



INFORMATION TECHNOLOGY  
Working program of the academic discipline (Syllabus)

Details of the academic discipline

Level of higher education	<i>First (educational)</i>
Branch of knowledge	16 – Chemical engineering and bioengineering
Specialty	162 Biotechnology and bioengineering
Educational program	<i>Biotechnology</i>
Discipline status	Basic
Form of education	Full time
Year of training, semester	2nd year, autumn semester
Scope of the discipline	The total amount is 120 hours.
Semester control/ control measures	Final test / MCW/ HCW
Lessons schedule	<i>4 credits (120 hours), incl. lectures – 36 hours, practical – 144 hours, laboratory – 0 hours, SRS – 48 hours.</i>
Language of teaching	Ukrainian
Information about the head of the course / teachers	Lecturer and practical works: doctor of technical sciences, prof. Gorobets Svitlana Vasylivna, <a href="mailto:gorobetsv@gmail.com">gorobetsv@gmail.com</a> , telegram chat of discipline
Placement of the course	On the distance learning platform Google Class at the link <a href="https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm">https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm</a> course code <b>3ku5mbm</b> , permanent link to Google Meet <a href="https://meet.google.com/uom-mkaz-tem">https://meet.google.com/uom-mkaz-tem</a>

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

It is no longer possible to imagine the modern world without information technologies, which have entered all industries. A professional biotechnologist should currently know and be able to use modern programming languages. Leading such languages are Python and Biopython, a collection of non-commercial, open-source computational tools for biology and bioinformatics created by the International Python Developers Association is a set of Python modules that provide functions for working with DNA, RNA, and protein sequences, such as reverse complementation of the DNA chain, search for motifs in protein sequences, etc. It provides a variety of parsers to read all major genetic databases such as GenBank, SwissPort, PDB, and shells/interfaces to run other popular bioinformatics programs/tools such as NCBI BLASTN, Entrez, etc. in a Python environment. The course introduces students to the basics of Python programming and a number of Python programming language packages for the analysis of experimental data in biotechnology: the Matplotlib package for plotting graphs and charts, Python packages for regression analysis problems, the Opencv-python package for image analysis, etc. This course introduces future biotechnologists to the basics of bioinformatics, which is a leader in the development of medicinal products, introduces software tools used to analyze and compare genetic data, modern methods of genomics, proteomics, pharmacogenomics and teaches to gain new knowledge using bioinformatics methods.

The purpose of the educational discipline is to form students' abilities to:

- use of information and communication technologies;

- use knowledge of mathematics and physics to the extent necessary to achieve other results of the educational program;
- use modern automated systems for managing the production of biotechnological products for various purposes, their technical, algorithmic, informational and software support for solving professional tasks;
- learn and master modern knowledge; comprehensively analyze biological and biotechnological processes at the molecular and cellular levels; to use knowledge about the ways of biosynthesis of practically valuable metabolites to improve the biotechnologies of their production

The main tasks of the credit module.

According to the requirements of the program of the academic discipline, after mastering the credit module, students must demonstrate the following learning outcomes:

competences:

- basics of Python and Biopython programming language
- to know basic bioinformatics algorithms for alignment of nucleotide and amino acid sequences;
- know the principles of operation of software packages Matplotlib, Opencv-python, Sklearn.
- to program at a basic level in Python and Biopython;
- be able to implement basic bioinformatics alignment algorithms in Python;
- be able to use the Matplotlib Python package for constructing graphs and charts, analyzing experimental data;
- O PENCV-PYTHON package to analyze images obtained by methods of optical microscopy, scanning probe microscopy, scanning electron microscopy, etc.
- be able to use the Sklearn Python package for regression analysis problems;
- to be able to apply modern methods of bioinformatics to solve practical problems related to the research and design of biotechnological processes.

Program learning outcomes:

- To be able to apply modern mathematical methods for solving practical problems related to the research and design of biotechnological processes. Use knowledge of physics to analyze biotechnological processes.
- Based on knowledge of the laws of mechanical, hydromechanical, heat and mass exchange processes and the main design features, be able to choose the appropriate equipment in the process of designing the production of biotechnological products of various purposes to ensure their maximum efficiency.

