National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Department of Descriptive Geometry, Engineering and Computer Graphics

Informatics and programming. Part 1.

Python programming language

Working program of the credit module (educational component) (Syllabus)

Details of the academic discipline

Level of higher education	***First (bachelor's)***
Field of knowledge	*10 Natural sciences*
Speciality	*104 Physics and Astronomy*
Educational program	*Computer Simulation of Physical Processes*
Discipline status	*Normative*
Form of study	*Full-time*
Year of preparation, semester	*1st year, autumn semester*
Scope of the credit module	*4 ECTS credits (120 hours), including:* *classroom – 2.4 ECTS credits (72 hours), in particular – lectures – 1.2 ECTS credits (36 hours), practical classes – 0.6 ECTS credits (18 hours) and laboratory classes – 0.6 ECTS credits (18 hours);* *independent work – 1.6 ECTS credits (48 hours)*
Semester control / control measures	*Exam / ICR*
Timetable	*http://roz.kpi.ua/*
Language of instruction	*Ukrainian*
Information about Course Leader / Instructors	Lecturer: Ph.D., *Assoc. Prof., Yablonsky Petr Nikolaevich, e-mail: ypn@ukr.NET, mobile: +380679977575*
Course Placement	*https://do.ipo.kpi.ua/course/view.php?id=3194*

The program of the discipline

1. Description of the discipline, its purpose, subject andlearning outcomes

One of the components of the success of any technology is the community created around it. The community around Python is one of the most powerful in the IT field. This is evidenced by the fact that Python is supported by such IT giants as, for example: Google, Dropbox, Mozilla, Facebook, Microsoft (recently very actively, in particular with Visual Studio), Intel (actively conducting research work in the field of parallel computing in Python) and many others. As for large and popular projects written in Python, these are such "monsters" as: YouTube (most of the codebase is completely in Python), the first version of the Google search spider was written in Python, Instagram (500M users in Python) and many other interesting and popular applications.

Thus, it can be argued that large corporations are not afraid to build their business around Python, they are confident that the technology will live. There is a huge demand for Python programmers now

because there are fewer of these specialists than there are open vacancies. Moreover, the variety of applications indicates a wide range of tasks that are solved with the help of Python.

As a result of studying the credit module (educational component) "Informatics and Programming. Part 1. Python programming language*" students will master the basic basics* of functional and object-oriented programming using the Python programming language as an example, which allows them to quickly move on to solving problems in the relevant subject areas. Mastering the *Python programming language allows you to create both prototypes of software systems and the software systems themselves, helps in the integration of software to solve specific scientific and production tasks.*

The purpose of the credit module (educational component) "Informatics and Programming. Part 1. Python programming language" *is the formation of the following competencies in higher education applicants:*

- *ability to abstract thinking, analysis and synthesis (GC 1);*
- *skills in the use of information and communication technologies (GC 3);*
- *the ability to make informed decisions (GC 5);*
- *ability to use in practice the basic knowledge of mathematics as a mathematical apparatus of physics and astronomy in the study and study of physical and astronomical phenomena and processes (FC 2);*
- *ability to work with sources of educational and scientific information (FC 9).*

The subject of the credit module (educational component) "Informatics and Programming. Part 1. Python programming language" *is a concept of basic functional and object-oriented programming, algorithmization and creation of computer programs for solving applied problems using programming languages.*

Learning Outcomes

According to the requirements of the program of the discipline "Informatics and Programming", applicants for higher education after mastering the credit module (educational component) "Informatics and Programming. Part 1. Python programming language" should achieve certain programmatic learning outcomes, namely (PRN 9, PRN 12, PRN 18):

- *be able to apply basic mathematical knowledge used in physics and astronomy: analytic geometry, linear algebra, mathematical analysis, differential and integral equations, probability theory and mathematical statistics, group theory, methods of mathematical physics, theory of functions of a complex variable, mathematical modeling (PRN 9);*
- *be able to present the obtained scientific results, participate in discussions regarding the content and results of their own scientific research (PRN 12);*
- *be able to find the necessary information in printed and electronic sources, analyze, systematize, understand, interpret and use it to solve scientific and applied problems (PRN 18).*

