

# SCIENTIFIC AND PRACTICAL SEMINAR IN SOFTWARE ENGINEERING

## COURSE SYLLABUS

<b>Code and name of specialty</b>	121 Software Engineering	<b>Institute / faculty</b>	<b>Computer Sciences and Software Engineering</b>
<b>Program name</b>	"Software Engineering"	<b>Department</b>	Software Engineering and Management Information Technologies
<b>Type of program</b>	<b>Educational and Professional</b>	<b>Language of instruction</b>	<b>Ukrainian</b>

## LECTURER

Olga CHEREDNICHENKO

[Olga.Cherednichenko@kphi.edu.ua](mailto:Olga.Cherednichenko@kphi.edu.ua)



Doctor of Engineering Sciences, Professor of SEMIT Department.

Working experience 23 years. Number of scientific and educational publications is more than 70.

**Scientific interests:** methodological bases of information technologies of actual information monitoring, artificial intelligence, multiagent systems.

**Courses taught:** "Software of intelligent systems", "Models of artificial intelligence", "Modern models and methods of artificial intelligence"

## GENERAL DESCRIPTION OF THE COURSE

<b>Summary</b>	The discipline is aimed at the formation of knowledge, skills and abilities necessary for the use of modern methods and tools of software engineering for students to perform diploma theses of bachelor qualification level.
<b>Course objectives</b>	The purpose of the discipline is to study students of modern information technology, methodological and practical foundations of research work in the direction of the topic of their own thesis.
<b>Types of classes and control</b>	Seminars, consultations. Final control - credit test
<b>Term</b>	7,8

## Part 1

<b>Student workload (credits) / Type of course</b>	4 / Compulsory	<b>Lectures (hours)</b>	0	<b>Laboratory classes (hours)</b>	16	<b>Self-study (hours)</b>	104
--	----------------	-------------------------	---	-----------------------------------	----	---------------------------	-----

<b>Program competences</b>	<p>GC 01. Ability to abstract thinking, analysis and synthesis.</p> <p>GC 02. Ability to apply knowledge in practical situations.</p> <p>GC 03. Ability to communicate in the state language both orally and in written form.</p> <p>GC 04. Ability to communicate in a foreign language both orally and in written form.</p> <p>GC 05. Ability to learn and master modern knowledge.</p> <p>GC 06. Ability to search, process and analyze information from various sources.</p>
----------------------------	--

GC 07. Ability to work in a team.  
 GC08. Ability to act for ethical reasons.  
 PC13. Ability to identify, classify and formulate software requirements.  
 PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes.  
 PC15. Ability to develop architectures, modules and components of software systems.  
 PC20. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.  
 PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development.  
 PC25. Ability to reasonably select and master software development and maintenance tools.  
 PC26. Ability to algorithmic and logical thinking.PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
PO01. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO03. Know the basic processes, phases and iterations of the software life cycle.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO04. Know and apply professional standards and other regulatory documents in the field of software engineering.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO06. Ability to select and use the appropriate task methodology of software development.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO10. Conduct a pre-project survey of the subject area, systematic analysis of the design object.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO12. Put effective approaches to software design into practice.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment

