 - UML Behavioral diagrams	**2**
Lect. 6 - Object-oriented design	**2**
Lect. 7 - Methods of identifying classes and objects in the designed project	**2**
Lect. 8 - Software architecture	**2**
Lect. 9 - Introduction to the Design Patterns	**2**
Lect. 10 - The selected design patterns	**2**
Lect. 11 - The process of verification and validation of the software	**2**
Lect. 12 - Unit Tests	**2**
Lect. 13 - Agile programming techniques	**2**
Lect. 14 - Basics of manage the software projects	**2**
Lect. 15 - Software configuration management	**2**
LABORATORIES	Hours
Lab. 1 - Introduction to the software engineering	**2**
Lab. 2 - Acquainted with CASE tools (Enterprise Architect)	**2**
Lab. 3 - Specification of the requirements for the sample project	**2**
Lab. 4 - Designing of the use cases based on the requirements specification	**2**
Lab. 5 - The use case scenarios, alternative scenarios and exceptions	**2**
Lab. 6 - The sequence diagrams for the use cases.	**2**
Lab. 7 - The compounds of classes: generalization, association, aggregation and composition.	**2**
Lab. 8 – The class diagram	**2**
Lab. 9 - Creation of the documentation for the source code	**2**
Lab. 10 - The use of the UML diagrams in the software project	**2**
Lab. 11 - Example implementations of the selected design patterns	**2**
Lab. 12 – Using of the design patterns.	**2**
Lab. 13 – Model – View - Controller	**2**
Lab. 14 – Unit Tests	**2**
Lab. 15 – Introduction to the Control Version Systems (CVS)	**2**

TEACHING TOOLS

1. – lectures using multimedia presentations
2. – blackboard and chalk or whiteboards and pens
3. – laboratory guides
4. – reports from laboratory activities
5. – computers with CASE software (ie. Enterprise Architect) and IDE

LITERATURE

E. Gamma, R. Helm, R. Jonhson, J. Vlissides, *Design patterns : elements of reusable object-oriented softwar*, Addison Wesley; 1 edition 1994
R. Miles, K. Hamilton, *Learning UML 2.0*, O'Reilly Media; 1 edition 2006
R. S. Pressman, *Software Engineering: A Practitioner's Approach,* McGraw-Hill Higher Education; 7 edition 2009
I. Sommerville, *Software Engineering, Pe*arson; 9 edition 2010

G.Booch, J. Rumbaugh, J. Jacobson, *The Unified Modeling Language User,* Addison Wesley; 2 edition 2005

M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language (Object Technology Series)*, Addison Wesley; 3 edition 2003

S. McConnell, *Code Complete: A Practical Handbook of Software Construction, Second Edition,* Microsoft Press; 2nd edition 2004

TEACHERS

1. dr inż Grzegorz Michalski grzegorz.michalski@icis.pcz.pl

ADDITIONAL NOTES

Links to course unit teaching materials can be found on the http://icis.pcz.pl/~michalski website for current students.