MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
Lesya Ukrainka Volyn National University
Faculty of Information Technologies and Mathematics
Department of Computer Science and Cybersecurity

SYLLABUS
Normative Educational Component
ALGORITHM DEVELOPMENT METHODOLOGY
Preparations at the Second (Master’s) Level of Higher Education
Specialty: F3 Computer Science
Educational (Professional) Program: Computer Science and Information Technologies

Lutsk – 2025

Syllabus of the educational component «Algorithm Development Methodology» for Master's level training, field of knowledge F Information Technology, specialization F Computer Science, within the educational program Computer Science and Information Technology.

Developer: T.O. Hryshanovych, Associate Professor of the Department of Computer Science and Cybersecurity, PhD.

Approved by
Program Educational Guarantor: Bulatetskyy V.V.

The syllabus of the educational component was approved at the meeting of the Department of Computer Science and Cybersecurity

protocol № 2 of 17.09.2025

Head of the department: Hryshanovych T.O.

I. Description of the educational component

Name of indicators	**Field of knowledge, specialty, educational-professional** /educational-scientific/educational-creative **program, educational level**	**Characteristic educational component**
Full-time form of education	12 Information Technologies, 122 Computer Science, Computer Science and Information Technology, second (master's)	**Normative**
		Year of study – 1st
Number of hours/credits 150/5		**Semester** – 1st
		Lectures – 28 hours
		Laboratory – 44 hours
IERT: yes		**Independent work** – 10 hours
		Consultations – 68 hours
		Form of control: exam
Language of education – English, Ukrainian		

II. Information about the instructor

PIP: Hryshanovych Tetiana Oleksandrivna
Degree: PhD
Academic status: -
Position: Associate Professor of the Department of Computer Science and Cybersecurity
Contact Information: hryshanovych.tatiana@vnu.edu.ua
Days of classes: http://94.130.69.82/cgi-bin/timetable.cgi

III. Description of the educational component

1. **Abstract of the course.** The syllabus of the educational component "Algorithm Development Methodology" has been developed in accordance with the educational-professional program "Computer Science and Information Technologies" (2024) of the second level of higher education in the field of knowledge 12 - Information Technologies, specializing in 122 - Computer Science. The study of algorithms and, consequently, data structures, is one of the most important aspects of computer science. The discipline "Methods of Algorithm Development" aims to familiarize learners with classical methods of algorithm development, contemporary algorithms and their practical applications, teach them to choose a method for algorithm development based on complexity assessment, develop algorithms for tasks using various approaches. This course is designed to cultivate skills in developing new algorithms to solve non-trivial problems, selecting data structures for optimization, and describing the constraints imposed on the application of certain algorithms or data structures. Specifically, the course involves working directly with approaches to algorithm development, machine learning algorithms, algorithms for handling big data, and genetic algorithms, which are currently relevant.
2. **Prerequisites:** Knowledge of algorithms, data structures, and algorithm complexity at the Bachelor's level in the specialty *Computer Science*.

Postrequisites: IT Project Management, Fuzzy Models and Data Analysis Methods, Computer Systems and Network Administration, Teaching Practice in Higher Education Institutions, Industrial Practice, Pre-graduation Practice with Thesis Writing, Qualification Work.

3. **The purpose and tasks of the educational component**
   Acquisition of knowledge and competencies in the field of algorithm development, their analysis, and optimization; acquisition of skills in using and adapting classical algorithms to solve computer science problems; acquisition of skills in developing new algorithms to solve problems, and in selecting data structures to improve algorithm efficiency.

4. **Learning outcomes (Competencies). Soft skills.**
   Learning outcomes:

LO01. To have specialized conceptual knowledge that includes contemporary scientific achievements in the field of computer science and serves as the foundation for original thinking and research, critical examination of issues in the field of computer science, and at the boundaries of knowledge domains.

LO02. Have specialized computer science problem-solving skills necessary for conducting research and/or conducting innovative activities to develop new knowledge and procedures.

LO11. To develop new algorithms for solving problems in the field of computer science, evaluate their effectiveness, and assess limitations on their application.

LO16. To conduct research in the field of computer science.

LO19. To analyze the current state and global trends in the development of computer sciences and information technologies.

Competencies:

General competences

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. The ability to be critical and self-critical.

GC07. Ability to generate new ideas (creativity).

Special (professional) competences

SC01. Understanding the theoretical foundations of computer science.

SC02. The ability to formalize the subject area of a specific project into an appropriate information model.

SC03. The ability to use mathematical methods for analyzing formalized models of the subject domain.

