FUNDAMENTALS OF PROTOTYPING

COURSE SYLLABUS

Code and name of specialty	121 Software Engineering 122-Computer Science 126-Information Systems and technologies	**Institute / faculty**	Faculty of Computer Science and Software Engineering
Program name	Software Engineering Computer Science and Intelligent Systems Information Systems Software	**Department**	Software Engineering and Management Information Technologies
Type of program	Educational and Professional	**Language of instruction**	Ukrainian, English

LECTURER

Oleksandr Shmatko ***oleksandr.shmatko@khpi.edu.ua***

Ph.D., Ass. Prof. at the Department of Software Engineering and Management Information Technologies of NTU «KhPI». Prepared and published more than 70 publications, 3 collective monographs, 2 textbooks with the University stamp, 8 articles in publications indexed in Scopus. (h-index = 7, i10-index = 4 in Google Academy-https://scholar.google.com/citations?user=Wyv6esuaaaaaj&hl=ru; ORCID iD-https://orcid.org/0000-0002-2426-900x)

Leading lecturer of courses: *Software modeling and analysis (bachelors) (in English and Ukrainian), Advanced technologies and areas of development of intelligent software systems (Masters) (in English and Ukrainian), Modern technologies of web application development (PhD) (in Ukrainian)*

GENERAL DESCRIPTION OF THE COURSE

Summary	The course "Fundamentals of prototyping" is an academic discipline from the profiled package of disciplines 02 "Software Development and Startup". It is taught in the seventh semester in the amount of 120 hours.(4 ECTS credits), in particular: lectures – 16 hours., laboratory works – 16 hours., independent work-88 hours. The course provides two content modules and one modular Control work. The discipline ends with a test.
Course objectives	Training students in the methodology of System Analysis and modeling, which allow at the stage of creating software to solve the following main tasks: providing the necessary functionality of the software and adaptability to constantly changing conditions of its functioning; designing data objects implemented in the system; designing interface tools (screen forms, reports) that will ensure the execution of data requests; choosing a specific environment or technology for project implementation.
Types of classes and control	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment, control work. Final assessment – tests.
Term	7

Student workload (credits) / Type of course	4 / elective	**Lectures (hours)**	16	**Laboratory classes (hours)**	16	**Self-study (hours)**	88

Program competences	*General competencies:* Ability to abstract thinking, analysis and synthesis. Ability to apply knowledge in practical situations.

	Knowledge and understanding of the subject area and understanding of professional activity. Ability to learn and master modern knowledge. Ability to work in a team. Ability to search, process and summarize information from various sources. Ability to develop and manage projects. Ability to evaluate and ensure the quality of work performed.*Professional competencies of the speciality:* 121-PC13. Ability to identify, classify and formulate software requirements. 121-PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes. 122-PC8. 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 management mechanisms. 122-PC9. Ability to implement a multi-tier computing model based on the client-server architecture, including databases, knowledge bases, 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. 122-PC10. Ability to apply methodologies, technologies, and tools to manage the life cycle processes of information and software systems, information technology products and services according to customer requirements. 122-PC12. Ability to ensure the organization of computational processes in information systems of various purposes, taking into account the architecture, configuration, performance indicators of operating systems and system software 122-PC19. Ability to comprehensively use for the creation of intelligent management systems methods of mathematical modelling and analysis of complex systems, methods of modelling and analysis of business processes, information technologies for the management of business systems. 122-PC20. Ability to develop the architecture of software systems and their particular components during the design of intelligent management systems in various fields, to manage the life cycle of intelligent management systems software. 126-PC 1. Ability to analyze the object of design or operation and its subject area. 126-PC 2. Ability to apply standards in the field of information systems and technologies in the development of functional profiles, construction and integration of systems, products, services and infrastructure elements of the organization. 126-PC 4. Ability to design, develop and use tools for the implementation of information systems, technologies and infocommunications (methodological, informational, algorithmic, technical, software and others).

Learning outcomes	**Teaching and learning methods**	**Forms of assessment (continuous assessment CAS, final assessment FAS)**
121-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. 121-PO03. Know the basic processes, phases and iterations of the software life cycle. 121-PO09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements. 121-PO11. Choose source data for design, guided by formal methods of describing requirements and modelling. 121-PO23. Be able to document and present the results of	Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning	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)

