“Халықаралық ақпараттық технологиялар университеті” АҚ

АО “Международный университет информационных технологий”

Faculty of Digital Technologies, Creative Industries and Business

Department of Computer Engineering

«APPROVED»
Dean of faculty
Mukhamediyeva A.G.
«27» 08 2026

WORKING CURRICULUM OF THE DISCIPLINE (SYLLABUS)

Course: SFT6301 Algorithmization and Programming

Group of educational programs: B057 – Information Technologies

Educational program: 6B06106 – Computer Systems and Software Engineering, 6B06110 – Software Engineering, 6B06107 – Cyber-Physical Systems, 6B06118 – Immersive Technologies

Year: 1 **Semester:** 1 **Number of academic credits**: 6 ECTS

Lectures:	30	hours
Laboratory work:	30	hours
IWST:	15	hours
IWS:	105	hours
TOTAL:	180	hours
Cycle of discipline:	**Basic**	
Form of control:	**Written Exam**	

Almaty 2026

The working curriculum of the discipline (syllabus) SFT6301 Algorithmization and Programming has been developed on the basis of the educational program 6B06106 – Computer Systems and Software Engineering, 6B06110 – Software Engineering, 6B06107 – Cyber-Physical Systems, 6B06118 – Immersive Technologies.

The working curriculum of the discipline (syllabus) has been reviewed at the meeting of Computer Engineering department.

Minutes № 1 dated "27" August 2026

Head of the Department _____ **Chinibayeva T.T., assoc. prof, PhD**
signature *full name*

Author _____ **Balgabek A.A., senior-lecturer, MSc**
signature *full name, title, degree*

The working curriculum of the discipline (syllabus) was approved at a meeting of the Faculty's Academic Quality Council.

Minutes № 1 dated "27" August 2026

Agreed:

Head of the Department of the Educational and Methodological activities _____ **Ajibayeva A.**

Library _____ **Sarsenbayeva Zh.A.**

1. General information	
Faculty	Digital Technologies, Creative Industries and Business
Code and name of the educational program (EP)	6B06106 – Computer Systems and Software Engineering, 6B06110 – Software Engineering, 6B06107 – Cyber-Physical Systems, 6B06118 – Immersive Technologies
Program level (bachelor's, master's, PhD)	Bachelor
Year, semester	1st year, 1st semester
Name of the discipline	Algorithmization and Programming (SFT6301)
Cycle of the discipline	Basic
Number of academic credits	6 ECTS credits
Prerequisites	N/A
Postrequisites	"Object Oriented Programming", "Algorithms and Data Structure"
Lecturer	**Lecturer**: Askar A. Balgabek, senior lecturer, Department of Computer Engineering. **E-mail**: a.balgabek@iitu.edu.kz **Office:** 409, IITU. **Office hours:** Thursday, 12:00-13:00(by schedule).
Teachers who conduct practical or laboratory classes	Mamanova Symbat , Senior Lecturer, Department of Computer Engineering. **Office:** 409, IITU. **E-mail:** s.mamanova@iitu.edu.kz Barlykbay Nazym, Senior Lecturer, Department of Computer Engineering. **Office:** 409, IITU. **E-mail:** n.barlykbay@iitu.edu.kz Sydykbek Zhanibek, Assistant, Department of Computer Engineering. **Office:** 409, IITU. **E-mail:** zh.sydykbek@iitu.edu.kz Tleuova Gaini, senior lecturer, Department of Computer Engineering. **Office:** 409, IITU. **E-mail:** .tleuova@iitu.edu.kz
2. Goals, objectives and learning outcomes of the course	
The course goal is to form students' systematic knowledge of fundamental programming concepts, algorithms and data structures and to develop practical skills in solving programming problems using the C++ programming language, including algorithm analysis, dynamic memory management and efficient program development.	
The objectives of the course are: – to develop students' understanding of the fundamental concepts and principles of programming in C++; – to develop practical skills in implementing and applying basic sorting and searching algorithms; – to introduce the concepts of algorithm complexity and Big O notation and develop the ability to evaluate the efficiency of algorithms; – to develop an understanding of pointers, dynamic memory management and their application in C++ programs;	

