

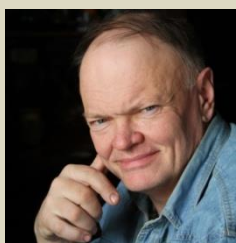
# PROGRAMMING BASICS

## 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

## LECTURER

**Full name, e-mail** Lev Ivanov, [Lev.Ivanov@khpi.edu.ua](mailto: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

<b>Summary</b>	The course "Programming Basics" is a course in the cycle of professional compulsory training of the specialty 121 "Software Engineering". It is taught in the first semester in the amount of 300 hours (10 ECTS credits), in particular: lectures – 80 hours, laboratory classes – 80 hours, independent work – 140 hours. The course includes four content modules and one course work. Both parts of discipline are completed by exams.
<b>Course objectives</b>	Assimilation of the necessary knowledge of mastering the theoretical bases of programming languages C++ and Java and the acquisition of practical skills of their use during the development of programs based on the principles of structural, procedural-oriented and object-oriented programming.
<b>Types of classes and control</b>	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment, course work. Final assessment – exams.
<b>Term</b>	1, 2

<b>Student workload (credits) / Type of course</b>	10 / Mandatory	<b>Lectures (hours)</b>	80	<b>Workshops (hours)</b>	80	<b>Self-study (hours)</b>	140
--	----------------	-------------------------	----	--------------------------	----	---------------------------	-----

<b>Program competences</b>	<p>GC01. Ability to abstract thinking, analysis and synthesis.                  GC02. Ability to apply knowledge in practical situations.                  GC05. Ability to learn and master modern knowledge.                  GC06. Ability to search, process and analyze information from various sources.                  PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes.                  PC25. Ability to reasonably select and master software development and maintenance tools.                  PC26. Ability to algorithmic and logical thinking.</p>
----------------------------	--

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
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.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS), course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO07. Know and apply in practice the fundamental concepts, paradigms and basic principles of operation of language, tools and computing software engineering	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS), course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO08. Be able to develop a human-machine interface	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS), course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO15. Being motivated to choose programming languages and development technologies to solve problems of software design and maintenance	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS), course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)

## 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	
	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)	

**100% Final assessment** as a result of Final exam (20%) and Continuous assessment (80%).  
**20% Final exam:** semester exam, according to the schedule of the educational process  
**80% Continuous assessment:**

- 50% of assessment of tasks in laboratory works;
- 10% intermediate control (colloquium)
- 20% Course work

**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

<b>Topic 1</b>	Basics of algorithmization	<b>Laboratory work 1</b>	Development of algorithms	<b>Self-study</b>	Graphical representation of cyclic algorithms
<b>Topic 2</b>	Basic C++ syntax	<b>Laboratory work 2</b>	C++ operators and statements		Comparison of cyclic structures C++
<b>Topic 3</b>	Functions and references	<b>Laboratory work 3</b>	Using functions		Using static local variables
<b>Topic 4</b>	Arrays and pointers	<b>Laboratory work 4</b>	Using arrays and pointers		Disadvantages and advantages of address arithmetic
<b>Topic 5</b>	Working with files and strings	<b>Laboratory work 5</b>	Working with strings, structures, and files		Differences between text files and binary files
<b>Topic 6</b>	Modular programming	<b>Laboratory work 6</b>	Pointers to functions and header files		Using namespaces
<b>Topic 7</b>	User Defined Types	<b>Laboratory work 7</b>	Creation and use of C++ classes		Purpose and advantages of handling exceptions
<b>Topic 8</b>	Inheritance, polymorphism, and templates in C++	<b>Laboratory work 8</b>	Using polymorphism and templates in C++		Compilation time and runtime polymorphism
<b>Topic 9</b>	Using the Standard C++ library	<b>Laboratory work 9</b>	Working with containers, algorithms, and iterators		Application of new C++ syntax features
<b>Topic 10</b>	Using basic Java	<b>Laboratory work 10</b>	Procedural programming in Java		The main problems of cross-platform programming
<b>Topic 11</b>	Reference types in Java	<b>Laboratory work 11</b>	Working with arrays and strings. Creation of Java classes		Differences between Java and C++ classes

