Python Programming

Module 1: Introduction to Python

- History and features of Python
- Applications of Python (Web, AI, Data Science, etc.)
- Installing Python and setting up environment (IDLE, VS Code, PyCharm)
- Python interpreters and code execution
- Writing and running your first Python program

Module 2: Basic Syntax and Data Types

- Keywords, identifiers, comments
- Variables and constants
- Data types:
  - Numbers (int, float, complex)
  - Strings
  - Booleans
- Type conversion and type casting
- `input()` and `print()` functions
- Operators:
  - Arithmetic, Relational, Logical, Bitwise
  - Assignment, Membership, Identity

Module 3: Control Flow Statements

- Conditional statements: `if`, `if-else`, `elif`
- Looping:
  - `for` loop
  - `while` loop
- Loop control:
  - `break`, `continue`, `pass`
- `range()` and `enumerate()`

Module 4: Data Structures

- **Lists**:
  - Creation, indexing, slicing, list methods
- **Tuples**:
  - Immutable sequences, packing/unpacking
- **Sets**:
  - Set operations and methods
- **Dictionaries**:
  - Key-value pairs, methods, loops
- Nested data structures

Module 5: Functions and Modules

- Defining and calling functions
- Arguments: positional, keyword, default, `*args`, `**kwargs`
- Return statement
- Lambda functions
- Built-in vs user-defined functions
- Modules and packages (`import`, `from`, `as`)
- Python Standard Library overview (`math`, `random`, `datetime`, etc.)

Module 6: File Handling

- Opening and closing files
- File modes (`r`, `w`, `a`, `x`)
- Reading and writing to text files
- Working with binary files
- `with` statement and context manager
- File methods (`read()`, `readline()`, `write()`)

Module 7: Exception Handling

- Types of exceptions
- Try, except blocks
- `else`, `finally`, `raise`
- Built-in vs user-defined exceptions

Module 8: Object-Oriented Programming (OOP)

- Classes and objects
- `__init__` method and constructor
- Instance and class variables
- Methods and method types (`staticmethod`, `classmethod`)
- Inheritance (single, multiple, multilevel)
- Method overriding
- Encapsulation and abstraction
- Special methods (`__str__`, `__len__`, etc.)

Module 9: Iterators, Generators, and Comprehensions

- Iterators and `__iter__`, `__next__`
- Generators and `yield`
- List, dictionary, and set comprehensions
- Generator expressions

Module 10: Advanced Python Topics

- Decorators
- Closures

- Context managers
- Regular Expressions (`re` module)
- Date and time handling
- JSON data handling

Module 11: Working with External Libraries

- Installing packages with `pip`
- Popular libraries:
    - `requests` (HTTP requests)
    - `os` and `shutil` (file system operations)
    - `sys`, `argparse` (command-line arguments)
    - `logging`
    - `sqlite3` (basic database access)

Module 12: Testing and Debugging

- Writing unit tests using `unittest` or `pytest`
- Test discovery and test cases
- Debugging tools and techniques

Logging vs print statements

==========

END

==========