

SYLLABUS

INTERNATIONAL EUROPEAN
UNIVERSITY



EUROPEAN SCHOOL
OF BUSINESS



SYLLABUS



Course Name 

 Software architecture and design

Lecturer (s) 

 Borys Shevchuk, associate professor at the Department of Information Technology, PhD in Pedagogy, associate professor

Lecturer's profile 

 <https://business.ieu.edu.ua/kafedry/kafedra-informatsiinykh-tekhnologii#zzz-063>

Consultations

online consulting

 Monday 3:10 p.m. – 4:30 p.m.; Friday 3:10 p.m. – 4:30 p.m.
<https://meet.google.com/brj-dvft-eqq>

offline consulting

 –

Contact number 

 (093) 806 84 58

E-mail 

 borys_shevchuk@ieu.edu.ua

Discipline page 

 <https://dist.ieu.edu.ua/course/view.php?id=632>

Form of final control

test

def. test

exam



SYLLABUS



1 Brief discipline annotation

The Software architecture and design academic discipline is designed for Bachelors of the knowledge area: 12 Information Technology, specialty: 121 Software Engineering. It is one of the professional disciplines for future software developers.

2 Background for studying discipline

The program is based on the knowledge acquired while studying such disciplines as Software quality and testing, Fundamentals of programming, Software modeling and analysis.

3 Goal and objectives of the discipline

The goal of the Software architecture and design discipline is the study and practical learning of software design methods and tools, design patterns, development and architectures for their application on software product development processes.

Key objectives of the discipline:

- to master basic methods of object-oriented software design and be able to apply these methods in practice;
- to acquire knowledge of the main existing architectures, software tools for their implementation, their characteristics and particularities, as well as to be able to apply these particularities in practice;
- to understand concepts of software architecture, approaches to their analysis and construction;
- to apply UML (Universal Modeling Language) for modeling and designing software

4 Learning outcomes

PLO 10. To conduct a pre-project examination of the subject area and the system analysis of the design object.

PLO 11. To select design input data using formal methods of requirements description and modeling.

PLO 12. To apply efficient software design approaches in practice.

After learning the discipline, students should

know:

- content of basic concepts, subject matter and methods of the course;
- basic design patterns;
- methodology and technology of collecting basic customer requirements and methods of analyzing them;
- basic stages, principles and styles of developing complex software systems;
- software life cycle, stages in different life cycle models;
- basic architectural patterns and lower-level design patterns.

be able to:

- analyze requirements for developed software;
- estimate labor intensity and select adequate approaches to software development;
- design software architecture using visual modeling tools;
- design components of an architectural solution;
- define software requirements;
- make decisions on the choice of software system architecture;
- develop program modules interface, perform decomposition;
- make decisions on application and implementation of patterns of different abstraction levels;
- analyze software architecture for compliance with quality attributes.

5 ECTS credits

5

6 Course Content

Modules and topics	Amount of hours (full-time mode of study)				
	Total	including			
		Lectures	Practical classes	Laboratory classes	Independent work
MODULE I. SOFTWARE ARCHITECTURE					
<i>Content module I. Introduction to the discipline. Software structure</i>					
<u>Topic 1.</u> Introduction to the Software architecture and design discipline. Basic approaches to software development	20	2	6	-	12
<i>Content module II. Software architecture</i>					
<u>Topic 2.</u> Application of visual modeling using UML language in software development	18	2	6	-	10
<u>Topic 3.</u> Software development rules and cycle. Evolution of software life cycle models	18	2	6	-	10
<u>Topic 4.</u> Analysis, characteristics and structure of software. Software architecture	24	2	8	-	14
MODULE II. SOFTWARE DESIGN					
<i>Content module III. Software design</i>					
<u>Topic 5.</u> Software architecture design	10	2	2	-	6
<u>Topic 6.</u> Software design technologies.	14	4	2	-	8
<u>Topic 7.</u> Strategies and methods of software design. Standards and tools in software architecture selection	10	2	2	-	6
<u>Topic 8.</u> Software development patterns.	10	2	2	-	6
<u>Topic 9.</u> Architectural patterns and style.	12	2	4	-	6
<u>Topic 10.</u> Quality analysis and evaluation of software design. Methods of architecture analysis	14	2	4	-	8



SYLLABUS



7 List of obligatory tasks

1. Requirements collection and analysis. Brief description of the project.
2. Planning the process of software system development using Ms Project.
3. Development of the terms of reference for the system project.
4. UML design. Structural diagrams. Behavior diagram. Interaction diagrams.
5. Analysis of web application architecture based on subject area using CMS.
6. Web application interface design.
7. Analysis of web application structure.
8. Web application deployment.
9. Quality analysis and evaluation of web applications.
10. Extending the functionality of web applications using modules.
11. Evaluation and selection of software components.
12. Preliminary design of software architecture.
13. Modeling and visualization of software architecture.
14. Software architecture analysis.
15. Creating and exploring a software architectural prototype.
16. Software implementation and deployment.
17. Exploration of design patterns.

