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

Academy of Engineering
(name of the main educational unit (MEU) that developed the educational program of higher education)

WORKING PROGRAM OF THE DISCIPLINE

COMPUTER SCIENCE AND PROGRAMMING
(name of discipline/module)

Recommended for the field of study/specialty:

27.03.04 CONTROL IN TECHNICAL SYSTEMS
(code and name of the training area/specialty)

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

DATA SCIENCE AND SPACE SYSTEMS
(name (profile/specialization) of the educational institution of higher education)

2025

1. THE GOAL OF MASTERING THE DISCIPLINE

The discipline "Computer Science and Programming" is included in the bachelor's program "Data Science and Space Systems" in the direction 27.03.04 "Control in Technical Systems" and is studied in semesters 1, 2, 3, 4, 5 of the 1st, 2nd, 3rd years. The discipline is implemented by the Department of Mechanics and Control Processes. The discipline consists of 29 sections and 101 topics and is aimed at studying the theoretical and practical foundations of information technology and programming. Particular attention is paid to the analysis of methods for solving typical problems and the analysis of the area of their application in professional activities.

The purpose of mastering the discipline is to develop fundamental knowledge and skills in applying programming technologies to solve a wide range of problems necessary for professional activity and mastering subsequent disciplines.

2. REQUIREMENTS TO THE RESULTS OF MASTERING THE DISCIPLINE

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

Table 2.1. List of competencies developed in students while mastering the discipline (results of mastering the discipline)

Cipher	Competence	**Indicators of Competence Achievement** (within the framework of this discipline)
GPC-6	Capable of developing and using algorithms and programs, modern information technologies, methods and means of control, diagnostics and control, suitable for practical application in the field of his professional activity	GPC-6.1 Knows the basic algorithms and programs, modern information technologies, methods and means of control, diagnostics and control, suitable for practical application in the field of his professional activity; GPC-6.2 Able to apply algorithms and programs, modern information technologies, methods and means of control, diagnostics and control, suitable for practical application in the field of his professional activity; GPC-6.3 Confidently uses algorithms and programs, modern information technologies, methods and means of control, diagnostics and control, suitable for practical application in the field of his/her professional activity;
GPC-9	Capable of performing experiments according to specified methods and processing the results using modern information technologies and technical means	GPC-9.1 Knows modern information technologies and technical means; GPC-9.2 Able to apply modern information technologies and technical means to process experimental results; GPC-9.3 Possesses modern information technologies and technical means for performing experiments and processing results;
PC-1	Capable of collecting, processing and interpreting modern scientific research data necessary to draw conclusions on relevant scientific research, including Earth remote sensing data	PC-1.1 Knows modern methods of collecting, processing and interpreting data from modern scientific research necessary for drawing conclusions on relevant scientific research; PC-1.2 Able to apply modern methods and tools for processing and interpreting scientific research data; PC-1.3 Possesses the basic skills of collecting, processing and interpreting data from modern scientific research necessary for drawing conclusions on relevant scientific research;

3. PLACE OF THE DISCIPLINE IN THE STRUCTURE OF THE EDUCATIONAL EDUCATIONAL EDUCATION

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

As part of the higher education program, students also master other disciplines and/or practices that contribute to the achievement of the planned results of mastering the discipline "Computer Science and Programming".

Table 3.1. List of components of the educational program of higher education that contribute to the achievement of the planned results of mastering the discipline

Cipher	Name of competence	Previous courses/modules, practices*	Subsequent disciplines/modules, practices*
GPC-6	Capable of developing and using algorithms and programs, modern information technologies, methods and means of control, diagnostics and control, suitable for practical application in the field of his professional activity		Research work / Scientific research work; Undergraduate Training; Automatic Control Theory; Space Flight Mechanics;
GPC-9	Capable of performing experiments according to specified methods and processing the results using modern information technologies and technical means		Undergraduate Training; Technological Training; Optimal Control Methods; Analysis of Geoinformation Data;
PC-1	Capable of collecting, processing and interpreting modern scientific research data necessary to draw conclusions on relevant scientific research, including Earth remote sensing data		Research work / Scientific research work; Technological Training; Undergraduate Training; Space Flight Mechanics; Automatic Control Theory; *Virtual and Augmented Reality Technology***; *Virtual and augmented reality technologies***; Optimal Control Methods; Analysis of Geoinformation Data;

