



Syllabus Course Program



Project workshop

Specialty

121 – Software Engineering
122 – Computer Science

Institute

Institute of Computer Science and Information
Technology

Educational program

Software Engineering
Computer Science and Intelligent Systems

Department

Software Engineering and Management Intelligent
Technologies (321)

Level of education

Bachelor's level

Course type

Special (professional), Elective

Semester

7

Language of instruction

English, Ukrainian

Lecturers and course developers



Dmytro Dvukhhlavov

dmytro.dvukhhlavov@khiu.edu.ua

Ph.D., Associate Professor, Associate Professor of Software Engineering and
Information Technology Management.

Google Scholar: <https://scholar.google.com/citations?user=OAzyFg8AAAAI&hl>

ORCID: <https://orcid.org/0000-0002-3361-3212>

Scopus: <https://www2.scopus.com/authid/detail.uri?authorId=57211294555>

Web of Science: <https://www.webofscience.com/wos/author/record/E-8279-2019>).

[More about the lecturer on the department's website](#)

General information

Summary

Studying the discipline gives the student the opportunity to decide on a technology stack for developing their own web applications, install it on a computer and use it to create a prototype of the software that will be developed as a result of diploma projecting.

Course objectives and goals

Teaching the discipline is aimed at deepening knowledge and improving skills in designing and creating a web application, choosing and applying effective technologies for its implementation.

Format of classes

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

Competencies

121 - Software Engineering

GC02. Ability to apply knowledge in practical situations.

PC14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and functioning processes.

- PC15. Ability to develop architectures, modules and components of software systems.
- PC16. Ability to formulate and provide software quality requirements in accordance with customer requirements, specifications and standards.
- PC17. Ability to adhere to specifications, standards, rules and recommendations in the professional field in the implementation of life cycle processes.
- 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.

122 - Computer Science and Intelligent Systems

- GC01. Ability to abstract thinking, analysis and synthesis.
- GC02. Ability to apply knowledge in practical situations.
- GC03. Knowledge and understanding of the subject area and understanding of professional activity.
- GC06. Ability to learn and master modern knowledge.
- GC08. Ability to generate new ideas (creativity).
- GC09. Ability to work in a team.
- GC10. Ability to be critical and self-critical.
- GC11. Ability to make informed decisions.
- GC12. Ability to evaluate and ensure the quality of work performed.
- GC13. Ability to act on ethical considerations.
- GC14. Ability to exercise their rights and responsibilities as a member of society, to realize the values of civil (free democratic) society and the need for its sustainable development, the rule of law, human and civil rights and freedoms in Ukraine.
- GC15. 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 leading a healthy lifestyle.
- PC01. Ability to mathematically formulate and study continuous and discrete mathematical models, justify the choice of methods and approaches for solving theoretical and applied problems in the field of computer science, analysis and interpretation.
- PC07. Ability to apply the theoretical and practical foundations of methodology and modeling technology to study the characteristics and behavior of complex objects and systems, to conduct computational experiments with processing and analysis of results.
- PC08. Ability to design and develop software using different programming paradigms: generalized, object-oriented, functional, logical, with appropriate models, methods and algorithms of calculations, data structures and control mechanisms.
- PC09. Ability to implement a multi-tier computing model based on client-server architecture, including databases, knowledge and data warehouses, perform distributed processing of large data sets on clusters of standard servers to meet the computing needs of users, including cloud services.
- PC10. Ability to apply methodologies, technologies and tools to manage the life cycle processes of information and software systems, information technology products and services in accordance with customer requirements.
- PC11. Ability to data mining based on methods of computational intelligence, including large and poorly structured data, their prompt processing and visualization of analysis results in the process of solving applied problems.
- PC14. Ability to apply methods and means of information security, to develop and operate special software for protection of information resources of critical information infrastructure.
- PC15. Ability to analyze and functional modeling of business processes, construction and practical application of functional models of organizational, economic and production-technical systems, methods of risk assessment of their design.

Learning outcomes

121 - Software Engineering

PLO 03. Know the basic processes, phases and iterations of the software life cycle.

