

**Federal State Autonomous Educational Institution of Higher Education "Moscow  
Institute of Physics and Technology  
(National Research University)"**

**APPROVED**

**Head of the Phystech School of  
Applied Mathematics and  
Informatics**

**A.M. Raygorodskiy**

**Work program of the course (training module)**

**course:** Python/Язык программирования Python  
**major:** Information Science and Computer Engineering  
**specialization:** Computer Science/Информатика  
Phystech School of Applied Mathematics and Informatics  
Chair of Algorithms and Programming Technologies  
**term:** 2  
**qualification:** Bachelor

Semester, form of interim assessment: 3 (fall) - Grading test

Academic hours: 30 АН in total, including:

lectures: 0 АН.

seminars: 0 АН.

laboratory practical: 30 АН.

Independent work: 105 АН.

In total: 135 АН, credits in total: 3

Author of the program: V.V. Yakovlev, candidate of physics and mathematical sciences

The program was discussed at the Chair of Algorithms and Programming Technologies 21.05.2020

## Annotation

The course covers Python programming language topics related to programming language itself and widely used libraries and frameworks.

During the course students will study Jupyter Notebook programming environment and explore Python language featured. Additional libraries to be learned are related to linear algebra (NumPy), generic data representation (.csv and text files) and data visualization (matplotlib).

### 1. Study objective

#### Purpose of the course

To introduce students to the Python programming language and prepare them for practical activities in the positions of analysts and software programmers.

#### Tasks of the course

- Build knowledge about the correct use of the Python language in development;
- build knowledge about popular Python libraries and frameworks

### 2. List of the planned results of the course (training module), correlated with the planned results of the mastering the educational program

Mastering the discipline is aimed at the formation of the following competencies:

Code and the name of the competence	Competency indicators
Gen.Pro.C-1 Apply fundamental knowledge acquired in the physical and mathematical fields and/or natural sciences and use it in professional settings	Gen.Pro.C-1.1 Analyze the task in hand, outline the ways to complete it

### 3. List of the planned results of the course (training module)

As a result of studying the course the student should:

know:

- Python programming language syntax generally accepted methods for solving basic problems using language features core Python libraries and frameworks;
- the principle of execution of programs in Python;
- Python data types;
- flow control in Python;
- features of the standard library;
- rules for working with exceptions;
- the internal structure of the containers of the standard library and the time complexity of operations with them;
- how garbage collection works in Python;
- encodings used when storing text data (ASCII, Windows-1250/1251, UTF-8, UTF-16).

be able to:

- Implement a general-purpose library in Python using the specified interfaces;
- solve data processing tasks in Python.

master:

- Core Python libraries and developer tools.

### 4. Content of the course (training module), structured by topics (sections), indicating the number of allocated academic hours and types of training sessions

4.1. The sections of the course (training module) and the complexity of the types of training sessions

	Types of training sessions, including independent work
--	--

№	Topic (section) of the course	Lectures	Seminars	Laboratory practical	Independent work
1	Python programming language syntax			10	40
2	Python libraries and tools			10	45
3	Modules and Packages			10	20
AH in total				30	105
Exam preparation		0 AH.			
Total complexity		135 AH., credits in total 3			

#### 4.2. Content of the course (training module), structured by topics (sections)

Semester: 3 (Fall)

##### 1. Python programming language syntax

Variables Expressions. Functions Conditional statements and loops. Collection Methods. Enumeration of collections. Conditional expressions and collections. Comparison of collections. Garbage collector.

##### 2. Python libraries and tools

Syntax errors. Exceptions Throwing and catching exceptions. Keywords try, except, finally, raise. The thread of execution when throwing and catching an exception. The open and close methods. Construction with as. Reading and writing data in various encodings. Directories. Writing variables to a file (pickle module). Reading and writing in csv format. Read and write in json format.

##### 3. Modules and Packages

Connecting modules with the import statement. Various import syntaxes. Execution of the module as a script. Search paths. dir () "Compilation" of modules.Packages. Import inside the module (. And ..)

#### 5. Description of the material and technical facilities that are necessary for the implementation of the educational process of the course (training module)

Classroom equipped with computers for each student.

#### 6. List of the main and additional literature, that is necessary for the course (training module) mastering

##### Main literature

1. Основы операционных систем [Текст] : Курс лекций : учеб. пособие для вузов / В. Е. Карпов, К. А. Коньков .— 2-е изд., доп. и испр. — М. : Интернет - Ун-т информац. технологий, 2005 .— 536 с.
2. Программирование [Текст] : учеб. пособие для вузов / Н. А. Давыдова, Е. В. Боровская .— М. : БИНОМ. Лаб. знаний, 2009 .— 238 с.

##### Additional literature

1. Алгоритмы : построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест .— М. : МЦНМО, 2001 .— 960 с.