– to develop practical skills in selecting and implementing appropriate data structures for solving programming problems; – to develop students' logical and critical thinking and their ability to design, analyse, debug and improve software solutions.
Learning outcomes of the course. On successful completion of the discipline the student will be able to: LO1 (knowledge and understanding): explain the fundamental concepts of programming in C++, including program structure, functions, pointers, dynamic memory and basic data structures; LO2 (applying knowledge): implement and apply basic searching, sorting and data-structure algorithms to solve practical programming problems; LO3 (making judgements): analyse and compare algorithms in terms of time and memory complexity and select an appropriate algorithm and data structure for a given task; LO4 (practical and technical skills): develop, debug, modify and test C++ programs using modern compiler and integrated development environments; LO5 (communication and learning skills): write clear, structured and maintainable program code, explain programming solutions and independently apply acquired programming concepts to new software development problems.
Competencies formed as a result of mastering the discipline: – programming competence — ability to develop, debug and modify programs in C++ using fundamental programming principles; – algorithmic and computational thinking competence — ability to design, analyse and compare algorithms and evaluate their time and memory efficiency; – data management competence — ability to select and implement appropriate data structures and manage dynamic memory using pointers; – problem-solving competence — ability to decompose practical problems and develop effective software solutions; – critical thinking and analytical competence — ability to evaluate alternative programming solutions and justify the selection of appropriate algorithms and data structures; – self-directed learning competence — ability to independently acquire and apply new programming techniques, tools and approaches. – awareness of lifelong learning and development of relevant ICT skills (SDG 4 – Quality Education).
3. Course description
Course Description: Algorithmization and Programming is a fundamental discipline of the educational program that develops students' algorithmic and computational thinking and provides the foundations for software development. The course introduces the principles of algorithm design and implementation and develops practical programming skills using the C++ programming language. The course covers fundamental programming concepts, problem-solving techniques, basic searching and sorting algorithms, algorithm complexity and Big O notation, pointers and dynamic memory management, and basic data structures. Particular attention is paid to analysing the efficiency of algorithms and selecting appropriate algorithms and data structures for solving practical problems. The course combines theoretical knowledge with practical programming activities, enabling students to develop, debug, analyse and improve software solutions. The acquired knowledge and skills provide a foundation for further study of software engineering, data structures and algorithms, object-oriented programming, software architecture and other advanced computing disciplines. The course contributes to Sustainable Development Goal 4 (Quality Education), particularly Target 4.4, by developing students' relevant technical and information and communications technology (ICT) skills.

4. Teaching methods

Teaching methods: Teaching methods. The course combines theoretical and practice-oriented approaches aimed at developing students' algorithmic thinking and practical programming skills. Lectures introduce fundamental programming concepts, algorithms and problem-solving techniques through explanations, examples and analysis of program execution. Laboratory classes and workshops follow a learning-by-doing approach and focus on implementing, testing, debugging and analysing C++ programs, as well as applying searching, sorting and basic data-structure algorithms to practical problems. Problem-based learning is used to develop students' ability to decompose complex tasks, select appropriate algorithms and data structures, and justify their solutions. Interactive discussions and case-based tasks encourage critical thinking and comparison of alternative programming solutions. Digital tools and development environments, including modern C++ compilers, IDEs, MS Teams and Platonus, are used to support programming practice, collaboration and feedback. Independent work reinforces the material covered in class and develops students' ability to search for information, practise programming techniques and solve problems independently. The combination of these methods is aimed at achieving the learning outcomes LO1–LO5 and developing programming, algorithmic, analytical and problem-solving competencies.

5. Course policy

Attendance Policy: Attendance is checked at every class. Being more than 5 minutes late may be recorded as an absence. If the absence rate exceeds 20%, the student automatically receives an 'F', in accordance with the course requirements.

Deadline Policy: All laboratory and independent assignments must be submitted by the deadlines specified in the course schedule and syllabus. Submissions are made through the Platonus system, which records the submission date and time.

Late Submission Policy: Assignments submitted after the established deadline are not accepted for regular grading unless otherwise approved by the instructor or justified by a documented valid reason. A submission uploaded on time but not defended within the required timeframe is subject to the applicable defense and resubmission rules.

Resubmission Policy: If a student is required to defend an assignment for a second time, 15 points are deducted from the final grade for that assignment. The resubmission must be completed within the timeframe specified by the instructor.