PLO 04. Know and apply professional standards and other legal documents in the field of software engineering.

PLO 06. Ability to choose and use the methodology of creating software appropriate to the task.

PLO 09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.

PLO 10. Conduct a pre-project survey of the subject area, systematic analysis of the design object.

PLO 11. Choose source data for design, guided by formal methods of describing requirements and modeling.

PLO 12. Apply effective approaches to software design in practice.

PLO 13. Know and apply methods of algorithm development, software design and data and knowledge structures.

PLO 14. Apply in practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.

PLO 15. Motivated to choose programming languages and development technologies to solve problems of software creation and maintenance.

PLO 17. Be able to apply methods of component software development.

PLO 19. Know and be able to apply methods of software verification and validation.

PLO 20. Know approaches to evaluating and ensuring software quality.

PLO 23. Be able to document and present the results of software development.

122 - Computer Science and Intelligent Systems

PLO 09. Develop software models of subject environments, choose a programming paradigm from the standpoint of convenience and quality of application for the implementation of methods and algorithms for solving problems in the field of computer science.

PLO 10. Use tools for developing client-server applications, design conceptual, logical and physical models of databases, develop and optimize queries to them, create distributed databases, repositories and showcases of databases, knowledge bases, including on cloud services, using web languages - programming.

PLO 11. Have the skills to manage the life cycle of software, products and services of information technology in accordance with the requirements and restrictions of the customer, be able to develop project documentation (feasibility study, terms of reference, business plan, agreement, contract, contract).

PLO 13. Know the languages of system programming and methods of program development that interact with the components of computer systems, know network technologies, computer network architectures, have practical skills in the technology of computer network administration and their software.

PLO 15. Understand the concept of information security, the principles of secure software design, ensure the security of computer networks in conditions of incomplete and uncertain source data.

PLO 16. Perform parallel and distributed calculations, apply numerical methods and algorithms for parallel structures, parallel programming languages in the development and operation of parallel and distributed software.

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 16 hours, laboratory classes - 16 hours, self-study - 88 hours.

Course prerequisites

121 - Software Engineering:

The study of this discipline is directly based on: "Fundamentals of Programming", "Computer Architecture and Operating Systems", "Algorithm Theory", "Introductory Practice at the Innovation Campus".

122 - Computer Science:

The study of this discipline is directly based on: "Algorithmization and Programming", "Operating Systems", "Algorithms and Data Structures", "Introductory Practice at Innovation Campus".

Object-oriented programming
Basic Java programming course
Advanced Java programming course

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

Teaching and learning methods

The main teaching method during lectures is the explanatory and illustrative method. In order to activate cognitive activity, student performances and the organization of discussions on specific issues of the lectures are provided. The laboratory course implements a project-based approach to learning. This involves creating from lesson to lesson by the student an individually themed web application that implements the manipulation of data stored in several linked tables of a client-server database using web pages.

Forms of assessment

The mastery of the theory is tested in the form of an express survey during the lectures (CAS), a survey or an automated test at the beginning of the laboratory work (CAS). The level of practical skills is checked by reviewing the results of laboratory works according to separate assessment methods (CAS). Final control - credit. The final grade is issued based on the sum of points earned during the semester (FAS).

Program of the course

Topics of the lectures

Topic 1:

Content and stages, features and software tools of web application development.

Topic 2:

Designing a relational database and organizing its filling with test data.

Topic 3:

Web Application Architecture Design Features. Architectural Patterns and Design Patterns in defining the structure of a web application project.

Topic 4:

Peculiarities of design and implementation of the web application interface.

Topic 5:

Providing ease of use of the web application interface: pagination, data validation, handling of exceptional situations.

Topic 6:

Providing access control and data security in the web application.

Topic 7:

Organization of web application logging.