* - filled in in accordance with the competency matrix and the SUP EP HE

** - elective disciplines/practices

4. SCOPE OF THE DISCIPLINE AND TYPES OF STUDY WORK

The total workload of the discipline "Computer Science and Programming" is 20 credit units.

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

Type of academic work	**TOTAL,**ac.h.		**Semester(s)**				
			1	**2**	**3**	**4**	**5**
Contact work, academic hours	*299*		*36*	*68*	*72*	*51*	*72*
Lectures (LC)	141		18	34	36	17	36
Laboratory work (LW)	158		18	34	36	34	36
Practical/seminar classes (SC)	0		0	0	0	0	0
Independent work of students, academic hours	*313*		*72*	*94*	*81*	*21*	*45*
Control (exam/test with assessment), academic hours	*108*		*0*	*18*	*27*	*36*	*27*
General complexity of the discipline	**ac.h.**	***720***	*108*	*180*	*180*	*108*	*144*
	credit.ed.	***20***	*3*	*5*	*5*	*3*	*4*

5. CONTENT OF THE DISCIPLINE

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

Section number	Name of the discipline section	Section Contents (Topics)		Type of academic work*
Section 1	Information and computer science	1.1	Basic concepts. Subject and tasks of informatics	LC
		1.2	Information and its properties	LC, LW
		1.3	Arithmetic and logical principles of computer operation	LC, LW
		1.4	Information coding	LC, LW
		1.5	Prospects for the development of informatics	LC
		1.6	Modern aspects of programming. Classification and areas of application of modern programming languages	LC
Section 2	Computing technology	2.1	History of development and classification of computers	LC
		2.2	Computer architecture. Composition of the computing system	LC, LW
		2.3	Principles of operation of elements of a computing system	LC, LW
		2.4	Computer networks. Client-server architecture	LC, LW
Section 3	Software	3.1	System software	LC, LW
		3.2	Application software	LC, LW
Section 4	Basic concepts of modeling and algorithmization	4.1	Stages of solving a problem using a computer	LC
		4.2	Models and their classification	LC, LW
		4.3	Concept and properties of the algorithm. Methods of describing the algorithm	LC, LW
Section 5	Python programming language	5.1	Interpreter. Basic syntax. Memory model. Data types	LC, LW
		5.2	Logical constructions. Cycles and branches	LC, LW
		5.3	Functions. Passing arguments. Scope. Call stack	LC, LW
		5.4	Working with files. Properties and types of files. Data serialization	LC, LW
		5.5	Block organization of the program. Modules and packages. The pip package manager	LC, LW
Section 6	Python libraries for solving scientific and applied problems	6.1	Data Visualization with Matplotlib Library	LC, LW
		6.2	Solving Statistics and Linear Algebra Problems with NumPy and Pandas Libraries	LC, LW
Section 7	Programming paradigms	7.1	Main paradigms and their features: procedural programming, object-oriented programming, functional programming	LC
		7.2	Object-oriented programming in Python. Encapsulation, polymorphism, inheritance. Classes and objects. Class inheritance	LC, LW
		7.3	Functional programming in Python. Anonymous functions: syntax and context of use. Function decorators	LC, LW
		7.4	Visual block programming as a tool for creating and managing VR worlds	LW
Section 8	Data structures	8.1	Basic data structures and their properties	LC, LW
		8.2	Standard data structures of the Python language and features of working with them	LC, LW
		8.3	Graph data structure. Python libraries implementing graph data structure and features of working with them	LC, LW
Section 9	Algorithms	9.1	The concept of computation and computability. Classification of algorithms. Turing machines.	LC
		9.2	Evaluation of algorithm complexity	LC, LW
		9.3	Sorting algorithms	LC, LW
		9.4	Search algorithms	LC, LW

