BASICS OF SOFTWARE ENGINEERING

COURSE SYLLABUS

Code and name of specialty	121 Software Engineering	**Institute / faculty**	Faculty of Computer Science and Software Engineering
Program name	“Software Engineering”	**Department**	Software Engineering and Management Information Technologies
Type of program	Educational and Professional	**Language of instruction**	Ukrainian, English

LECTURER

Name and surname, email	Iryna Liutenko, iryna.liutenko@khpi.edu.ua

Ph.D., Associate Professor, Associate Professor of Software Engineering and Information Technology Management. Author (co-author) of more than 60 publications, 1 collective monograph, 1 textbook with the stamp of the university, 3 articles in Scopus (Google Scholar - https://scholar.google.com/citations?hl=ru&user=9EhcsRcAAAAJ ; ORCID ID is ORCID: https://orcid.org/0000-0003-4357-1826).
Leading lecturer of the courses: Basics of Software Engineering (Bachelors) (in Ukrainian), Formal methods of research of software systems (Masters) (in English)

GENERAL DESCRIPTION OF THE COURSE

Summary	The discipline "BASICS OF SOFTWARE ENGINEERING" is a discipline from the cycle of professional compulsory training in the specialty 121 "Software Engineering". It is taught in the first semester in the amount of 120 hours. (4 ECTS credits), in particular: lectures - 32 hours, laboratory - 32 hours, independent work - 56 hours. The course provides two content modules. The discipline ends with a test.
Course objectives	Assimilation by students of the necessary knowledge about software engineering as one of the main activities in software projects, study of basic methods and tools of software engineering in a systematic way for their application in the processes of analysis, design, construction and testing of software systems.
Types of classes and control	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment, course work. Final assessment – test.
Term	1

Student workload (credits) / Type of course	4 / Mandatory	Lectures (hours)	32	Workshops (hours)	32	Self-study (hours)	56

Program competences	K01. Ability to abstract thinking, analysis and synthesis. K04. Ability to communicate in a foreign language both orally and in writing. K05. Ability to learn and master modern knowledge. K06. Ability to search, process and analyze information from various sources. K07. Ability to work in a team. K08. Ability to act on ethical considerations. K09. The desire to preserve the environment. K12. Ability to preserve and multiply moral, cultural, scientific values and achievements of society based on understanding the history and patterns of development of the subject area, its place in the general system of knowledge about nature and society and in the development of society, techniques and technologies. Active recreation and a healthy lifestyle. K13. Ability to identify, classify and formulate software requirements. K14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and functioning processes. K16. Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards. K17. Ability to adhere to specifications, standards, rules and recommendations in the professional field in the implementation of life cycle processes. K22. Ability to accumulate, process and systematize professional knowledge on the creation and maintenance of software and recognition of the importance of lifelong learning. K25. Ability to reasonably select and use software development and maintenance tools.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
PR01. 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.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR03. Know the basic processes, phases and iterations of the software life cycle.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)

PR04. Know and apply professional standards and other regulatory documents in the field of software engineering.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR06. Ability to choose and use the methodology of creating software according to the task.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, project training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, project training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR14. Apply in practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR15. Motivated to choose programming languages and development technologies to solve problems of software creation and maintenance.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR19. Know and be able to apply methods of software verification and validation.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR20. Know approaches to evaluating and ensuring software quality	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, problem training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PR23. Be able to document and present the results of software development.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, project training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)

ASSESSMENT AND GRADING

Ranges of points correspondin	core (points) for all types of learning activities	ECTS grading scale	The national grading scale	Allocation of grade	**100% Final assessment** as a result of Final exam (20%)
	90-100	A	excellent		

g to grades	82-89	B	good	**points**	and Continuous assessment (80%). **20% Final exam** **80% Continuous assessment**: • 50% of assessment of tasks in laboratory works; • 30% intermediate control (2 modular control works)
	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)		

Course policy	Students must attend all classes according to the study schedule and adhere to the norms of academic ethics. To study the course, students need to have their personal computer and (or) use computers of the computer center at the department. Students must work with compulsory and recommended reading, including Internet resources. Students must complete and submit all laboratory works during the semester in which the course is taught, before the examination session. The final assessment is not carried out without the personal presence of students.

COURSE STRUCTURE AND CONTENT

Topic 1	The core of knowledge in software engineering.	**Laboratory work 1**	Introduction with the Visual Paradigm for UML environment.	**Self-study**	Development of UML-diagrams according to the individual task
Topic 2	System bases of modern technologies of software engineering. Profiles of software systems life cycle standards. Standards of open systems that regulate the structure and interfaces of software systems. Flexible software development methodologies.	**Laboratory work 2**	Development of software requirements and program modeling (on the example of small tasks).		Specification of functional requirements for IP software
Topic 3	Requirements analysis, design and construction of software systems	**Laboratory work 3**	Software design and development. Application of visual modeling tools during software development		Creating unit tests.
Topic 4	Software systems quality engineering. Systems quality management standards. Inherited systems.	**Laboratory work 4**	Software testing in accordance with the formulated requirements		Software project management - SMMI.

RECOMMENDED READING

Compulsory	1.Ian Sommerville Software Engineering. (6th Edition). Retrieved from https://www.academia.edu/6826193/Ian_Sommerville_Software_Engineering_6th_Edition 2.Лавріщева, К. М. (2008). Програмна інженерія. Київ. Retrieved from http://www.cyb.univ.kiev.ua/library/books/lavrishcheva-6.pdf 3.Карл І. Вігерс, Джой Бітті Розробка вимог до програмного забезпечення. Retrieved from http://www.twirpx.com/file/1073169/ 4.Левус, Є., Мельник, Н. (2017). Вступ до інженерії програмного забезпечення. Львів: Видавництво Львівської політехніки. Retrieved from https://shron1.chtyvo.org.ua/Levus_Yevheniia/Vstup_do_inzhenerii_prohramnoho_zabezpechennia.pdf?PHPSESSID=c7f247v55v542l158balt97e81 5.Зайцев, Є. О. (2017). Основи програмної інженерії. К.:КНТЕУ. Retrieved from https://www.researchgate.net/publication/322028393_Navcalnij_posibnik_Teoreticni_osnovi_programnoi_inzenerii 6.Жулковський, О. О., Жулковська, І. І. (2017). Конспект лекцій з дисципліни «Основи програмної інженерії». Дніпро: ДДТУ. Retrieved from https://vo.uu.edu.ua/pluginfile.php/188142/mod_resource/content/1/%D0%BA%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9.pdf	**Recommended**	7.Петрик, М. Р., Петрик, О. Ю. (2015). Моделювання програмного забезпечення: науково-методичний посібник. Тернопіль: Вид-во ТНТУ. 8.Табунщік, Г. В, Каплієнко, Т. І., Петрова, О. А .(2016). Проектування та моделювання програмного забезпечення сучасних інформаційних систем: навч. Посібник. Запоріжжя. 9.The Unified Modeling Language user guide. Retrieved from https://www.researchgate.net/publication/234785986_Unified_Modeling_Language_User_Guide_The_2nd_Edition_Addison-Wesley_Object_Technology_Series 10.ISO/IEC 9126-1:2001 Software engineering-Product quality. Part 1: Quality model. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749. 11.Орлик, С. Введення в програмну інженерію і управління життєвим циклом програмного забезпечення Guide to Software Engineering Base of Knowledge (SWEBOK). [Електронний ресурс]. Retrieved from sorlik.blogspot.com/

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.