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

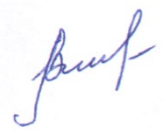
SYLLABUS

Normative Educational Component
METHODS OF ALGORITHM DEVELOPMENT
Preparations at the Second (Master's) Level of Higher Education
Specialty: 122 Computer Science
Educational (Professional) Program: Computer Science and Information Technologies

Lutsk – 2023

Syllabus of the educational component «Methods of Algorithm Development» for Master's level training, field of knowledge 12 Information Technology, specialization 122 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 №1 of 30.08. 2023

Head of the department:  (Hryshanovych T.O.)

I. Description of the educational component

<table>
<tr><td>Name of indicators</td><td>Field of knowledge, specialty, educational-professional /educational-scientific/educational-creative program, educational level</td><td>Characteristic educational component</td></tr>
<tr><td>Full-time form of education</td><td rowspan="8">12 Information Technologies, 122 Computer Science, Computer Science and Information Technology, second (master's)</td><td>Normative</td></tr>
<tr><td>Number of hours/credits 150/5</td><td>Year of study – 1st</td></tr>
<tr><td>IERT: yes</td><td>Semester – 1st</td></tr>
<tr><td></td><td>Lectures – 32 hours</td></tr>
<tr><td></td><td>Laboratory – 40 hours</td></tr>
<tr><td></td><td>Independent work – 10 hours</td></tr>
<tr><td></td><td>Consultations – 68 hours</td></tr>
<tr><td></td><td>Form of control: exam</td></tr>
<tr><td colspan="2">Language of education – English, Ukrainian</td><td></td></tr>
</table>

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 "Methods of Algorithm Development" has been developed in accordance with the educational-professional program "Computer Science and Information Technologies" (2023) 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. **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.

3. Learning outcomes (Competencies)

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.

GC04. Ability to communicate in a foreign language.

GC05. Ability to learn and master modern knowledge.

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.

4. 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 2
Topic 3. Algorithm Development Using Dynamic Programming.	8	2	2	4		SP/C 2
Topic 4. Algorithm Development Using the Iterative Method.	8	2	2	4		SP/C 2
Topic 5. Algorithm Development Using the Backtracking.	8	2	2	4		SP/C 2
Topic 6. Algorithm Development Using Solution Trees.	10	2	2	4	2	SP/C 2
Topic 7. Algorithm Development Using the Trial and Error Method.	8	2	2	4		SP/C 2
Topic 8. Algorithm Development Using the Branch and Bound Method.	8	2	2	4		SP/C 2
Topic 9. Algorithm Development Using the Alpha-Beta Pruning Method.	8	2	2	4		SP/C 2
Topic 10. Development of Greedy Algorithms.	8	2	2	4		SP/C 2
Topic 11. Evaluation of Algorithm Complexity. Classes of Algorithm Complexity.	10	2	2	4	2	SP/C 2
Module 1 Total	92	22	20	46	4	20
Content module 2. Applied algorithms and their complexity						
Topic 12. Algorithms on Graphs.	10	2	4	4		SP/C 4
Topic 13. Maze Generation Algorithms. Algorithms for Finding the Exit from a Maze.	10	2	4	4		SP/C 4
Topic 14. Machine Learning Algorithms.	12	2	4	4	2	SP/C 4
Topic 15. Algorithms for Big Data Processing.	14	2	4	6	2	SP/C 4
Topic 16. DNA Sequence Algorithms.	12	2	4	4	2	SP/C 4
Module 2 Total	58	10	20	22	6	20
Types of final works						Points
Test 1						10
Test 2						10
IERT 1						15
IERT 2						25
Total hours/Points	150	32	40	68	10	100

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

5. 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	Hash Functions.	1
2	BlockChain.	2
3	Algorithms for fast search of similar documents.	1
4	Matrix multiplication algorithms. Parallel computing processes.	2
5	Classes *P* and *NP*. The relationship between classes *P* and *NP*.	2
6	Quantum Computing.	2

IV. Evaluation policy

Instructor's Policy Regarding the Student's Education.

All participants in the educational process must adhere to the requirements of the current legislation of Ukraine, the Statute, and the Rules of Internal Regulations of Lesya Ukrainka Volyn National University, as well as commonly accepted moral principles, rules of conduct, and corporate culture. An atmosphere of friendliness, responsibility, honesty, and tolerance is expected. Tardiness to classes, the use of mobile phones, tablets, or other mobile devices during classes, and cheating are not allowed.

It is expected that all students will participate in the distance course "Methods Of Algorithm Development", which is hosted on the Moodle distance learning platform (https://moodle-cs.vnu.edu.ua/course/view.php?id=148). Practical assignments (labs, homework, individual assignments) and final assessments (tests, program development assignments) are submitted through the distance course.