## 2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

A place in the structural and logical scheme of education is provided by disciplines such as the general technical discipline "Higher Mathematics", as well as a basic level of English language proficiency not lower than A2. In the structural and logical plane of the bachelor's training program in biotechnology, the discipline is based on previously studied disciplines at school. The study of the discipline affects the further path of higher education seekers. On the basis of the discipline "Information technologies", the following disciplines "Methods of analysis in biotechnology" and the block of disciplines related to the design of biotechnological productions are studied in the future.

## 3. Content of the academic discipline

### Topic 1. Interpreted, object-oriented programming language Python.

Lecture 1. Different programming languages and their areas of application. Object-oriented and system approaches. Comparison of object-oriented and procedural (traditional) approaches. Features of the Python language. Integrated development environments (IS): Pycharm Community, IDLE, Spyder.

Lecture 2. Basic concepts of the Python language: basic syntax, data types, mathematical operations, logical operators, conditional operators, loops.

Lecture 3. Complex data structures of the Python language. Work with strings, lists, dictionaries, tuples, sets (sets).

Lecture 4. Working with files in the Python language.

Lecture 5. Fundamentals of the numpy package of the Python programming language.

Lecture 6. Functions in Python.

Lecture 7. Classes and projects in Python.

### **Topic 2. Using the Python programming language to analyze experimental data in biotechnology.**

Lecture 8. Matplotlib package for plotting graphs and charts.

Lecture 9 . Python packages for regression analysis problems. Linear regression of experimental data.

Lecture 10. Opencv-python package for analysis of images obtained by methods of optical microscopy, scanning probe microscopy, scanning electron microscopy. Reading images of biological objects from files, writing them to a file.

### **Topic 3. Basics of bioinformatics. Basic algorithms of bioinformatics in Python.**

Lecture 11. Formation of bioinformatics as an independent science. The subject and problems of bioinformatics.

Lecture 12. Basics of the Biopython package for working with databases. Pairwise and multiple sequence alignment in Biopython. Organization of access to the NCBI Entrez database in the Biopython package.

Lecture 13. Dynamic programming methods. The algorithm of global alignment of two sequences and an example of the implementation of the algorithm of global alignment of two sequences in the Python programming language .

Lecture 14. Modifications of the genetic sequence alignment algorithm. Algorithms for local, pseudo-global and fast alignment of two sequences.

Lecture 15. Hashing. Using hash tables in databases to find matches on a list of words. Algorithm for constructing a dot matrix.

Lecture 16. Peculiarities of protein sequence alignment algorithms. PAM and BLOSUM matrices of amino acid substitutions, PSSM matrices.

Lecture 17. System of bonuses and penalties for alignment of sequences. Algorithm of the general penalty function.

Lecture 18. Bioinformatics as a tool for obtaining new knowledge.

*A list of sections and topics of the entire discipline is provided .*

### **Educational materials and resources**

Basic literature:

1. Gorobets O.Yu. The python language: for engineering and scientific problems [Electronic resource]: a textbook for bachelor's degree holders in the specialty 162 "Biotechnology and Bioengineering", master's degree in the specialty 162 "Biotechnology and Bioengineering", bachelor's degree in the specialty 104 "Physics and Astronomy", and master's degree in the specialty 104 "Physics and Astronomy" / O. Yu. Gorobets, S. V. Gorobets, K. Yu. Khakhno; KPI named after Igor Sikorsky. - 1st edition. - Electron. text. data (1 file , 13.7 auto. sheets) - Kyiv: KPI named after Igor Sikorskyi, 2024. – 277 p.
2. S. V. Gorobets Basics of bioinformatics [Electronic resource]: a textbook for students of the 6.051401 "Industrial Biotechnology" course of the Faculty of Biotechnology and Biotechnology / S. V. Gorobets, O. Yu. Gorobets, T. A. Khomenko; NTUU "KPI". - Electronic text data (1 file: 2.72 MB). - Kyiv: NTUU "KPI", 2010.

