**SYLLABUS**
**Fall semester 2025-2026 academic year**
**Educational program " 6B06102 – Information systems"**

| ID and name of course | Independent work of the student (IWS) | Number of credits | | | General number of credits | Independent work of the student under the guidance of a teacher (IWST) |
|---|---|---|---|---|---|---|
| | | Lectures (L) | Practical classes (PC) | Lab. classes (LC) | | |
| 64290, Algorithms, Data Structures and Programming | 2 | 1.70 | 0 | 3.30 | 5 | 6 |

**ACADEMIC INFORMATION ABOUT DISCIPLINE**

| Training format | Cycle, component | Lecture types | Types of practical classes | Form and platform final control |
|---|---|---|---|---|
| *Offline* | B, EC | Introductory , explanatory, concluding | Practical classes | Test, IS Univer |
| **Lecturer - (s)** | Mussina Aigerim Bolatovna | | | |
| **e-mail :** | mussina.aigerim95@gmail.com | | | |
| **Phone :** | +7 775 929 5274 | | | |
| **Assistant - (s)** | Barat Bekzat Talgatuly | | | |
| **e-mail :** | bekzat.barat@mail.ru | | | |
| **Phone :** | 87756679198 | | | |

**ACADEMIC COURSE PRESENTATION**

| Purpose of the course | Expected learning outcomes (LO) As a result of studying the discipline the undergraduate will be able to: | Indicators of LO achievement (ID) |
|---|---|---|
| The purpose of the discipline is to form ideas about the structure of data, about how to save and retrieve them, about the classification of programming languages, and the basic principles of software design. The following will be studied: calculation systems, methods of recording and retrieving data, creating and distributing tasks among program methods, methods of generating flowcharts and diagrams for representing algorithms of computational procedures. | 1. Analyze data structures and algorithms for solving various problems of searching, sorting, and graph processing | 1.1 Compares various search and graph algorithms |
| | | 1.2 Solves practical problems that require the use of specific algorithms and data structures |
| | | 1.3 Develops efficient algorithms for solving various problems |
| | 2. Define algorithms and data structures based on their effectiveness and scope of application | 2.1 Classifies the time and space complexity of algorithms for different types of problems |
| | | 2.2 Specifies the appropriate data structure for a particular task, taking into account the characteristics of the data |
| | 3. Evaluate the time and space complexity of algorithms and the choice of data structure depending on the task | 3.1 Integrate optimization to improve the performance of algorithms and data structures |
| | | 3.2 Proves the effectiveness of algorithms using mathematical and computational methods |
| | 4. Build basic data structures and algorithms in the chosen programming language | 4.1 Explains how basic data structures and algorithms work. |
| | | 4.2 Implements work programs using various data structures and algorithms |
| | 5. Develop data structures and algorithms in practice using a programming language | 5.1 Tests software code to verify the effectiveness of solutions |
| | | 5.2 Creates correct and optimized programs that implement the studied algorithms and data structures |
| | | 5.3 Applies principles of structured and modular programming to algorithm development. |
| **Prerequisites** | "School computer science course", Operating Systems[18070] | |
| **Post-requisites** | Programming Technologies[84567], Programming in Java Language[100842] | |

| | |
|---|---|
| **Learning Resources** | **Literature:** about the main one.<br>1. Lippman, Stanley B.; Lajoie, Josée; Moo, Barbara E. C++ Primer. 5th Edition. — Addison-Wesley, 2019. — 976 p.<br>2. Stroustrup, Bjarne. Programming: Principles and Practice Using C++. 2nd Edition. — Addison-Wesley, 2019. — 1312 p.<br>3. Bhargava, Aditya. Grokking Algorithms: An Illustrated Guide. — Manning, 2016. — 256 p.<br>4. Weiss, Mark Allen. Data Structures and Algorithm Analysis in C++. 4th Edition. Pearson, 2013. 656 p.<br>5. Knuth, Donald E. The Art of Computer Programming, Volumes 1–3. Addison-Wesley, 2011. —2200 p.<br>**Research infrastructure**<br>1. Visual Studio Code<br>**Internet resources**<br>1. VisuAlgo — https://visualgo.net<br>2. GeeksforGeeks — https://www.geeksforgeeks.org<br>3. LeetCode — https://leetcode.com<br>4. C++ Reference — https://cppreference.com |

