

# OBJECT-ORIENTED PROGRAMMING. Part

## 1. Python Programming Fundamentals

### Work program of the credit module (educational component) (Syllabus)

#### Details of the discipline

<b>Level of higher education</b>	<i>First (bachelor's)</i>
<b>Field of knowledge</b>	<i>11 Mathematics and Statistics</i>
<b>Speciality</b>	<i>111 Mathematics</i>
<b>Educational program</b>	<i>Insurance and Financial Mathematics</i>
<b>Discipline status</b>	<i>Normative</i>
<b>Form of study</b>	<i>Full-time</i>
<b>Year of preparation, semester</b>	<i>1st year, autumn semester</i>
<b>Scope of the credit module</b>	<i>5 ECTS credits (150 hours), including: classroom – 2.4 ECTS credits (72 hours), including lectures – 0.6 ECTS credits (72 hours) and laboratory classes – 1.8 ECTS credits (54 hours); independent work – 2.6 ECTS credits (78 hours)</i>
<b>Semester control / control measures</b>	<i>3Alick / R&amp;D / MKR</i>
<b>Timetable</b>	<a href="http://roz.kpi.ua/">http://roz.kpi.ua/</a>
<b>Language of instruction</b>	<i>Ukrainian</i>
<b>Information about Course Leader / Instructors</b>	Lecturer: Ph.D., Assoc. Prof., Yablonsky Petr Nikolaevich, e-mail: ypn@ukr.NET, mobile: +380679977575; Laboratory: Ph.D., Assoc. Prof., Yablonsky Petro Mykolayovych, e-mail: ypn@ukr.net.
<b>Course Placement</b>	<a href="https://do.ipk.kpi.ua/course/view.php?id=3193">https://do.ipk.kpi.ua/course/view.php?id=3193</a>

#### The program of the discipline

##### 1. Description of the discipline, its purpose, subject of study and learning 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) "Object-oriented programming. Part 1. Fundamentals of Python Programming" 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 industrial problems.

**The purpose of the credit module (educational component) "Object-oriented programming. Part 1. Fundamentals of Python Programming"** is the formation of the following competencies in higher education applicants:

- skills in the use of information and communication technologies (GC 6);
- ability to learn and master modern knowledge (GC 7);
- ability to search, process and analyze information from various sources (GC 8);
- the ability to make informed decisions (GC 9);
- ability to carry out production or applied activities in an international environment (GC 18);
- ability to apply specialized programming languages and application packages (SC 9);
- ability to use computational tools for numerical and symbolic calculations (SC 10).

**The subject of the credit module (educational component) "Object-oriented programming. Part 1. Fundamentals of Python Programming"** is a concept of 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 "Object-Oriented Programming", applicants for higher education after mastering the credit module (educational component) "Object-Oriented Programming. Part 1. Fundamentals of Python Programming" should achieve certain programmatic learning outcomes, namely, have the skills to use specialized software tools of computer and applied mathematics and use Internet resources (PH 5).

## **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) "Object-oriented programming. Part 1. Fundamentals of Python Programming" is the initial stage for the study of the following credit modules and disciplines in which programming for mathematical calculations is used (studied), in particular "Software for Mathematical Calculations".

## **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. Working with files.

Topic 12. Routines. User functions.

Topic 13. Data structures.

Topic 14. Working with lists (arrays).

Topic 15. Ways to sort structured data types.

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](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](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](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](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](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](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.ipk.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](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, laboratory classes and independent work of students are provided. The materials necessary for mastering the educational component are posted on the Sikorsky distance learning platform (<https://do.ipk.kpi.ua/course/view.php?id=3193>), 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	<p><u>Lecture 1.</u> Introduction to the Computer Science and Programming Course Familiarity with Python and programming environments.</p> <p>Subject and objectives of the course. Number systems. History of Programming Languages. Compilation and interpretation. Installing Python. Python language and interpreter. Comparison Of Python with other programming languages.</p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>CPC:</u> To consolidate in practice the skills of converting numbers into different number systems and the skills of installing Python..</p>