<https://ela.kpi.ua/bitstream/123456789/774/1/10-11-146.pdf>

3. Gorobets S. V. Bioinformatics. Practicum [Electronic resource]: training. manual for students specialty 162 "Biotechnologies and bioengineering" / S. V. Gorobets, O. Yu. Gorobets, I.V. Demyanenko; KPI named after Igor Sikorsky. – Electronic text data (1 file 5.49 MB). – Kyiv: KPI named after Igor Sikorskyi, 2020. – 86 p. <https://ela.kpi.ua/handle/123456789/38813>
4. Gorobets S. V. Bioinformatics databases [Electronic resource]: academician. manual for students specialty 162 "Biotechnologies and bioengineering" / S. V. Gorobets, O. Yu. Gorobets, M. O. Bulaevska; KPI named after Igor Sikorsky. – Electronic text data (1 file: 3.86 MB). – Kyiv: KPI named after Igor Sikorskyi, 2020. – 117 p. <https://ela.kpi.ua/handle/123456789/36457>
5. Zamuruyeva O. V., Krymus A. S., Olkhova N. V. Object-oriented programming in Python: a course of lectures. Lutsk: Vezha-Druk, 2018. – 64 p.
6. Gorobets S. V. Information technologies. Home control work [Electronic resource]: teaching. manual for bachelor's degree holders in education. program "Biotechnology" special. 162 Biotechnology and bioengineering / KPI named after Igor Sikorsky - Electron. text. data (1 file, 0.5 auto. sheet). – Kyiv: KPI named after Igor Sikorskyi, 2024. – 18 p.

Additional:

1. Kostyuchenko A.O. Fundamentals of Python Programming: A Study Guide. Ch.: FOP Balykina S.M., 2020. 180 p.
2. Yakovenko A.V. Fundamentals of programming. Python. Part 1 [Electronic resource]: textbook for students. specialty 122 "Computer science", specialization "Information technologies in biology and medicine" / A. V. Yakovenko; KPI named after Igor Sikorsky. – Electronic text data (1 file: 1.59 MB). – Kyiv: KPI named after Igor Sikorskyi, 2018. – 195 p.
3. Bioinformatics . - editor: Nakaya Helder I., PhD, editor Information Brisbane (AU): 2021. – 188 p.
4. Lesk A.M. Introduction to Bioinformatics. - Oxford. – 2002. – 255 p.

### Information resources

1. <https://www.ncbi.nlm.nih.gov/>
2. <https://www.python.org/>
3. <https://matplotlib.org/>
4. <https://scikit-learn.org/stable/>
5. [https://docs.opencv.org/4.5.2/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.5.2/d6/d00/tutorial_py_root.html)
6. <http://www.biopython.org>

### Educational content

#### 4. Methods of mastering an educational discipline (educational component)

1	2
	<i>Topic 1. Interpreted, integrated, object-oriented programming language Python.</i>
1	<p><b>Lecture 1.</b> Different programming languages and their areas of application. Object-oriented and system approaches. Comparison of object-oriented and procedural (traditional) approaches. Features of the Python language. Integrated development environments (IS): Pycharm Community, IDLE, Spyder.</p> <p>Literature: basic [1, 5], additional [1-2], information sources [1-6]</p>
2	<p><b>Lecture 2.</b> Basic concepts of the Python language: basic syntax, data types, mathematical operations, logical operators, conditional operators, loops.</p> <p>Literature: basic [1, 5], additional [1-2], information sources [2]</p>