Academic Conduct and Ethics Policy: Students are expected to attend classes on time, actively participate in the learning process and respect the instructor and their peers. Mobile devices should not be used in a way that disrupts the learning process. The use of unauthorised materials during assessments is prohibited. Students must follow the principles of academic integrity and maintain a respectful learning environment.

Respect for Differences of Opinion: Different technical viewpoints and approaches are welcomed when supported by appropriate reasoning and evidence. Students are expected to discuss differing opinions constructively and respectfully and to avoid offensive or discriminatory remarks.

Communication and Ethics of Interaction: Official course-related communication is conducted through MS Teams and the university LMS, including Platonus. Communication should remain professional, respectful and relevant to the subject matter.

Inclusion and Individual Needs Policy: Students with individual educational needs or circumstances requiring reasonable accommodation are encouraged to inform the instructor in advance. The instructor will consider such needs within the applicable university policies and available arrangements.

6. Academic Integrity

Plagiarism and other violations policy: All submitted work must be original and independently prepared by the student. Plagiarism, copying another student's work, unauthorised collaboration or submitting externally sourced work as one's own is prohibited. If an assignment or report submitted through Platonus is found to have been copied from another student or external source, the student who submitted the copied work receives a grade of 0 for the assignment. A student who knowingly provides their work for copying may also receive a grade of 0. Cases may be escalated in accordance with university regulations.

Code of academic integrity: Students are expected to follow the IITU Code of Academic Integrity and the principles of honesty, fairness, responsibility and respect. Cheating during tests, control works and examinations is strictly prohibited. The use of unauthorised materials or external resources during assessments is not permitted.

Use of AI: AI tools, including ChatGPT and other AI assistants, may be used as learning-support tools for understanding course concepts, obtaining explanations, exploring alternative approaches and debugging programs, unless otherwise specified by the instructor for a particular assessment. However, AI tools must not be used to generate code for laboratory assignments, independent assignments, tests, examinations or other assessed programming tasks. Students are responsible for understanding, writing and being able to explain the code they submit. Submitting AI-generated code as one's own work is considered an academic-integrity violation and may result in a grade of 0 for the respective work.

7. Literature

Basic literature

1. Deitel, P., C++ How To Program (Программирование на C++): study quide / U.S.A : Prentice Hall, 2002. - 1321 p. 004.43

2. Lospinoso, J., C++ Crash Course : tutorial / San Francisco : No Starch Press, 2019. - 733 p. 004.43

Supplementary literature:

1. Thinking In C++. Second Edition by Bruce Eckel, 1995

2. The Art of Computer Programming. 3: Sorting and Searching (2nd ed.), 1998

Online resources:

1. Coursera – « Learning with AI », Politecnico di Milano (IWST course): https://www.coursera.org/learn/learning-with-ai

2. C++ basics – https://education.yandex.ru/handbook/cpp

3. Tutorial notes (english) – https://www.programiz.com/cpp-programming

8. Forms of control and assessment

№	Type of work	Form of control	Evaluation criteria
1	Continuous Assessment 1: Defences	Practical examination	Timely submission of the report in Platonus. Successful defense of the work in English. Ability to answer additional questions from the instructor. Criterion: a report uploaded without defense = the work is not counted.

2	Mid-term	Practical examination	The assessment consists of problem sets assigned by variants. Tasks must be solved and submitted within the given time. Full credit is awarded for correct and complete solutions. Partial credit may be given for incorrect or non-functional solutions if the logical reasoning and approach are sound. No credit is awarded for missing or irrelevant work. Use of unauthorized resources results in a score of zero.
3	Continuous Assessment 2: Defences	Practical examination	Timely submission of the report in Platonus. Successful defense of the work in English. Ability to answer additional questions from the instructor. Criterion: a report uploaded without defense = the work is not counted.
4	End-of-term	Practical examination	The assessment consists of problem sets assigned by variants. Tasks must be solved and submitted within the given time. Full credit is awarded for correct and complete solutions. Partial credit may be given for incorrect or non-functional solutions if the logical reasoning and approach are sound. No credit is awarded for missing or irrelevant work. Use of unauthorized resources results in a score of zero.
5	Independent Student Work (ISW) (List of topics in Table 9)	Oral defence	Students are provided with a set of problems for laboratory and homework assignments. Completion of these tasks is mandatory. An oral defense of the submitted solutions is required. Successful defense serves as the prerequisite for admission to the final exam. Failure to complete or defend the tasks results in denial of exam entry.
6	Final Assessment	Exam	The assessment is conducted through an online platform with an automatic compiler. Each task must be completed and submitted within the specified time. Solutions are automatically tested; the percentage of passed tests determines the score. Full credit is awarded for solutions passing all tests. Partial credit is awarded proportionally to the percentage of passed tests, provided the solution demonstrates coherent logic. No credit is given for missing submissions or irrelevant code. Use of unauthorized resources results in a score of zero.

