Python Programming Syllabus

Python

Introduction

- ✓ **History and Overview**: Introduction to Python, its history, and key features.
- ✓ **Python Installation**: Installing Python, setting up the environment (IDEs like PyCharm, Jupyter Notebook, or VS Code).
- ✓ **First Python Program**: Writing and running the first Python program (print("Hello, World!")).
- ✓ **Python Interpreter**: Understanding the interactive mode, running Python scripts.

Basic Syntax and Data Types

Variables and Data Types:

- ✓ Numeric types: int, float, complex.
- ✓ Sequence types: list, tuple, range.
- ✓ Mapping type: dict.
- ✓ Set types: set, frozenset.
- ✓ Boolean type: True, False.
- ✓ Binary types: bytes, bytearray, memoryview.

Basic Operators:

- ✓ Arithmetic operators, assignment operators, comparison operators, logical operators, bitwise operators, membership and identity operators.

Control Structures

- ✓ **Conditional Statements:** if, elif, else, and nested conditions.
- ✓ **Loops:** for loop, while loop, break, continue, and else with loops**.**
- ✓ **Comprehensions:** List comprehension, dictionary comprehension, and set comprehension.

Functions

- ✓ **Defining Functions:** Syntax, function parameters, return values.
- ✓ **Arguments and Parameters:** Positional arguments, keyword arguments, default parameters, arbitrary arguments (*args, **kwargs).
- ✓ **Lambda Functions:** Anonymous functions using lambda.
- ✓ Recursion: Understanding recursive functions, recursion depth.
- ✓ **Function Scopes:** Local and global variables, the global and nonlocal keywords.

Data Structures

- ✓ **Lists:** Definition, indexing, slicing, adding, removing, sorting, and manipulating lists.
- ✓ **Tuples**: Immutable sequences, indexing, slicing, and operations on tuples.
- ✓ **Dictionaries:** Key-value pairs, accessing values, adding/removing elements, nested dictionaries.
- ✓ **Sets:** Unordered collections of unique elements, set operations like union, intersection, difference.
- ✓ **Strings:** String methods, slicing, formatting, escaping characters, and multi-line strings.
- ✓ **File Handling**: Reading from and writing to files, file modes, handling file exceptions.

Object-Oriented Programming (OOP)

- ✓ **Classes and Objects:** Defining classes, creating objects, and accessing class members.
- ✓ **Methods:** Instance methods, class methods, static methods.
- ✓ **Constructor and Destructor**: __init__, __del__ methods.
- ✓ **Inheritance:** Single and multiple inheritance, method overriding, super() function.
- ✓ **Polymorphism:** Method overloading, method overriding, duck typing.
- ✓ **Encapsulation:** Public, protected, and private members, getter and setter methods.
- ✓ **Abstraction:** Abstract classes and methods using abc module.

Error and Exception Handling

- ✓ **Try, Except Block:** Syntax for handling exceptions, catching multiple exceptions.
- ✓ **Else and Finally:** Executing code after try-except, cleanup operations in finally.
- ✓ **Raising Exceptions**: Raise statement to generate exceptions manually.
- ✓ **Custom Exceptions:** Creating user-defined exception classes.

File Handling

- ✓ **File Operations**: fopen, fclose, fread, fwrite, fprintf, fscanf.
- ✓ **File Modes**: Reading, writing, and appending to files.
- ✓ **Error Handling**: Handling errors in file operations.
- ✓ **Binary Files**: Reading and writing binary data.

Modules and Packages

- ✓ **Modules**: Importing standard and custom modules using import and from-import.
- ✓ **Packages:** Organizing code into packages, creating and importing packages.

Advanced Python Concepts

- ✓ **Decorators:** Function decorators, class decorators, using @decorator.
- ✓ **Generators:** Using yield, creating generator functions, and understanding iterator behavior.
- ✓ **Iterators:** Creating and using iterators, iter(), next().
- ✓ **Context Managers:** Using with statement for file handling and resource management.
- ✓ **Regular Expressions**: re module for pattern matching and text manipulation**.**

Python Libraries and Frameworks

- ✓ **NumPy:** Introduction to NumPy arrays, operations on arrays, matrix manipulation.
- ✓ **Pandas:** Data manipulation with DataFrames, reading and writing CSV files, filtering and transforming data.
- ✓ **Matplotlib:** Basic data visualization, plotting graphs and charts.
- ✓ **Flask/Django**: Introduction to web development using Flask or Django for building web applications.

Data Science and Machine Learning (Introductory)

- ✓ **Data Analysis with Pandas:** Data cleaning, data aggregation, merging datasets, handling missing values.
- ✓ **Basic Machine Learning with Scikit-learn:** Introduction to machine learning concepts, supervised and unsupervised learning, classification algorithms (e.g., Logistic Regression), regression algorithms, evaluation metrics**.**
- ✓ **Visualization:** Using matplotlib and seaborn for data visualization.