2	<p><u>Lecture 2.</u> The alphabet of the Python programming language. Structure of the program. Code organization.</p> <p>Mathematical operators. Intelligent calculator. Relation operators. Priorities of operations. Structure of the program. Python Program Writing Style. Comments.</p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>SRS:</u> To consolidate in practice the skills of using mathematical operators and relation operators. Execution of the program. Varieties of execution.</p>
3	<p><u>Lecture 3.</u> Types of data in programming. Variables and literals. Arithmetic and logical expressions. Basic Mathematics.</p> <p>Types of data. Variables and literals. Arithmetic operations. Comparison operations. Priorities of operations. Basic logical operations.</p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>CPC:</u> Data Type Check. Bit operations.</p>
4	<p><u>Lecture 4.</u> Data input/output. Conditional statement. Branching.</p> <p>Formatted output. Data entry. Exceptions and exception handling (try-except). Logical operations. Binary operators. The if.</p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>CPC:</u> Random module. Random number generation. Condition check: True and False.</p>
5	<p><u>Lecture 5.</u> Loops. Lines and text.</p> <p>While and for statements. Interrupting and continuing loops. Splitting and merging strings. Working with substrings. Uppercase and lowercase.</p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>CPC:</u> Empty pass statement. Check for numbers and whitespace.</p>
6	<p><u>Lecture 6.</u> Working with files. Subroutines. User functions.</p> <p>Read and write files. Routines. User functions and their applications. Scope of variables. Passing arguments.</p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>CPC:</u> Functions for working with directories. Global and local variables.</p>
7	<p><u>Lecture 7.</u> Data structures. Working with lists (arrays). Ways to sort structured data types.</p>

	<p><i>Lists. Tuples. Dictionaries. Working with vectors and matrices. Filling the list with random data. Ways and methods of sorting. Working with slices.</i></p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>SRS:</u> <i>Plurals. Matrices: determinant calculation, inverse of matrices. Familiarize yourself with alternative sorting methods.</i></p>
8	<p><u>Lecture 8.</u> <i>The concept of object-oriented programming.</i></p> <p><i>Advantages and disadvantages of OOP. Classes and definitions of constructs.</i></p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>SRS:</u> <i>OOP in programming languages. Use of special methods.</i></p>
9	<p><u>Lecture 9.</u> <i>Popular modules and extensions. Data visualization.</i></p> <p><i>Some library modules. Data visualization. Numpy. Matplotlib</i></p> <p><u>Didactic aids:</u></p> <p><a href="https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp">https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp</a></p> <p><u>Recommended reading:</u> [1-5].</p> <p><u>SRC:</u> <i>Working with the Turtle module.</i></p>

### **L Practice Classes**

The main tasks of the series of laboratory 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	<p><b>Topic 1. Introduction to Python (introductory lab).</b></p> <p><i>Installing Python. Python interpreter. IDLE (Integrated Development and Learning Environment). Structure of the program.</i></p>	3
2	<p><b>Topic 2. Calculation of arithmetic expressions (laboratory work No 1).</b></p> <p><i>Code organization. Types of data. Calculation of arithmetic expressions. Priorities of operations.</i></p>	5
3	<p><b>Topic 3. Conditional statement. Branching (laboratory work No 2).</b></p> <p><i>Data entry. Exceptions and exception handling (try-except). The if-elif-else statement.</i></p>	6
4	<p><b>Topic 4. Cycles (laboratory work No 3).</b></p> <p><i>While and for statements. Replace one loop statement with another.</i></p>	6
5	<p><b>Modular test.</b></p>	2

6	<b>Topic 5. Working with one-dimensional lists (laboratory work No 4).</b> <i>Fill, sort, add, and delete list items.</i>	8
7	<b>Topic 5. Work with multidimensional lists (laboratory work No 5).</b> <i>Populating the list with random numbers. Working with slices. Use of user functions.</i>	8
8	<b>Topic 6. Rows. Working with files (laboratory work No 6).</b> <i>Working with strings and substrings. Writing information to a file. Read data from a file.</i>	8
9	<b>Topic 7. Data visualization (laboratory work No 7).</b> <i>Using numpy and matplotlib modules. Building surfaces, graphs, etc.</i>	8
<b>JUST:</b>		<b>54</b>

### **Homework**

The purpose of the home test is to master the methods of constructing images using the capabilities of the Turtle module. Homework is done during the semester. Requirements for the homework are posted on the Sikorsky distance learning platform: <https://do.ipk.kpi.ua/course/view.php?id=3193>. Doing homework encourages students to:

- independently find, analyze, systematize, understand, interpret and use information from various sources to solve scientific and applied problems;
- creatively and reasonably approach the solution of tasks;
- use practical skills in implementing algorithms in Python in scientific and professional activities.

