

**Syllabus
Academic Year 2022 -2023**

1. General information	
Course title	Advanced Programming 1(Python)
Degree cycle (level)/major	Elective Cybersecurity
Year, semester	2, 1
Number of credits	5
Language of delivery:	English
Prerequisites	-
Postrequisites	Advanced Programming II
Lecturer(s)	<p>Shynggys Alshynov, MSc in IT, senior-lecturer, CE Dept. Emails: shyngys.alshynov@astanait.edu.kz shynggys.alshynov@astanait.edu.kz</p> <p>Magzhan Ikram, MSc in Applied Math and Computer Science, senior-lecturer, CE Dept. Email: magzhan.ikram@astanait.edu.kz</p> <p>Zhantileuov Eldiyar, MSc in Cyber Physical and Social Systems, senior-lecturer, CE Dept.</p> <p>Tleubayeva Arailym, MSc in Technical Science (Information Systems), senior-lecturer, CE Dept.</p> <p>Astana IT University, Expo, C1 block</p>
2. Goals, objectives and learning outcomes of the course	
1. Course description	<p>This course is aimed to introduce students with a Python programming language and its libraries. A structure of the course is oriented on procedural programming, algorithm designing, working application forms (libraries), object-oriented programming, asynchronous programming, creating database applications. During the process students will also gain knowledge of creation of web apps and RESTful APIs and use of ORM library (SQLAlchemy or similar). Additionally, this course provides students understanding of using non-strict types of variables. Upon completion,</p>

	students should be able to code, test, create and debug fully functional and complicated applications on Python programming language.
2. Course goal(s)	Course goal is to provide a solid background in developing RESTful API with Python for students of Software Engineering, in preparation either for a job in industry or for more advanced courses at the graduate level.
3. Course objectives:	<p>Course objectives include:</p> <ul style="list-style-type: none"> - To introduce with regular expressions; - To introduce multiprocessing and multithreading; - To understand and demonstrate the concepts of Python programming and applying that knowledge to create web-services; - To understand the concept of both authorized and unauthorized requests; - To become familiar with web sockets and asynchronous programming;
4. Skills & competences	<p>The course designed for people who are already familiar with programming and requires basic understandings of programming paradigms</p> <ul style="list-style-type: none"> - Solve problems using various Python language and library features. - Understand and use web service development tools, machine learning and deep learning libraries.
5. Course learning outcomes:	<p>By the end of this course the students will be able to:</p> <ul style="list-style-type: none"> - Display understanding of programming paradigms in Python. - Demonstrate knowledge of how Python code is built and executed. - Apply obtained skills to write truly well-designed and stable software in Python. - Design a flexible architecture that serves as a program module or package.
6. Methods of assessment	<ul style="list-style-type: none"> - Quiz - Practical tasks - Assignments
7. Reading list	<p>Basic literature:</p> <ul style="list-style-type: none"> • www.python.org – Python official web site with fully provided documentation. [Online] • https://docs.sqlalchemy.org/en/14/ – SQLAlchemy ORM official web site with fully provided documentation. [Online] • www.fastapi.tiangolo.com – FastAPI. official web site with fully provided documentation. [Online] • https://pandas.pydata.org/docs/ – Pandas official web site with fully provided documentation. [Online] • https://flask.palletsprojects.com/en/2.2.x/ – Flask web framework official web site with fully provided documentation. [Online] • Kuhlman, D. (2009). <i>A Python Book: Beginning Python, Advanced Python, and Python Exercises</i>. 1st ed. http://www.opensource.org/licenses/mitlicense.php. • Mueller, J.P. (2018). <i>Beginning Programming with Python for dummies</i>. 2nd editions. https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html <p>Supplementary literature:</p> <ul style="list-style-type: none"> • Cisco Corp., <i>Programming Essentials in Python</i>, 2018, www.netacad.com Edube[Online]
8. Resources	<p>Basic literature:</p> <ul style="list-style-type: none"> • www.python.org – Python official web site with fully provided documentation. [Online]

	<ul style="list-style-type: none"> • https://docs.sqlalchemy.org/en/14/ – SQLAlchemy ORM official web site with fully provided documentation. [Online] • www.fastapi.tiangolo.com – FastAPI. official web site with fully provided documentation. [Online] • https://pandas.pydata.org/docs/ – Pandas official web site with fully provided documentation. [Online] • https://flask.palletsprojects.com/en/2.2.x/ – Flask web framework official web site with fully provided documentation. [Online] • Kuhlman, D. (2009). <i>A Python Book: Beginning Python, Advanced Python, and Python Exercises</i>. 1st ed. http://www.opensource.org/licenses/mitlicense.php. • Mueller, J.P. (2018). <i>Beginning Programming with Python for dummies</i>. 2nd editions. https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html <p>Supplementary literature: Cisco Corp., <i>Programming Essentials in Python</i>, 2018, www.netacad.com Edube[Online]</p>
9. Course policy	<p>Course and university policies include:</p> <p>Attendance: Students are expected to attend all scheduled class sessions with all required reading and supplementary materials. Readings are to be completed prior to class.</p> <p>The student won't obtain additional points for course attendance, but the attendance is important to pass the course. In case the student is not able to attend the classes for some reasons, he/she must inform the dean's office in advance and the student itself is responsible for learning all materials, which were given during unattended lessons.</p> <p>In case if the student did not attend more than 20% of the classes without any reasonable excuses, the teacher has a right to mark him as "not graded", and the student wouldn't be admitted to the exam. In other words, students must participate in at least 80% of all class time, otherwise he/she fails the course.</p> <p>Preparation for Class: Class participation is a very important part of the learning process in this course. Although not explicitly grade, students will be evaluated on the QUALITY of their contributions and insights. Quality comments possess one or more of the following properties:</p> <ul style="list-style-type: none"> - Offers a different and unique, but relevant, perspective; - Contributes to moving the discussion and analysis forward; - Builds on other comments. <p>Class work: The duration of each lecture and practical lesson is 50 minutes. Students are expected to complete all readings and assignments ahead of</p>

