Federal State Autonomous Educational Institution of Higher Education
"Russian Peoples' Friendship University named after Patrice Lumumba"

Engineering Academy
(name of the main educational unit (POU) - developer of the EP HE)

DISCIPLINE WORK PROGRAM

PROGRAMMING
(name of discipline/module)

Recommended by MSSN for the following areas of training/specialty:

01.04.02 APPLIED MATHEMATICS AND INFORMATION SCIENCE
(code and name of the area of training/specialty)

The discipline is mastered as part of the implementation of the main professional educational program of higher education (EP HE):

BALLISTIC DESIGN OF SPACE COMPLEXES AND SYSTEMS
(name (profile/specialization) EP HE)

2024

1. GOAL OF DISCIPLINE MASTERING

The discipline "Programming" is included in the master's program "Ballistic design of space complexes and systems" in the direction of 01.04.02 "Applied mathematics and computer science" and is studied in the 1st semester of the 1st year. The discipline is implemented by the Department of Mechanics and Control Processes. The discipline consists of 26 sections and 65 topics and is aimed at studying basic sorting and searching algorithms, graph algorithms, dynamic programming methods, modern programming paradigms, approaches to parallel and distributed programming technologies

The goal of mastering the discipline is for students to master practical skills in algorithmization and programming.

2. REQUIREMENTS FOR THE RESULTS OF MASTERING THE DISCIPLINE

Mastering the discipline "Programming" is aimed at developing the following competencies (parts of competencies) in students:

Table 2.1. List of competencies formed in students when mastering the discipline (results of mastering the discipline)

Cipher	**Competence**	**Indicators of Competency Achievement** (within this discipline)
UK-1	Able to critically analyze problem situations based on a systematic approach and develop an action strategy	UK-1.1 Analyzes the task, highlighting its basic components;; UK-1.2 Identifies and ranks the information required to solve the task;; UK-1.3 Searches for information to solve a given problem using various types of requests;; UK-1.4 Offers options for solving a problem, analyzes the possible consequences of their use;;
UK-7	Able to search for the necessary sources of information and data, perceive, analyze, remember and transmit information using digital means, as well as using algorithms when working with data received from various sources in order to effectively use the information received to solve problems; evaluate information, its reliability, build logical conclusions based on incoming information and data	UK-7.1 Searches for the necessary sources of information and data, perceives, analyzes, remembers and transmits information using digital means, as well as using algorithms when working with data received from various sources in order to effectively use the received information to solve problems;; UK-7.2 Evaluates information, its reliability, builds logical conclusions based on incoming information and data.;
OPK-1	Able to solve current problems of fundamental and applied mathematics	GPC-1.1 Analyzes problems in the field of fundamental and applied mathematics;; GPC-1.2 Formulates the research objectives;; OPK-1.3 Solves current problems of fundamental and applied mathematics.;
OPK-2	Able to improve and implement new mathematical methods for solving applied problems	GPC-2.1 Uses the results of applied mathematics to master and adapt new methods for solving problems in the field of professional interests;; OPK-2.2 Implements and improves new methods for solving applied problems in the field of professional activity;; OPK-2.3 Conducts qualitative and quantitative analysis of the resulting solution in order to construct the optimal option.;
OPK-3	Able to develop mathematical models and analyze them when solving problems in the field of professional activity	OPK-3.1 Develops mathematical models in the field of applied mathematics and computer science;; OPK-3.2 Analyzes mathematical models for solving applied problems of professional activity;;