## RECOMMENDED READING

### Compulsory

1. Stroustrup, B. (2013). The C++ Programming Language. (4th ed.). Addison-Wesley.
2. Lippman, S. B., Lajoie, J., Moo, B. E. (2011). C++ Primer. (6th ed.). Addison-Wesley Professional.
3. Schildt, H. (2002). C++: The Complete Reference. (4th ed.). McGraw-Hill Education.
4. Трофименко, О. Г. , Прокоп, Ю. В., Логінова, Н. І., Задерейко, О. В. (2019). С++. Алгоритмізація та програмування : підручник. (2-ге вид.). Одеса: Фенікс.
5. Пекарський, Б. Г. (2018). Основи програмування: навчальний посібник. Кондор.
6. Джордж Хайнеман, Гері Полліс, Стенлі Селков. (2017). Алгоритми. Довідник з прикладами на С, С++, Java і Python. Діалектика.
7. Грицюк, Ю. І., Рак, Т. Є. (2011). Програмування мовою С++: навчальний посібник. Львів : Вид-во Львівського ДУ БЖД.
8. Schildt, H. (2018). Java: A Beginner's Guide. (8th ed.). McGraw-Hill Education.
9. Horstmann, C. S. (2018). Core Java. Vol. I.–Fundamentals. (11th ed.). Prentice Hall.
10. Ратушняк, Т. В. (2017). Програмування мовою JAVA: практикум: навчальний посібник. Державна фіскальна служба України, Університет державної фіскальної служби України. Ірпінь.

### Recommended

1. Eckel, B. (2000). Thinking in C++. Vol. 1: Introduction to Standard C++. (2nd ed.). Prentice Hall.
2. Eckel, B. (2006). Thinking in Java. (4th ed.). Pearson.
3. Bloch, J. (2017). Effective Java. (3rd ed.). Addison Wesley.
4. Воловщиков, В. Ю., Иванов, Л. В., Рубін, Е. Ю., Гончаренко, Т. Г. (2017). Мова С++ в програмуванні та комп'ютерних науках. Харків: ФОРМ Мезіна В.В.

### INTERNET RESOURCES

1. International Standard ISO/IEC 14882:2014(E)-Programming Language C++. Retrieved from <https://isocpp.org/std/the-standard>
2. C/C++ language and standard libraries reference. Retrieved from <https://msdn.microsoft.com/en-us/library/hh875057.aspx>
3. Освоюємо Java-Вікіпідручник. Retrieved from [http://uk.wikibooks.org/wiki/Освоюємо\\_Java](http://uk.wikibooks.org/wiki/Освоюємо_Java)
4. Програмування на Java. Retrieved from <http://javaland.com.ua>
5. Брнакевич, І. Є., Вагін, П. П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. Retrieved from [http://blues.franko.lviv.ua/ami/books/ami/Java\\_fundamental.pdf](http://blues.franko.lviv.ua/ami/books/ami/Java_fundamental.pdf)
6. Начинающим Java программистам. Retrieved from <http://habrahabr.ru/blogs/java/43293/>
7. Java Tutorials. Retrieved from <http://docs.oracle.com/javase/tutorial> (англ.)
8. Java Tutorial. Retrieved from <http://www.java2s.com/Tutorial/Java/CatalogJava.htm>
9. Bruce Eckel. Thinking in Java. (4th ed.). Retrieved from [http://sd.blackball.lv/library/Thinking\\_in\\_Java\\_4th\\_edition.pdf](http://sd.blackball.lv/library/Thinking_in_Java_4th_edition.pdf) (англ.)
10. Java programming notes. Retrieved from <http://lepoint.net/notes-java>

### 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.