3	<b>Lecture 3.</b> Complex data structures of the Python language. Work with strings, lists, dictionaries, tuples, sets (sets). Literature: basic [1, 5], additional [1-2], information sources [2]
4	<b>Lecture 4.</b> Working with files in the Python language . Literature: basic [1, 5], additional [1-2] , information sources [2]
5.	<b>Lecture 5.</b> Fundamentals of the numpy package of the Python programming language. Literature: basic [1, 5, 6], additional [1-2], information sources [2]
6.	<b>Lecture 6.</b> Functions in Python. Types of functions, function arguments, d documentation of functions, etc. Literature: basic [1, 5], additional [1-2], information sources [2]
7.	<b>Lecture 7.</b> Classes and projects in Python. Literature: basic [1, 5], additional [1-2], information sources [2]
<i>Topic 2. Using the Python programming language to analyze experimental data in biotechnology.</i>	
14	<b>Lecture 8.</b> Matplotlib package for plotting graphs and charts. Literature: basic [1, 5], information sources [3]
15	<b>Lecture 9.</b> Python packages for regression analysis problems. Linear regression of experimental data. Literature: basic [1, 5], additional [ 3 ], information sources [2-5]
16	<b>Lecture 10.</b> O PENCV-PYTHON package for the analysis of images obtained by methods of optical microscopy, scanning probe microscopy, scanning electron microscopy. Reading images of bioobjects from files, writing them to a file, etc. Literature: basic [1, 5], information sources [2,5]
<i>Topic 3. Basics of bioinformatics. Basic algorithms of bioinformatics in Python.</i>	
8	<b>Lecture 11.</b> Formation of bioinformatics as an independent science. The subject and problems of bioinformatics. Literature: basic [1-4].
9	<b>Lecture 12.</b> Basics of the Biopython package for working with databases. Pairwise and multiple sequence alignment in Biopython. Organization of access to the NCBI Entrez database in the Biopython package. Literature: basic [1-4], information sources [1, 6]
10	<b>Lecture 13.</b> Dynamic programming methods. The algorithm of global alignment of two sequences and an example of the implementation of the algorithm of global alignment of two sequences in the Python programming language . Literature: basic [2-4], additional [ 4 ] , information sources [1]
11	<b>Lecture 14.</b> Modifications of the genetic sequence alignment algorithm. Algorithms for local, pseudo-global and fast alignment of two sequences. Literature: basic [2-4], information sources [1]
12	<b>Lecture 15.</b> Hashing. Using hash tables in databases to find matches on a list of words. Algorithm for constructing a dot matrix. Literature: basic [2-4], information sources [1]
13	<b>Lecture 16.</b> Peculiarities of protein sequence alignment algorithms. PAM and BLOSUM matrices of amino acid substitutions, PSSM matrices. Literature: basic [2-4], information sources [1]
17	<b>Lecture 17.</b> System of bonuses and penalties for alignment of sequences. Algorithm of the general penalty function. Literature: basic [2-4], information sources [1]
18	<b>Lecture 18.</b> Bioinformatics as a tool for obtaining new knowledge. literature : basic [2-4] , additional [ 3 ] , information sources [1]

## Practical training

The main tasks of the cycle of practical classes:

- development of elementary programs in Python
- work with software packages used in biotechnology;
- practical work with modern biotechnological databases and their main algorithms.