Cipher	Competence	Indicators of Competency Achievement (within this discipline)
		GPC-3.3 Develops and analyzes new mathematical models for solving applied problems of professional activity in the field of applied mathematics and computer science.;
OPK-4	Able to combine and adapt existing information and communication technologies to solve problems in the field of professional activity, taking into account information security requirements	GPC-4.1 Analyzes problems of applied mathematics and computer science using information technology;; OPK-4.2 Takes into account the basic requirements of information security;; OPK-4.3 Uses modern information and communication technologies to solve problems in the field of applied mathematics and computer science, taking into account information security requirements.;
PC-1	Able to formulate goals and objectives of scientific research in the field of applied mathematics and computer science, computer technology and modern programming technologies, select methods and means for solving problems	PC-1.1 Has fundamental knowledge acquired in the field of mathematical and (or) natural sciences, programming and information technology;; PC-1.2 Can find, formulate and solve standard problems in their own research activities in the field of applied mathematics and computer science, computer technology and modern programming technologies;; PC-1.3 Has practical experience in research activities in the field of applied mathematics and computer science, computer technology and modern programming technologies.;
PC-2	Able to apply modern theoretical and experimental methods for developing mathematical models of studied objects and processes related to professional activities in the field of training and participate in their implementation in the form of software products	PC-2.1 Knows modern theoretical and experimental methods for developing mathematical models, innovative design tools and elements of architectural solutions of information systems;; PC-2.2 Can develop and implement algorithms for mathematical models based on languages and application packages for modeling;; PC-2.3 Has practical experience in developing options for implementing information systems using innovative tools.;

3. PLACE OF DISCIPLINE IN THE STRUCTURE OF HE EP

Discipline "Programming" refers to the mandatory part of block 1 “Disciplines (modules)” of the educational program of higher education.

As part of the educational program of higher education, students also master other disciplines and/or practices that contribute to achieving the planned results of mastering the “Programming” discipline.

Table 3.1. List of components of EP HE that contribute to achieving the planned results of mastering the discipline

Cipher	Name of competency	Previous disciplines/modules, practices*	Subsequent disciplines/modules, practices*

UK-7	Able to search for the necessary sources of information and data, perceive, analyze, remember and transmit information using digital means, as well as using algorithms when working with data received from various sources in order to effectively use the information received to solve problems; evaluate		Advanced Methods of Remote Sensing and Geoinformation Systems;
			Pre-Graduation Internship in Industry;
			Practical Training and Research in Dynamics and Control of Space Systems (online from RUDN Mission Control Center) / Research work;
			Technological Training;

Cipher	Name of competency	Previous disciplines/modules, practices*	Subsequent disciplines/modules, practices*
	information, its reliability, build logical conclusions based on incoming information and data		

UK-1	Able to critically analyze problem situations based on a systematic approach and develop an action strategy		Practical Training in Receiving Remote Sensing Data from Satellites and its Interpretation (online from RUDN Mission Control Center) / Research;
			Practical Training and Research in Dynamics and Control of Space Systems (online from RUDN Mission Control Center) / Research work;
			Technological Training;
			Pre-Graduation Internship in Industry;
			Advanced Methods of Remote Sensing and Geoinformation Systems;
			Structures & Materials Modeling;
			System Design;
			Dynamics and Control of Space Systems;
			Project "Drone Systems Engineering. Part 1";

OPK-1	Able to solve current problems of fundamental and applied mathematics		Pre-Graduation Internship in Industry;

OPK-2	Able to improve and implement new mathematical methods for solving applied problems		Aerospace Systems;
			Structures & Materials Modeling;
			System Design;
			On-board Energy;
			Dynamics and Control of Space Systems;
			Pre-Graduation Internship in Industry;
			Technological Training;

OPK-3	Able to develop mathematical models and analyze them when solving problems in the field of professional activity		Pre-Graduation Internship in Industry;
			Technological Training;
			Aerospace Systems;
			Structures & Materials Modeling;
			System Design;
			On-board Energy;
			Dynamics and Control of Space Systems;
			Project "Drone Systems Engineering. Part 1";
			Project "Drone Systems Engineering. Part 2";

Cipher	Name of competency	Previous disciplines/modules, practices*	Subsequent disciplines/modules, practices*

OPK-4	Able to combine and adapt existing information and communication technologies to solve problems in the field of professional activity, taking into account information security requirements		Project "Drone Systems Engineering. Part 1";
			Project "Drone Systems Engineering. Part 2";
			Pre-Graduation Internship in Industry;
			Practical Training in Receiving Remote Sensing Data from Satellites and its Interpretation (online from RUDN Mission Control Center) / Research;
			Practical Training and Research in Dynamics and Control of Space Systems (online from RUDN Mission Control Center) / Research work;
			Technological Training;