2. Prerequisites and post-requisites of the discipline / credit module (educational component) (place in the structural and logical scheme of education according to the relevant educational program)

Credit module (educational component) "Informatics and Programming. Part 1. Python programming language" is the *initial stage for the study of the following credit modules and disciplines in which programming is used (studied), including for research and computer modeling of various phenomena and processes in physics, in particular: "Special sections of computer science", "Computer modeling in physics", etc.*

3. The content of the discipline

Chapter 1. Introduction to Programming:

Topic 1. Introduction to the Computer Science and Programming Course

Topic 2. Familiarity with Python and programming environments.

Chapter 2. Python Programming:

Topic 3. Python Alphabet.

Topic 4. Structure of the program. Code organization.

Topic 5. Data types in programming. Variables and literals.

Topic 6. Arithmetic and logical (comparison operations) expressions. Basic Mathematics.

Topic 7. Data input/output.

Topic 8. Conditional statement. Branching.

Topic 9. Cycles.

Topic 10. Lines and text.

Topic 11. Routines. User functions.

Topic 12. Data structures.

Topic 13. Working with lists (arrays).

Topic 14. Ways to sort structured data types.

Topic 15. Working with files.

Topic 16. The concept of object-oriented programming (OOP).

Topic 17. Popular modules and extensions.

4. Training Materials & Resources

Basic literature:

1. Vasiliev, *Alexey Nikolaevich. Python programming / O.M. Vasiliev.- Ternopil: Publishing House "Educational Book-Bogdan", 2021. - 503 p.*

Order in the KPI Library:
https://opac.kpi.ua/F/?func=direct&doc_number=000637846&local_base=KPI01

2. Vysotskaya, Viktoriia Anatoliivna. PYTHON : Algorithmization and programming: textbook / V.A. Vysotska, O.V. Oborska; Ministry of Education and Science of Ukraine, Lviv Polytechnic National University. Lviv: Novyi Svit-2000 Publishing House, 2021. – 514 p.

Order in the KPI Library:
https://opac.kpi.ua/F/?func=direct&doc_number=000637149&local_base=KPI01

3. Mattes, Eric. Python Crash Course: A Practical, Project-Oriented Introduction to Programming / Eric Mattes ; translated from English by Olga Belova. Lviv:Old Lion Publishing House, 2021. – 556 p.

Order in the KPI Library:
https://opac.kpi.ua/F/?func=direct&doc_number=000633837&local_base=KPI01

4. Melnyk, Igor Vitaliyovych. Fundamentals of programming in the Python language: a comprehensive textbook: in 2 volumes / I.V. Melnyk. - Kyiv: Department, 2020. – 2 vols. – Volume 1, Basic principles of building the Python programming language and its main syntactic constructions. – 2020. – 372 p. Volume 2, Advanced Tools of the Python Programming Language. – 2020. – 491 p.
Order in KPI Library: https://opac.kpi.ua/F/?func=direct&doc_number=000633555&local_base=KPI01

5. Kopey, V. B. Python programming language for engineers and scientists: textbook /V.B. Kopei; Ministry of Education and Science of Ukraine, Ivano-Frankivsk National Technical University of Oil and Gas, Department of Computerized Mechanical Engineering. Ivano-Frankivsk: IFNTUOG, 2019. – 267 p.

Order in the KPI Library:

https://opac.kpi.ua/F/?func=direct&doc_number=000610834&local_base=KPI01

Supporting Literature:

1. Anisimov, Anatoly Vasilyevich. Programming of numerical methods in Python: textbook / A. V. Anisimov, A. Y. Doroshenko, S. D. Pogorilyi, Y. Y. Dorogyi; edited by A. V. Anisimov; Ministry of Education and Science of Ukraine, Taras Shevchenko National University of Kyiv. - Kyiv: VPC "Kyiv University", 2015. - 639 p.

