FUNDAMENTALS OF JAVA PROGRAMMING

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

Full name, e-mail	**Lev Ivanov, *Lev.Ivanov@khpi.edu.ua***

Senior Lecturer at the Department of Software Engineering and Management Information Technologies. Work experience – since 1981. Author (co-author) of more than 30 research papers and textbooks (Google Scholar: https://scholar.google.com/citations?user=ADPHLAsAAAAJ).

Leading lecturer of the courses: “Programming Basics (Part 1, Part 2) (English and Ukrainian)”, “Object-oriented programming. Introductory practice” (English and Ukrainian), “Fundamentals of Java programming” (English and Ukrainian).

GENERAL DESCRIPTION OF THE COURSE

Summary	The course “Fundamentals of Java programming” is a course in the cycle of professional selective training. It is taught in the third semester in the amount of 180 hours (6 ECTS credits), in particular: lectures – 32 hours, laboratory classes – 32 hours, independent work – 116 hours. The course includes two content modules. The study of the discipline ends with the test.
Course objectives	Assimilating the syntax of the Java object-oriented programming language, as well as the use of methods and techniques for creating programs using the tools of the Java Standard Edition platform.
Types of classes and control	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment. Final assessment – test.
Term	3

Student workload (credits) / Type of course	6 / Elective	**Lectures (hours)**	32	**Workshops (hours)**	32	**Self-study (hours)**	116

Program competences	*General competencies:* 121-GC01. Ability to abstract thinking, analysis and synthesis. 121-GC01. Ability to abstract thinking, analysis and synthesis. 121-GC02. Ability to apply knowledge in practical situations. 121-GC03. Ability to communicate in the state language both orally and in written form. 121-GC05. Ability to learn and master modern knowledge. 121-GC06. Ability to search, process and analyze information from various sources. 122-GC1. Ability to abstract thinking, analysis and synthesis. 122-GC2. Ability to apply knowledge in practical situations. 122-GC6. Ability to learn and master modern knowledge. 122-GC7. Ability to search, process and analyze information from various sources. 122-GC11. Ability to make justified decisions. 126-GC1. Ability to abstract thinking, analysis and synthesis. 126-GC2. Ability to apply knowledge in practical situations. 126-GC5. Ability to learn and master modern knowledge. 126-GC6. Ability to search, process and summarize information from various sources. *Professional competencies of the specialty:* 121-PC13. Ability to identify, classify and formulate software requirements. 121-PC15. Ability to develop architectures, modules and components of software systems. 121-PC19. Knowledge of information data models, the ability to create software for data storage, retrieval and processing. 121-PC22. Ability to accumulate process and systematize professional knowledge on the creation and maintenance of software and recognition of the importance of lifelong learning. 121-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. 121-PC25. Ability to reasonably select and master software development and maintenance tools. 121-PC26. Ability to algorithmic and logical thinking. 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-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. 126-PC1. Ability to analyze the object of design or operation and its subject area. 126-PC4. Ability to design, develop and use tools for the implementation of information systems, technologies and infocommunications (methodological, informational, algorithmic, technical, software and others). 126-PC10. Ability to select, design, deploy, integrate, manage, administer and maintain information systems, technologies and infocommunications, services and infrastructure of the organization.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
121-PO03. Know the basic processes, phases and iterations of the software life cycle	Interactive lectures with presentations, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
121-PO08. Be able to develop a human-machine interface		
121-PO23. Be able to document and present the results of software development.		
122-PLO1. 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		
122-PLO5. Design, develop and analyze algorithms for solving computational and logical problems, evaluate the efficiency and complexity of algorithms based on the use of formal models of algorithms and computational functions.		
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.		
126-PLO 2. Apply knowledge of basic and natural sciences, systems analysis and modeling technologies, standard algorithms and discrete analysis in solving problems of design and use of information systems and technologies.		
126-PLO 6. Demonstrate knowledge of the current level of information systems technology, practical skills of programming and use of applied and specialized computer systems and environments for their implementation in professional activities.		
126-PLO 7. Justify the choice of technical structure and develop appropriate software that is part of information systems and technologies.		

