SYLLABUS
Fall semester 2023-2024 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 (PT)	Lab. Classes (LC)		
76215 Parallel programming	4	1,7	0	3,3	5	7

Academic Course Information

Learning Format	Cycle, Component	Lecture types	Types of practical classes	Form and platform final control
Offline	MD, UC	Problem-oriented	Learning the concepts of object-oriented programming and implementing programs to practice practical skills	Oral, offline

Lecturer - (s)	Karyukin Vladislav Igorevich
e-mail:	vladislav.karyukin@gmail.com vladislav.karyukin@kaznu.kz
Phone:	+77019405992
Assistant – (s)	–
e-mail:	–
Phone:	–

ACADEMIC COURSE PRESENTATION

Purpose of the course	Expected Learning Outcomes (LO) *	Indicators of LO achievement (ID)
This course is aimed at studying the concepts of object-oriented programming, as well as understanding their practical implementation by solving real-life practical problems of varying complexity. Within the discipline, the following aspects will be considered: principles of construction of parallel computing systems, modeling and analysis of parallel computations, principles of developing parallel algorithms and programs	1. (cognitive) Know the theoretical and methodological concepts of parallel programming	1.1 ability to create basic and advanced programs using parallelisms
		1.2 Know the peculiarities of working with cores and memory
		1.3 Knowledge of Concurrency Application Development Methods
	2. (functional) Apply knowledge of mpi4py, Celery, PyCUDA, and NumbaPRO libraries	2.1 Develop programs for parallel computing
		2.2 Using Libraries for Parallel Computing
		2.3developing multifunctional applications that both developers and users understand
	3. (functional) Developing programs of various complexity levels: from a simple console to a product of academic and industrial importance.	3.1 Be able to implement program code using parallelism
		3.2 Be able to configure applications
		3.3 Be able to create applications on Python
	4. (system) Building complex, multifunctional applications	4.1 Creating Parallel Computing Modules
		4.2 constructing interaction of various structural elements between each other
		4.3 Modifying and Editing Applications
	5. (system) Creating Parallel Computing Applications on the GPU	5.1 Create a New Application
		5.2 Implement Parallel Computing on Video Cards
		5.3 Test Applications
Prerequisites	Programming on Python language, Programming in Java	
Post requisites		
Learning Resources	**Literature:** **Main:**	

	- Python parallel programming cookbook by Giancarlo Zaccone. Packt publishing, 2015. - Python for Everybody: Exploring Data in Python 3 by Dr. Charles Russell Severance, Sue Blumenberg, Elliott Hauser, Aimee Andrion, 2016 - Python Cookbook: Recipes for Mastering Python 3 3rd Edition, Kindle Edition by David Beazley, Brian K. Jones, 2013 - Advanced Python Development: Using Powerful Language Features in Real-World Applications 1st ed. Edition by Matthew Wilkes, 2021. - Learning Python 5ed: Powerful Object-Oriented Programming, Mark Lutz, 2013. - Fluent Python: Clear, Concise, and Effective Programming, Luciano Ramalho, 2015. **Additional:** 1. Natural Language Processing with Python and spaCy: A Practical Introduction, Yuli Vasiliev, 2021 2. Learning Scientific Programming with Python, Christian Hill, 2021 **Internet resources:** - Parallel programming. https://aaltoscicomp.github.io/python-for-scicomp/parallel/ - Parallel Processing in Python. https://www.geeksforgeeks.org/parallel-processing-in-python/ - A guide to Python multiprocessing and parallel programming. https://www.sitepoint.com/python-multiprocessing-parallel-programming/ **Software and Internet Resources:** Python IDE, Anaconda Navigator Python, Microsoft Visual Studio, PyCharm, Microsoft Office Word, WinRAR, WordPad, Power Point, Adobe Reader, Paint.
Academic course policy	The academic policy of the course is determined by the Academic Policy and the Policy of Academic Integrity of Al-Farabi Kazakh National University . Documents are available on the main page of IS Univer. **Integration of science and education.** The research work of students, undergraduates and doctoral students is a deepening of the educational process. It is organized directly at the departments, laboratories, scientific and design departments of the university, 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 obtaining new knowledge using modern research and information technologies. A research university teacher integrates the results of scientific activities into the topics of lectures and seminars (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 assignments. **Attendance.** The deadline for each task is indicated in the calendar (schedule) for the implementation of the content of the course. Failure to meet deadlines results in loss of points. **Academic honesty.** Practical/laboratory classes, IWS develop the student's independence, critical thinking, and creativity. Plagiarism, forgery, the use of cheat sheets, cheating at all stages of completing tasks are unacceptable. Compliance with academic honesty during the period of theoretical training and at exams, in addition to the main policies, is regulated by the "Rules for the final control" , "Instructions for the final control of the autumn / spring semester of the current academic year" , "Regulations on checking students' text documents for borrowings". Documents are available on the main page of IS Univer. **Basic principles of inclusive education.** The educational environment of the university is conceived as a safe place where there is always support and equal attitude from the teacher to all students and students to 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, progress is more about what they can do than what they can't. Diversity enhances all aspects of life. All students, especially those with disabilities, can receive counseling assistance by vladislav.karyukin@gmail.com / +77019405992 or via video link in MS Teams

