



## Syllabus Course Program



# Algorithmization and Programming

### Specialty

122 – Computer Science

### Institute

Institute of Computer Modelling, Applied Physics and Mathematics

### Educational program

Computer Science. Modeling, Design, and Computer Graphics

### Department

Computer Modeling of Processes and Systems (162)

### Level of education

Bachelor's degree

### Course type

Special (Professional), Mandatory

### Semester

1

### Language of instruction

English

---

## Lecturers and course developers



### Senko Alyona

[Alyona.Senko@khpi.edu.ua](mailto:Alyona.Senko@khpi.edu.ua)

Doctor of Philosophy (PhD), senior lecturer

The main scientific results were obtained in the direction of computer modeling of the processes of high-temperature fracture of structural elements.

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



### Oleksiy D'yomin

[oleksii.domin@khpi.edu.ua](mailto:oleksii.domin@khpi.edu.ua)

Postgraduate student of the Department of Mathematical Modeling and Intelligent Computing in Engineering at NTU 'KPI'

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

## General information

### Summary

The course is designed to develop students' knowledge, skills, and abilities to program in high-level languages and write simple algorithms. The structural and logical part of the C++ language is studied. The basic control operators of the language, work with arrays and structures are considered. The basic operations of the language, working with basic and derived types, programming data in statically and dynamically allocated memory are taught. The work with functions is analyzed. Algorithms for working with numerical sequences, one-dimensional and two-dimensional arrays are considered

## Course objectives and goals

To develop students' knowledge, skills and abilities necessary for understanding and rational use of concepts, methods and algorithms of programming as a subject of study and as a means of studying other subject areas, in particular, object-oriented programming, numerical methods, etc. Learn to effectively apply the theoretical framework and practical skills of creating program modules for program development.

## Format of classes

Lectures, laboratory work, independent work, consultations. The final control is an exam.

## Competencies

GC06. Ability to think abstractly, analyze and synthesize.

PC04. Ability to develop algorithms and data structures, software tools and program documentation.

PC08. Ability to use modern technologies of programming and software testing.

## Learning outcomes

PO04. Perform mathematical description, analysis and synthesis of discrete objects and systems using the concepts and methods of discrete mathematics and algorithm theory.

PLO05. To be able to develop and use in practice algorithms related to the approximation of functional dependencies, numerical differentiation and integration, solving systems of algebraic, differential and integral equations, solving boundary value problems, finding optimal solutions.

PO11. Be able to apply modern technologies of programming and software development, software implementation of numerical and symbolic algorithms.

## Student workload

The total volume of the discipline is 180 hours (6 ECTS credits): lectures - 32 hours, laboratory work - 48 hours, independent work - 100 hours.

## Course prerequisites

Knowledge of the school curriculum in mathematics, general computer skills and the ability to think analytically.

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

For the labs, we recommend using the free integrated development environment (IDE) Code:: Blocks and online software Draw.io (<https://www.drawio.com>) to create flowcharts.

Lectures are conducted interactively with the use of multimedia technologies, where explanatory and illustrative method, critical thinking method and discussions are used to present the theory and analyze the code. Laboratory work emphasizes practical and partially exploratory methods, which contributes to the development of practical programming skills and solving specific problems.

Learning materials are available to students on Microsoft OneDrive.

## Program of the course

### Topics of the lectures

**Topic 1: Concepts and basic approaches of structural logic programming. Introduction to the C language as a basic component of the C++ language**

The concept and basic approaches of structural and logic programming.

A simple program in C as a basic component of C++

**Topic 2. Development of algorithms and flowcharts. The concept of a loop and conditional execution. Programming of numerical sequences. Arrays in C**

The concept of algorithm and flowcharts.

The concept of a loop and conditional execution.

Input and output operators in C. Formats.  
Programming of numerical sequences.  
Arrays in the C language

### Topic 3. Basic data types. Constants. Blocks. Visibility rules

Basic data types. Constants. Visibility rules.  
Flowcharts. A graphical way to describe algorithms.  
Algorithm for determining the minimum positive element of an array

### Topic 4. The system of data types. Expressions .Operations in C. Two-dimensional arrays - processing algorithms

Expressions. Operations. Two-dimensional arrays. Algorithm for processing two-dimensional arrays.