Section number	Name of the discipline section	Section Contents (Topics)		Type of academic work*
		9.5	Graph Algorithms	LC, LW
Section 10	Python libraries for solving scientific and applied problems	10.1	SciPy library functionality and features of working with them	LC, LW
		10.2	Functional capabilities of the SymPy library and features of working with them	LC, LW
Section 11	Operating Systems Basics	11.1	History of development and main functions of operating systems	LC
		11.2	Basics of working in the command interpreter	LW
		11.3	Architectural features of operating systems	LC
		11.4	Process and memory control	LC, LW
		11.5	Input/output control	LC, LW
Section 12	Version control systems (VCS)	12.1	History of development of SLE. Basic concepts and terms. Classification and modern SLE	LC
		12.2	Using Git and Organizing Your Software Development Workflow	LW
Section 13	Basics of the C programming language	13.1	History of development, features and scope of application of the C language	LC
		13.2	Declaration and definition of variables. Variable types. Type conversion.	LC, LW
		13.3	Arithmetic and logical operators. Bitwise operators. Precedence and order of calculation.	LC, LW
		13.4	Control structures. Branching and loops, unconditional jump and multiple choice operators	LC, LW
Section 14	Functions and structure of the program	14.1	Functions. Syntactic constructions for working with functions: declaration, definition, call. Recursion. Call stack. Block structure of the program	LC, LW
		14.2	External variables and scope. Static and register variables. Header files.	LC, LW
		14.3	The process of compiling programs. Preprocessor, file inclusion, macro substitution, conditional compilation	LC, LW
Section 15	Pointers and arrays	15.1	Pointers and Addresses. Pointers and Function Arguments	LC, LW
		15.2	Arrays. Address arithmetic	LC, LW
		15.3	Pointers to pointers. Multidimensional arrays	LC, LW
		15.4	Command Line Arguments. Function Pointers. Complex Declarations	LC, LW
Section 16	Structures	16.1	Basics of working with structures. Structures and functions. Pointers to structures	LC, LW
		16.2	Defining new types	LC, LW
		16.3	Unions and bit fields	LC, LW
Section 17	Input/output operations	17.1	Standard Input/Output Facilities	LC, LW
		17.2	Variable Length Argument Lists Formatted Input	LC, LW
		17.3	Reading and writing files	LC, LW
		17.4	Error handling	LC, LW
Section 18	Standard Library	18.1	String Operations. Analysis, Classification, and Conversion of Characters	LC, LW
		18.2	Command Execution. Memory Control	LC, LW
		18.3	Mathematical functions. Random number generator	LC, LW
Section 19	Basics of the C++ programming language	19.1	History of development, features and scope of application of the C language. Differences between the C and C++ languages	LC
		19.2	Types and Declarations. Namespaces. Pointers, References, Arrays, and Structures	LC, LW
		19.3	Expressions and operators. Functions	LC, LW
		19.4	Exceptions. Key words throw, catch	LC, LW