1.	<b><i>Practical work 1</i></b> Acquaintance and the basics of working in the databases of the National Center for Biotechnology Information ( NCBI). Work with PMC and PubMed databases. Literature: basic [1-4], information resources [1].
2.	<b><i>Practical work 2.</i></b> Setting up the development environment and running the program on python. Data input and output. Mathematical operations. Conditional operators, loops. Literature: basic [1,5], additional [1,2] information sources [2]
3.	<b><i>Practical work 3.</i></b> Work with data types of the Python language: strings, lists, dictionaries, tuples, sets (sets). Literature: basic [1,5], additional [1,2] information sources [2]
4.	<b><i>Practical work 4.</i></b> Working with files in Python. Reading sequences from NCBI databases and writing them to files. Literature: basic [1,5], additional [1,2] information sources [2]
5.	<b><i>Practical work 5</i></b> Basics of the numpy package of the Python programming language. Functions of the numpy package , working with arrays. Literature: basic [1,5], additional [1,2] information sources [2]
6.	<b><i>Practical work 6</i></b> Functions in Python, types of functions, local and global variables. Literature: basic [1,5], additional [1,2] information sources [2]
7.	<b><i>Practical work 7</i></b> Classes and projects in Python. Literature: basic [1,5], additional [1,2] information sources [2]
8.	<b><i>Practical work 8</i></b> Working with Matplotlib packages for plotting graphs and charts. Literature: basic [1], information resources [3].
9.	<b><i>Practical work 9</i></b> Python programming language packages for regression analysis problems. Linear regression of experimental data. Literature: basic [1], additional [ 3 ] , information resources [2-5].
10.	<b><i>Practical work 10</i></b> O PENCV-PYTHON package for the analysis of images obtained by methods of optical microscopy, scanning probe microscopy, scanning electron microscopy. Reading images of biological objects from files, writing them to a file. Literature: basic [1], information sources [2-5]
11.	<b><i>Practical work 11.</i></b> Comparison of sequences using the BLAST program. Programs of the BLAST family. Literature: basic [2-4], additional [ 4 ] , information resources [1].
12.	<b><i>Practical work 12.</i></b>

	Basics of the Biopython package for working with databases. Organization of access to the NCBI Entrez database in the Biopython package. Literature: basic [1], information sources [6]
13.	<b>Practical work 13</b> Global alignment algorithm and its implementation in Python. Literature: basic [2-4], information sources [2]
14.	<b>Practical work 14</b> Algorithm for local, pseudo-global and fast alignment of two sequences and implementation of the local alignment algorithm in Python. Literature: basic [2-4], additional [1,2] information sources [2]
15.	<b>Practical work 15</b> Hashing algorithm. Construction of hash tables. Construction of dot matrices in NCBI . Literature: basic [2-4], information resources [1].
16.	<b>Practical work 16.</b> Peculiarities of protein sequence alignment algorithms. Calculation of the matrix of amino acid substitutions PAM and matrix PSSM . Literature: basic [2-4] , information sources [2]
17.	<b>Practical work 17</b> <b>MKR</b>
18.	<b>Practical work 18</b> <b>Test</b>

#### Independent work of student

The student's independent work in the discipline includes preparation for classroom classes (48 hours), modular test (4 hours), writing the DKR (10 hours), preparation for the assessment (6 hours).

#### Policy and control

##### 5. Policy of academic discipline (educational component)

The study of the discipline "Information technologies" takes place in lectures and practical classes. Clarity of training sessions is provided by the use of a significant amount of illustrative material (diagrams, tables, slides). During the teaching of this discipline, the teacher conducts a survey of students in order to determine the level of assimilation of the material taught by them, the activity of students is important. Practical classes are held using computer equipment and appropriate software.

The regulations on the rating system of evaluation in the discipline "Information technologies" for this work training program are presented in clause 9 of the syllabus.

*The system of requirements that the teacher sets for the student/graduate student is indicated:*

- *rules for attending classes (both lectures and practical/laboratory);*  
Attendance at lectures, practical classes and laboratory works, as well as absence from them, are not evaluated. However, students are recommended to attend classes, as they teach theoretical material and develop skills necessary for the formation of competencies defined by the education standard. The evaluation system is focused on receiving points for the student's activity, as well as the performance of tasks that can develop practical skills and abilities. For objective reasons (for example, illness, employment, international internship, etc.), training can take place online upon agreement with the course leader.
- *rules of behavior in classes (activity, preparation of short reports or texts, turning off phones, using means of communication to search for information on the teacher's Google Drive or on the Internet, etc.);*



In classroom classes, the student must respect the teacher and the discipline he is listening to; Follow basic rules and norms of behavior; During the class, it is forbidden to use mobile phones, except for emergencies. Standards of ethical behavior of students and employees are defined in Chapter 2 of the Code of Honor of the National Technical University of Ukraine "Ihor Sikorskyi Kyiv Polytechnic Institute". More details: <https://kpi.ua/code>.