PC-1	Able to formulate goals and objectives of scientific research in the field of applied mathematics and computer science, computer technology and modern programming technologies, select methods and means for solving problems		Pre-Graduation Internship in Industry;
			Practical Training in Receiving Remote Sensing Data from Satellites and its Interpretation (online from RUDN Mission Control Center) / Research;
			Practical Training and Research in Dynamics and Control of Space Systems (online from RUDN Mission Control Center) / Research work;
			Technological Training;
			Advanced Methods of Remote Sensing and Geoinformation Systems;
			System Design;
			Dynamics and Control of Space Systems;

PC-2	Able to apply modern theoretical and experimental methods for developing mathematical models of studied objects and processes related to professional activities in the field of training and participate in their implementation in the form of software products		Advanced Methods of Remote Sensing and Geoinformation Systems;
			System Design;
			Project "Drone Systems Engineering. Part 1";
			Pre-Graduation Internship in Industry;
			Practical Training in Receiving Remote Sensing Data from Satellites and its Interpretation (online from RUDN Mission Control Center) / Research;
			Practical Training and Research in Dynamics and Control of Space Systems (online from RUDN Mission

Cipher	Name of competency	Previous disciplines/modules, practices*	Subsequent disciplines/modules, practices*

			Control Center) / Research work;
			Technological Training;

* - to be filled out in accordance with the competency matrix and SUP EP VO
** - elective disciplines/practices

4. SCOPE OF DISCIPLINE AND TYPES OF STUDY WORK

The total labor intensity of the “Programming” discipline is “4” credit units.

Table 4.1. Types of educational work by periods of mastering the educational program of higher education for full-time study.

Type of educational work	**TOTAL,**ac.ch.		**Semester(s)**
			1
Contact work, ac.ch.	*20*		*20*
Lectures (LK)	10		10
Laboratory work (LR)	10		10
Practical/seminar sessions (SZ)	0		0
Independent work of students, ac.ch.	*97*		*97*
Control (exam/test with assessment), academic degree.	*27*		*27*
Total labor intensity of the discipline	**ac.ch.**	***144***	*144*
	credit units	***4***	*4*

5. CONTENT OF DISCIPLINE

Table 5.1. Contents of the discipline (module) by type of academic work

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
Section 1	Basic elements of Python syntax	1.1	Basic Python 3 syntax. Memory model and basic data types	LK, LR
		1.2	Loops and lists. Functions.	LK, LR
Section 2	Elements of the theory of algorithms	2.1	The concept of an algorithm. Turing machine. Computability. Complexity theory. Exponentiation: analysis of the algorithm (smart exponentiation).	LK, LR
		2.2	The backpack problem. Greedy algorithm. The gradient descent method is an example of a greedy algorithm. “Divide and conquer” strategy. Recursive algorithm.	LK, LR
Section 3	Programming paradigms. Object Oriented Programming	3.1	Basic principles of programming. Procedural programming.	LK, LR
		3.2	Object-oriented programming (OOP). Functional programming.	LK, LR
		3.3	Features of OOP. Classes and objects. Inheritance. Implementation of OOP in Python	LK, LR
Section 4	Sorting and searching algorithms	4.1	Sorting by selection. Insertion sort. Sorting using the Bubble Method. Merge sort. Quick sort	LK, LR
		4.2	Finding the median. Sequential search. Methods for narrowing the area. Sorting in Python.	LK, LR
Section 5	Graph Algorithms	5.1	Graphs and their analysis. Graph representation. Depth-first and breadth-first graph traversal	LK, LR
		5.2	Restoring the shortest path. The problem of moving a chess knight	LK, LR
		5.3	Dijkstra's algorithm. Queue and stack. Queue and stack in Python	LK, LR
Section 6	Dynamic programming	6.1	Bellman's optimality principle. Concept of bottom-up and top-down decision.	LK, LR
		6.2	Problem about the number of routes. Similarities and differences between dynamic programming and the concept of “divide and conquer”	LK, LR