Topic 8:

Organization of web application deployment.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1: A meaningful description of the object (process) being automated. Defining functional requirements for a web application and choosing a technology stack for its development

Topic 2: Designing a data schema, creating a database and filling it with test data

Topic 3: Development and testing of web application code

Topic 4: Implementation of web page pagination, data validation and exception handling

Topic 5: Implementation of multi-user data access

Topic 6: Implementation of the event log

Topic 7: Development of a web application deployment scheme

Self-study

Software tools for creating and deploying web applications. Rationale for choosing a technology stack for developing your own application. Software tools to support data model development and database creation. Database tools in the chosen technology stack. Architectural patterns and design patterns. Peculiarities of the implementation of design templates when designing the structure of a web application using the selected technology stack. Software to support the development of web pages Software tools for creating web pages of rich web applications. Web page pagination, data validation, and exception handling using the chosen technology stack. Peculiarities of implementing authentication and authorization, distinguishing user access rights, encrypting passwords, the concept of captcha using the selected technology stack. Features of the implementation of logging using the selected technology stack. Variants of web applications deployment.

Course materials and recommended reading

Key literature

1. Ramonyai J. (2021) Software Engineering: Skill Development, Architecture, Design Patterns, Testing, Product Management Concepts, Project Lifecycle, Programming, Quality Assurance, Security Systems, Service & Practice / J. Ramonyai. – 545 p.
2. Мулеса О.Ю. (2018) Інформаційні системи та реляційні бази даних. Навч.посібник. - 118 с.
3. Хайрова Н.Ф., Петрасова С.В., Панов А.М. (2020) Сучасні технології Web-програмування. - НТУ "ХПІ". Харків. - 112 с.
4. Newman, S. (2021). Building Microservices: Designing Fine-Grained Systems; 2nd ed. - O'Reilly Media/ - 616 p.
5. Duggan, D. (2012). Enterprise Software Architecture and Design: Entities, Services, and Resources. - Wiley-IEEE Computer Society Pr. - 821 p.
6. Mullin, S. (2021). Coding Activities for Building Databases with SQL (Code Creator). - Rosen Publishing Group. - 64 p.
7. Duckett Jon. HTML and CSS: Design and Build Websites, 2020. - 514 p.
8. Graham D., Veenendaal E., Black R. (2018) Foundations of Software Testing ISTQB Certification, 4th edition. - Cengage Learning. - 288 p.
9. Munos P. G. The 5 Secrets of Good Software Testing / P. G. Munos, A. Redwood. - PG Publishing, 2021. – 21 p.
10. Євсєєв С.П, Остапов С.Е., Король О.Г. (2019) Кібербезпека: сучасні технології захисту. Навчальний посібник для студентів вищих навчальних закладів. Львів: "Новий Світ - 2000". - 678 с.

Additional literature

1. Nixon R. (2018) Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. - O'Reilly Media, Inc. - 832 p.
2. Adrian W.W., Steve Prettyman S. (2018) Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites. - Apress. - 546 p.
3. Karthik P. (2019) Web Applications using JSP (Java Server Page): Develop a fully functional web application. - BPB Publications. - 988 p.
4. Mehta C. (2018) MySQL 8 Administrator's Guide: Effective guide to administering high-performance MySQL 8 solutions. - Packt Publishing Ltd. - 510 p.
5. Miguel Grinberg. . (2018) Flask Web Development: Developing Web Applications with Python; 2nd edition. - O'Reilly Media. - 474 p.
6. Kunal Relan. (2019) Building REST APIs with Flask: Create Python Web Services with MySQL – Apress. - 218 p.
7. Мартін Р. С. (2019) Чистий код.

Internet resources

8. <https://www.mysqltutorial.org/>

9. https://www.tutorialspoint.com/mvc_framework/index.htm
10. <https://www.refactoring.guru>
11. <https://w3schoolsua.github.io/html/>
12. <https://www.metanit.com/>
13. <https://o7planning.org/>
14. <https://www.json.org/json-en.html>

Assessment and grading

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

The student will receive points for obtaining a final grade for performing laboratory work (up to 80 points) and taking a test (up to 20 points) or for preparing documentation for a project created during laboratory work, including:

- justified choice of technology stack for web application development;
- justified choice of web application deployment option;
- instructions for deploying and using the web application.

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 Ihor HAMAIUN
	08.06.2023	Guarantors of the educational programs Andrii KOPP Uliya LITVINOVA