### Topic 5. Working with text files.

Working with text files. Examples of data processing programs

### Topic 6. Functions in the C language.

Functions in C. Declaring a function. Scope and scope. Localization rules. Parameters of functions.  
Examples of programs with functions.

### Topic 7. Working with symbolic variables.

Working with character variables. Arrays of characters. Strings. Functions for working with strings.  
Examples of string processing.

### Topic 8: Dynamic memory allocation. Programming work with one-dimensional arrays in dynamically allocated memory

Pointers in the c language. Operations on pointers. The relationship between arrays and pointers.  
Dynamic memory allocation. Arrays in dynamically allocated memory. One-dimensional arrays as parameters of functions. Two-dimensional arrays in dynamically allocated memory

### Topic 9: Dynamic memory allocation. Continuation.

Dynamic memory allocation. Programming the operation of functions with two-dimensional arrays in dynamically allocated memory

### Topic 10. Passing arrays to functions

Passing arrays to functions. Examples of programs

### Topic 11: User-defined types.

User-defined types. Structures. Arrays of structures. Structures and fields of structures as parameters of functions

### Topic 12: User-defined types. Continuation

Bit operations. Bit fields. Unions. Data type - enumeration. Renaming types -typedef

### Topic 13. Preprocessor directives. Macros

The language preprocessor c. Directives. Macros.

### Topic 14. Developing algorithms for programs that use multiple functions

Development of algorithms for programs that use several functions. Example 1. Example 2

### Topic 15. Developing programs that use several functions that return values using pointers.

Developing programs that use several functions that return values using pointers. Example 1. Example 2.

### Topic 16. Designing programs based on a modular approach

Designing programs based on a modular approach. Modular programming. Advantages and disadvantages of modular programming

## Topics of the workshops

It is not provided by the curriculum.

## Topics of the laboratory classes

Topic 1: Introduction to the integrated program development environment. Creating a simple program.

Topic 2. Programs that use a conditional statement. Branching of the algorithm.

Topic 3. Programs that use loop statements. Development of flowcharts.

Topic 4. Working with one-dimensional arrays. Finding the minimum (maximum) elements, other characteristics of arrays, etc.

Topic 5. Working with two-dimensional arrays and text files. Finding the minimum (maximum) elements, other characteristics of arrays, etc.

Topic 6. Functions in C. Function parameters. Typed functions

Topic 7. Working with character arrays.

Topic 8: Using functions to work with strings.

Topic 9. Working with pointers. Functions that return values using pointers.

Topic 10. Functions that operate with arrays.

Topic 11. Location of variables in dynamically allocated memory. Programs with one-dimensional arrays in dynamic memory as function parameters.

Topic 12: Location of variables in dynamically allocated memory. Programs with two-dimensional arrays in dynamic memory.

Topic 13. Programs with one-dimensional and two-dimensional arrays in dynamic memory as function parameters. Development of flowcharts.

Topic 14: Programming of structures.

## Self-study

The subject of programming. A brief history of the development of programming.

Input and output operators in C. Formats.

Application of C programming skills for programming tables.

Standard libraries in C.

Recursive functions.

Application of structures to create simple databases.

Working with unions.

Access to individual bits.

Performing an individual task (10 points). Create a flowchart of the algorithm and write a program for processing two-dimensional arrays

## Course materials and recommended reading

Modern C for Absolute Beginners. A Friendly Introduction to the C Programming Language. Slobodan D. Mitrovic. 2023. - 350.

2. Programming for Problem-solving with C. Formulating algorithms for complex problems. / Kamaldeep. 2023. - 480

3. Mastering C. A Comprehensive Guide to Programming Excellence. 2023. - 188
4. Maxwell Rivers C++ Programming for Beginners. 2023. - 200
5. Ryan Campbell. Computer Programming- C++ Programming. 2023. - 112
6. Prachi Manish Patil. Code in Style - The C Programming. 2023. - 198

## Assessment and grading

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

Points are awarded according to the following ratio:

- defense of all labs: 60 points
- defense of the calculation task: from 0 to 10 points

The exam consists of two parts:

- theoretical and practical part of the exam: from 0 to 20 points
- program part of the exam with algorithms of increased complexity (after successful completion of the 1st part and obtaining at least 15 points): from 0 to 10 points

### 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<br>(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

Date, signature

**Head of the department**  
Dmytro BRESLAVSKYI

Date, signature

**Guarantor of the educational program**  
Gennadiy LVOV