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

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

COURSE SYLLABUS

COMPUTER SCIENCE AND PROGRAMMING
(name of discipline/module)

Recommended by the Didactic Council for the Education Field of:

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

The course instruction is implemented within the professional education programme of higher education:

DATA ENGINEERING AND SPACE SYSTEMS CONTROL
(name (profile/specialization) EP HE)

2024

1. GOAL OF DISCIPLINE MASTERING

The discipline “Computer Science and Programming” is included in the bachelor’s program “Data Engineering and Space Systems Control” in the direction of 27.03.04 “Control in Technical Systems” and is studied in the 1st, 2nd, 3rd, 4th, 5th semesters of the 1st, 2nd, 3rd courses. 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 analysis of the scope of their application in professional activities.

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

2. REQUIREMENTS FOR THE RESULTS OF MASTERING THE DISCIPLINE

Mastering the discipline “Informatics and 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)
GPC -6	Able to develop and use algorithms and programs, modern information technologies, methods and means of control, diagnostics and management, suitable for practical application in the field of his professional activity	GPC-6.1 Knows basic algorithms and programs, modern information technologies, methods and means of control, diagnostics and management, suitable for practical use in the field of their professional activities; GPC -6.2 Can use algorithms and programs, modern information technologies, methods and means of control, diagnostics and management, suitable for practical use in the field of their professional activities; GPC-6.3 Confidently masters algorithms and programs, modern information technologies, methods and means of control, diagnostics and management, suitable for practical application in the field of his professional activity;
GPC -9	Able to perform experiments using given methods and process the results using modern information technologies and technical means	GPC -9.1 Knows modern information technologies and technical means; GPC -9.2 Able to use modern information technologies and technical means to process the results of experiments; 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 how to collect, process and interpret modern scientific research data necessary to form 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 modern scientific research data necessary to form conclusions on relevant scientific research;

3. PLACE OF DISCIPLINE IN THE STRUCTURE OF HE EP

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

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***

GPC -6	Able to develop and use algorithms and programs, modern information technologies, methods and means of control, diagnostics and management, suitable for practical application in the field of his professional activity		Research work / Scientific research work;
			Undergraduate practice / Pre-graduate practice;
			Automatic Control Theory;
			Space Flight Mechanics;

GPC -9	Able to perform experiments using given methods and process the results using modern information technologies and technical means		Undergraduate practice / Pre-graduate practice;
			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 practice / Pre-graduate practice;
			Space Flight Mechanics;
			Automatic Control Theory;
			*Virtual and Augmented Reality Technology***;
			*Virtual and augmented reality technologies***;
			Optimal Control Methods;
			Analysis of Geoinformation Data;

* - 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 “Computer Science and Programming” discipline is “21” 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	**2**	**3**	**4**	**5**
Contact work, ac.ch.	*316*		*36*	*68*	*72*	*68*	*72*
Lectures (LC)	158		18	34	36	34	36
Laboratory work (LR)	158		18	34	36	34	36
Practical/seminar sessions (SZ)	0		0	0	0	0	0
Independent work of students, ac.ch.	*368*		*36*	*58*	*81*	*76*	*117*
Control (exam/test with assessment), academic degree.	*72*		*0*	*18*	*27*	*0*	*27*
Total labor intensity of the discipline	**ac.ch.**	***756***	*72*	*144*	*180*	*144*	*216*
	credit units	***21***	*2*	*4*	*5*	*4*	*6*