PO15. Being motivated to choose programming languages and development technologies to solve problems of software design and maintenance.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment					
PO16. Have the skills of team development, approval, design and release of all types of software documentation.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment					
PO18. Know and be able to apply information technology processing, storage and transmission of data.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment					
PO19. Know and be able to apply methods of software verification and validation.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment					
PO23. Be able to document and present the results of software development.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment					
<b>Tutorial 1-2</b>	The main purpose of writing a thesis. Topics of graduation theses and tasks for their implementation for the "bachelor" educational and qualification level.	<b>Self-study</b>					
<b>Tutorial 3</b>	Formation of the calendar plan-schedule of the diploma work performance.						
<b>Tutorial 4</b>	The general structure and approximate content of the main sections of the thesis to the diploma work of the bachelor's level.						
<b>Tutorial 5</b>	The relevance of software development and software systems for various subject areas - in industry, economics, production, society development at the present stage, the purpose and methods of the bachelor's thesis.						
<b>Tutorial 6-7</b>	A concise qualitative (verbal) description of your subject area. The main problems of modern software systems development used in the subject area, which is selected in the thesis.						
<b>Tutorial 8</b>	Analytical review of some existing software systems that can be used to solve the problems identified in the thesis and determine your own approach to solving the problem and the motivation for its implementation.						
<b>Part 2</b>							
<b>Student workload (credits) / Type of course</b>	3 / Compulsory	<b>Lectures (hours)</b>	0	<b>Laboratory classes (hours)</b>	20	<b>Self-study (hours)</b>	70
<b>Program competences</b>	GC 01. Ability to abstract thinking, analysis and synthesis. GC 02. Ability to apply knowledge in practical situations. GC 03. Ability to communicate in the state language both orally and in written form. GC 04. Ability to communicate in a foreign language both orally and in written form.						

GC 05. Ability to learn and master modern knowledge.  
 GC 06. Ability to search, process and analyze information from various sources.  
 GC 07. Ability to work in a team.  
 GC08. Ability to act for ethical reasons.  
 PC13. Ability to identify, classify and formulate software requirements.  
 PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes.  
 PC15. Ability to develop architectures, modules and components of software systems.  
 PC20. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.  
 PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development.  
 PC25. Ability to reasonably select and master software development and maintenance tools.  
 PC26. Ability to algorithmic and logical thinking.PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
PO01. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO03. Know the basic processes, phases and iterations of the software life cycle.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO04. Know and apply professional standards and other regulatory documents in the field of software engineering.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO06. Ability to select and use the appropriate task methodology of software development.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO10. Conduct a pre-project survey of the subject area, systematic analysis of the design object.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO12. Put effective approaches to software design into practice.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment

PO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO15. Being motivated to choose programming languages and development technologies to solve problems of software design and maintenance.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO16. Have the skills of team development, approval, design and release of all types of software documentation.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO18. Know and be able to apply information technology processing, storage and transmission of data.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO19. Know and be able to apply methods of software verification and validation.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment
PO23. Be able to document and present the results of software development.	Practical classes, seminars and discussions, brainstorming, presentations, independent work with literature sources; project learning methods	Final assessment

<b>Tutorial 1-3</b>	Construction of the general system architecture of the target software system.	<b>Self-study</b>	Development of algorithmic and information support for the implementation of models (methods), which are proposed in the second section of the thesis.
<b>Tutorial 4-6</b>	Motivated choice of software tools and information technologies to be used in the thesis.		Finding ways that exist to continue working on your own topic of the thesis and possible improvement of the results
<b>Tutorial 7</b>	Features of direct software implementation of developed systems, processes of their testing and maintenance. Assessing the developed software whether it is possible to achieve the ultimate goal of the thesis.		Compilation of the list of the basic information sources taking into account specificity of own diploma work. Formation of conclusions to the qualification work.
<b>Tutorial 8-9</b>	Use of additional modern information sources during the design of the work: textbooks, monographs, articles in scientific and technical publications and Internet resources.		Create your own multimedia presentation and present it to defend the thesis.
<b>Tutorial 10</b>	Recommendations for the development and presentation of a multimedia presentation and the structure of the student's report on the defense of the thesis. The logical connection between the slides of the presentation and the student's report on the defense of the thesis.		Preparation for the defense of qualifying work.

## ASSESSMENT AND GRADING

Ranges of points corresponding to grades	core (points) for all types of learning activities	ECTS grading scale	The national grading scale	Allocation of grade points
	90-100	A	excellent	
	82-89	B	good	
	74-81	C		
	64-73	D	satisfactory	
	60-63	E		
	35-59	FX	Unsatisfactory (with the exam retake option)	
	0-34	F	Unsatisfactory (with mandatory repetition of the course)	

**100% Final assessment** as a result of credit test (40%) and Current assessment (60%).  
**40% Final exam**  
**60% Current assessment**

### Course policy

To cover all aspects of preparation, design and defense of qualification work, which is aimed at mastering by applicants all program competencies and obtaining training results in the specialty "Software Engineering".