| | |
|---|---|
| **Academic discipline policy** | The academic policy of the discipline is determined by the Academic Policy and the Academic Integrity Policy of Al-Farabi KazNU.<br>Documents are available on the main page of the Univer IS .<br>**Integration of science and education.** Research work of students, undergraduates and doctoral students is a deepening of the educational process. It is organized directly in departments, laboratories, scientific and design departments of the university, and in student scientific and technical associations. Independent work of students at all levels of education is aimed at developing research skills and competencies based on acquiring new knowledge using modern research and information technologies. A teacher at a research university integrates the results of scientific activity into the topics of lectures and seminar (practical) classes, laboratory classes and into the tasks of the IWST, IWS, which are reflected in the syllabus and are responsible for the relevance of the topics of training sessions and tasks.<br>**Attendance.** The deadline for each task is indicated in the calendar (schedule) for the implementation of the discipline content. Failure to meet deadlines will result in loss of points.<br>**Academic integrity.** Practical/laboratory classes and SRL develop the student's independence, critical thinking, and creativity. Plagiarism, forgery, use of cheat sheets, and cheating at all stages of assignments are unacceptable.<br>In addition to the main policies, the observance of academic integrity during theoretical training and exams is regulated by the "Rules for conducting final control" , "Instructions for conducting final control of the autumn/spring semester of the current academic year" , "Regulations on checking students' text documents for the presence of borrowings".<br>Documents are available on the main page of the Univer IS .<br>**Basic principles of inclusive education.** The educational environment of the university is conceived as a safe place where there is always support and equal treatment on the part of the teacher towards all students and students towards each other, regardless of gender, race/ethnicity, religious beliefs, socio-economic status, physical health of the student, etc. All people need the support and friendship of peers and fellow students. For all students, making progress is more about what they can do than what they can't do. Variety enhances all aspects of life.<br>All students, especially those with disabilities, can receive advice by phone/e- mail +77759295274 / mussina.aigerim95@gmail.com , or via video call in MS Teams https://teams.microsoft.com/l/team/19%3AEXobN2cQvJpEY0Z6XkVPfMuU_rMyQ-Pwn3fFsUhLcGo1%40thread.tacv2/conversations?groupId=776342c7-342f-453a-a5bd-332259d38eac&tenantId=b0ab71a5-75b1-4d65-81f7-f479b4978d7b. |

**INFORMATION ABOUT TEACHING, LEARNING AND ASSESSMENT**

**Score-rating letter system of assessment of accounting for educational achievements**

| Grade | Digital equivalent points | points, % content | Assessment according to the traditional system |
|---|---|---|---|
| A | 4.0 _ | 95-100 | Great |
| A- | 3.67 | 90-94 | |
| B+ | 3.33 | 85-89 | Fine |

**Assessment Methods**

**Criteria-based assessment** is the process of correlating actual learning outcomes with expected learning outcomes based on clearly defined criteria. Based on formative and summative assessment.

**Formative assessment is** a type of assessment that is carried out in the course of daily learning activities. It is the current measure of progress. Provides an operational relationship between the student and the teacher. It allows you to determine the capabilities of the student, identify difficulties, help achieve the best results, timely correct the educational process for the teacher. The performance of tasks, the activity of work in the classroom during lectures, seminars, practical exercises (discussions, quizzes, debates, round tables, laboratory work, etc.) are evaluated. Acquired knowledge and competencies are assessed.

**Summative assessment** - type of assessment, which is carried out upon completion of the study of the section in accordance with the program of the course. Conducted 2 times per semester when performing IWS. This is the assessment of mastering the expected learning outcomes in relation to the

| | | | | | |
|---|---|---|---|---|---|
| | | | | descriptors. Allows you to determine and fix the level of mastering the course for a certain period. Learning outcomes are evaluated. | |
| B | 3.0 | 80-84 | | **Formative and summative assessment** | **Points % content** |
| C+ | 2.33 | 70-74 | | Work in laboratory classes | 42 |
| C | 2.0 | 65-69 | Satisfactorily | Independent work | 18 |
| D+ | 1.33 | 55-59 | | Final control (exam) | 40 |
| D | 1.0 | 50-54 | | TOTAL | 100 |
| FX | 0,5 | 25-49 | Unsatisfactory | | |
| F | 0 | 0-24 | | | |