Order in the KPI Library:

https://opac.kpi.ua/F/?func=direct&doc_number=000603361&local_base=KPI01

2. Yakovenko, A. V. Osnovy programming. Python. Part 1 [Electronic resource]: textbook for students studying in the specialty 122 "Computer Science" with the specialization "Information Technologies in Biology and Medicine" / A. V. Yakovenko; KPI them. Igor Sikorsky. - Electronic text data (1file: 1.71 MB). – Kyiv : KPI them. Sikorsky's Game, 2018. – 195 p.

https://ela.kpi.ua/handle/123456789/25111

3. Fundamentals of Programming: Methodical Instructions for Performing Computer Practicums in PYTHON on the Academic Discipline "Fundamentals of Programming" for Students of the Specialty 122 "Computer Science" with the Specialization "Information Technologies in Biology and Medicine" /National Technical University of Ukraine "Kyiv Polytechnic Institute"; compiled by L. M. Dobrovska; editor A. V. Yakovenko. – Kyiv : KPI them. Igor Sikorsky, 2017. – 254 p. (in Russian).

http://ela.kpi.ua/handle/123456789/19094

Information Resources:

1. A set of methodical materials. Sikorsky Distance Learning Platform: https://do.ipo.kpi.ua/course/view.php?id=3193.

2. Fundamentals of programming in Python - YouTube – *Oksana Pasichnyk's distance course on YouTube. – https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp.*

3. https://www.python.org/ (Python's official website).

Educational content

5. Methods of mastering the discipline (educational component)

To master the educational component, lectures, practical, laboratory classes and independent work of students are provided. The materials necessary for mastering the educational component are posted on the distance learning platform "Sikorsky" (https://do.ipo.kpi.ua/course/view.php?id=3194), which gives students the opportunity to remotely access the materials at a convenient time for them, as well as in the case of organizing distance learning.

Lectures

Salary No.	*Title of the lecture topic and list of main questions (list of didactic aids, tasks for SRS with reference to literature)*
1	*Lecture 1. Introduction to the Computer Science and Programming Course*

	Subject and objectives of the course. Number systems. History of Programming Languages. Compilation and interpretation. *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRS: To consolidate in practice the skills of converting numbers into different number systems.*
2	*Lecture 2. Familiarity with Python and programming environments.* *Installing Python. Python language and interpreter. Comparison of Python with other programming languages.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: To consolidate the skills of installing Python in practice.*
3	*Lecture 3. The alphabet of the Python programming language.* *Mathematical operators. Intelligent calculator. Relation operators. Priorities of operations.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: To consolidate in practice the skills of using mathematical operators and relation operators.*
4	*Lecture 4. Structure of the program. Code organization.* *Structure of the program. Python Program Writing Style. Comments.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRS: Execution of the program. Varieties of execution.*
5	*Lecture 5. Data types in programming. Variables and literals.* *Types of data. Variables and literals.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: Data Type Check.*
6	*Lecture 6. Arithmetic and logical expressions. Basic Mathematics.* *Arithmetic operations. Comparison operations. Priorities of operations. Basic logical operations.* *Didactic aids:*

	https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp *Recommended reading: [1-5].* *CPC: Bit Operations.*
7	*Lecture 7. Data input/output.* *Formatted output. Data entry. Exceptions and try-except.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: Module random. Random number generation.*
8	*Lecture 8. Conditional statement. Branching.* *Logical operations. Binary operators. The if* statement. *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: Condition check: True and False.*
9	*Lecture 9. Cycles.* *While* and *for* statements. *Interrupt and continue loops.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: Empty pass* statement.
10	*Lecture 10. Lines and text.* *Splitting and merging strings. Working with substrings. Uppercase and lowercase.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5]. CPC: Check for numbers and whitespace.*
11	*Lecture 11. Subroutines. User Functions.* *Routines. User functions and their applications. Scope of variables. Passing arguments.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: Global and Local Variables.*
12	*Lecture 12. Data structures.* *Lists. Tuples. Dictionaries.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRS: Plurals.*