SC06. The ability to apply existing and develop new algorithms for problem-solving in the field of computer science.

SC10. The ability to assess and ensure the quality of IT projects, information and computer systems of various purposes, apply international standards for software quality assessment in information and computer systems, and models for assessing the maturity of development processes in information and computer systems.

SC11. The ability to initiate, plan, and implement the processes of developing information and computer systems and software, including its design, analysis, testing, system integration, deployment, and maintenance.

Analytical thinking: the ability to break down a complex problem into simpler parts; the ability to see multiple solution paths and choose the optimal one.

Critical thinking: evaluating algorithm efficiency; comparing different data structures in terms of resource usage (time, memory).

Problem-solving: finding unconventional solutions; the ability to bring a problem to completion, even if it involves several “dead ends.”

Time management: working within time constraints (both when solving algorithmic problems and during competitions or exams); planning the stages of problem-solving.

Presentation and argumentation skills: the ability to defend the choice of an algorithm or data structure; explaining solutions in simple terms to a non-technical audience.

Self-organization and continuous learning: constant improvement, as new algorithms and data structures need to be learned throughout one’s professional career; developing the habit of systematic practice.

Creativity: finding new approaches to classical problems; developing original algorithmic ideas.

5. **The structure of the educational component**

Content module 1. The main approaches to the development and analysis of algorithms
Content module 2. Applied algorithms and their complexity

Names of content modules and topics	Total	Lec.	Lab.	IW	Cons.	Control form/ Points
Content module 1. The main approaches to the development and analysis of algorithms						
Topic 1. Introduction. General Overview of Algorithm Development Methods.	8	2		6		DS
Topic 2. Algorithm Development Using the Method of Partial Goals. Brute Force Method.	8	2	2	4		SP/C 3
Topic 3. Algorithm Development Using Dynamic Programming.	8	2	2	4		SP/C 3
Topic 4. Algorithm Development Using the Iterative Method.	8	2	2	4		SP/C 3
Topic 5. Algorithm Development Using the Backtracking.	8	2	2	4		SP/C 3
Topic 6. Algorithm Development Using Solution Trees.	10	2	2	4	2	SP/C 3

Topic 7. Algorithm Development Using the Trial and Error Method.	8	2	2	4		SP/C 3
Topic 8. Algorithm Development Using the Branch and Bound Method, the Alpha-Beta Pruning Method.	16	2	6	8		SP/C 9
Topic 9. Development of Greedy Algorithms.	6		2	4		SP/C 3
Topic 10. Evaluation of Algorithm Complexity. Classes of Algorithm Complexity.	10	2	2	4	2	SP/C 3
Module 1 Total	92	18	22	46	4	33
Content module 2. Applied algorithms and their complexity						
Topic 12. Algorithms on Graphs.	10	2	6	4		SP/C 9
Topic 13. Maze Generation Algorithms. Algorithms for Finding the Exit from a Maze.	10	2	4	4		SP/C 6
Topic 14. Machine Learning Algorithms.	12	2	4	4	2	SP/C 6
Topic 15. Algorithms for Big Data Processing.	14	2	4	6	2	SP/C 8
Topic 16. DNA Sequence Algorithms.	12	2	4	4	2	SP/C 8
Module 2 Total	58	10	22	22	6	37
Types of final works						Points
Modular control work 1						10
Modular control work 2						10
IERT						10
Total hours/Points	150	28	44	68	10	100

Control methods*: DS – discussion, SP/C – solving problems/cases, IERT – individual task, MCW – modular control work.

6. **Tasks for independent study**

Independent work of students includes:

Studying lecture materials. 10 hrs.

Assessment is carried out during laboratory sessions and is evaluated when assigning grades for the content module.

Preparation for practical sessions, completing homework assignments. 20 hrs.

Assessment is conducted during practical sessions.

Organizing and reviewing the material learned before the exam. 10 hrs.
Assessment is conducted during the exam.

Studying topics that are not covered in the lecture course. 10 hrs.
Assessment is conducted during module assessment activities and is evaluated based on the corresponding number of points.

Preparation IERT. 18 hrs.
Assessment is carried out during the submission of individual assignments.

№	**Topic**	**Hours**
1	Geometric algorithms.	2
2	BlockChain.	2
3	Heuristics algorithms and metaheuristics.	2
4	Matrix multiplication algorithms. Parallel computing processes.	2
5	Cryptography algorithms/	2

IV. Evaluation policy

Instructor's Policy Regarding the Student's Education.