software development.		
122-PLO9. Develop software models of subject areas, choose a programming paradigm from the standpoint of convenience and quality of its application to implement methods and algorithms that solve problems in the computer science field.		
122-PLO10. Use tools for developing client-server applications, design conceptual, logical, and physical models of databases, develop and optimize database queries, create distributed databases, repositories and showcases of databases, and knowledge bases, including those based on cloud services, using web programming languages.		
122-PLO11. Have the skills to manage the life cycle of software, products, and services of information technology under the requirements and restrictions of the customer, be able to develop project documentation (feasibility study, technical task, business plan, agreement, contract).		
122-PLO19. Create intelligent management systems using methods of mathematical modelling and analysis of complex systems, methods of modelling and analysis of business processes, information technologies for the management of business systems.		
122-PLO20. Develop the architecture of software systems and their particular components during the construction of intelligent management systems in various fields, as well as manage the life cycle of intelligent management systems software.		
122-PLO11. Have the skills to manage the life cycle of software, products, and services of information technology under the requirements and restrictions of the customer, be able to develop project documentation (feasibility study, technical task, business plan, agreement, contract).		
126-PLO 3. To use basic knowledge of informatics and modern information systems and technologies, programming skills, technologies of safe work in computer networks, methods of creation of databases and Internet resources, technologies of development of algorithms and computer programs in high-level languages with application of project-oriented programming to solve problems of design and use of information systems and technologies.		

126-PLO 8. Apply the rules of design materials of information systems and technologies, know the composition and sequence of design work, taking into account the requirements of relevant legal documents for implementation in professional activities.		

ASSESSMENT AND GRADING

Ranges of points corresponding to grades	**ıl score (points) for all types of learning activities**	**ECTS grading scale**	**The national grading scale**	**Allocation of grade points**	100% final assessment in the form of an exam (30%) and a current assessment (70%). 30% exam: semester exam, according to the schedule of the educational process 70% current rating: * 40% evaluation of tasks in laboratory work; * 30% intermediate control (2 modular Control works)
	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)		
	0-34Students are required to attend classes according to the schedule and adhere to the ethics of behavior. In case of absence, students will need to complete all tasks to compensate for missed classes. Participation in practical classes requires preliminary preparation and early processing of all the necessary materials for productive discussions during the lesson. Written assignments must be submitted before the deadline.	F	Unsatisfactory (with mandatory repetition of the course)		

Course policy	Students are required to attend classes according to the schedule and adhere to the ethics of behavior. In case of absence, students will need to complete all tasks to compensate for missed classes. Participation in practical classes requires preliminary preparation and early processing of all the necessary materials for productive discussions during the lesson. Written assignments must be submitted before the deadline.

COURSE STRUCTURE AND CONTENT

Topic 1	Introduction to the design process. Design constraints and choices.	**Laboratory work 1**	Create early design ideas - scripts, actors, and storyboards.	**Self-study**	Create design views according to individual tasks.
Topic 2	Conceptual foundations of user interaction by introducing the basic building blocks of user interaction.	**Laboratory work 2**	Create low-and high-precision prototypes to evaluate and improve the design.		Create low-and high-precision prototypes according to an individual task.
Topic 3	Low-to high-precision prototyping. Various forms of prototyping.	**Laboratory work 3**	Creating a functional prototype of a low-precision design.		Create a functional prototype of a low-precision design according to an individual task.
Topic 4	Wireframe design of the main program screens.	**Laboratory work 4**	Explore the features and limitations of interface formats.		Explore the possibilities and limitations of interface formats according to an individual task.
Topic 5	Conceptual issues of prototyping and design. Prototyping and mental models.				
Topic 6	Functional prototype design with low precision.				

RECOMMENDED READING

Compulsory

1. Barrier, T. (2002). Human Computer Interaction Developments and Management. IRM Press, 336 p.
2. Becker Christopher Reid. Learn Human-Computer Interaction: Solve human problems and focus on rapid prototyping and validating solutions through user testing / Packt, 2020, 322 p.
3. Dahl, D. A. (Ed.). (2017). Multimodal Interaction with W3C Standards: Toward Natural User Interfaces to Everything. *Springer International Publishing*. Switzerland, 430 p.
4. Dasgupta Ritwik. (2019). Voice User Interface Design: Moving from GUI to Mixed Modal Interaction. Apress, 108 p.
5. Mara Andrew. (2021). UX on the Go: A Flexible Guide to User Experience Design. Routledge. Taylor & Francis, 242 p.

Recommended

6. Варфел Тодд. (2013). Прототипирование: практич. руководство. Москва: Манн, Иванов и Фербер, 389 с.
7. Marian, P., van der Hoek, A. (2016). Software Design Decoded: 66 Ways Experts Think. Cambridge (Mass.); London: The MIT Press.
8. Utesheva Anastasia. (2020). Designing Products for Evolving Digital Users: Study UX Behavior Patterns, Online Communities, and Future Digital Trends. Apress Media LLC., 128 p.
9. Нейгард Майкл. (2016). Release it! Проектирование и дизайн ПО для тех, кому не все равно. Санкт-Петербург: Питер, 320 с.
10. Scheible, J., Tuulos, V. (2017). Mobile Python: Rapid prototyping of applications on the mobile platform. Wiley Publshing, Inc., 348 c.

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.