13	*Lecture 13. Working with lists (arrays).* *Working with vectors and matrices. Filling the list with random data.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRS: Matrices: calculation of the determinant, inverse of matrices.*
14	*Lecture 14. Ways to sort structured data types.* *Ways and methods of sorting. Working with slices.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRS: Familiarize yourself with alternative sorting methods.*
15	*Lecture 15. Working with files.* *Read and write files.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *CPC: Functions for working with directories.*
16	*Lecture 16. The concept of object-oriented programming.* *Advantages and disadvantages of OOP. Classes and definitions of constructs.* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRS: OOP in programming languages. Use of special methods.*
17	*Lecture 17. Popular modules and extensions. Data visualization.* *Some library modules. Data visualization. Numpy. Matplotlib* *Didactic aids:* *https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp* *Recommended reading: [1-5].* *SRC: Working with the Turtle* module.
18	*Lecture 18. Preliminary preparation for the exam. Answers to questions.*

Practical and laboratory classes

The main tasks of the cycle of laboratory and practical (computer practicum) classes are devoted to the acquisition of practical programming skills in Python to solve specific scientific and applied problems.

Salary No.	*Name of the topic of the lesson*	*Number of aud. Hours*
1	***Topic 1. Introduction to Python***	*2*

	Installing Python. Python interpreter. IDLE (Integrated Development and Learning Environment). Structure of the program.	
2	***Topic 2. Calculation of arithmetic expressions (laboratory work No 1).*** *Code organization. Types of data. Calculation of arithmetic expressions. Priorities of operations.*	*4*
3	***Topic 3. Conditional statement. Branching (laboratory work No 2).*** *Data entry. Exceptions and exception handling (try-except). The* if-elif-else statement.	*4*
4	***Topic 4. Cycles (laboratory work No 3).*** *While and for* statements. *Replace one loop statement with another.*	*4*
5	***Modular test.***	*2*
6	***Topic 5. Working with one-dimensional lists (laboratory work No 4).*** *Fill, sort, add, and delete list items.*	*5*
7	***Topic 5. Work with multidimensional lists (laboratory work No 5).*** *Populating the list with random numbers. Working with slices. Use of user functions.*	*5*
8	***Topic 6. Rows. Working with files (laboratory work No 6).*** *Working with strings and substrings. Writing information to a file. Read data from a file.*	*5*
9	***Topic 7. Data visualization (laboratory work No 7).*** *Using numpy and matplotlib* modules. *Building surfaces, graphs, etc.*	*5*
	JUST:	***36***

Modular test

The modular test is held in the 8th-9th week of the academic semester. The modular test includes the task of representing (converting) numbers in different number systems. The results of the modular test are communicated to students at the next practical lesson, and, in case of receiving an unsatisfactory grade, the modular test can be rewritten at the consultation.

6. Independent work of a higher education applicant

Salary No.	*Name of the topic, tasks for the CPC*	*Number of hours on CPS*
1	*Topic 1. Introduction to the Computer Science and Programming Course* *SRS: Elaboration and consolidation of the materials of the lecture, practical and laboratory classes, in particular, to consolidate in practice the skills of converting numbers into different number systems.*	*1*
2	*Topic 2. Familiarity with Python and programming environments.* *CPC: Working out and consolidating the materials of the lecture, practical and laboratory classes, in particular, to consolidate in practice the skills of installing Python.*	*1*
3	*Topic 3. The alphabet of the Python programming language.*	*2*

