Ala-Too International University
Department of Computer Systems and complexes
Course Syllabus for
2025-2026 academic year
Fall semester

Title of the Course	Basics of algorithmization and programming
Course Code	None
Credit	2
Hours in a week	2
Prerequisites	Basic facilities of information skills
Type of the course	Practical
Recommended for	Software of computer engineering and automated systems
Language of instruction	English
Mode of education	Offline
LMS (Learning Management System)	None
LMS enrollment code	None
Online conference room	None
Physical classroom	C001
Lecturer	Nurbekov Mirlan Nurbekovich
Office	H 209
Office hours	Tu 15:00-17:00; We/Th 16:00-17:00; Fr 08:30-17:00
Email	mirlan.nurbekov@alatoo.edu.kg

Course Description

This course introduces the fundamental principles of algorithmization and programming as the foundation of computational problem-solving. Students will learn how to design, analyze, and implement algorithms using the Java programming language. The course emphasizes structured and logical thinking, covering essential programming constructs, flowcharts, pseudocode, recursion, and basic data structures. Through a combination of lectures, practical coding exercises, and algorithmic problem-solving activities, students will develop the ability to transform real-world problems into efficient, executable solutions. By the end of the course, learners will be able to implement algorithms with clarity and efficiency.

Student Learning Outcomes (SLOs)

SLO.1. Understand the fundamental concepts of algorithms, programming logic, and computational problem-solving.

SLO.2. Develop and analyze algorithms using flowcharts, pseudocode, and structured programming techniques.

SLO.3. Implement algorithms in the Java programming language, applying proper syntax, control structures, and data handling methods.

SLO.4. Apply modular and object-oriented approaches to design efficient, maintainable, and reusable code.

SLO.5. Evaluate algorithm performance and efficiency using basic complexity analysis and optimization techniques.

SLO.6. Demonstrate problem-solving skills through hands-on programming labs, debugging exercises, and small-scale algorithmic projects.

Course Content

Week	Topic	Focus/Lab/Activity
1	Introduction and Java revision part 1	Course overview, grading policies, setup of IDE (IntelliJ), review of Java basics: variables, data types, input/output, operators
2	Java revision part 2	Control structures (if, switch, loops), methods, arrays, and simple problem-solving using loops and conditionals
3	Introduction to Algorithms and Flowcharts	What is an algorithm; pseudocode conventions; drawing flowcharts; translating logic into code
4	Problem Solving and Algorithm Design	Step-by-step problem-solving techniques, modular programming, basic debugging practices
5	Searching and Sorting Algorithms	Linear and binary search, bubble sort, selection sort, insertion sort, implementing and comparing.
6	Functions, Recursion, and Complexity	Function decomposition, recursion principles, introduction to time complexity (Big-O notation)
7	Arrays, Strings, and Data Structures	Multi-dimensional arrays, string manipulation, intro to stacks and queues using arrays
8	Algorithmic thinking project	Students design algorithms for real-world problems (e.g., route optimization, grading system, ATM simulation)
9	Midterm assessment	Offline written + coding test covering weeks 1–8
10	Object-oriented approach in algorithms	Applying OOP concepts (classes, objects, inheritance) to algorithm design and implementing

11	Advanced sorting and searching	Merge sort, quick sort, recursion-based search; performance comparison and analysis
12	Introduction to graphs and trees	Basic concepts of graphs and trees, BFS/DFS algorithms and practical examples (social networks, file systems)
13	Algorithms in real applications	Pathfinding (Dijkstra), data compression, encryption logic and discussion of algorithms in real systems
14	Team Activity: algorithm optimization challenge	Teams analyze and improve inefficient algorithms provided by the instructor, students need to provide proof of a runtime improvement.
15	Algorithm analysis and debugging Tools	Profiling code performance, using debugging tools, understanding algorithmic bottlenecks
16	Final exam	Offline exam covering materials from week 1–15

Grading rubric and assessments

Midterm assessment - Test	100 points
Final exam	100 points
Average = Midterm assessment * 0.4 + Final assessment * 0.6	

Course policies and academic integrity

This is a 16-week course instructed by Mirlan Nurbekov. The course instructor reserves the right to make changes to any portion of the syllabus at any time. Any modifications will be communicated in writing to students via the Learning Management System (LMS).

- ➢ Students are expected to communicate with the instructor professionally. Mobile phones, social media, and messaging apps (e.g., WhatsApp) should not be used for course-related inquiries. Instead, students should send emails, post comments in the LMS, or visit the instructor during office hours. All course-related announcements and materials will be shared through the LMS.

- ➢ Attendance is mandatory. Students are expected to attend at least 70% of the course sessions to be eligible for exams. Active participation in lectures, labs, and discussions is strongly encouraged.

- ➢ Students are required to uphold the highest ethical standards in all aspects of the course. Academic dishonesty includes, but is not limited to, cheating on exams, completing work for another student, and plagiarism.

- ➢ Plagiarism is a serious academic offense. To avoid plagiarism:

    - Do not copy words from any source without proper quotation and citation.

    - Do not use ideas, concepts, or opinions from any source without citing the source. This includes technical terms, original views, and key concepts.

    - Paraphrasing is allowed, but the source must be cited correctly, and the original meaning must not be misrepresented.

All cases of academic dishonesty will result in a failing grade for the course and will be reported to the Head of the Management Department for administrative review.

All projects, presentations, essays, and assignments must follow the Harvard referencing style. Submissions must be uploaded to the LMS before the specified deadline.

- Grading and Assessment:
    - Midterm, final, and makeup exams should account for 40–50% of the total course grade, with the remainder coming from projects, presentations, essays, and assignments.

    - To pass the course, students must achieve an average grade of at least 50 points, with a minimum of 40 points on the final exam.

    - Students who do not pass the final exam are eligible for a makeup exam, which carries the same weight as the final assessment.
- Appeals

  Students have the right to appeal grades within three working days from the date grades are announced. Appeals may include reviewing exam papers, requesting re-evaluation, requesting grading rubrics, or notifying the instructor of any errors in grade aggregation. Once the grade submission system is closed, grades cannot be changed.

Main Resources

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2022) *Introduction to Algorithms.* 4th edn. Cambridge, MA: MIT Press.

Horstmann, C.S. (2022) *Big Java: Early Objects.* 8th edn. Hoboken, NJ: Wiley.

Liang, Y. Daniel. (2022) *Introduction to Java Programming and Data Structures.* 13th edn. Pearson.

Knuth, D.E. (1997) *The Art of Computer Programming, Volume 1: Fundamental Algorithms.* 3rd edn. Addison-Wesley.

Additional Resources

GeeksforGeeks (no date) *Data Structures and Algorithms.* Available at: https://www.geeksforgeeks.org/ (Accessed: 3 August 2025).

Khan Academy (no date) *Computer Science – Algorithms and Programming.* Available at: https://www.khanacademy.org/computing/computer-science (Accessed: 3 August 2025).

W3Schools (no date) *Java Tutorial.* Available at: https://www.w3schools.com/java/ (Accessed: 12 August 2025).

HackerRank (no date) *Algorithms Practice Problems.* Available at: https://www.hackerrank.com/domains/tutorials/ (Accessed: 12 August 2025).

Coursera (no date) *Algorithmic Thinking (Part 1 & 2) – Rice University.* Available at: https://www.coursera.org/learn/algorithmic-thinking-1 (Accessed: 12 August 2025).