**Python Programming**

**Course Duration: 40 Hrs**

**Course Overview**

The *Python Programming Fundamentals* course provides a structured introduction to Python, covering essential programming concepts, control structures, functions, data structures, file handling, object-oriented programming (OOP), and advanced topics such as regular expressions and working with system modules. Participants will gain hands-on experience in writing Python scripts, handling data, managing files, and utilizing Python’s built-in libraries for various real-world applications. The course ensures a strong foundation in Python programming, preparing learners for data analysis, automation, and software development.

**What you’ll learn?**

* Set up and run Python using Anaconda, PyCharm, or VS Code
* Understand Python syntax, keywords, data types, and variables
* Work with operators, expressions, and control flow structures
* Define and call functions, use recursion, and import modules
* Manipulate strings, lists, tuples, dictionaries, and sets
* Handle file operations, including reading and writing CSV/JSON files
* Implement object-oriented programming (OOP) concepts like classes, inheritance, and encapsulation
* Use exception handling for error management
* Work with regular expressions, system modules (os), and datetime functions

**Target Audience**

This course is designed for beginners, aspiring programmers, data analysts, and software developers looking to build a solid foundation in Python. It is suitable for students, professionals, and individuals transitioning into programming, automation, or data science roles.

**Pre-Requisites**

No prior programming experience is required, making this course ideal for beginners. However, familiarity with basic computer operations and logical thinking will be beneficial for a smoother learning experience.

**Course content**

1. Introduction to Python

* Overview of Python
* Python Installation and Setup (Anaconda, PyCharm, VS Code)
* Python Syntax and Interactive Mode
* Running Python Scripts

2. Basic Python Concepts

* Keywords, Identifiers, and Variables
* Data Types (int, float, str, bool, None)
* Input and Output in Python
* Comments and Documentation Strings

3. Operators and Expressions

* Arithmetic, Relational, Logical, and Assignment Operators
* Conditional (Ternary) Operator
* Operator Precedence and Associativity
* Working with Expressions

4. Control Structures

* Conditional Statements: if, ifelse, and nested if
* Looping Constructs: for, while, nested loops
* break, continue, and pass statements

5. Functions and Modules

* Defining and Calling Functions
* Function Parameters and Return Values
* Recursion and Lambda Functions
* Importing Modules and Standard Libraries (math, random)

6. Strings and String Operations

* String Manipulation and Formatting
* String Methods and Functions
* Slicing and Indexing Strings
* Regular Expressions

7. Lists and Tuples

* Creating and Accessing Lists and Tuples
* List and Tuple Methods
* Slicing, Indexing, and Looping through Lists and Tuples
* Nested Lists and Tuples

8. Dictionaries and Sets

* Creating and Accessing Dictionaries and Sets
* Dictionary and Set Methods
* KeyValue Pairs, Hashing, and Looping through Dictionaries
* Use Cases of Dictionaries and Sets

9. File Handling in Python

* Opening, Reading, and Writing Files
* Working with File Modes (r, w, a, etc.)
* Handling CSV and JSON Files
* Error Handling in File Operations

10. Introduction to ObjectOriented Programming

* Understanding Classes and Objects
* Defining Classes and Creating Objects
* Attributes and Methods
* The `self` Keyword and Constructor (\_\_init\_\_)

11. Inheritance and Polymorphism

* Single and Multiple Inheritance
* Overriding Methods and Using Super()
* Polymorphism: Method Overloading and Overriding
* Abstract Classes and Interfaces

12. Encapsulation and Data Hiding

* Access Specifiers: Public, Protected, and Private
* Getters and Setters
* Property Decorators
* Encapsulation Best Practices

13. Exception Handling

* Understanding Exceptions
* try, except, else, and finally Blocks
* Raising Exceptions
* Creating Custom Exceptions

14. Working with Modules and Packages

* Importing and Using Modules
* Creating and Managing Packages
* Understanding the Python Package Index (PyPI)
* Using pip for Package Management

15. Working with Dates and Times

* Using datetime and time Modules
* Formatting and Parsing Dates
* Time Arithmetic and Comparisons
* Time Zones and Localization

16. Introduction to Regular Expressions

* Understanding Regular Expressions
* Using the `re` Module
* Matching, Searching, and Replacing Patterns
* Practical Examples and Use Cases

17. Working with the `os` Module

* Introduction to the `os` Module
* File and Directory Operations
* Environment Variables and System Information
* Running System Commands using `os.system()`
* Path Manipulations with `os.path`