Section number	Name of the discipline section	Section Contents (Topics)		Type of academic work*
		19.5	Source files and programs. Separate compilation	LC
Section 20	Abstraction mechanisms (OOP)	20.1	Classes and objects. Class members. Constructors and destructors. Class composition. Access modifiers. Overloading class methods.	LC, LW
		20.2	Operator overloading. Function operators. Type casting operators. Class friends.	LC, LW
		20.3	Class inheritance. Derived classes. Virtual functions. Class hierarchies and abstract classes	LC, LW
		20.4	Templates. Definition of a template. Specification of templates. Type checking. Function templates. Specialization	LC, LW
Section 21	Exception handling	21.1	Error Handling. Exception Grouping	LC, LW
		21.2	Catching Exceptions. Resource Control	LC, LW
		21.3	Exception specification	LC, LW
		21.4	Exceptions and Efficiency: Alternatives to Error Handling	LC
Section 22	Class Hierarchies	22.1	Designing a Class Hierarchy. Traditional Class Hierarchies	LC, LW
		22.2	Multiple Inheritance and Access Control	LC, LW
Section 23	Standard Library STL	23.1	Standard containers	LC, LW
		23.2	Algorithms and classes of functional objects	LC, LW
		23.3	Iterators and allocators	LC, LW
		23.4	Strings and Streams	LC, LW
		23.5	Classes for mathematical calculations	LC, LW
Section 24	Programming technology	24.1	Basic concepts and approaches	LC
		24.2	Problems of developing complex software systems	LC
		24.3	Block-hierarchical approach to creating complex systems	LC, LW
		24.4	Life cycle and stages of development	LC, LW
		24.5	Evaluation of the quality of software development processes	LC
Section 25	Techniques for ensuring the technological effectiveness of software products	25.1	Software technology. Modules and their properties	LC, LW
		25.2	Top-down and bottom-up development	LC
		25.3	Structured and "non-structured" programming. Means of describing structured algorithms	LC
		25.4	The style of the program design. Efficiency and technology	LC, LW
Section 26	Defining software requirements	26.1	Classification of software products by functional feature. Basic operational requirements	LC
		26.2	Development of technical specifications. Fundamental solutions for the initial stages of design	LC, LW
Section 27	Structural approach	27.1	Requirements analysis and specification definition in a structured approach. State transition diagrams, functional diagrams, data flow diagrams. Data structures and data component relationship diagrams. Mathematical models of problems	LC
		27.2	Software design with a structured approach. Structural and functional diagrams. Step-by-step detailing. Constantine maps. Designing data structures. Designing based on data decomposition. Case technologies	LC, LW
Section 28	Object approach	28.1	Requirements analysis and specification definition in the object approach. UML. Identification of use cases. Construction of a conceptual model of the subject area. Description of behavior	LC
		28.2	Software design with an object approach. Designing the structure. Defining relationships	LC, LW

Section number	Name of the discipline section	Section Contents (Topics)		Type of academic work*
			between objects and classes. Designing classes. Layout. Placing distributed software systems. Spiral development model	
Section 29	Software testing	29.1	Types of quality control. Manual control. Structural and functional testing	LC, LW
		29.2	Modular, complex and evaluation testing	LC, LW

* - filled in only for FULL-TIME education: LC – lectures; LW – laboratory work; PW – practical/seminar classes.

6. LOGISTIC AND TECHNICAL SUPPORT OF DISCIPLINE

Table 6.1. Material and technical support of the discipline

Audience type	Equipping the auditorium	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; a board (screen) and technical means for multimedia presentations.	
Computer class	A computer room for conducting classes, group and individual consultations, ongoing monitoring and midterm assessment, equipped with personal computers (15 units), a board (screen) and technical means for multimedia presentations.	
For independent work	A classroom for independent work of students (can be used for conducting seminars and consultations), equipped with a set of specialized furniture and computers with access to the Electronic Information System.	

* - the audience for independent work of students MUST be indicated!

7. EDUCATIONAL, METHODOLOGICAL AND INFORMATIONAL SUPPORT OF THE DISCIPLINE

Main literature:

1.

- Computer Science. Basic course. Simonovich S.V., St. Petersburg: Peter, 2011 - 640 p.

- Learning Python. Volume 1. 5th edition. M. Lutz, St. Petersburg: Dialectics, 2019 — 832 p.

- Python 3. The Essentials. PrLChorenLC N., Dronov V., St. Petersburg: BHV-Petersburg, 2019 — 610 p.

2.

- C Programming Language Brian W. Kernighan, D.M. Ritchie, M.: Williams, 2019 - 288 p.