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	Information and computer science	1.1	Basic concepts. Subject and tasks of computer science	OK
		1.2	Information and its properties	LK, LR
		1.3	Arithmetic and logical foundations of computer operation	LK, LR
		1.4	Encoding information	LK, LR
		1.5	Prospects for the development of computer science	LK
		1.6	Modern aspects of programming. Classification and areas of application of modern programming languages	LK
Section 2	Computer technology	2.1	History of development and classification of computers	LK
		2.2	Computer architecture. Composition of the computing system	LK, LR
		2.3	Principles of functioning of elements of a computer system	LK, LR
		2.4	Computer networks. Client-server architecture	LK, LR
Section 3	Software	3.1	System software	LK, LR
		3.2	Application software	LK, LR
Section 4	Basic concepts of modeling and algorithmization	4.1	Stages of solving a problem using a computer	LK
		4.2	Models and their classification	LK, LR
		4.3	Concept and properties of the algorithm. Ways to describe the algorithm	LK, LR
Section 5	Python programming language	5.1	Interpreter. Basic syntax. Memory model. Data types	LK, LR
		5.2	Logical constructions. Loops and branches	LK, LR
		5.3	Functions. Passing arguments. Area of visibility. Call stack	LK, LR
		5.4	Working with files. Properties and types of files. Data Serialization	LK, LR
		5.5	Block organization of the program. Modules and packages. pip package manager	LK, LR
Section 6	Python libraries for solving scientific and applied problems	6.1	Data visualization using Matplotlib library	LK, LR
		6.2	Solving statistics and linear algebra problems using NumPy and Pandas libraries	LK, LR
Section 7	Programming Paradigms	7.1	Basic paradigms and their features: procedural programming, object-oriented programming, functional programming	LK
		7.2	Object-oriented programming in Python. Encapsulation, polymorphism, inheritance. Classes and objects. Class inheritance	LK, LR
		7.3	Functional programming in Python. Anonymous functions: syntax and context of use. Function decorators	LK, LR
		7.4	Visual block programming as a tool for creating and managing VR worlds	LR
Section 8	Data structures	8.1	Basic data structures and their properties	LK, LR
		8.2	Standard Python data structures and features of working with them	LK, LR
		8.3	Graph data structure. Python libraries implementing the graph data structure and features of working with them	LK, LR
Section 9	Algorithms	9.1	The concept of calculation and computability. Classification of algorithms. Turing machines.	LK

Section number	**Name of the discipline section**	**Contents of the section (topic)**		**Type of educational work***
		9.2	Algorithm complexity assessment	LK, LR
		9.3	Sorting algorithms	LK, LR
		9.4	Search algorithms	LK, LR
		9.5	Graph Algorithms	LK, LR
Section 10	Python libraries for solving scientific and applied problems	10.1	Functionality of the SciPy library and features of working with them	LK, LR
		10.2	Functionality of the SymPy library and features of working with them	LK, LR
Section 11	Operating Systems Basics	11.1	History of development and main functions of operating systems	LK
		11.2	Command Sheet Basics	LR
		11.3	Architectural features of operating systems	OK
		11.4	Process and memory management	LK, LR
		11.5	I/O Management	LK, LR
Section 12	Version control systems (VCS)	12.1	History of the development of SLE. Basic concepts and terms. Classification and modern SLE	LK
		12.2	Using Git and organizing your software creation workflow	LR
Section 13	C Programming Language Basics	13.1	History of development, features and scope of the C language	LK
		13.2	Declaration and definition of variables. Variable types. Type conversion.	LK, LR
		13.3	Arithmetic and logical operators. Bitwise operators. Priority and order of calculation.	LK, LR
		13.4	Control structures. Branching and looping, unconditional branch and multiple choice operators	LK, LR
Section 14	Functions and structure of the program	14.1	Functions. Syntactic constructions for working with functions: declaration, definition, call. Recursion. Call stack. Block program structure	LK, LR
		14.2	External variables and scope. Static and register variables. Header files.	LK, LR
		14.3	The process of compiling programs. Preprocessor, file inclusion, macro substitution, conditional compilation	LK, LR
Section 15	Pointers and Arrays	15.1	Pointers and addresses. Pointers and Function Arguments	LK, LR
		15.2	Arrays. Address arithmetic	LK, LR
		15.3	Pointers to pointers. Multidimensional arrays	LK, LR
		15.4	Command line arguments. Function pointers. Complex declarations	LK, LR
Section 16	Structures	16.1	Basics of working with structures. Structures and functions. Pointers to structures	LK, LR
		16.2	Defining New Types	LK, LR
		16.3	Unions and bitfields	LK, LR
Section 17	I/O Operations	17.1	Standard I/O	LK, LR
		17.2	Variable length argument lists. Formatted input	LK, LR
		17.3	Reading and writing files	LK, LR
		17.4	Error processing	LK, LR
Section 18	Standard Library	18.1	Operations with strings. Analysis, classification and transformation of symbols	LK, LR
		18.2	Executing commands. Memory management	LK, LR
		18.3	Mathematical functions. Random number generator	LK, LR