8 List of selective tasks

1. Application of UML in programming
2. Technologies for creating electronic glossaries.
3. SOLID principles and patterns of object-oriented programming

9 Discipline features

Period of teaching	Semester	International disciplinary integration	Year of study	Courses: general training/ professional training/elective
1 semester	5 semester	available	3rd year	Professional training

10 Assessment system and requirements. General system of discipline assessment

Personal computer (PC), Windows OS (7, 8.1, 10, 11), office software packages (Microsoft: Word, Excel Project; Adobe Acrobat), online services.

11 Admission to final control

As part of discipline teaching, one carries out the current and final control of students' knowledge. The final grade is given according to the total rating of students.

The results of the current control of students' knowledge is assessed in general between 0 and 60 scores. Students are admitted to the final control if they fulfil the requirements of the training program and obtain at least 36 scores for the current learning activity.

Final assessment of students' knowledge is conducted in the form of exam.

The maximum amount of scores that can be obtained during the exam is 40 scores.

The overall score of the discipline is 100. The total grade for the discipline is given according to the national and European scale.



SYLLABUS



12 Discipline policy

Teaching of the discipline is based on cutting-edge educational technologies aimed at increasing the level of students' interest in the course, providing theoretical and practical knowledge of the discipline. To activate the learning and cognitive activity of students, the discipline includes the consolidation of knowledge obtained at the lecture and acquisition of practical skills in lecture topics during laboratory classes and independent work.

13 Policy of absence and late task performance

Scores are not given for missed lectures. If students miss a laboratory work, they should perform all tasks of the missed laboratory work before the next laboratory work and present the results to the lecturer. Students who have missed classes without valid reasons and have not participated in current control activities are not admitted to the final semester control. In this case, a mark 'non-admission' is put in the exam record on the day of the exam. Repeated taking of the exam of the discipline is appointed in case of accomplishing all types of educational, individual work stipulated by the working program of the academic discipline and is carried out according to the approved schedule of academic failure liquidation.

14 Academic integrity policy

Tasks and laboratory works submitted later are assessed with a lower grade. The grade is reduced by one point for each week of lateness.

15 Recommended sources of information

PRIMARY (BASIC) LITERATURE:

1. Eric J. Braude. Software Engineering Technology. Translated from English. St. Petersburg: Peter, 2004.
2. K. Wiegers. Software Requirements. Translated from English. 2004.
3. Andy Carmichael, Dan Haywood. Better Software Faster. Translated from English. M.: Williams, 2003.
4. Alistair Cockburn. Writing Effective Use Cases. Translated from English. M.: Lori, 2002.
5. Butch G., Rambo J., Jacobson A. The language of UML. Users guide. Translated from English. M.: DMK, 2000.

Additional

1. Donald F. Shafer, Robert T. Futrell, Linda Isabell Shafer. Quality Software Project Management. Translated from English. M.: Williams, 2003.
2. Boggs W. UML and Rational Rose. Translated from English. M.: Lori Press, 2004.
3. Leszek A. Matsyashek, Requirements analysis and system design, Development of information systems using UML. Translated from English. M.: Williams, 2002. 432 p.
4. Leffingwell D. and Widrig D. Managing Software Requirements. Translated from English. M.: Williams, 2002.
5. Jacobson A., Butch G., Rambo J. Unified Software Development Process. Translated from English. St. Petersburg: Peter, 2002. 496 p.

Internet resources

1. Microsoft website. – Available at: <https://www.microsoft.com/uk-ua/>
2. UML Tutorial. – Available at: <https://www.tutorialspoint.com/uml/index.htm/>
3. IT website. – Available at: <https://dou.ua/>
4. Rational Requisite Pro 7.1.4. – Available at: <https://www.ibm.com/support/pages/rational-requisitepro-714>
5. Problems of Programming Scientific and Technical Journal. – Available at: <https://pp.isofts.kiev.ua/index.php/ojs1>



SYLLABUS



16

Tips on successful study during the course

Note: examine lecture materials and perform tasks and laboratory works synchronously with the curriculum. Thus, your abilities and insistence will be the key to success!