

## Python Programming - Beginner to Pro Course Policies & Syllabi, 2024

Course Name	Python Programming - Beginner to Pro
Duration of the Course in Weeks	26 weeks
Duration of the Course in Academic Hours	156 (of which 60% practice hours)
Tuition Fee	600,000 Kazakhstani Tenge (100,000 KZT per month)
Learning Format	Online – pre-recorded lectures + live practice hours (Google Meet)
Learning Platform	In-house LMS + Google Meet
Frequency of Classes	Weekly lectures + bi-weekly practice (combined 6 hours)
Skills / Results Upon Completion of Training	Python Fundamentals: Syntax, data types, and control structures.
Qualification Upon Completion of Training	Certificate of Completion in "Python Programming - Beginner to Pro"
Methodological Basis of the Course	Sequential learning progression. Hands-on practice
The Ratio of Theory to Practice	60-40 (60% practice, 40% theory)
Teaching Staff	Yedyge Radolda Sufyan Mustafa bin Uzayr
Tracking the Development of Students	Assessment and Evaluation Feedback Mechanisms Coding Portfolios Peer and Self-Assessment
Student Performance Evaluation Methods	Midterms + Coding Tests + Final Exam

## Course Description

This comprehensive Python programming course takes students from beginner to pro level, covering fundamental syntax, data types, and control structures, progressing to advanced topics like OOP, file handling, and external libraries. Through hands-on projects, students master data structures, functional programming, web development basics, and introductory data science and machine learning concepts. By course end, students have the skills and confidence to develop scalable applications and pursue careers in software development, data science, or machine learning.

## Course Duration

Number of weeks	26
Number of academic hours	156 (of which 60% practice hours)

## Pre-requisites

- As the course materials, lectures, and instructions are primarily delivered in English, a strong understanding of the English language is essential for effective comprehension and communication.
- Fundamental logical and reasoning abilities are required to understand programming concepts and solve problems efficiently.

## Skills Taught

- Python Fundamentals: Syntax, data types, and control structures
- Object-Oriented Programming (OOP): Classes, objects, inheritance, polymorphism, encapsulation, and abstraction
- Error Handling: Implementing exception handling techniques
- Debugging Techniques: Identifying and fixing errors in code

- Working with Modules and Packages: Organizing code into reusable modules, utilizing external libraries
- Web Development Basics: Using frameworks like Flask or Django
- Data Analysis and Visualization: NumPy, Pandas, and Matplotlib
- Project-Based Learning: Applying learned concepts to real-world projects

## Learning Plan

Week	Topic	Notes
1	Introduction to Python. Overview of Python language and its applications. Introduction to Python interpreter and basic syntax.	
2	Setting up Python Environment. Installation of Python interpreter and relevant tools. Setting up integrated development environment (IDE) for Python programming.	
3	Data Types: Basic Syntax and Variables. Syntax rules and conventions in Python. Understanding variables, data types, and basic operations, Strings, and Booleans.	
4	Data Types: Numbers, Strings, and Booleans. Detailed exploration of numeric, string, and boolean data types. Manipulating and formatting strings, performing arithmetic operations.	
5	Lists and Tuples. Understanding lists and tuples as ordered collections of elements. Accessing and manipulating list and tuple elements.	
6	Dictionaries and Sets. Exploring dictionary and set data structures for efficient data organization. Performing operations like insertion, deletion, and	CT-1

	retrieval in dictionaries and sets.	
<b>7</b>	Control Structures: Conditionals and Loops. Implementing conditional statements (if, elif, else) for decision-making. Utilizing loop structures (for, while) for iteration and repetitive tasks.	
<b>8</b>	Functions and Scope. Defining and calling functions with parameters and return values. Understanding function scope and variable visibility.	
<b>9</b>	File Input and Output. Reading from and writing to files using Python. Handling file operations, such as opening, closing, and reading/writing data.	
<b>10</b>	Exception Handling. Handling runtime errors using try-except blocks. Implementing error handling strategies to gracefully manage exceptions.	
<b>11</b>	Debugging Techniques. Identifying and resolving errors in Python code. Utilizing debugging tools and techniques for effective troubleshooting.	
<b>12</b>	Introduction to Object-Oriented Programming (OOP). Understanding OOP principles and concepts. Introduction to classes, objects, and methods.	
<b>13</b>	Midterm-1	Midterm-1
<b>14</b>	Classes and Objects. Creating and using classes to model real-world entities. Instantiating objects and accessing their attributes and methods.	
<b>15</b>	Inheritance and polymorphism. Implementation of inheritance for code reuse and extension. The study of polymorphic behavior in Python.	