9. System for evaluating student performance in a discipline:

Each type of academic work is assessed on a 100-point scale in accordance with the table.

Example

Period	Assignments	Score
1st attestation	Laboratory works 1-6	100
	Continuous assessment 1	100
	Continuous assessment 2	100

	Mid-term exam	100
2nd attestation	Laboratory works 7-12	100
	Continuous assessment 3	100
	Continuous assessment 4	100
	End-of-term exam	100
Exam		**100**
TOTAL	0,3*1Ат + 0,3*2Ат + 0,4*Ex.	**100**

10. Course schedule

Week/ date	Course topics	References	Lectures (h/w)	Practical sessions (h/w)	Lab. sessions (h/w)	IWST (h/w)	IWS (h/w)
1.	Introduction to C++ programming. Variables, datatypes	B1,S2	2		2	1	7
2.	Consitional statement and logical operators	B1,S2	2		2	1	7
3.	Loops: for, while, do…while	B1,S2	2		2	1	7
4.	The basics of array	B1, B3	2		2	1	7
5.	Sorting algorithms and Big-O notation	B1, B3	2		2	1	7
6.	Multidimensional Arrays	B1, B3	2		2	1	7
7.	Introduction to functions, recursion	B3, S2	2		2	1	7
8.	Introduction to pointers	B3, B4, S2	2		2	1	7
9.	Double pointers	B3, B4, S2	2		2	1	7
10.	Vector and String	B3, B4, S2	2		2	1	7
11.	Introduction to structures	B4, S2	2		2	1	7
12.	Nested structures	B3, B4, S2	2		2	1	7
13.	Streams, file input/output stream	B3, B4, S2	2		2	1	7
14.	Introduction to classes. Incapsulation	B4	2		2	1	7
15.	Review of the course material	S2	2		2	1	7
Total hours:			**30**		**30**	**15**	**105**

11. List of topics/assignments for laboratory classes

№	Topic Title	Number of hours	References	Form of reporting	Deadline
1.	Writing simple program, variables and data types, libraries, system functions	2	B1,S2	Report	Week 2

2.	Selection statement. If-Else statement. Switch case statement, logical operator	2	B1,S2	Report	Week 3
3.	Loops. For, while, do...while statements.	2	B1,S2	Report	Week 4
4.	One-dimensional arrays	2	B1, B3, S2	Report	Week 5
5.	Searching algorithms, bubble sort, selection sort	2	S2		Week 6
6.	Multi-dimensional arrays, nested cycles	2	B1, B3, S2	Report	Week 7
7.	Functions, return statement, void function, function overloading, usage of Recursion logic algorithms	2	B1, B3, S2	Report	Week 8
8.	Pointers, iteration, dynamic memory, resizing arrays.	2	B1, B3, S2, S3	Report	Week 9
9.	Double pointing, memory control	2	B3, B4, S2	Report	Week 10
10	String, string functions, vectors	2	B3, B4, S2	Report	Week 11
11	Structures	2	B4, S2	Report	Week 12
12	Nested structures, array of structures	2	B4, S2	Report	Week 13
13	Working with txt files, parsing	2	B3, B4, S2	Report	Week 14
14	Introduction to classes and basic principles of Object Oriented Programming	2	B3, B4, S2	Report	Week 15
15	Associative Containers	2	B1, B3, S2	Report	Week 15

12. List of topics/assignments for independent work with the teacher (IWST) — online support course

Course: « Learning with AI» — Politecnico di Milano (Coursera), beginner level, ≈15 hours, English with subtitles. Link: https://www.coursera.org/learn/learning-with-ai. In accordance with the department's decision, the following online course is included in the discipline as guided independent work with the teacher (IWST). Form of reporting: a Coursera Certificate of Completion together with a short reflective summary (0.5–1 page) describing the key concepts learned and reflecting on how AI tools can be used to support independent learning, critical thinking, and the development of programming skills. Deadline: by the end of the 2nd attestation (Week 14). Assessed within IWST.