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
		6.3	ATM problem. Dynamic programming and games	LK, LR
Section 7	Parallel Algorithms	7.1	Prerequisites. Classification of computing systems. CPU and GPU processors.	LK, LR
		7.2	Characteristics of parallel algorithms. Types of Inconsistent Programming in Python.	LK, LR
		7.3	Processes and Threads in Python. Asynchronous programs.	LK, LR
Section 8	Program optimization	8.1	Methods for optimizing and accelerating Python programs. Profiling programs in Python.	LK, LR
		8.2	Line_profiler module. Python compilation: Ahead-of-time and Just-in-time compilation. Numba module.	LK, LR
		8.3	Cython as an extension of the Python language. Features of program development in Cython	LK, LR
Section 9	C\C++. Introduction	9.1	C and C++ language features, history and evolution. Machine-oriented programming languages and principles of computer operation.	LK, LR
		9.2	Code translation. Types of broadcast. Differences between interpreters and compilers.	LK, LR
		9.3	Comparison of programs in Python and C/C++. Scope of application and C/C++ languages.	LK, LR
Section 10	Basic Syntax Elements	10.1	Block structure of programs in C/C++ languages, syntactic rules for selecting blocks and their types.	LK, LR
		10.2	Basic instructions: branch (or conditional statement), loops (while, do while and for), unconditional jump operator, multiple choice operator.	LK, LR
		10.3	Syntactic constructions for working with functions: declaration, definition, call. Call stack. Comparison of goto and return	LK, LR
Section 11	Arrays and Pointers	11.1	Pointers and addresses. Working with pointers and addresses. Array as a data	LK, LR

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
			structure: storage in memory, access to elements.	
		11.2	Creating static arrays. Address arithmetic	LK, LR
Section 12	Static and dynamic memory.	12.1	Rules for creating static arrays, its initialization and use. One-dimensional and multidimensional static arrays. Dynamic memory (C style).	LK, LR
		12.2	Dynamic memory (C++ style). Functions for working with dynamic memory, memory allocation and deallocation operations	LK, LR
		12.3	Creating one-dimensional and multidimensional dynamic arrays	LK, LR
Section 13	Structured Data Types	13.1	Arrays, character strings, structures, union, enumerated data type, bit fields. Syntactic features of declaration, initialization and operation.	LK, LR
		13.2	Features of “packaging” in memory. Examples of using. Dynamic data structures: vector, queue (stack), list, as examples of organizing work with structured data in a dynamic mode.	LK, LR
Section 14	Error catching	14.1	Syntax of the exception handling operation. Examples of using.	LK, LR
Section 15	Data input/output	15.1	The concept of stream and buffer. Keyboard, screen and file as a source and receiver of data. Organizing input and output streams in C++.	LK, LR
		15.2	Writing data to a stream and reading data from a stream. Positioning data in a stream. File working modes: read-write, character-text format and combinations thereof	LK, LR
		15.3	Text and binary files, and the peculiarity of data storage in them. Direct Files	LK, LR
Section 16	Object-oriented programming in C++	16.1	Creating classes and objects. Setting up access modifiers: public, private and protected. Friendly functions and classes.	LK, LR
		16.2	The keyword is this. Organization of inheritance operation in C++ language.	LK, LR

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
			Virtual functions and overloading of functions and operators.	
Section 17	Using Libraries	17.1	Overview and examples of using STL and BOOST	LK, LR
Section 18	Parallel algorithms and systems	18.1	Classification of computing systems. CPU and GPU processors. Characteristics of parallel algorithms	LK, LR
		18.2	Types of inconsistent programming. Parallel computing standards: interaction between supercomputer nodes, interaction between cores of one CPU within one node, accelerators within one node	LK, LR
Section 19	Algorithms in external memory	19.1	Organization of calculations taking into account the hierarchical structure of memory. Buffering when reading and writing.	LK, LR
		19.2	Complex and dynamic data structures. Algorithms on graphs in external memory (BFS, DFS, search for connected components, MST)	LK, LR
Section 20	OpenMP technology	20.1	Parallel computing using the OpenMP standard.	LK, LR
		20.2	Basic information. Threads and processes. Parallel and serial areas	LK, LR
		20.3	Parallel loops and parallel regions. Automatic parallelization of cycles.	LK, LR
Section 21	MPI technology	21.1	Parallel computing using the MPI standard. Basic information. Basic MPI procedures.	LK, LR
		21.2	MPI data types. Methods of transmitting messages. Reception and transmission of messages by processes	LK, LR
Section 22	OpenACC technology	22.1	Parallel computing using the OpenACC standard	LK, LR
		22.2	Review of GPU performance in various applications. Comparison of computing accelerators. Basic principles for achieving high productivity	LK, LR
		22.3	Benefits of OpenACC. Execution model: gangs, workers, vectors. Parallel, kernels, loop directives	LK, LR
		22.4	Data attributes. Data regions: data, enter data,	LK, LR