	SRS: Elaboration and consolidation of the materials of the lecture, practical and laboratory classes, in particular, to consolidate in practice the skills of using mathematical operators and relation operators.	
4	*Topic 4. Structure of the program. Code organization.* *SRS: Elaboration and consolidation of the materials of the lecture, practical and laboratory classes, in particular, to pay attention to the specifics of the program and the types of implementation.*	*2*
5	*Topic 5. Data types in programming. Variables and literals.* *SRS: Processing and consolidation of lecture, practical and laboratory lesson materials, in particular, checking the type of data.*	*2*
6	*Topic 6. Arithmetic and logical expressions. Basic Mathematics.* *SRS: Processing and consolidation of lecture, practical and laboratory materials, including bit operations.*	*2*
7	*Topic 7. Data input/output.* *CPC: Elaboration and consolidation of materials of lectures, practical and laboratory classes, in particular, the work* of the random module *and the generation of random numbers.*	*3*
8	*Topic 8. Conditional statement. Branching.* *CPC: Processing and consolidation of lecture, practical and laboratory lesson materials, in particular, checking the conditions: True and False.*	*2*
9	*Topic 9. Cycles.* *SRS: Processing and consolidation of lecture, practical and laboratory lesson materials, including an empty pass* instruction.	*3*
10	*Topic 10. Lines and text.* *SRS: Processing and consolidation of lecture, practical and laboratory lesson materials, including checking for numbers and whitespaces.*	*2*
11	*Topic 11. Subroutines. User Functions.* *SRS: Elaboration and consolidation of lecture, practical and laboratory lesson materials, in particular global and local variables.*	*3*
12	*Topic 12. Data structures.* *SRS: Elaboration and consolidation of materials of lectures, practical and laboratory classes, in particular the data structure – sets.*	*4*
13	*Topic 13. Working with lists (arrays).* *SRS: Elaboration and consolidation of the materials of the lecture, practical and laboratory lesson, in addition to the calculation of the determinant of the matrix and the inverse of the matrices.*	*4*
14	*Topic 14. Ways to sort structured data types.* *SRS: Elaboration and consolidation of lecture, practical and laboratory lesson materials, in particular, to get acquainted with alternative methods of sorting.*	*2*
15	*Topic 15. Working with files.* *SRS: Elaboration and consolidation of lecture, practical and laboratory lesson materials, in particular, to consider the functions for working with catalogs.*	*2*
16	*Topic 16. The concept of object-oriented programming.*	*3*

	SRS: Elaboration and consolidation of lecture, practical and laboratory lesson materials, in particular OOP in programming languages and the use of special methods.	
17	*Topic 17. Popular modules and extensions. Data visualization.* *CPC: Elaboration and consolidation of the materials of the lecture, practical and laboratory classes, in particular, to get acquainted with the Turtle* module.	*2*
18	*Preparation for the exam.*	*8*
	JUST:	***48***

Policy & Control

7. Academic discipline policy (educational component)

Successful mastering of the credit module "Informatics and Programming. Part 1. Python programming language" requires the applicant for higher education:

- ✓ *compliance with academic integrity, in particular, the applicant's work must demonstrate signs of independence in the performance of tasks, the absence of signs of repetition and plagiarism;*
- ✓ *compliance with the schedule of the educational process, including attending classes;*
- ✓ *systematic elaboration of theoretical material;*
- ✓ *compliance with the schedule for the defense of laboratory work.*

In addition, if the higher education applicant was absent from the class (including for a valid reason), then he should work out this lesson at another time (with another group, at a consultation, independently, using methodological materials and materials that are posted on the Sikorsky distance learning platform, etc.).

