

# A COMPREHENSIVE PROGRAM OF PRACTICAL TRAINING

## COURSE SYLLABUS

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

### GENERAL DESCRIPTION OF THE COURSE

<b>Summary</b>	The comprehensive program of practical training of students is a document, the main purpose of which (according to the Regulations on the procedure of practical training of applicants for the higher education of the National Technical University "Kharkiv Polytechnic Institute" p. 8.1.1) is to acquaint applicants for higher education and other participants in the educational process with a holistic system of practical training of the educational program "Computer Science and Intelligent Systems" (Innovation Campus). An integrated approach to the organization of practical training of students is provided by the presence of 2 stages of practical training, which are listed in the table.			
	Code	Components of the educational program	Number of credits	Form of final control
				Exams (terms)      Credits (terms)
	PT 25	Project (practice)	6.0	6
PT 26	Pre-graduation practice	6.0	8	
<b>Course objectives</b>	Improving efficiency and quality in the development, implementation, maintenance, and research of intelligent systems requires a rational combination of theoretical knowledge of specialists with the ability to solve practical problems. Achieving the goal of the educational program is based on the principles of continuity and individualization of learning, fundamentality, and integrity of knowledge, practical orientation and awareness of the place of acquired competencies, the symbiosis of scientific and systematic approaches. In the process of practical training is the formation and consolidation of general and special competencies, which are listed in the educational program.			
<b>Types of classes and control</b>	In the process of internship, lectures, seminars, and excursions can be held, the topics of which are made taking into account the peculiarities of the specialty of training and the base of practice.			
<b>Term</b>	Project (practice) – 6 Pre-graduation practice – 8			

<b>Student workload (credits) / Type of course</b>	6 / Mandatory	<b>Lectures (hours)</b>	-	<b>Laboratory classes (hours)</b>	-	<b>Self-study (hours)</b>	360
--	---------------	-------------------------	---	-----------------------------------	---	---------------------------	-----

<b>Program competences</b>	<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 05. Ability to learn and master modern knowledge.</p> <p>GC 06. Ability to search, process and analyze information from various sources.</p> <p>GC 07. Ability to work in a team.</p> <p>GC08. Ability to act for ethical reasons.</p> <p>GC09. The desire to preserve the environment.</p> <p>GC10. The ability to act socially responsibly and consciously.</p> <p>PC13. Ability to identify, classify and formulate software requirements.</p>
----------------------------	---

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.  
 PC17. Ability to adhere to specifications, standards, rules and recommendations in the professional field in the implementation of life cycle processes.  
 PC18. Ability to analyze, select and apply methods and tools to ensure information security (including cybersecurity).  
 PC19. Knowledge of information data models, the ability to create software for data storage, retrieval and processing.  
 PC20. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.  
 PC22. Ability to accumulate, process and systematize professional knowledge on the creation and maintenance of software and recognition of the importance of lifelong learning.  
 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.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
<p>PLO 1. 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.            PLO 2. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.            PLO 3. Know the basic processes, phases and iterations of the software life cycle.            PLO 5. Know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modelling for software development.            PLO 7. Know and apply in practice the fundamental concepts, paradigms and basic principles of operation of language, tools and computing software engineering.            PLO 8. Be able to develop a human-machine interface.            PLO 9. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.            PLO10. Conduct a pre-project survey of the subject area, systematic analysis of the design object.            PLO12. Put effective approaches to software design into practice.            PLO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.            PLO15. Being motivated to choose programming</p>	<p>Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning</p>	<p>Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester test, according to the schedule of the educational process (FAS)</p>

languages and development technologies to solve problems of software design and maintenance.

PLO16. Have the skills of team development, approval, design and release of all types of software documentation.

PLO17. Be able to apply methods of component software development.

PLO19. Know and be able to apply methods of software verification and validation.

PLO22. Know and be able to apply methods and tools of project management.

PLO23. Be able to document and present the results of software development.

PLO24. Be able to calculate the economic efficiency of software systems.

PLO25. Apply the principles of moral, cultural, scientific values and increase the achievements of society, use different types and forms of physical activity to lead a healthy lifestyle and professional activities in the field of information technology.

**Structural and logical scheme of studying the discipline**

Before the practical training, students must have previously studied several disciplines. For project practice, there are SP1 - Programming basics, SP2 - Basics of software engineering, SP4 - Computer Architecture Fundamentals and Operating Systems, SP5 - Fundamentals of the theory of algorithms, SP7 - Data models and structures, SP 8 - Object-oriented programming, SP9 - Fundamentals of computer networks, SP10 - Fundamentals of web development , , SP12 - Software requirements engineering, SP14 - Architecture and design of software, SP19 - Scientific and practical seminar in software engineering.

To conduct undergraduate practice, students must study the disciplines of general and special training in full.

**ASSESSMENT AND GRADING**

Range s of points corres pondi ng to grades	core (points) for all types of learning activities	ECTS grading scale	The national grading scale	Allocation of grade points	At the end of the internship, students submit to the commission, which is approved by the head of the department, a report and a diary of practice. Credit (with the differentiated assessment) is carried out following the Regulations on the procedure for practical training of higher education applicants of the National Technical University "Kharkiv Polytechnic Institute" (p. 6. Summarizing the practice).
	90-100	A	excellent		
	82-89	B	good		
	74-81	C			
	64-73	D	satisfactory		
	60-63	E			
	35-59	FX			
	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.

### THE CONTENT OF PRACTICAL TRAINING

#### Content

During the internship, the student gets acquainted with the main forms of activity in his future specialty.  
Practical training helps to increase students' motivation to learn competencies, individualize learning, create conditions for the development of student's creative thinking, ability to generate new ideas, deepen and consolidate theoretical knowledge gained in higher education.  
During the internship, students master new technologies of project development and implementation, acquire teamwork skills, improve communication skills in areas of professional activity.

#### Self-study

In the process of internship, students must:

- to get acquainted with perspective directions of development of intelligent systems;
- get acquainted with the standards and other regulatory documentation used in software development;
- perform the task using systematic and critical thinking;
- draw up the final documentation.

#### Academic integrity

Students must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to show discipline, politeness, friendliness, honesty, responsibility

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