**Syllabus**
Course Program

# Fundamentals of software engineering

Specialty
121 – Software Engineering

Educational program
Software Engineering

Level of education
Bachelor's level

Semester
1

Institute
Institute of Computer Science and Information Technology

Department
Software Engineering and Management Intelligent Technologies (321)

Course type
Special (professional), Mandatory

Language of instruction
English, Ukrainian

## Lecturers and course developers

### First name and surname

Karina.Melnyk@khpi.edu.ua
Ph.D., Associate Professor, Associate Professor of Software Engineering and Information Technology Management

Author (co-author) of more than 70 publications, 5 collective monographs, 10 articles in publications indexed in Scopus and Web of Science. (h-index = 5, i10-index = 1 in Google Scholar -https: //scholar.google.com/citations? user = xCU7GMgAAAAJ & hl = ru; ORCID ID https://orcid.org/0000-0001- 9642-5414; Scopus Author ID https://www.scopus.com/authid/detail.uri?authorId=57195074119).
Leading lecturer of the courses: Basics of Software Engineering (Bachelors) (in English), Methods of Empirical Information Processing (Bachelors) (in English and Ukrainian), Basics of Intelligent Systems Design (Masters) (in English and Ukrainian)
More about the lecturer on the department's website

## General information

### Summary

The discipline "Fundamentals of software engineering" is an educational discipline from the cycle of professional mandatory training in the specialty 121 "Software engineering", aimed at forming the professional skills and abilities of students. It is taught in the first semester in the amount of 120 hours. (4 credits), in particular: lectures – 32 hours, laboratory hours – 32 hours, independent work – 56 hours. The course includes two content modules. The discipline ends with a credit.

### Course objectives and goals

Students' assimilation of the necessary knowledge about software engineering as one of the main areas of activity in software projects, studying the main methods and tools of software engineering in a systematic form for their application in the processes of analysis, design, construction and testing of software systems.

## Format of classes

Lectures, laboratory classes, consultations, self-study. Final control in the form of a credit.

## Competencies

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. The ability to act on the basis of ethical considerations.
K09. The desire to preserve the environment.
K12. The 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, technology and technologies, to use various types and forms of motor activity for active recreation and leading a healthy lifestyle.
K13. Ability to identify, categorize and formulate software requirements.
K14. Ability to participate in the design of software, 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 when implementing life cycle processes.
K22. The ability to accumulate, process and systematize professional knowledge about creating and maintaining software and recognizing the importance of lifelong learning.
K25. The ability to reasonably choose and master software development and maintenance tools

## Learning outcomes

PR01. Analyze, purposefully search for and select the information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology.
PR02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.
PR03. Know the main processes, phases and iterations of the software life cycle.
PR04. Know and apply professional standards and other regulatory documents in the field of software engineering.
PR06. Ability to choose and use software development methodology appropriate to the task.
PR09. Know and be able to use methods and tools for gathering, formulating and analyzing software requirements.
PR14. Apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement and documentation of software.
PR15. Motivated to choose programming languages and development technologies to solve the tasks of creating and maintaining software.
PR19. Know and be able to apply software verification and validation methods.
PR20. Know approaches to evaluation and quality assurance of software
PR23. Be able to document and present the results of software development

## Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 32 hours, laboratory classes - 32 hours, self-study - 56 hours.

## Course prerequisites

Students should have the basic knowledge and skills they have acquired in a school computer science course.

### Features of the course, teaching and learning methods, and technologies

The teaching of the course involves the use of problem-based and interactive lectures with presentations, discussions, laboratory classes, teamwork, the method of feedback from students, problem-based learning. In the process of performing laboratory work, students will receive an individual task, after performing laboratory work, the student prepares report documentation. Students' knowledge is assessed for each laboratory work. In addition, online tests are conducted in laboratory classes

## Program of the course

### Topics of the lectures

Theme 1. Basic Principles of Software Engineering
Theme 2. Software Development Life Cycle
Theme 3. Requirements Engineering
Theme 4. Domain Modeling
Theme 5. Software Design
Theme 6. Information Architecture
Theme 8. Software testing
Theme 9. Measuring Software Quality

### Topics of the workshops

Practical classes within the discipline are not provided.

### Topics of the laboratory classes

Topic 1. Learning Use Case and Activity Diagrams by using Visual paradigm
Topic 2. Learning basic principles of C++
Topic 3. Domain modeling with different methodologies
Topic 4. Development of Software Requirement Specification
Topic 5. Learning of Working with project tracking system
Topic 6. Learning basic principles of testing C++ code

### Self-study

Information on self-study and individual assignments (reports, course projects, etc.), if it is necessary according to the plan. Also, methods of control and assessment of self-study.

## Course materials and recommended reading

1. Ian Sommerville Software Engineering, Global Edition. – Pearson Higher Ed. – 2018. – 816p.
2. Ian Sommerville Engineering Software Products: An Introduction to Modern Software Engineering, Global Edition. – Pearson. – 2020. – 368p.
3. Martin Fowler Refactoring. Addison-Wesley. – 2018. – 448p.
4. David Thomas, Andrew Hunt The Pragmatic Programmer, 20-th Anniversary Edition. Addison-Wesley. – 2019. – 352p.
5. Joy Beatty & Karl Wiegers. Software Requirements (3rd Edition). Microsoft Press Redmond, WA, USA 2013 ISBN:0735618798 9780735618794
6. Methodical recommendation to basics of software engineering" laboratory practice (part 1) / K. V. Melnyk, N. V. Borisova, I. V. Lyutenko, S. I. Yershova, P. O. Smolin, M. A. Grinchenko - Kharkiv: NTU "KhPI", 2019.–22 p.
7. Methodical recommendation to basics of software engineering" laboratory practice (part 2) / K. V. Melnyk, N. V. Borisova, I. V. Lyutenko, S. I. Yershova, P. O. Smolin, M. A. Grinchenko - Kharkiv: NTU "KhPI", 2019. – 20 p.
8. Methodical recommendation to basics of software engineering" laboratory practice (part 3) / K. V. Melnyk, N. V. Borisova, I. V. Lyutenko, S. I. Yershova, P. O. Smolin, M. A. Grinchenko - Kharkiv: NTU "KhPI", 2019. – 17 p.

# Assessment and grading

### Criteria for assessment of student performance, and the final score structure

-100% final assessment in the form of credit (10%) and current assessment (90%).
10% credit: semester credit, according to the schedule of the educational process
90% current assessment:
70% assessment of tasks in laboratory works;
20% intermediate control (2 independent individual works)

### Grading scale

| Total points | National | ECTS |
|---|---|---|
| 90–100 | Excellent | A |
| 82–89 | Good | B |
| 75–81 | Good | C |
| 64–74 | Satisfactory | D |
| 60–63 | Satisfactory | E |
| 35–59 | Unsatisfactory (requires additional learning) | FX |
| 1–34 | Unsatisfactory (requires repetition of the course) | F |

# Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

# Approval

| | | |
|---|---|---|
| Approved by | 08.06.2023 | Head of the department<br>Ihor HAMAIUN |
| | 08.06.2023 | Guarantor of the educational program<br>Uliya LITVINOVA |