All participants in the educational process must comply with the current legislation of Ukraine, the Statute and Internal Regulations of Lesya Ukrainka Volyn National University, generally accepted moral principles, rules of conduct, and corporate culture. They must maintain an atmosphere of friendliness, responsibility, integrity, and tolerance. The classroom environment should be creative and open to constructive criticism.

Being late for class is not acceptable, nor is the use of mobile phones, tablets, or other mobile devices during class, or cheating. Students are expected to attend all lectures and laboratory sessions of the course.

Each student must be enrolled in the online course *"Methodology of Algorithm Development"* hosted on the Moodle distance learning platform (https://moodle-cs.vnu.edu.ua/course/view.php?id=148

Academic Integrity Policy

Adherence to academic integrity by higher education seekers includes: independent completion of educational tasks, current and final assessment tasks (for individuals with special educational needs, this requirement is applied with consideration of their individual needs and abilities); citing sources of information when using ideas, statements, or facts; compliance with copyright laws; providing truthful information about the results of their own educational (scientific, creative) activities.

During the assessment of learning outcomes, higher education seekers do not use prohibited means (mobile phones, tablets, notes, educational literature, other sources of information, including internet resources); they independently complete the proposed tasks. When completing laboratory work for the course, higher education seekers have the right to use their own laptops if they support the required software

Policy on Deadlines and Retaking

If a student is absent from classes for any reason, he or she must study the theoretical material independently using textbooks, lecture notes, and materials from the distance learning course *"Methodology of Algorithm Development"* (https://moodle-cs.vnu.edu.ua/course/view.php?id=84) available on the Moodle platform, and complete all homework assignments.

Students can report on the completion of assignments through the *"Methodology of Algorithm Development"* distance course or during consultations, where they can also clarify unclear points and ask questions to the instructor. The use of the course forum is also available.

Retaking tests and control works is **prohibited**. Assignments submitted after the deadline without a valid reason will receive a lower grade.

Recognition of Learning Outcomes Obtained Through Formal, Non-formal, and Informal Education

According to Clause 3.3 of the Regulation on the Recognition of Learning Outcomes Obtained Through Formal, Non-formal, and/or Informal Education at Lesya Ukrainka Volyn National University, the recognition of such learning outcomes is **not provided** for the discipline *"Methodology of Algorithm Development"*.

Possibility of Earning Additional (Bonus) Points

Students have the opportunity to earn additional points in the educational component *"Methodology of Algorithm Development"* by participating in the All-Ukrainian Student Programming Olympiad. A link to the Methodological Council will be provided.

V. Final Assessment

The final form of assessment for the educational component *"Methodology of Algorithm Development"* is an exam. Student performance is evaluated on a 100-point scale. The total grade consists of:

- Ongoing assessment – evaluated through class participation, timely and high-quality completion of homework assignments.
- Final assessment – includes independent individual assignments, control works, testing of theoretical knowledge, and individual projects.

The maximum number of points a student can receive during the semester for ongoing assessment is 70 points.

The maximum number of points for the final assessment is 30 points. Students are required to complete individual assignments. An individual task may consist of a set of problems covering one or several related topics, or a single comprehensive task that requires independent study of several specific topics.

IERT example. Let's assume there is a finite set $Q\{q1,\ q2,\ ...\ qn\}$, and for each $q_i \in Q$, we know the cost c_i and the volume a_i. There is also a knapsack with a volume limit B. The task is to pack the knapsack in such a way that the total value of the items packed is maximized, while their total volume does not exceed B. Traditionally, c_i, a_i, and B are considered non-negative integers. Develop an algorithm to solve the given problem using a greedy algorithm, dynamic programming, and the branch-and-bound method. Evaluate the complexity of the developed algorithms.

If a student accumulates at least 75 points by the end of the semester, and they agree with this result, the grade for the semester may be assigned without taking the exam. Otherwise, the student takes the exam, with the maximum number of points that can be earned on the exam being 60 points. This replaces the module semester control scores, while the continuous semester control is retained.

The exam for the educational component "Algorithm Development Methodology" includes fundamental questions, typical and complex tasks, situations that require creative answers, and the ability to synthesize acquired knowledge and apply it to practical problems.

The exam for the educational component "Algorithm Development Methodology" involves oral responses to theoretical questions and practical task execution. The exam consists of 2 theoretical questions and 1 task. Each task is worth 20 points. The instructor reserves the right to ask clarifying questions during the student's response and may request an explanation of the program code. The student is free to choose the programming language for implementing the practical task.

The questions and form of the exam are defined in this syllabus.