- *rules for assigning incentive and penalty points;*  
for active work in lectures and practical classes, a student can receive up to 10 points in total during the semester.
  - *policy of deadlines and rescheduling;*  
The deadline for submitting each type of work is discussed in class when the assignment is issued and depends on the type of work.
  - *academic integrity policy;*  
defined in section 3 of the Code of Honor of the National Technical University of Ukraine "Ihor Sikorskyi Kyiv Polytechnic Institute". More details: <https://kpi.ua/code>. The use of additional sources of information during knowledge assessment is prohibited (including mobile devices). Mobile devices are allowed to be used only during online testing and calculations.
  - *other requirements that do not contradict the legislation of Ukraine and normative documents of the University.*
  - The student must register on the distance learning platform Google Workspace for Education Fundamentals (formerly G Suite For Education ) on the @LLL.kpi.ua domain and join the Google Class "Information Technologies" via the link <https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm>. For this, the student must first get an account in Google Workspace for Education Fundamentals on the @LLL.kpi.ua domain. To get an account in Google Workspace for Education Fundamentals on the @LLL.kpi.ua domain, you need to fill out the Google Form: [https://sikorsky-distance.kpi.ua/reg\\_gsuite/](https://sikorsky-distance.kpi.ua/reg_gsuite/). After registration and moderation of the student's application, the administrator will send the student a password and login to the account, with which the student will be able to use all available tools and services of Google Workspace for Education Fundamentals. Google Workspace for Education Fundamentals is a suite of specialized cloud software, tools for collaboration and distance learning from Google. The main component of the package is the Google Class learning management system, which allows the teacher to create classes, evaluate assignments, provide feedback to students, publish announcements, and distribute educational materials. The teacher can see who has completed the task and who is still working on it, as well as read students' questions and comments. To join the "Information Technologies" educational course, the student needs to go to Google Classroom via the link <https://classroom.google.com>, click the "+" icon in the upper right corner of the browser, select "Join the class" and enter **the course code 3ku5mbm**. Accounts of students who joined Google Class from an account not on the @LLL.kpi.ua domain will be removed from the Google Class "Information Technologies" training course because automatic import of test grades is possible only from the Information Technologies account on the @LLL domain. kpi.ua The Google Class system automatically sends scores for each type of control to each student by e-mail. Therefore, in order to get acquainted with the points for each individual type of control, the student needs to change his email settings so that these emails do not end up in spam. The student must upload all completed tasks for teacher verification through the Google Class system (results of tasks sent through the Telegram channel will not be verified);
  - Correspondence with students on organizational matters will be carried out through telegram chat of discipline
6. Types of control and rating system for evaluating learning outcomes (RSO)

**Current control** : points for work in practical classes, namely for answering questions and writing code in the language python (total 10 points per semester); testing students' knowledge of lecture material and practical work, 14 tests of 5 points each per semester (70 points in total); completion of homework test (10 points) and modular test work (10 points). The total amount of points for the semester's work is 100 points. More detailed information on current control and evaluation criteria is given in RSO on discipline (Appendix 1).



**Calendar control** : twice per semester. More detailed information on the conduct and assessment is given in the RSO on the discipline.

**Semester control** : credit. The total amount of points in the assessment is 100 points. More detailed information on the conduct and assessment is given in the RSO on the discipline.

**Conditions for admission to the semester control** : a semester rating of not less than 50 points, work in practical classes, writing an express test on MKR, writing an express test DKR and all express tests on practical works. If the semester rating is higher than 60 points, provided that all types of assignments are passed, the student can receive credit automatically.

The verification of the implementation of the SRS will be carried out by evaluating the test results on the distance learning platform Google Workspace for Education Fundamentals on the domain @LLL.kpi.ua in the Google Class system using the link:

<https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm>.

Each of the tests can be passed only once. The tests include 10 questions of varying difficulty, the points for each question depend on its difficulty. Rating (weighted) points of classes and rating evaluations for all types of control in Google Class are equal to the corresponding points in the syllabus with a factor of 10 for the convenience of calculating points (so as not to use fractional numbers). Accordingly, before each certification, as well as at the end of the semester, all the points scored by the student in Google Class will be divided by 10 and entered into the AIS system "Electronic Campus" of Igor Sikorsky KPI.