INFORMATION ABOUT TEACHING, LEARNING AND ASSESSMENT

Score-rating letter system of assessment of accounting for educational achievements				Assessment Methods
Grade	Digital equivalent points	points, % content	Assessment according to the traditional system	**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
A	4.0	95-100	Great	
A-	3.67	90-94		

B+	3.33	85-89	Fine	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 3-4 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**
B-	2.67	75-79			
C+	2.33	70-74			
C	2.0	65-69		Activity at lectures	0
C-	1.67	60-64		Work in practical classes	25
D+	1.33	55-59	Satisfactorily	Independent work	25
D	1.0	50-54		Design and creative activity	10
FX	0.5	25-49		Final control (exam)	40
F	0	0-24	Unsatisfactory	TOTAL	100

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

A week	Topic name	Number of hours	Max. score * * *
	MODULE 1 Foundations of parallel computing		
1	**L 1.** Getting started with parallel computing	1	0
	LC 1. Python in a parallel world	2	5
2	**L 2.** Memory organization	1	0
	LC 2. Thread-based Parallelism	2	5
	IWST 1. Consultation on the implementation of IWS 1		
3	**L 3.** Parallel programming models	1	0
	LC 3. Thread synchronization with Lock and RLock	2	10
	IWS 1. Implementation of project with parallel operations in Python		20
4	**L 4.** Python in parallel programming	1	0
	LC 4. Thread synchronization with a condition	2	10
	IWST 2. Acceptance of IWS 1		
5	**L 5.** Thread-based parallelism	1	0
	LC 5. Process-based parallelism	2	10
	IWST 3. Consultation on the implementation of IWS 2		
	Module 2 Multiprocessing with Python		
6	**L 6.** Thread synchronization with Lock and RLock	1	0
	LC 6. Using a process in a subclass	2	10
	IWS 2. Creating the application with distributed processes		20
7	**L 7.** Multiprocessing	1	0
	Lab 7. Using the mpi4py Python module	2	10
	IWST 4. Acceptance of IWS 2		
Midterm control 1			100
8	**L 8.** Subclass processes	1	0
	LC 8. Distributed python	2	5
	IWST 5. Consultation on the implementation of IWS 3		
9	**L 9.** mpi4py Python module	1	0
	LC 9. Scientific computing with SCOOP	2	5
	IWS 3. Test		5
10	**L 10.** Distributed python	1	0
	LC 10. GPU programming with Python	2	20
	IWST 6. Acceptance of IWS 3		
	Module 3 Parallel programming on GPU		
11	**L 11.** GPU programming with python	1	0
	LC 11. Matrix multiplication with PyCUDA	2	5
	IWST 7. Consultation on the implementation of IWS 4		
12	**L 12.** PyCUDA	1	0
	LC 12. GPU programming with Numba	2	10

	IWS 4. Developing the parallel application with CUDA		20
13	**L 13.** PyCUDA matrix multiplication	1	0
	LC 13. Using the PyOpenCL module	2	10
14	**L 14.** NumbaPRO	1	0
	LC 14. Evaluating element-wise operations with PyOpenCL	2	10
15	**L 15.** PyOpenCL application	1	0
	Lab 15. Creating the parallel computing application	2	10
Midterm control 1			**100**
Final control (exam)			**100**
TOTAL for course			**100**

Dean _____ **Urmashev B.A.**

Head of the Department of _____ **Mussiraliyeva Sh. Zh.**

Lecturer _____ **Karyukin V.I.**

RUBRICATOR OF THE SUMMATIVE ASSESSMENT

CRITERIA EVALUATION OF LEARNING OUTCOMES

IWS 1. Implementation of project with parallel operations in Python (20% of 100% of MC1)