### **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	<u>Topic 1.</u> Introduction to the Computer Science and Programming Course <i>SRS: Elaboration and consolidation of the materials of the lecture and laboratory lesson, in particular, to consolidate in practice the skills of converting numbers into different number systems.</i>	2
2	<u>Topic 2.</u> Familiarity with Python and programming environments. <i>CPC: Working out and consolidating the materials of the lecture and laboratory lesson, in particular, to consolidate in practice the skills of installing Python.</i>	2
3	<u>Topic 3.</u> The alphabet of the Python programming language.	3

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



16	<u>Topic 16.</u> <i>The concept of object-oriented programming.</i> <i>SRS: Elaboration and consolidation of lecture and laboratory materials, in particular OOP in programming languages and the use of special methods.</i>	4
17	<u>Topic 17.</u> <i>Popular modules and extensions. Data visualization.</i> <i>CPC: Elaboration and consolidation of lecture and laboratory materials, in particular, to get acquainted with the Turtle module.</i>	3
18	<i>Doing homework.</i>	15
19	<i>Preparation for the test.</i>	6
<b>JUST:</b>		<b>78</b>

## Policy & Control

### 7. Academic discipline policy (educational component)

Successful mastering of the credit module "Object-Oriented Programming. Part 1. Fundamentals of Python Programming" requires the following from the applicant:

- ✓ 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 rating of a higher education applicant is calculated on a 100-point scale. Rating consists of the points that the higher education applicant receives (during the semester) for:

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

8.2. Criteria for awarding points:

8.2.1. Performance and defense of laboratory work is evaluated from 10 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 – 10 points;
- ✓ the work was completed at a high level (at least 90% of the requirements were met), submitted on time and a full answer was provided during the defense with minor inaccuracies – 9 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 – 7-8 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 – 6 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, 2 penalty points are awarded (but in total not more than 7 penalty points for all laboratory work).

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

8.2.2. Homework is evaluated out of 15 points according to the following criteria:

- ✓ the task is completely solved and the work is impeccably designed – 14-15 points;

- ✓ at least 75% of the task is solved and the work is properly formatted (minor inaccuracies are possible) – 11-13 points;
- ✓ at least 60% of the task has been solved, the work is designed with shortcomings – 9-10 points;
- ✓ The task is solved by less than 60%, the work is not properly designed – 0 points.

Passed homework is a prerequisite for admission to the test.

8.2.3. 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.

The passed modular test is a prerequisite for admission to the test.

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 test is the enrollment of all laboratory work, homework and modular control work.

8.5. Applicants who have fulfilled all the conditions for admission to the test and have a rating of 60 or more points receive a grade corresponding to the scored rating (see Figure 1). Tables of correspondence of rating points to grades on the university scale :) without additional tests.

8.6. With applicants who have fulfilled all the conditions for admission to the test and have a rating score of less than 60 points, as well as with those applicants who want to improve their rating, the teacher conducts semester control in the form of a test work.

8.6.1. In this case, the starting rating from a 100-point scale is reduced to a 70-point multiplication by a coefficient of 0.7:

Laboratory work	Homework	Modular test	Maximum Points	Coefficient 0.7
70	15	15	100	$100 \times 0.7 = 70$

8.6.2. Applicants for higher education perform individual tests in the control work, which consists of two theoretical questions and one practical task. Each question (task) is evaluated out of 10 points according to the following criteria:

- ✓ complete answer, at least 95% of the required information (complete, error-free solution of the problem) – 10 points;
- ✓ complete answer (with some minor inaccuracies), not less than 90% of the required information (complete solution of the problem with some minor inaccuracies) – 9 points;
- ✓ sufficiently complete answer, at least 75% of the required information or minor inaccuracies (complete solution of the problem with minor inaccuracies) – 7-8 points;
- ✓ incomplete answer, which contains at least 60% of the required information, and some errors (the task was completed with certain shortcomings) – 6 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.

8.6.3. The sum of the starting points determined according to the algorithm described in clause 8.6.1 and the points received for individual credits in the control work is converted to the grades according to the Tables and the correspondence of the rating points to the grades on the university scale:

Table of correspondence of rating points to grades on the university scale:

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

8.6.4. The option of a "hard" rating system of evaluation is applied - the previous rating of the applicant is canceled and he receives a grade taking into account the results of the individual test work.

### 9. Additional information on the discipline (educational component)

- A list of theoretical questions and examples of practical tasks that are put to the test in the work is posted on the distance learning platform "Sikorsky" (<https://do.ipk.kpi.ua/course/view.php?id=3193>).

### 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 of 27.06.2023)