time, attend class regularly and participate in class discussions. In case of systemic student's misconduct, the student would be dispensed from the classes.

Being late on class: When students come to class late, it can disrupt the flow of a lecture or discussion, distract other students, impede learning, and generally erode class morale. Moreover, if left unchecked, lateness can become chronic and spread throughout the class. By the policy of this course, students who come late to class for more than 5 minutes are not allowed to get in to class and consequently, they will be marked as "absent" for the specific hour.

Home work / Assignments: The assignments are designed to acquaint students with the theoretical knowledge and practical skills required for the course. The textbook readings will be supplemented with materials collected from recent professional articles and journals. In case of using someone's work (papers, articles, any publications), all works must be properly cited. Failure to cite work will be resulted as a cheating from the students and may be a subject of additional disciplinary measures. **Cheating from other students will be strictly punished with a mark of 0 for every student with the same work.**

Late assignments: Late submissions will not be accepted and will be graded as 0.

Final exam: Project

Laptops and mobile devices can only be used for classroom purposes when directed by the teacher. Misuse of laptops or handheld devices will be considered a breach of discipline and appropriate action will be initiated by the teacher.

Cheating and plagiarism are defined in the Academic conduct policies of the university and include:

1. Submitting work that is not your own papers, assignments, or exams;
2. Copying ideas, words, or graphics from a published or unpublished source without appropriate citation;
3. Submitting or using falsified data;
4. Submitting the same work for credit in two courses without prior consent of both instructors.

	<p>Any student who is found cheating or plagiarizing on any work for this course will receive 0 (zero) for that work and further actions will also be taken regarding academic conduct policies of the university.</p> <p>Academic Conduct Policies of the university: The full texts of all the academic conduct code will be posted to the students using the learning management system (moodle.astanait.edu.kz).</p> <p>Contacting the Instructor (Teacher): The easiest and the most reliable way to get in touch with the teacher is by private message in MS Teams. Students must feel free to send messages if they have a question related to the course. The teacher will respond as soon as he can but not always instantaneously. Besides that, students are also welcome to arrange one-to-one meeting with the teacher during office hours to discuss the class.</p>
--	---

3. Course Content

#	Abbreviation	Meaning
1	TSIS	Teacher-supervised independent work
2	SIS	Students' independent work
3	IP	Individual project
4	PA	Practical assignment
5	LW	Laboratory work
6	MCQ	Multiple choice quiz

3.1 Lecture, practical/seminar/laboratory session plans

Week No	Course Topic	Lectures (H/W)	Practice sessions (H/W)	Lab. sessions (H/W)	TSIS (H/W)	SIS (H/W)

1	Introduction to Python Programming: <ul style="list-style-type: none"> ● Review, history, goals and versions ● Python3 installation ● Programming Environments ● Built-in instruments and pip ● Useful links and official documentation ● First program creation and running on different OSs ● Python print() function ● Short description of python programming components <ul style="list-style-type: none"> ○ Data types ○ Object referencing ○ Collections ○ Logical operations ○ Conditionals and loops ○ Error (exception) handling ○ Arithmetic operators ○ Input and Output ○ Function creation & call 	2	3	0	1	9
---	--	---	---	---	---	---

2	<p>Data types and operations over them:</p> <ul style="list-style-type: none"> ● Identifiers and keywords ● Numbers ● Strings <ul style="list-style-type: none"> ○ Indexing ○ Slicing ○ Comparison ○ Formatting ○ Methods ○ Encoding <p>Working with collections:</p> <ul style="list-style-type: none"> ● Sequences <ul style="list-style-type: none"> ○ Tuples ○ Named tuples ○ Lists ● Sets <ul style="list-style-type: none"> ○ Set ○ Frozenset ● Data Mapping <ul style="list-style-type: none"> ○ Dictionary ○ Dictionary with default values ● Loop avoidance and collections copying <ul style="list-style-type: none"> ○ Iterators, functions and operators to work with iterating objects ○ Collection copying 	2	3	0	1	9
---	---	---	---	---	---	---