## RECOMMENDED READING

Compulsory		Recommended	
	<ol style="list-style-type: none"> <li>Годлевський, М. Д., Ткачук, М. В., Сокол, В. Є., Чередніченко, О. Ю., Шматко, О. В. (2018). Методичні вказівки до виконання дипломних робіт освітньо-кваліфікаційного рівня–бакалавр у галузі знань 12 «Інформаційні технології» для студентів спеціальності 121 «Інженерія програмного забезпечення». Харків: НТУ «ХПІ».</li> <li>Зайцев, Є. О. (2017). Основи програмної інженерії: навчальний посібник. Ки: КНТЕУ.</li> <li>Бородкіна, І. Л., Бородкін, Г. О. (2018). Інженерія програмного забезпечення: посібник для студентів вищих навчальних закладів. Київ.</li> <li>Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12–Інформаційні технології, спеціальність 121–Інженерія програмного забезпечення. Затверджено і введено в дію наказом Міністерства освіти і науки України від 29.10.2018 № 1166.(2018).</li> <li>Актуальність і новизна наукового дослідження [Електронний ресурс]. Retrieved from <a href="http://pidruchniki.com/70330/buhgalter-skiy_oblik_ta_audit/aktua lnist_novizna_naukovogo_doslidzhennya">http://pidruchniki.com/70330/buhgalter-skiy_oblik_ta_audit/aktua lnist_novizna_naukovogo_doslidzhennya</a>.</li> </ol>		<ol style="list-style-type: none"> <li>Черткова, Е. А. (2018). Комп'ютерні технології навчання. Retrieved from <a href="https://stud.com.ua/174105/informatika/kompyuterni_tehnologiyi_navchannya">https://stud.com.ua/174105/informatika/kompyuterni_tehnologiyi_navchannya</a>.</li> <li>Положення про організацію освітнього процесу в Національному технічному університеті «Харківський політехнічний інститут». Retrieved from <a href="http://blogs.kpi.kharkov.ua/v2/nv/dokumentintu-hpi-2/">http://blogs.kpi.kharkov.ua/v2/nv/dokumentintu-hpi-2/</a></li> <li>Кодекс етики академічних взаємовідносин та доброчесності Національного технічного університету «Харківський політехнічний інститут». Retrieved from <a href="http://blogs.kpi.kharkov.ua/v2/nv/wp-content/uploads/sites/17/2019/11/04_code_ethics.pdf">http://blogs.kpi.kharkov.ua/v2/nv/wp-content/uploads/sites/17/2019/11/04_code_ethics.pdf</a></li> <li>Положення про систему запобігання та виявлення академічного плагіату у випускних кваліфікаційних роботах здобувачів вищої освіти Національного технічного університету «Харківський політехнічний інститут». Retrieved from <a href="http://blogs.kpi.kharkov.ua/v2/nv/wp-content/uploads/sites/17/2019/11/05_polozhennya-proekt-plagyat-1.pdf">http://blogs.kpi.kharkov.ua/v2/nv/wp-content/uploads/sites/17/2019/11/05_polozhennya-proekt-plagyat-1.pdf</a></li> <li>Положення про Електронний репозитарій кваліфікаційних випускних робіт здобувачів вищої освіти у Національному технічному університеті «Харківський політехнічний інститут». Retrieved from <a href="http://blogs.kpi.kharkov.ua/v2/nv/wp-content/uploads/sites/17/2019/11/07_repozitarij_dipl_rabot_2018.pdf">http://blogs.kpi.kharkov.ua/v2/nv/wp-content/uploads/sites/17/2019/11/07_repozitarij_dipl_rabot_2018.pdf</a></li> </ol>

### Academic integrity

Graduate students are expected to adhere to the Code of Ethics of Academic Relations and Integrity” of NTU “KhPI”.

The content of this syllabus is consistent with the course program.