13. List of topics/assignments for student's independent work

Proper organization of students' independent work is the key to the formation of skills in mastering, learning, assimilation and systematization of acquired knowledge, ensuring a high level of academic performance in the learning process.

№	Topic/Assignment title	Number of hours	References	Form of reporting	Deadline
1.	Problem-solving task set: independent solution of a set of programming problems requiring algorithm selection and implementation	15	B1, S2	Code	Week 2
2.	Debugging and code improvement: analysis, debugging and optimisation of existing C++ programs	8	B1, S2	Report	Week 4
3.	Algorithm comparison: development of two alternative solutions for the same	8	B1, B3	Code + Report	Week 5

	problem and justification of the more efficient approach				
4.	Searching algorithms analysis, implementation and comparison	10	B1, B3	Report	Week 6
5.	Algorithm efficiency study: experimental evaluation of time and memory efficiency of selected algorithms	10	B1, B3	Report	Week 7
6.	Sorting algorithms analysis, implementation and experimental comparison of basic sorting algorithms	12	B1, B3	Report + Presentation	Week 8
7.	Pointers and dynamic memory: practical tasks involving pointers, arrays and dynamic memory allocation in C++	15	B3, B4, S2	Code + Report	Week 10
8.	Data structures: implementation and application of basic data structures to practical problems	12	B4, S2	Code	Week 12
9.	Comprehensive programming assignment: independent solution of an applied programming problem	15	B3, B4, S2	Code + Report	Week 14

14. Assessment criteria

The point-rating letter system for assessing the educational achievements of students with their interpretation in the traditional grading scale:

Letter system assessment	The digital equivalent of points	Percentage content	Traditional system assessment	General description of grading criteria
A	4,0	95-100	Excellent	The student has knowledge of the subject in the full scope of the curriculum, understands the discipline deeply enough; shows a high level of knowledge that exceeds the volume provided by the syllabus, gives an exhaustive answer
A-	3,67	90-94		The student has knowledge of the subject in the full scope of the curriculum, understands the discipline deeply enough; gives an exhaustive answer
B+	3,33	85-89	Good	The student shows a complete, well-founded knowledge of the subject, but the answers did not always highlight the main idea, rational methods of calculation were not always used; the answers were mostly brief and
B	3,0	80-84		
B-	2,67	75-79		
C+	2,33	70-74		

				sometimes unclear.
C	2,0	65-69	Satisfactory	The student demonstrates sufficient knowledge of the subject, but without proper depth and justification, the answers are unclear and without proper logical sequence.
C-	1,67	60-64		
D+	1,33	55-59		
D	1,0	50-54		
FX	0,5	25-49	Unsatisfactory	The student demonstrates insufficient knowledge of the subject, positive answers were not given to individual questions.
F	0	0-24		The student demonstrates a very low level of knowledge of the subject.

15. Assessment and evaluation materials (exam questions)

Form of assessment (exam): written.

Example of examination question

Task #1(40 points)
Given a matrix N by M
Fill matrix by positive and negative numbers
You need to **zero out** (transform to a zero) all negative elements, except for minimum

Task #2(30 points)
create structure Candidate
name (string with spaces. Ex: Fiona Mccenley)
work (string. Ex: businessman)
income (int)

We have a private elite club
Create an array of candidate to enter our super private party
There are some rules
We do not like banker
We do not take people that take less than 1000$
We do not like people with surname "Johnson" or "Edisson"
(Note that name and surname is one string. You need to divide it in the program)
Print the list of candidates, which we can pass to the club

Task #3(30 points)
create structure Investor
name (string)
salary (int)
invest_free (boolean)

create structure Company
company_name (string)
cost (int)
investors_number (int)

Investor* investors

We have some companies, that need money to be created.
Each company has an amount of money that needs to be collected **(cost)**
Each company have an array of investors **(investors)**, with some size **(investors_number)**
Investor has name and salary. Each of them, pay 10 percent of their salary per year to create company
Also, there is an investor that doesn't pay anything **(invest_free == true)**
We have some companies
You need to list companies that can collect enough money (cost) from their investors by the 3 years

16. Introducing students to the syllabus

I have read and agree with the requirements of the discipline «_____».

№	Full name of the student	Signature	Date
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			