3	Controlling structures and functions: <ul style="list-style-type: none"> ● Conditionals and Loops ● Error (exception) handling & own exceptions ● Functions <ul style="list-style-type: none"> ○ Function names and documenting ○ Document and parameters extraction ○ Recursion ○ Nested functions ○ Global variable access ○ Lambda functions ○ Asserts ● Asynchronous Programming 	2	3	0	1	9
4	Working with modules and files: <ul style="list-style-type: none"> ● Reading and writing of binary files ● Writing and analysis of txt files ● Context manager “with” ● Working with JSON files Object-oriented programming in Python: <ul style="list-style-type: none"> ● OOP principles ● Own classes ● Dynamic code execution and dynamic import ● Decorators and annotations of functions and methods ● Functors ● Descriptors ● Class decorators 	2	3	0	1	9
5	Working with databases and network interaction: <ul style="list-style-type: none"> ● Databases in python <ul style="list-style-type: none"> ○ PostgreSQL ○ MySQL ○ SQLite3 ● Networking in python <ul style="list-style-type: none"> ○ TCP server and client 	2	3	0	1	9

6	Flask Web Application Development <ul style="list-style-type: none"> • Environment setup & installation • REST API • Postman for testing • Endpoint implementation 	2	3	0	1	9
7	SQLAlchemy <ul style="list-style-type: none"> • Object Relational Mapper • Flask SQL Alchemy • CRUD 	2	3	0	1	9
8	FastAPI <ul style="list-style-type: none"> • Overview • Installation • HTTP Exceptions • Status Codes Authentication and Authorization (JWT) <ul style="list-style-type: none"> • Authentication • Authorization • JWT • OAuth 	2	3	0	1	9
9	Introduction to Data science. <ul style="list-style-type: none"> • Intro to numpy, • Intro to Pandas • Intro to Exploratory Data analysis 	2	3	0	1	9
10	Project week	2	3	0	1	9
	Total hours: 150	20	30	0	10	90

3.2 List of assignments for Student Independent Study

№	Assignments (topics) for Independent study	Hours	Recommended literature and other sources (links)	Form of submission
1	Lab 1 – Setting up environment	3	www.python.org www.netacad.com https://www.jetbrains.com/pycharm/download/ https://jupyterlab.readthe	-

			docs.io/en/stable/getting-started/installation.html http://pythonanywhere.com/	
2	Lab 2 – Working with print function and manipulations with numbers and strings	3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
3	Lab 3 – Working with collections	3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
4	Lab 4 – Working with controlling structures	3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
5	Lab 5 – Working with python files & OOP	3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
6		3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html	moodle.astanait.edu.kz

			www.netacad.com	
7		3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
8		3	www.python.org https://regex101.com/r/F8dY80/3 http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
9		3	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz
10		5	www.python.org http://www.opensource.org/licenses/mitlicense.php . https://www.pdfdrive.com/beginning-programming-with-python-for-dummies-e176211134.html www.netacad.com	moodle.astanait.edu.kz

4. Student performance evaluation system for the course

Period	Assignments	Number of points	Total
--------	-------------	------------------	-------

1 st attestation	<i>Midterm Practical Task</i> <i>Midterm Theoretical Task</i>	50 50	100
2 nd attestation	<i>Assignment 1</i> <i>Assignment 2</i>	50 50	100
Final exam			100
Total	0,3 * 1st Att + 0,3 * 2nd Att + 0,4*Final		100

Achievement level as per course curriculum shall be assessed according to the evaluation chart adopted by the academic credit system.

Letter Grade	Numerical equivalent	Percentage	Grade according to the traditional system
A	4,0	95-100	Excellent
A-	3,67	90-94	
B+	3,33	85-89	Good
B	3,0	80-84	
B-	2,67	75-79	
C+	2,33	70-74	
C	2,0	65-69	
C-	1,67	60-64	Satisfactory
D+	1,33	55-59	
D	1,0	50-54	
FX	0	25-49	
F	0	0-24	Fail

5. Methodological Guidelines

Assessment is administered continuously throughout the course. The students are rated against their performance in continuous rating administered throughout the semester (60%) and summative rating done during the examination session (40%), total 100%. Continuous rating is students' on-going performance in class and independent work. Class work is assessed for attendance, laboratory works' defense and in- class assessments.

- **TSIS (Teacher Supervised Student Independent Study)** -comprises presentation to be done by students independently and checked by instructor.
- **Mid-term and End-term** is a review of the topics covered and assessment of each student's knowledge. The form of the midterm and endterm exams is complex.
- **Final assessment** for the course “Python Programming I” includes two parts. The first part is a multiple-choice test for one hour which covers the most theoretical part of the course. The second part is programming part where students will show their practical skills. At the completion of the exam, all works must be submitted in the Learning Management System (moodle.astanait.edu.kz). No late submissions are allowed in the exam

6. Lecturer (lecturers) approvals

Full name	Job title	Date	Sign
Shynggys Alshynov	Senior lecturer		
Magzhan ikram	Senior lecturer		
Zhantileuov Eldiyar	Senior lecturer		