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

### Appendix 1

#### The system of rating (weighted) points of classes and rating evaluations by types of control for the year

No	type of control	Mark	Number	Total points
1.	Work in practical classes	1	10	10
2.	Testing students' knowledge of lecture material and practical work	5	14	70
3.	Modular control work			
	-weight score $r_k$	10	1	10
4.	DKR			
	-weight score $r_k$	10	1	10
5.				100

#### Calculation of the rating scale (R).

The sum of the weighted points of control measures during the semester is:

$$R = 10+70+10+10 = 100 \text{ points.}$$

The rating scale for the discipline is  $R = 100$  points.

A necessary condition for receiving automatic credit is the inclusion of all proposals submitted for implementation discussion for a positive evaluation of the modular control work and a total rating of more than 60 points. In order to improve the assessment, credit work is carried out. At the same time, the previous rating is canceled.

**Calendar control** : twice per semester.

Frontier (scheduled attestations). The student must score points: 1st certification - "credited" - 20 points (40 - maximum), 2nd certification - 40 points (80 - maximum).

The final assessment of the quality of knowledge in the discipline is determined according to a traditional 6-level scale based on individual current assessments according to the following scale:

Scores	Rating
100-95	Perfectly
94-85	Very good

84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily

**Semester control** : credit. The total amount of credit points is 100 points. Conditions for admission to semester control: semester rating of at least 50 points, writing MKR and DKR, performance of practical work.

The assessment consists of 10 questions, 1 question is valued at 10 points.

flawless work	- 9-10 points
there are certain shortcomings in the performance of work	- 7-8 points
there are significant shortcomings in the performance of work	- 2-6 points
incorrect answers	- 0-1 point.

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

<i>Scores</i>	<i>Rating</i>
100-95	Perfectly
94-85	Very good
84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily

## Appendix 2

### Approximate list of questions for the DKR from the "Information Technologies" course

1. Built-in Python functions
2. Built-in Python data types
3. The Python numpy package
4. Python package matplotlib
5. Working with arrays of the numpy package
6. Analysis of experimental data using the numpy package:
  - the average value of the value;
  - root mean square error of the mean;
  - mean square deviation of the value;
  - value dispersion;
  - the maximum and minimum value of a quantity, etc.
7. Creating functions in Python.
8. Construction of diagrams based on experimental data.
9. Layout of the axes of the diagram.
10. Using a dictionary when designing a diagram.
11. Using the numpy package to prepare data for data visualization.