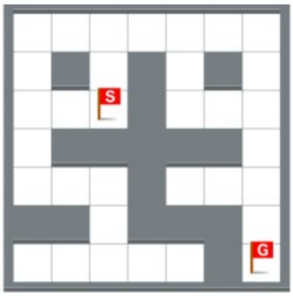
Exam questions:

1. General Overview of Algorithm Development Methods.
2. Method of Partial Goals: General Description, Example Demonstration.
3. Dynamic Programming: General Method Description, Example Demonstration.
4. Hill Climbing Method: General Principle, Example Demonstration.
5. Backtracking Method: General Description, Example Demonstration.
6. Solution Trees: General Method Description, Example Demonstration.
7. Trial and Error Method: General Description, Example Demonstration.
8. Branch and Bound Method: General Description, Example Demonstration.
9. Alpha-Beta Pruning Method: General Description, Example Demonstration.
10. Greedy Algorithms: General Method Description, Example Demonstration.
11. Algorithm Complexity Assessment. Main Approaches to Assessing Complexity.
12. Complexity Classes of Algorithms.
13. Hash Functions. Rules for Formation. Practical Applications of Hash Functions.
14. Blockchain Algorithms.
15. Graph Spanning Tree Algorithms. Greedy Algorithms.
16. Maze Generation Algorithms.
17. Maze Exit Search Algorithms. Their Complexity and Limitations.
18. Shortest Path Algorithms Between Two Graph Nodes. Greedy Algorithms.
19. Knapsack Problem: Solution Methods.
20. Machine Learning Algorithms: Supervised, Unsupervised, and Reinforcement Learning.
21. Big Data Algorithms: Exploratory Data Analysis, Linear Classification (Perceptron & Logistic Regression), Linear Regression, C4.5, Decision Tree, Apriori, K-means Clustering, EM Algorithm, PageRank & HITS, Collaborative Filtering.
22. DNA Sequence Algorithms.
23. Matrix Multiplication Algorithms. Parallelization of the Algorithm.

24. Similar Document Retrieval Problem: Main Solution Methods, Examples.

Examples of exam tasks and cases

Given a maze, white cells represent free paths, and gray cells represent blocked ones. The 'S' symbol marks the starting point, while 'G' indicates the exit from the maze. Using the wave algorithm, construct an algorithm to exit the maze.



VI. Rating scale

A scale for evaluating the knowledge of education seekers from educational components, where the form of control is an exam

Rating in points	**Linguistic assessment**	**Evaluation on the ECTS scale**	
		rating	**explanation**
90–100	Perfect	A	excellent performance
82–89	Very good	B	above average level
75–81	Good	C	overall good job
67–74	Satisfactorily	D	not bad
60–66	Enough	E	performance meets the minimum criteria
1–59	Unsatisfactorily	Fx	Recompletion is required

VI. Recommended literature and Internet resources

1. Hryshanovych T. O. Alhorytmy ta struktury danykh: navchalnyi posibnyk. Lutsk : VNU im. Lesi Ukrainky, 2021. 150 s. (Rekomendovano Vchenoiu radou VNU imeni Lesi Ukrainky, protokol №13 vid 28.12.2021)
2. Hryshanovych T. O. Laboratornyi praktykum z dyscypliny «Alhorytmy ta struktury danykh» dlia studentiv spetsialnosti 122 Kompiuterni nauky. Elektronnyi resurs, 2021. 49 s. Rezhym dostupu https://evnuir.vnu.edu.ua/handle/123456789/20006
3. Alhorytmy ta struktury danykh: navchalnyi posibnyk / Kovalenko O. O., Tkachenko O. M., Chekhmestruk R. Yu., Vinnytsia : VNTU, 2025. 114 s.

4. Krenevych A. P. Alhorytmy i struktury danykh: Pidruchnyk. Kyiv : VPTs "Kyiv. Un-t", 2021. 200 s.
5. Kublii L. I. Alhorytmy ta struktury danykh. Osnovy alhorytmizatsii: Pidruchnyk dlia zdobuvachiv stupenia bakalavra za spetsialnistiu 121 «Inzheneriia prohramnoho zabezpechennia». Kyiv : KPI im. Ihoria Sikorskoho, 2022. 528 s.
6. Padalko H., Chumachenko D., Kaur N., Investigation of machine learning approaches to classify war-related content during Russian full-scale invasion of Ukraine. ORIGINAL ARTICLE. Social Network Analysis and Mining (2025), https://doi.org/10.1007/s13278-025-01501-3