8. Types of Control and Rating System for Assessment of Learning Outcomes (CRO)

8.1. The student's rating in the credit module is calculated on a 100-point scale, of which 64 points are the starting scale (rating) and 36 points are individual written examination work. The starting rating consists of the points that the higher education applicant receives (during the semester) for:

- ✓ *performance and defense of 7 laboratory works;*
- ✓ *Performance of modular control work.*

8.2. Criteria for awarding points:

8.2.1. Performance and defense of laboratory work is evaluated from 7 points for each work:

- ✓ *the work was completed at a high level (at least 95% of the requirements were met), submitted on time and a full answer was provided during the defense – 7 points;*
- ✓ *the work was completed at a high level (at least 90% of the requirements were met), submitted on time and during the defense a full answer was provided with minor inaccuracies – 6 points;*
- ✓ *the work was completed at a sufficiently high level (at least 75% of the requirements were met), submitted on time and a fairly complete answer was provided during the defense – 5 points;*
- ✓ *the work was completed at a sufficient level (at least 60% of the requirements were met), submitted on time and an incomplete answer was provided during the defense – 4 points;*
- ✓ *the work was performed at an insufficient level (less than 60% of the requirements were met) – 0 points;*

For each week of delay in the completion and defense of laboratory work, 1 penalty point is awarded (but in total not more than 5 penalty points for all laboratory work).

The presence of all completed and defended laboratory work is a prerequisite for admission to the exam.

8.2.2. Modular control work is evaluated from 15 points according to the following criteria:

- ✓ *fully solved task (minor inaccuracies are possible) – 14-15 points;*
- ✓ *at least 75% of the task is solved – 11-13 points;*

- ✓ *at least 60% of the task is solved – 9-10 points;*
- ✓ *The task is solved by less than 60% – 0 points.*

In case of re-passing (retake) of the modular test, 2 penalty points are awarded.
Passing the modular test is a prerequisite for admission to the exam.

8.3. Calendar control is carried out twice a semester as a monitoring of the current state of compliance with the requirements of the syllabus: the condition of the first calendar control is the performance and defense of 3 laboratory works, and the condition of the second is 5 laboratory works.

8.4. The condition for admission to the exam is the enrollment of all laboratory work, modular control work and a starting rating of at least 40 points.

8.5. At the exam, higher education applicants perform an individual written examination work, which consists of two theoretical questions and one practical task. Each question (task) is evaluated out of 12 points according to the following criteria:

- ✓ *complete answer, at least 95% of the required information (complete, error-free solution of the problem) – 11-12 points;*
- ✓ *sufficiently complete answer, at least 75% of the required information or minor inaccuracies (complete solution of the problem with minor inaccuracies) – 9-10 points;*
- ✓ *incomplete answer, which contains at least 60% of the required information, and some errors (the task was completed with certain shortcomings) – 7-8 points;*
- ✓ *An answer that contains less than 60% of the required information (the task is performed incorrectly or with gross errors and shortcomings) – 0 points.*

8.6. The sum of starting points and points received for individual written examination work is converted to grades on the university scale according to the table:

Score	*Score*
100-95	Perfectly
94-85	Very good
84-75	Well
74-65	Satisfactory
64-60	Enough
Less than 60	Disappointing
Admission conditions are not met	Not allowed

8.7. In the case of carrying out the educational process in a mixed or distance mode, it is allowed to give the final grade for the exam by proportional recalculation of semester grades into the final grade "automatically" on a 100-point scale in *accordance with the Procedurefor* conducting *semester control in distance mode (https://osvita.kpi.ua/sites/default/files/downloads/Reglament%20semestr%20control.pdf).*

9. Additional information on the discipline (educational component)

- *The list of theoretical questions and examples of practical tasks that are submitted for semester control is posted on the Sikorsky distance learning platform (https://do.ipo.kpi.ua/course/view.php?id=3194).*

Work program of the credit module (syllabus):

Compiled by Associate Professor of the Department of Descriptive Geometry, Engineering and Computer Graphics,
Cand. Tech. Sci., Assoc. Prof. Yablonsky Petr Nikolaevich

Approved by the Department of Descriptive Geometry, Engineering and Computer Graphics (Minutes No. 5 of 28.03.2023)

Approved by the Methodological Commission of the Faculty of Physics and Mathematics (Minutes No. 10 dated 27.06.2023)