ASSESSMENT AND GRADING

Ranges of points corresponding to grades	score (points) for all types of learning activities	ECTS grading scale	The national grading scale	Allocation of grade points	
	90-100	A	excellent		**100% Final assessment** as a result of Final exam (20%) and Continuous assessment (80%). **20% Final test:** semester test, according to the schedule of the educational process **80% Continuous assessment**: • 70% of assessment of tasks in laboratory works; • 10% intermediate control (colloquium)
	82-89	B	good		
	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	Encapsulation, Inheritance, and Polymorphism in Java	**Laboratory work 1**	Using Encapsulation, Inheritance, and Polymorphism in Java	**Self-study**	Disadvantages and advantages of using lambda expressions.
Topic 2	Work with generics and collections	**Laboratory work 2**	Using container classes		General container functions
Topic 3	Working with files in Java	**Laboratory work 3**	Working with exceptions and files in Java		Disadvantages and advantages of using exceptions
Topic 4	graphical user interface applications	**Laboratory work 4**	Create graphical user interface programs		Implementation of event-driven programming in Java FX
Topic 5	Reflection and metaprogramming. Multithreading	**Laboratory work 5**	Use of reflection and metaprogramming.		Interpretation and compilation of code in metaprogramming

RECOMMENDED READING

Compulsory

1. Bloch, J. (2017). Effective Java: 3rd Edition, Addison Wesley, 412 p.
2. Schildt, H. (2018). Java: A Beginner's Guide: 8th Edition. McGraw-Hill Education, 684 p.
3. Schildt, H. (2018). Java: The Complete Reference: 11th Edition. McGraw-Hill Education, 1208 p.
4. Horstmann, C. S. (2018). Core Java Volume I Fundamentals: 11th Edition. Prentice Hall, 889 p.
5. Horstmann, C. S. (2017). Core Java SE 9 for the Impatient: 2nd Edition Addison-Wesley Professional, 576 p.
6. Eckel, B. (2006). Thinking in Java 4th Edition: Pearson, 1150 p.
7. Deitel, P., Deitel, H. (2017). Java How to Program, Early Objects: 11th Edition. Pearson, 1296 p.
8. Deitel, P., Deitel, H. (2017). Java How To Program, Late Objects: 11th Edition. Pearson, 1248 p.
9. Ратушняк, Т. В. (2017). Програмування мовою JAVA: практикум: навч. посібник / Державна фіскальна служба України, Університет державної фіскальної служби України. Ірпінь, 212 с.

Recommended

1. Deitel, P., Deitel, H. (2017). Java 9 for Programmers: 4th Edition. Pearson, 1120 p.
2. Копитко, М. Ф., Іванків, К. С. (2002). Основи програмування мовою Java: тексти лекцій. Львів: Видавничий центр ЛНУ ім. І. Франка, 83 с.
3. Брнакевич, І. Є., Вагін, П. П. (2002). Програмування мовою Java: використання фундаментальних класів: тексти лекцій. Львів: Видавничий центр ЛНУ ім. І. Франка, 75 с.
4. Horstmann, C. S. (2019). Core Java, Volume II. Advanced Features (Core Series): 11th Edition. Pearson, 1040 p.

INTERNET RESOURCES

1. Java Tutorials. [Electronic resource]. Access mode: http://docs.oracle.com/javase/tutorial
2. Java Tutorial. [Electronic resource]. Access mode: http://www.java2s.com/Tutorial/Java/CatalogJava.htm
3. Bruce Eckel. Thinking in Java, 4th Edition. [Electronic resource]. Access mode: http://sd.blackball.lv/library/Thinking_in_Java_4th_edition.pdf
4. Java programming notes. [Electronic resource]. Access mode: http://leepoint.net/notes-java
5. Освоюємо Java: Вікіпідручник. [Електронний ресурс]. Ресурс доступу: http://uk.wikibooks.org/wiki/Освоюємо_Java
6. Програмування на Java. [Електронний ресурс]. Ресурс доступу: http://javaland.com.ua
7. Брнакевич, І. Є., Вагін, П. П. Програмування мовою Java: використання фундаментальних класів: тексти лекцій. [Електронний ресурс]. Ресурс доступу: http://blues.franko.lviv.ua/ami/books/ami/Java_fundamental.pdf

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.