In accordance with section 3.3 of the Regulations on the Recognition of Learning Outcomes obtained through formal, non-formal, and/or informal education at Lesya Ukrainka Volyn National University, no assessments will be conducted for the "Methods Of Algorithm Development" discipline in the first semester.

During the study of the educational component, the recognition of other learning outcomes obtained through formal, non-formal, and/or informal education is possible. The procedure for recognizing learning outcomes for higher education seekers acquired through formal education (academic mobility of students within Ukraine or abroad, transfers, and re-enrollments from other higher education institutions, both domestic and foreign) as well as non-formal and/or informal education is carried out according to the "Regulations on the Recognition of Learning Outcomes

obtained through formal, non-formal, and/or informal education at Lesya Ukrainka Volyn National University".

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 Sentences and Resubmissions

If a higher education seeker is absent from classes for any reason, they should independently study the theoretical material using educational manuals, lecture notes, and materials from the distance learning course "Methods Of Algorithm Development" (https://moodle-cs.vnu.edu.ua/course/view.php?id=148), hosted on the Moodle distance learning platform. They should also complete all homework assignments. Reporting on the completion of tasks can be done using the distance learning course "Methods Of Algorithm Development" or during consultations with the instructor. During this time, students can clarify any unclear points and ask questions. The use of the distance course forum is also an option. Re-taking control tests and exams is prohibited.

V. Final control

The final form of assessment for the educational component "Methods Of Algorithm Development" is an exam. The assessment is conducted on a 100-point scale. The evaluation includes continuous assessment (grading based on in-class work and the timely and high-quality completion of homework assignments) and final assessment (independent completion of individual tasks, midterm tests, evaluation of theoretical knowledge through testing, and individual homework assignments). The maximum number of points that a learner can earn during continuous assessment for the semester is 40 points. The maximum number of points a learner can earn for the final assessment for the semester is 60 points.

The completion of individual tasks is expected. The individual homework assignment (IERT) includes a set of problems that cover one or more closely related topics or a single task that requires independent exploration of smaller topics.

IERT example. Let's assume there is a finite set $Q\{q1, q2, ... qn\}$, and for each $q_i \square Q$, 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 “Methods Of Algorithm Development” 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 “Methods Of Algorithm Development” 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.

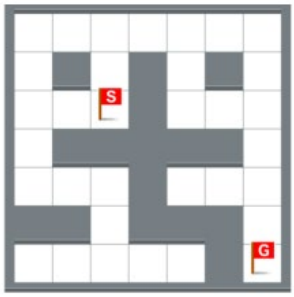
The questions and form of the exam are defined in this syllabus.

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.
25. Combinatorial Problems. Complexity Classes P and NP. Their Relationship.

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.



V. 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. Гришанович Т. О. Методи розробки алгоритмів [Електронний ресурс] : електронний курс навчальної дисципліни, затверджений НМР ВНУ імені Лесі Українки, протокол № 10 від 21.06.2023. ВНУ ім. Лесі Українки, 2023. URL: https://moodle-cs.vnu.edu.ua/course/view.php?id=148
2. Кормен Т. Г., Лейзерсон Ч. Е., Рівест Р. Л. Вступ до алгоритмів. Київ : К.І.С., 2019. 1285 с.
3. Креневич А.П. Алгоритми і структури даних. Підручник. К.: ВПЦ "Київський Університет", 2021. 200 с.
4. A Review of Yolo Algorithm Developments / P. Jiang et al. *Procedia Computer Science*. 2022. Vol. 199. P. 1066–1073. URL: https://doi.org/10.1016/j.procs.2022.01.135 (date of access: 20.08.2023).
5. Dinneen M. J., Henderson A., Nicolescu R. Sublinear P system solutions to NP-complete problems. *Theoretical Computer Science*. 2023. P. 113848. URL: https://doi.org/10.1016/j.tcs.2023.113848 (date of access: 10.09.2023).
6. Dynamic Programming, Greedy Algorithms | Coursera | Online Courses & Credentials From Top Educators. Join for Free | Coursera. *Coursera*. URL: https://www.coursera.org/learn/dynamic-programming-greedy-algorithms/home/week/1.
7. Skiena S. S. The Algorithm Design Manual (Texts in Computer Science). 3rd ed. NY : Stony Brook, 2020. 769 p.
8. Terletskyi D. O. and Provotar O. I. "Algorithm for Intersection of Fuzzy Homogeneous Classes of Objects," *2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT)*, 2020, pp. 314-317, doi: 10.1109/CSIT49958.2020.9321914.