



S Y L L A B U S

Python Programming for Beginners

Duration: 5 Days (Total: 20 Hours)

Contents

1.0 Course Overview	3
1.1 Learning Outcomes	3
1.2 Training Methodology	3
2.0: Introduction to Python & Programming Concepts (3 Hours)	4
2.1 Objectives:	4
2.2 Topics Covered:	4
2.3 Activities:	4
3.0: Data Types, Variables, and Operators (3 Hours)	5
3.1 Objectives:	5
3.2 Topics Covered:	5
3.3 Activities:	5
4.0: Control Structures – Conditions & Loops (4 Hours)	6
4.1 Objectives:	6
4.2 Topics Covered:	6
4.3 Activities:	6
5.0: Functions and Modular Programming (3 Hours)	7
5.1 Objectives:	7
5.2 Topics Covered:	7
5.3 Activities:	7
6.0: Working with Lists, Tuples, and Dictionaries (3 Hours)	7
6.1 Objectives:	7

6.2 Topics Covered:.....	8
6.3 Activities:.....	8
7.0: Basic File Handling & Error Handling (2 Hours)	8
7.1 Objectives:.....	8
7.2 Topics Covered:	8
7.3 Activities:.....	9
8.0: Conclusion and Wrap-Up (1 Hour).....	9
8.1 Key Takeaways:	9
8.2 Final Activities:.....	9

1.0 Course Overview

This beginner-level training module is designed for individuals with no prior programming experience. It introduces learners to the world of coding through Python, one of the most versatile and beginner-friendly programming languages. Through hands-on activities and real-world examples, learners will develop foundational skills in writing, understanding, and applying Python code.

1.1 Learning Outcomes

By the end of this course, participants will be able to:

Understand the basic concepts of programming and Python.

Write Python scripts using variables, data types, and operators.

Implement logical decisions and loops.

Create reusable code using functions.

Work with collections such as lists, tuples, and dictionaries.

Handle files and manage basic error situations in Python.

1.2 Training Methodology

Interactive Coding Lectures

Live Demos and Code Walkthroughs

Hands-On Practice Exercises

Pair Programming & Debugging Sessions

Group Challenges

Online Tools (e.g., Replit, Jupyter Notebook)

2.0: Introduction to Python & Programming Concepts (3 Hours)

2.1 Objectives:

Understand what programming is and why Python is a good first language.

Learn how to set up the development environment.

Write and run your first Python program.

2.2 Topics Covered:

What is Programming?

Programming logic and thinking

Role of programming in modern software

Why Python?

Simplicity, readability, and wide usage

Use cases: web, automation, data science, etc.

Installing Python & Setting Up Environment

Installing Python and VS Code / Replit

Writing Your First Python Script

Using `print()` function

Syntax and indentation basics

2.3 Activities:

Exercise: Install Python and write a "Hello, World!" script

Group Discussion: Python's use in different industries

Pair Activity: Code a basic text output using variables

3.0: Data Types, Variables, and Operators (3 Hours)

3.1 Objectives:

Understand how data is stored and manipulated in Python.

Learn to use different data types and perform operations.

3.2 Topics Covered:

- Variables & Assignment
- Naming conventions
- Dynamic typing

Data Types

- Integers, floats, strings, booleans
- Type Conversion
- Casting between data types

Operators

- Arithmetic, comparison, logical, assignment operators

3.3 Activities:

Quiz: Identify the correct data type

Exercise: Simple calculator using variables and operators

Group Task: String formatting and basic math operations

4.0: Control Structures – Conditions & Loops (4 Hours)

4.1 Objectives:

Learn to write programs that can make decisions and repeat actions.

Understand the role of logic in programming flow.

4.2 Topics Covered:

Conditional Statements

- if, elif, else

Logical Expressions

- AND, OR, NOT operators

Loops

- for and while loops
- Range function

Loop Control

- break, continue, pass

4.3 Activities:

Scenario Practice: Write a program that grades scores

Code Exercise: Multiplication table generator

Challenge: Guess the Number game using while loop

5.0: Functions and Modular Programming (3 Hours)

5.1 Objectives:

Create reusable code using Python functions.

Understand parameters, return values, and scope.

5.2 Topics Covered:

Defining Functions

- Syntax and indentation

Function Arguments

- Positional, keyword, default values

Return Statements

- Using return to send data back

Scope and Lifetime

- Local vs global variables

5.3 Activities:

Code Along: Simple calculator using functions

Group Task: BMI calculator

Practice: Write a function to check for prime numbers

6.0: Working with Lists, Tuples, and Dictionaries (3 Hours)

6.1 Objectives:

Work with collections to store and access groups of data.

Perform common operations on Python data structures.

6.2 Topics Covered:

- Lists
- Creating, accessing, updating, slicing
- Tuples
- Immutable collections
- Dictionaries
- Key-value pairs
- Adding, removing, and accessing data
- Iteration over collections

6.3 Activities:

Exercise: Student grades tracker using lists

Activity: Store personal information using a dictionary

Challenge: Word frequency counter from input string

7.0: Basic File Handling & Error Handling (2 Hours)

7.1 Objectives:

Learn to read from and write to text files.

Understand and handle basic runtime errors.

7.2 Topics Covered:

File Handling

- `open()`, `read()`, `write()`, and `close()`
- File modes: read, write, append

Error Handling

- Try-except blocks
- Handling common errors (e.g., file not found, type errors)

7.3 Activities:

Hands-On: Read and display content from a .txt file

Group Practice: Simple log file generator

Exercise: Exception handling during user input

8.0: Conclusion and Wrap-Up (1 Hour)

8.1 Key Takeaways:

Review of major concepts covered.

Next steps in the programming journey.

Discussion of Python libraries and future learning paths (e.g., pandas, tkinter, Flask).

8.2 Final Activities:

Group Presentation: “Mini Project Demo”

Feedback Forms & Certificate Distribution