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
			exit data.Additional data management constructs: cache, update, declare	
		22.5	Asynchronous execution - async and wait. Atomic operations. Global variables. OpenACC in C++	LK, LR
Section 23	Software and hardware architecture CUDA	23.1	GPU architecture. GPU memory hierarchy. CUDA software model.	LK, LR
		23.2	Using C++ libraries for OpenCL and CUDA programming	LK, LR
Section 24	Introduction to Distributed Object Technologies	24.1	The concept of a distributed information processing system. Types and properties of distributed systems.	LK, LR
		24.2	Software architecture of information systems. Managing the interaction of heterogeneous applications	LK, LR
		24.3	Basic mechanisms of distributed object technologies.	LK, LR
Section 25	Basic models of distributed object technologies	25.1	Types of distributed applications. Cloud technologies. Definition of cloud computing. Multilayer architecture of cloud applications. Components of cloud applications.	LK, LR
		25.2	Advantages and disadvantages of cloud computing. The most common cloud platforms. GRID technologies.	LK, LR
		25.3	GRID architecture. GRID standards. Parametric models of GRID performance. Comparison of GRID and Cloud Computing.	LK, LR
Section 26	Application integration issues	26.1	Application integration problems. Comprehensive application integration. Message Brokers. Publish/subscribe interaction model	LK, LR
		26.2	Workflow management systems. Application servers.	LK, LR

* - to be filled out only for full-time education: LC – lectures; LR – laboratory work; SZ – practical/seminar classes.

6. MATERIAL AND TECHNICAL SUPPORT OF DISCIPLINE

Table 6.1. Material and technical support of the discipline

Audience type	Auditorium equipment	Specialized educational/laboratory equipment, software and materials for mastering the discipline (if necessary)
Lecture	An auditorium for conducting lecture-type classes, equipped with a set of specialized furniture; board (screen) and technical means of multimedia presentations.	
Computer class	A computer class for conducting classes, group and individual consultations, ongoing monitoring and intermediate certification, equipped with personal computers ([Parameter] pcs.), a whiteboard (screen) and technical means for multimedia presentations.	
For independent work	An auditorium for independent work by students (can be used for seminars and consultations), equipped with a set of specialized furniture and computers with access to EIOS.	

* - the audience for independent work of students is MANDATORY!

7. EDUCATIONAL, METHODOLOGICAL AND INFORMATIONAL SUPPORT OF DISCIPLINE

Main literature:

1. Python 3. Essentials. Prokhorenok N., Dronov V., BHV-Petersburg, 2019 – 610 pp.;
2. Python. Express course. Seder N., St. Petersburg: Peter, 2019 – 480 pp.;
3. Algorithms. Reference book with examples in C, C++, Java and Python. Heineman J., Pollis G., Selkov S., St. Petersburg: Alpha Book LLC, 2017 – 432 pp.;
4. Programming in a high-level language. C/C++. Khabibullin I.Sh., St. Petersburg: BHVPePeterburg, 2006 – 512 p.
5. C-C++. Programmer's Handbook. G. Shildt, Williams, 2003 - 429 pp.;
6. Programming in C++ in Visual Studio 2010 Express. Prokhorenok N.A., 2010 – 71 p.
7. C Programming Language Brian W. Kernighan, D.M. Ritchie, Williams, 2015 – 288 s
8. C++ programming language. Stroustrup B., Martynov N.N., Moscow: Binom, 2011. - 1135 p.
9. Parallel computing. Voevodin V.V., Voevodin Vl. V., St. Petersburg: BHV-Petersburg, 2002
10. Parallel and distributed programming using C++. Hughes K., Hughes T., M.: Williams Publishing House, 2004;
11. Parallel programming using OpenMP. Levin M.P. M.: Binom. Knowledge Laboratory, 2008
12. Parallel programming with OpenACC. Farber R., Newnes, 2016 – 316 pp.;
13. CUDA technology in examples: an introduction to programming GPUs. Sander J., Kendrot E. M.: DMK Press, 2011 - 232 p.
14. Distributed systems. Principles and paradigms. Tanenbaum E., van Steen M. St. Petersburg: Peter, 2003. - 877 p.
15. Development of distributed applications on the .NET Framework platform. M. Sarah, R. Bill, H. Shannon, B. Mark. St. Petersburg: Peter, 2008. - 608 p.

