

Python Programming: 6-Week Course Outline

This course provides a comprehensive introduction to Python programming, focusing on core programming concepts, problem solving, and practical applications. Over six weeks, learners will gain hands-on experience with Python and develop the skills needed to build scripts, work with data, and create small applications.

Week 1: Introduction to Python and Programming Basics

- Introduction to Programming Concepts
- What is Python and why it is widely used
- Installing Python and Setting up the Development Environment (VS Code / Jupyter Notebook)
- Python Syntax and Structure
- Variables and Data Types: Integers, Floats, Strings, Booleans
- Basic Operators: Arithmetic, Comparison, Logical
- Writing and running your first Python program

Week 2: Control Flow and Functions

- Conditional Statements: if, elif, else
- For loops and While loops
- Loop control statements: break, continue, pass
- Functions: defining functions
- Function parameters and return values
- Writing reusable code with functions

Week 3: Data Structures in Python

- Introduction to Data Structures
- Lists: creating, modifying, and list methods
- Tuples
- Dictionaries
- Sets
- Iterating through data structures
- Practical exercises using collections

Week 4: File Handling and Error Handling

- Reading and writing files in Python
- Working with text files and CSV files
- File operations: open(), read(), write()
- Exception Handling: try, except, finally
- Debugging techniques

Week 5: Python Libraries and Data Processing

- Introduction to Python Libraries
- Working with NumPy for numerical computation
- Introduction to Pandas for data analysis
- DataFrames and Series
- Basic data manipulation with Pandas
- Simple data visualization with Matplotlib

Week 6: Practical Applications and Final Project

- Introduction to Object-Oriented Programming (OOP): Classes, Objects, Methods
- Building simple Python applications
- Automating tasks using Python scripts
- Introduction to APIs and working with external data
- Final Project Examples: Student Grade System, Simple Calculator, Data Analysis Mini Project
- Project steps: problem definition, coding, testing, presenting results