## Appendix 3

### List of questions for the modular control work from the "Information Technologies" course

1. Development history and main programming languages.
2. Object-oriented approach to programming.

3. A procedural approach to programming.
4. Key benefits of Python. Problems for which Python is used.
5. Data types in Python.
6. Boolean operations in Python.
7. Mathematical operators and functions in Python.
8. Functions in Python.
9. Classes in Python.
10. Basic Python packages.
11. The main functions of working with lists.
12. The main functions of working with dictionaries.
13. The main functions of working with tuples.
14. Basic functions of working with sets (sets).
15. Definition of the concept of bioinformatics.
16. Subject of bioinformatics.
17. Goals of bioinformatics.
18. The purpose of the "Human Genome" program.
19. When did the Human Genome program start and its main achievements?
20. When is the local alignment algorithm used to compare two sequences?
21. When is the global alignment algorithm used to compare two sequences?
22. When is the pseudo-global alignment algorithm used to compare two sequences?
23. When is the fast alignment algorithm used to compare two sequences?
24. What does each cell of the prefix weight matrix  $a(i,j)$  mean?
25. The main idea of dynamic programming algorithms?
26. Databases (general definition) and bioinformatics databases. In which areas of molecular biology is BIBD used.
27. The history of the emergence of BIBD. The most famous organizations are curators of BIBD and data banks created by them.
28. Classification of BIBD. Give examples.
29. What methods are used to align 2 sequences, when and by whom were these methods developed?
30. When is the general penalty function algorithm used to compare two sequences?
31. Algorithm of the general penalty function.
32. Initialization of the array of prefix weights  $a(i,j)$ ;  $b(i,j)$ ;  $c(i,j)$  when using the general penalty function.
33. Recursive formula for global alignment of two sequences.
34. A recurrence formula for the local alignment of two sequences.
35. Recurrent formula for pseudo-global alignment of two sequences.
36. A recursive formula for the general penalty function.
37. Initialize column zero and row zero to globally align the two sequences.
38. Initialize column zero and row zero to locally align the two sequences.
39. Initialize column zero and row zero for pseudo-global alignment of two sequences.
40. Initialize zero column and zero term for the general penalty function.
41. Hash algorithm, in which applications is it used?
42. Form a hash table to examine the sequence (word length 4).
43. The main functions of the Matplotlib package for plotting graphs.
44. The main functions of the Opencv-python package for working with images of biological objects of graphs.
45. Basic tools for working with genetic sequences in the Biopython package.
46. Functions for linear regression in the Sklearn package.

1. Development history and main programming languages.
2. Object-oriented approach to programming.
3. A procedural approach to programming.
4. Key benefits of Python. Problems for which Python is used.
5. Data types in Python.
6. Boolean operations in Python.
7. Mathematical operators and functions in Python.
8. Functions in Python.
9. Classes in Python.
10. Basic Python packages.
11. The main functions of working with lists.
12. The main functions of working with dictionaries.
13. The main functions of working with tuples.
14. Basic functions of working with sets (sets).
15. Definition of the concept of bioinformatics.
16. Subject of bioinformatics.
17. Goals of bioinformatics.
18. The purpose of the "Human Genome" program.
19. When did the Human Genome program start and its main achievements?
20. When is the local alignment algorithm used to compare two sequences?
21. When is the global alignment algorithm used to compare two sequences?
22. When is the pseudo-global alignment algorithm used to compare two sequences?
23. When is the fast alignment algorithm used to compare two sequences?
24. What does each cell of the prefix weight matrix  $a(ij)$  mean?
25. The main idea of dynamic programming algorithms?
26. Databases (general definition) and bioinformatics databases. In which areas of molecular biology is BIBD used.
27. The history of the emergence of BIBD. The most famous organizations are curators of BIBD and data banks created by them.
28. Classification of BIBD. Give examples.
29. What methods are used to align 2 sequences, when and by whom were these methods developed?
30. When is the general penalty function algorithm used to compare two sequences?
31. Algorithm of the general penalty function.
32. Initialization of the array of prefix weights  $a(i,j)$ ;  $b(i,j)$ ;  $c(i,j)$  when using the general penalty function.
33. Recursive formula for global alignment of two sequences.
34. A recurrence formula for the local alignment of two sequences.
35. Recurrent formula for pseudo-global alignment of two sequences.
36. A recursive formula for the general penalty function.
37. Initialize column zero and row zero to globally align the two sequences.
38. Initialize column zero and row zero to locally align the two sequences.
39. Initialize column zero and row zero for pseudo-global alignment of two sequences.
40. Initialize zero column and zero term for the general penalty function.
41. Hash algorithm, in which applications is it used?
42. Form a hash table to examine the sequence (word length 4).
43. The main functions of the Matplotlib package for plotting graphs.
44. The main functions of the Opencv-python package for working with images of biological objects of graphs.
45. Basic tools for working with genetic sequences in the Biopython package.
46. Functions for linear regression in the Sklearn package.

Working program of the academic discipline (syllabus):

composed by Doctor of Technical Sciences, Prof. Gorobets S.V.

Approved by the Department of Bioenergy, Bioinformatics and Ecobiotechnology (Protocol No. 14 dated 27.05.2024 )

Agreed Methodical commission of the faculty (protocol No. 19 dated 28.06.24).