Criterion	**"Excellent" Max. weight in 16-20%**	**"Good" Max. weight in 11-15%**	**"Satisfactory" Max. weight in 5-10%**	**"Unsatisfactory" Max. weight in 1-4%**
Knowledge and understanding of basic concepts of parallel operations	Understanding the degree of relevance and reliability of the data found. Knowledge and understanding of all parallel operations	Understanding the degree of relevance and reliability of the data found. Knowledge of most operations with parallel operations	Limited understanding of the relevance and validity of parallel operations	Superficial understanding/lack of understanding of the degree of relevance and reliability of the data found. Lack of concept of parallel operations
Coding skills	Clear presentation of the program code, absence of syntax errors in the code	There are small logical errors in the program code	A large number of logical and syntax errors in the program code, which make it practically unworkable	No code or just a few lines of code
Writing a report	The writing demonstrates clarity, conciseness, and accuracy	The writing demonstrates clarity, conciseness and correctness. Mostly no errors	There are some key errors in the writing and the clarity needs improvement.	The writing is unclear and it is difficult to follow the content. Lots of errors in the text

IWS 2. Creating the application with distributed processes (20% of 100% of MC1)

Criterion	"Excellent" Max. weight in 16-20%	"Good" Max. weight in 11-15%	"Satisfactory" Max. weight in 5-10%	"Unsatisfactory" Max. weight in 1-4%
Knowledge and understanding of basic concepts of distributed processes	Understanding the degree of relevance and reliability of the data found. Knowledge and understanding of all distributed processes	Understanding the degree of relevance and reliability of the data found. Knowledge of most operations with distributed processes	Limited understanding of the relevance and validity of distributed processes	Superficial understanding/lack of understanding of the degree of relevance and reliability of the data found. Lack of concept of distributed processes
Coding skills	Clear presentation of the program code, absence of syntax errors in the code	There are small logical errors in the program code	A large number of logical and syntax errors in the program code, which make it practically unworkable	No code or just a few lines of code
Writing a report	The writing demonstrates clarity, conciseness, and accuracy	The writing demonstrates clarity, conciseness and correctness. Mostly no errors	There are some key errors in the writing and the clarity needs improvement.	The writing is unclear and it is difficult to follow the content. Lots of errors in the text

IWS 3. Writing a test **(20% of 100% MC2)**

Criterion	**"Excellent"** **Max. weight in 16-20%**	**"Good"** **Max. weight in 11-15%**	**"Satisfactory"** **Max. weight in 5-10%**	**"Unsatisfactory"** **Max. weight in 1-4%**
Knowledge of solutions to test tasks	Full understanding of all test tasks and the correct answers to them	Almost complete understanding of test tasks and answers to them	Partial understanding of test tasks	Lack of understanding of test tasks and answers to questions asked
Writing program code for test tasks	Clear presentation of the program code, absence of syntax errors in the code	There are small logical errors in the program code	A large number of logical and syntax errors in the program code, which make it practically unworkable	No code or just a few lines of code
Writing a report	The writing demonstrates clarity, conciseness, and accuracy	The writing demonstrates clarity, conciseness and correctness. Mostly no errors	There are some key errors in the writing and the clarity needs improvement.	The writing is unclear and it is difficult to follow the content. Lots of errors in the text

IWS 4. Developing the parallel application with CUDA (20% of 100% of MC2)

Criterion	**"Excellent" Max. weight in 16-20%**	**"Good" Max. weight in 11-15%**	**"Satisfactory" Max. weight in 5-10%**	**"Unsatisfactory" Max. weight in 1-4%**
Knowledge and understanding of basic concepts of parallel applications with CUDA	Understanding the degree of relevance and reliability of the data found. Knowledge and understanding of parallel applications with CUDA	Understanding the degree of relevance and reliability of the data found. Knowledge of most operations with parallel applications with CUDA	Limited understanding of the relevance and validity of parallel applications with CUDA	Superficial understanding/lack of understanding of the degree of relevance and reliability of the data found. Lack of concept of parallel applications with CUDA
Coding skills	Clear presentation of the program code, absence of syntax errors in the code	There are small logical errors in the program code	A large number of logical and syntax errors in the program code, which make it practically unworkable	No code or just a few lines of code
Writing a report	The writing demonstrates clarity, conciseness, and accuracy	The writing demonstrates clarity, conciseness and correctness. Mostly no errors	There are some key errors in the writing and the clarity needs improvement.	The writing is unclear and it is difficult to follow the content. Lots of errors in the text