Additional literature:

1. Automating Wounded Tasks with Python: A Practical Guide for Beginners. Sveirart El., M.: "ID Williams", 2017 - 592 p.

2. Numerical methods: Computational workshop. Vabishchevich P.N., M.: "LIBROKOM", 2010 – 320 pp.;

3. Programming language C. Lectures and exercises. S. Prata, M.: Williams Publishing House, 2013 – 960 pp.;

4. C++. Sacred knowledge. Dewhurst S., St. Petersburg: Symbol Plus, 2012 – 240 p.

5. Algorithms. Reference book with examples in C, C++, Java and Python. Heineman J., Pollis G., Selkov S., St. Petersburg: Alpha Book LLC, 2017 – 432 p.

6. Algorithms construction, analysis and implementation in the C programming language. Vorozhtsov A.V., Vinokurov N.A., Moscow: MIPT, 2007 – 452 p.

7. Programming and computer science. Antonyuk V.A., Ivanov A.P., Moscow: Faculty of Physics. Moscow State University named after M. V. Lomonosova, 2015 – 64 p.

8. Sequential and parallel algorithms. Miller R., Boxer L.M.: Binom. Knowledge Laboratory, 2006

9. Introduction to parallel methods for solving problems. Yakobovsky M.V.M.: Moscow University Publishing House, 2013 – 328 pp.;

10. Debugging applications for Microsoft .NET and Microsoft Windows. R. John. M.: Microsoft Press. Russian Edition. 2008.

11. XML. New perspectives of WWW. Bumfray F, Direnzo O, Duckett J. Publisher: DMK Press, 2006. - 688 p.

12. Chertovskoy V.D. Databases: theory and practice: a textbook for bachelors. Sovetov B. Ya., Tsekhanovsky V. V. M.: YURAYT, 2011. - 459 p.

Resources of the information and telecommunications network "Internet":

1. EBS of RUDN University and third-party EBS, to which university students have access based on concluded agreements

- Electronic library system of RUDN - EBS RUDN http://lib.rudn.ru/MegaPro/Web
- EBS "University Library Online" http://www.biblioclub.ru
- EBS Yurayt http://www.biblio-online.ru
- EBS "Student Consultant" www.studentlibrary.ru
- EBS "Trinity Bridge"

2. Databases and search engines

- electronic fund of legal and regulatory technical documentation http://docs.cntd.ru/
- Yandex search engine https://www.yandex.ru/
- Google search engine https://www.google.ru/
- SCOPUS abstract database http://www.elsevierscience.ru/products/scopus/

Educational and methodological materials for students' independent work when mastering a discipline/module:*

1. A course of lectures on the discipline "Programming".

* - all educational and methodological materials for students' independent work are posted in accordance with the current procedure on the discipline page in TUIS!

8. ASSESSMENT MATERIALS AND POINT-RATING SYSTEM FOR ASSESSING THE LEVEL OF COMPETENCIES FOR A DISCIPLINE

Evaluation materials and point-rating system* for assessing the level of development of competencies (parts of competencies) based on the results of mastering the discipline"Programming" are presented in the Appendix to this Work Program of the discipline.

* - OM and BRS are formed on the basis of the requirements of the relevant local regulatory act of RUDN University.

DEVELOPER:

Assistant professor		Karatunov Maxim Olegovich
Position, PBU	*Signature*	*Last name I.O.*

HEAD OF BUP:

Head of the department		Razumny Yuri Nikolaevich
Position PBU	*Signature*	*Last name I.O.*

HEAD OF OP VO:

Professor		Razumny Yuri Nikolaevich
Position, PBU	*Signature*	*Last name I.O.*