**Calendar (schedule) for the implementation of the content of the course. Methods of teaching and learning.**

| A week | Topic name | Number of hours | Max. points |
|---|---|---|---|
| | **MODULE 1 Basics of programming** | | |
| 1 | **L 1.** Introduction to the Course: The Concept of an Algorithm and Its Properties | 1 | |
| | **LC 1.** Installing and Configuring the Development Environment. C++ Program Structure | 2 | |
| 2 | **L 2.** Number Systems | 1 | |
| | **LC 2.** Conversion Between Number Systems | 2 | |
| 3 | **L 3.** Data Types and Variables | 1 | |
| | **LC 3.** Declaring and Initializing Variables. Input/Output Operators (cin/cout). Arithmetic Operations | 2 | |
| 4 | **L 4.** Control Structures | 1 | |
| | **LC 4.** If-else conditional statements. Multiple-choice switch-case statements. Loops: for, while, do-while | 2 | |
| | **IWST 1.** Consultation on completing IWS 1 on the topic: "Writing a Program Based on a Given Flowchart" | | |
| 5 | **L 5.** Arrays | 1 | |
| | **LC 5.** Declaring and Initializing Arrays. Working with Two-Dimensional Arrays (Matrices) | 2 | 15 |
| 6 | **L 6.** Functions and Modularity | 1 | |
| | **LC 6.** Creating and Calling Functions. By-Value and By-Reference Parameters. Return Values. Recursive Functions | 2 | 15 |
| | **IWST 2.** Analysis of the covered topics of the module | | |
| 7 | **L 7.** Pointers and Dynamic Memory | 1 | |
| | **LC 7.** Declaring and Using Pointers. Dynamic Arrays | 2 | 20 |
| 8 | **L 8.** Structures and Records | 1 | |
| | **LC 8.** Creating Structures (structs). Arrays of Structures. Nested Structures | 2 | 20 |
| | **IWST 3.** Reception and Defense of IWS 1. | | 30 |
| **Midterm control 1** | | | **100** |
| | **MODULE 2 Weighted Graph Algorithm** | | |
| 9 | **L 9.** Linked Lists | 1 | |
| | **LC 9.** Implementation of a Singly Linked List | 2 | 5 |
| 10 | **L10.** Stack and Queue | 1 | |
| | **LC 10.** Implementation of a Stack on an Array. Implementation of a Stack on a List | 2 | 10 |
| | **IWST 4.** Consultation on Completing IWS 2 on the Topic: "Comparison of Search and Sorting Algorithms by Efficiency and Applicability" | | |
| 11 | **L 11.** Search Algorithms | 1 | |
| | **LC 11.** Implementation of Linear Search. Implementation of Binary Search (Iterative and Recursive) | 2 | 10 |
| 12 | **L 12.** Sorting Algorithms I | 1 | |
| | **LC 12.** Bubble Sort. Selection Sort. Insertion Sort | 2 | 10 |
| 13 | **L 13.** Sorting Algorithms II | 1 | |
| | **LC 13.** QuickSort. MergeSort | 2 | 10 |
| | **IWST 5.** Analysis of the covered topics of the module | | |
| 14 | **L 14.** Introduction to Trees | 1 | |
| | **LC 14.** Binary Search Trees | 2 | 10 |
| 15 | **L 15.** Hash Functions | 1 | |
| | **LC 15.** Chaining Collision Resolution | 2 | 15 |
| | **IWST 6.** IWS 2 Acceptance and Defense | | 30 |
| **Midterm control 2** | | | **100** |
| **Final control (exam)** | | | **100** |
| **TOTAL for discipline** | | | **100** |

**RUBRICATOR OF THE SUMMATIVE ASSESSMENT**

**CRITERIA EVALUATION OF LEARNING OUTCOMES**

**IWS1 - "Writing a Program Based on a Given Flowchart" (30% of 100% MC )**

| **Criterion** | 25-30 % | 20-24% | 10-19% | 0-9% |
|---|---|---|---|---|
| **Understanding Theories and Concepts** | Demonstrates a deep understanding of algorithmic theories and their practical implications. | Shows a good grasp of algorithmic concepts, with mostly accurate explanations. | Presents a basic understanding of the theories, but explanations lack depth or have noticeable inaccuracies. | Fails to demonstrate a clear understanding of the theories. Explanations are vague, incorrect, or missing. |
| **Implementation and Code Quality** | Code is well-written, efficient, and follows best practices. | Code is mostly correct and functional, with some minor inefficiencies or coding practice issues. | The implementation is partially correct, with noticeable errors or inefficiencies. | The code is incorrect or incomplete, with significant logical errors, inefficiencies, or poor structure. |

**IWS2 - "Comparison of Search and Sorting Algorithms by Efficiency and Applicability" (30% of 100% MC)**

| **Criterion** | 25-30 % | 20-24% | 10-19% | 0-9% |
|---|---|---|---|---|
| **Time and Space Complexity Analysis** | A comprehensive analysis of the best-, average-, and worst-case time complexity is presented for all the algorithms considered. Plots of execution time versus input data size are constructed based on practical measurements. | A good time complexity analysis is provided for most cases. Practical measurements are presented, but the graphs or analysis of the results are insufficiently detailed. | A basic time complexity analysis is performed, but not all cases are considered. Practical measurements are conducted, but their analysis is superficial or contains inaccuracies. | Complexity analysis is missing or contains serious errors. Incorrect understanding of O-notation. Practical measurements are not conducted, or their results do not correspond to the theoretical analysis. |
| **Comparative analysis and recommendations for applicability** | A detailed comparison of the algorithms was conducted, taking into account various factors: data size, data ordering, data type, and operation frequency. Special cases and boundary conditions were considered. | A good comparison of the algorithms based on key parameters is provided. Applicability recommendations are provided, but may be too general or omit some important scenarios. | A basic comparison of the algorithms is performed, but not all important aspects are considered. Advantages and disadvantages are partially or inaccurately defined. Applicability recommendations are overly simplified or not fully substantiated. | A comparative analysis is absent or extremely superficial. |

**Dean _____T.S. Imankulov**

**Chair of the Academic Committee**

**on the Quality of Teaching and Learning_____ Zh.A. Buribayev**

**Head of Department _____N.M. Kassymbek**

**Lecturer _____A.B. Mussina**