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
Section 19	Basics of the C++ programming language	19.1	History of development, features and scope of the C language. Differences between the C and C++ languages	LK
		19.2	Types and declarations. Namespaces. Pointers, references, arrays and structures	LK, LR
		19.3	Expressions and operators. Functions	LK, LR
		19.4	Exceptions. Keywords throw, catch	LK, LR
		19.5	Source files and programs. Separate compilation	LK
Section 20	Abstraction mechanisms (OOP)	20.1	Classes and objects. Class members. Constructors and destructors. Composition of classes. Access modifiers. Overloading class methods.	LK, LR
		20.2	Operation overload. Functions-operations. Type conversion operations. Class friends	LK, LR
		20.3	Class inheritance. Derived classes. Virtual functions. Class hierarchies and abstract classes	LK, LR
		20.4	Templates. Definition of a template. Specification of templates. Type checking. Function templates. Specialization	LK, LR
Section 21	Exception Handling	21.1	Error processing. Grouping exceptions	LK, LR
		21.2	Catching exceptions. Resource management	LK, LR
		21.3	Exception Specification	LK, LR
		21.4	Exceptions and efficiency. Error Handling Alternatives	LK
Section 22	Class hierarchies	22.1	Designing a class hierarchy. Traditional class hierarchies	LK, LR
		22.2	Multiple inheritance and access control	LK, LR
Section 23	Standard STL Library	23.1	Standard containers	LK, LR
		23.2	Algorithms and classes of functional objects	LK, LR
		23.3	Iterators and allocators	LK, LR
		23.4	Strings and Streams	LK, LR
		23.5	Math classes	LK, LR
Section 24	Programming technology	24.1	Basic concepts and approaches	LK
		24.2	Problems of developing complex software systems	LK
		24.3	Block-hierarchical approach to creating complex systems	LK, LR
		24.4	Life cycle and development stages	LK, LR
		24.5	Assessing the quality of software creation processes	LK
Section 25	Techniques for ensuring the manufacturability of software products	25.1	Software manufacturability. Modules and their properties	LK, LR
		25.2	Top-down and bottom-up development	LK
		25.3	Structural and “non-structural” programming. Tools for describing structured algorithms	LK
		25.4	Program design style. Efficiency and technology	LK, LR
Section 26	Defining Software Requirements	26.1	Classification of software products based on functionality. Basic Operational Requirements	LK
		26.2	Development of technical specifications. Fundamental solutions for the initial stages of design	LK, LR
Section 27	Structural approach	27.1	Requirements analysis and specification determination using a structured approach. State transition diagrams, functional diagrams, data flow diagrams. Data structures and data component relationship diagrams. Mathematical models of problems	LK
		27.2	Software design using a structured approach. Structural and functional diagrams. Step by step	LK, LR