<b>16</b>	Encapsulation and Abstraction. Encapsulating data and behavior within classes. Achieving abstraction through class interfaces and method signatures.	
<b>17</b>	Advanced Functions: Lambda Functions, Map, Filter, and Reduce. Understanding lambda functions for concise and inline function definitions. Exploring map, filter, and reduce functions for functional programming paradigms.	
<b>18</b>	Working with Modules and Packages. Organizing code into modules and packages for modularity and reusability. Importing and using external modules and packages in Python programs.	CT-2
<b>19</b>	Introduction to External Libraries. Exploring popular external libraries and their functionalities. Installation and usage of external libraries for extended capabilities.	
<b>20</b>	Web Development Basics with Flask or Django. Introduction to web development frameworks. Flask or Django. Creating web applications and handling HTTP requests and responses.	
<b>21</b>	Data Analysis with NumPy and Pandas. Introduction to NumPy for numerical computing in Python. Exploring Pandas for data manipulation and analysis.	
<b>22</b>	Data Visualization with Matplotlib. Creating visualizations using Matplotlib for data analysis and presentation. Customizing plots and charts to convey insights effectively.	
<b>23</b>	Introduction to Machine Learning Concepts. Understanding fundamental concepts and terminology in machine learning. Overview of	

	supervised and unsupervised learning approaches.	
<b>24</b>	Using scikit-learn for Machine Learning Tasks. Introduction to scikit-learn library for machine learning in Python. Implementing common machine learning algorithms for classification, regression, and clustering.	
<b>25</b>	End Term (Midterm-2)	Midterm-2
<b>26</b>	Projects: Making a Voice Assistant in Python Task Manager Application Data Analysis Dashboard E-commerce Website with Flask Weather Forecasting Application	
<b>Finals Period</b>		Final Exam

## Student Evaluation Criteria

Component	Description	When	Points
Class Test	(a) Computer-based (b) Online	6th and 18th week	each worth 15 points → 30 points
Midterms	(a) Computer-based (b) Online	13th and 25th week	each worth 15 points → 30 points
Final Exam	(a) Computer-based (b) Online	Finals Period	40 points
Total			100 points

Letter Grade	Points	Grade Remarks
A	95-100	Excellent
A-	90-94	
B+	85-89	Good
B	80-84	
B-	75-79	
C+	70-74	
C	65-69	Satisfactory
C-	60-64	
D+	55-59	
D	50-54	
FX	25-49	Fail
F	0-24	

## Course Fee

600,000 Kazakhstani Tenge (100,000 KZT per month)

## Teaching Staff

Yedyge Radolda
Email - edige.rad@gmail.com
Linkedin Profile - <a href="https://www.linkedin.com/in/yedige-radolda/">linkedin.com/in/yedige-radolda/</a>

Yedyge Radolda is a dedicated educator and mentor who is educating over 1000 students in programming, algorithms, and data structures. With a passion for fostering excellence, they've mentored over 40 students to win prestigious awards in regional competitive programming Olympiads. Additionally, their contributions to the Kazakh IT community and their popular YouTube channel with over 3300 subscribers reflect their commitment to advancing tech education.

Sufyan Mustafa bin Uzayr

Telegram Username - @sufyanbinuzayr

Site: <https://sufyanism.com/>

Google Scholar Profile -

<https://scholar.google.com/citations?user=CzF2dJIAAAJ>

Sufyan Mustafa bin Uzayr is a skilled teacher with extensive experience in the IT industry, having taught for over 5 years. He is proficient in various software tools and programming paradigms. Having authored 60+ books, their expertise in Sufyan has taught and worked at universities and colleges in 3 different continents.

## Student Selection Criteria

To be admitted to the program, students are required to pass an Aptitude Test, consisting of 25 questions. The Test shall be conducted online, the dates of which shall be communicated to the students via email.

Additionally, the Test topics shall include General Reasoning and Logic, Mathematical Skills (School Grade-8 level).

It is a pre-requisite for students to possess general verbal and oral English skills.



## Training Format

Online – pre-recorded lectures + live practice hours (Google Meet)

## Use of Technology in Teaching

Zeba Academy uses its own flagship Learning Management System, that comes with features such as Pre-recorded Video Lessons, Advanced Quizzes, Flashcards, Image Hotspots Groups and Cohorts, AI-based adaptive learning algorithm, and guided practice materials in the form of ebooks and other content.

## Contact

Zeba Academy Help Desk for Students

Telegram and WhatsApp: **+7 705 435 2915**