#### 7. List of web resources that are necessary for the course (training module) mastering

<https://www.youtube.com/watch?v=VP2wRhwl6c&list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl>

**8. List of information technologies used for implementation of the educational process, including a list of software and information reference systems (if necessary)**

IDE PyCharm

**9. Guidelines for students to master the course**

You can study additional materials from similar courses:

- Python programming basics <https://www.coursera.org/learn/python-osnovy-programmirovaniya>
- Python programming language <http://www.intuit.ru/studies/courses/49/49/info>

**Assessment funds for course (training module)**

**major:** Information Science and Computer Engineering  
**specialization:** Computer Science/Информатика  
Phystech School of Applied Mathematics and Informatics  
Chair of Algorithms and Programming Technologies  
**term:** 2  
**qualification:** Bachelor

Semester, form of interim assessment: 3 (fall) - Grading test

**Author:** V.V. Yakovlev, candidate of physics and mathematical sciences

## 1. Competencies formed during the process of studying the course

Code and the name of the competence	Competency indicators
Gen.Pro.C-1 Apply fundamental knowledge acquired in the physical and mathematical fields and/or natural sciences and use it in professional settings	Gen.Pro.C-1.1 Analyze the task in hand, outline the ways to complete it

## 2. Competency assessment indicators

As a result of studying the course the student should:

### know:

- Python programming language syntax generally accepted methods for solving basic problems using language features core Python libraries and frameworks;
- the principle of execution of programs in Python;
- Python data types;
- flow control in Python;
- features of the standard library;
- rules for working with exceptions;
- the internal structure of the containers of the standard library and the time complexity of operations with them;
- how garbage collection works in Python;
- encodings used when storing text data (ASCII, Windows-1250/1251, UTF-8, UTF-16).

### be able to:

- Implement a general-purpose library in Python using the specified interfaces;
- solve data processing tasks in Python.

### master:

- Core Python libraries and developer tools.

## 3. List of typical control tasks used to evaluate knowledge and skills

Not provided.

## 4. Evaluation criteria

1. The concept of assignment in Python
2. Functions
3. The while loop. Cycle Management Instructions.
4. The if statement. Cascading conditional elif construction.
5. Logical operations in Python.
6. Type str. The length of the string is len (s). Line immutability.
7. Slices of lines.
8. String methods find, count, replace, startswith, endswith.
9. Reference data model in Python. Operators == and is. Copy objects.
10. Slices of lists. Assignment to a slice. List methods.
11. List of lines. The split and join methods for the string.
12. The for loop and its features in Python.
13. List comprehensions: generating lists.
14. Two-dimensional arrays (lists of lists). Nested Generation.
15. Type set. Dozens and work with them.
16. Tuples and their use.
17. Type dict. Dictionary (associative array) and operations with it.
18. Dict comprehensions: generating sets and dictionaries.

19. Generators, yield.
20. Work with files. Open, close methods. Benefits of using with as.
21. Reading and writing a file in csv format
22. Reading and writing a json file
23. Reading and writing a file in various encodings. Codecs module.
24. Modules in Python. Running the module as a script.
25. Various import options. Advantages and disadvantages.

1. An array of integers A [0..n-1] is given. It is known that on the interval [0, m] the values of the array strictly increase, and on the interval [m, n-1] they strictly decrease. Find m in O (log m).

$2 \leq n \leq 10000$ .

2. Two-dimensional arrays (lists of lists). Nested Generation.

excellent

10 comprehensive, systematized, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, free and correct justification of decisions made;

9 systematic, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, the correct justification of decisions made;

8 deep knowledge of the curriculum of the discipline and the ability to apply them in practice when solving specific problems, the correct justification of decisions made;

good

7 firmly knows the material, correctly and essentially sets out it, knows how to apply the knowledge gained in practice, but admits some inaccuracies in the answer or in solving problems;

6 knows the material, correctly presents it, knows how to apply the acquired knowledge in practice, but admits some inaccuracies in the answer or in solving problems;

5 knows the basic material, correctly presents it, knows how to apply the knowledge gained in practice, but admits inaccuracy in the answer or in solving problems;

satisfactorily

4 fragmented, fragmented nature of knowledge, insufficiently correct wording of basic concepts, violation of logical sequence in the presentation of program material, but at the same time he owns the main sections of the curriculum necessary for further training and can apply the acquired knowledge in the standard situation;

3 the nature of knowledge is sufficient for further training and can apply the acquired knowledge on the model in a standard situation;

unsatisfactory

2 does not know most of the main content of the curriculum of the discipline, makes gross errors in the wording of the basic concepts of the discipline and does not know how to correctly use the knowledge gained in solving typical practical problems.

1 does not know the wording of the basic concepts of the discipline and does not know how to use the knowledge gained in solving typical practical problems.

## **5. Methodological materials defining the procedures for the assessment of knowledge, skills, abilities and/or experience**

The test can be carried out based on the results of current performance and delivery of tasks, laboratory and other types of work provided for by the discipline program and (or) by organizing a special survey conducted in oral and (or) written form, as well as with the issuance of tasks for implementation on a computer.