Section number	Name of the discipline section	Contents of the section (topic)		Type of educational work*
			detailing. Maps of Constantine. Designing data structures. Design based on data decomposition. Case technologies	
Section 28	Object approach	28.1	Requirements analysis and specification determination using an object-based approach. UML. Define use cases. Construction of a conceptual model of the subject area. Description of behavior	LK
		28.2	Software design using an object-based approach. Development of the structure. Define relationships between objects and classes. Designing classes. Layout. Hosting distributed software systems. Spiral development model	LK, LR
Section 29	Software Product Testing	29.1	Types of quality control. Manual control. Structural and functional testing	LK, LR
		29.2	Unit, end-to-end and evaluation testing	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.	MS Visual Studio Code, NotePad++, Python, GIT, MS Visual Studio Community Edition, MinGW, Varwin Education, Open VR, Windows Subsystem for Linux (WSL2), Glasgow Haskell Compiler, Cabal, Haskell Tool Stack, haskell-language-server, PostgreSQL .
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.

- Computer science. Basic course. Simonovich S.V., St. Petersburg: Peter, 2011 - 640 p.
- We studyPython. Volume 1. 5th edition. M. Lutz, St. Petersburg: Dialectics, 2019 - 832 p.
-Python 3. Essentials. Prokhorenok N., Dronov V., St. Petersburg: BHV-Petersburg, 2019 - 610 p.

2.

- Programming languageC. Brian W. Kernighan, D.M. Ritchie, M.: Williams, 2019 - 288 p.
- How to program inC. 7th edition. X. Deitel, P. Deitel, M.: BINOM, 2017 - 1000 p.
- Programming languageC. Lectures and exercises. Stephen Prata. M.: Williams, 2015 - 928 p.
- 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.

3.

- Programming languageC++. Stroustrup B., Martynov N.N., M: Binom, 2011. - 1135 p.
- How to program inC++. 8th edition. X. Deitel, P. Deitel, M.: Binom, 2020 - 1032 p.
-C++. Sacred knowledge. Dewhurst S., St. Petersburg: Symbol Plus, 2012 – 240 p.
- Object-oriented design patterns. Gamma E., Helm R., Johnson R., Vlissides J., St. Petersburg: Peter, 2020 - 448 p.
- 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.
-

Additional literature:

1.

- The Computer Science Book: A complete introduction to computer science in one book. Johnson Thomas, Canada: Leanpub, 2020, - 410 p.
- Automate routine tasks usingPython: A Practical Guide for Beginners. Sveyrart El., M.: "ID Williams", 2017 - 592 p.
- Classic problemsComputer Science in Python. Kopets D. SPb.: Peter, 2020 - 256 p.
- Big book of projectsPython. Sweigart El. St. Petersburg: Peter, 2022 - 432 pp.;
- We studyPython: game programming, data visualization, web applications. Matiz E. SPb.: Peter, 2020 - 512 p.

2.

- Algorithms: construction, analysis and implementation in the C programming language.Vorozhtsov A.V., Vinokurov N.A., Moscow: MIPT, 2007 - 452 p.
- Programming and computer science. Antonyuk V.A., Ivanov A.P., Moscow: Faculty of Physics.Moscow State University named after M. V. Lomonosov, 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/book/en/v2

3.

- Object-oriented thinking. Weisfeld M., St. Petersburg: Peter, 2014 - 304 p.
- Object-oriented programming: Workshop. Pavlovskaya T.A., Shchupak Yu.A., St. Petersburg: Peter, 2006. - 265 p.

- Structures and algorithms for data processing: object-oriented approach and implementation inC++. Kubensky A.A. St. Petersburg: BHV-Petersburg, 2004 - 464 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 University - EBS RUDN Universityhttp://lib.rudn.ru/MegaPro/Web
    - EBS "University Library Online"http://www.biblioclub.ru
    - EBS Lawhttp://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 documentationhttp://docs.cntd.ru/
    - Yandex search enginehttps://www.yandex.ru/
    - search systemGoogle https://www.google.ru/
    - abstract databaseSCOPUS 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 "Computer Science and 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 "Computer Science and 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		Saltykova Olga Alexandrovna
Position	*Signature*	*Last name I.O.*

HEAD OF DEPARTMENT:

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

HEAD OF EP HE:

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