- How to Program in C. 7th edition. X. Deitel, P. Deitel, M.: BINOM, 2017 — 1000 p.

- The C Programming Language. Lectures and Exercises. Stephen Prata. Moscow: Williams, 2015 — 928 p.
- Algorithms. HandboLC with examples in C, C++, Java and Python. Heineman J., Pollis G., SeLCov S., St. Petersburg: OOO "Alfa-kniga", 2017 — 432 p.
3.
- C++ programming language. Stroustrup B., Martynov N.N., M: Binom, 2011. - 1135 p.
- How to Program in C++. 8th edition. X. Deitel, P. Deitel, M.: Binom, 2020 - 1032 p.
- C++. Sacred knowledge. Dewhurst S., St. Petersburg: Symbol Plus, 2012 – 240 p.
- Patterns of object-oriented design. Gamma E., Helm R., Johnson R., Vlissides J., St. Petersburg: Piter, 2020 - 448 p.
- Algorithms. HandboLC with examples in C, C++, Java and Python. Heineman J., Pollis G., SeLCov S., St. Petersburg: OOO "Alfa-kniga", 2017 — 432 p.
-

Further reading:

1.
- The Computer Science BoLC: A complete introduction to computer science in one boLC. Johnson Thomas, Canada: Leanpub, 2020, - 410 p.
- Automating Routine Tasks with Python: A Practical Guide for Beginners. Sveyart El., M.: "ID Williams", 2017 — 592 p.
- Classic Computer Science Problems in Python. Kopets D. SPb.: Piter, 2020 — 256 p.
- The Big BoLC of Python Projects. Sweigart El. SPb.: Piter, 2022 — 432 p.;
- Learning Python: Game Programming, Data Visualization, Web Applications. Matiz E. SPb.: Piter, 2020 — 512 p.
2.
- Algorithms: construction, analysis and implementation in the C programming language. Vorozhtsov A.V., VinLCurov N.A., Moscow: MIPT, 2007 — 452 p.
- Programming and computer science. AntonyUC V.A., Ivanov A.P., Moscow: Faculty of Physics. Moscow State University named after M. V. Lomonosova, 2015 - 64 p.
- Pro Git. Version 2.1.x. Scott Chacon, Ben Straub, USA, New York: Apress, 2020 - 506 p. URL: https://git-scm.com/boLC/en/v2
3.
- Object-oriented thinking. Weisfeld M., St. Petersburg: Piter, 2014 — 304 p.
- Object-oriented programming: Workshop. Pavlovskaya T.A., Shchupak Yu.A., St. Petersburg: Piter, 2006. - 265 p.
- Structures and algorithms of data processing: object-oriented approach and implementation in C++. Kubensky A.A. SPb.: BHV-Petersburg, 2004 — 464 p.

Resources of the information and telecommunications network "Internet":

1. RUDN University EBS and third-party EBSs to which university students have access on the basis of concluded agreements
- Electronic library system of RUDN - EBS RUDN http://lib.rudn.ru/MegaPro/Web
- Electronic library system "University library online" http://www.biblioclub.ru
- EBS Yurayt http://www.biblio-online.ru
- Electronic library system "Student Consultant" www.studentlibrary.ru
- Electronic library system "Troitsky Bridge"
2. Databases and search engines
- electronic fund of legal and normative-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 independent work of students in mastering a discipline/module:*

1. Lecture course on the subject "Computer Science and Programming".

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

DEVELOPER:

Associate Professor		Saltykova Olga Alexandrovna
Position, Department	*Signature*	*Surname I.O.*

HEAD OF THE DEPARTMENT:

Head of Department		Razumny Yuri Nikolaevich
Position of the Department	*Signature*	*Surname I.O.*

HEAD OF THE EP HE:

Head of Department		Razumny Yuri Nikolaevich
Position, Department	*Signature*	*Surname I.O.*