Python Basics – Module 1 Lesson Plan (10 Hours)

Session 1 (2 Hours): Introduction to Python & Variables

Topics:

- Python Interpreter / Shell
- Identifiers, Keywords
- Statements & Expressions
- Variables
- Operators (Arithmetic, Relational, Logical, Assignment, etc.)
- Operator Precedence & Associativity

Programs / Activities:

1. Write a Python program to declare variables, perform arithmetic operations, and display results.
2. Write a Python program to calculate the area and perimeter of a rectangle using variables and operators.

Session 2 (2 Hours): Data Types, Input/Output & Control Flow

Topics:

- Data Types (int, float, bool, str, complex)
- Indentation rules & Comments
- Reading Input using `input()`
- Printing Output with formatting
- Type Conversions, `type()`, `is` operator
- Python as Dynamically & Strongly Typed Language
- Conditional statements: `if`, `elif`, `else`, nested if
- Loops: `while`, `for`, `range()`
- Loop manipulation: `break`, `continue`, `else`, `pass`

Programs / Activities:

1. Create a program to check if a number is even or odd using if-else.
2. Write a Python program to print the first n Fibonacci numbers using a for loop.
3. Demonstrate the use of break, continue, and pass in loops.

Session 3 (2 Hours): Functions

Topics:

- Function Definition & Calling
- Built-in Functions
- Return Statement
- Default Parameters
- Scope and Lifetime of Variables
- Command Line Arguments (`sys.argv`)

Programs / Activities:

1. Implement a program to generate prime numbers up to n using a generator function.
2. Write a Python program to calculate the factorial of a number using recursion.

Session 4 (2 Hours): Strings

Topics:

- Creating and Storing Strings
- String Operations (concatenation, repetition, membership)
- Indexing & Slicing
- Joining & Splitting Strings
- Common String Methods (`upper`, `lower`, `find`, `replace`, `strip`, etc.)

Programs / Activities:

1. Implement a program that accepts a string and counts the number of vowels and consonants.
2. String Operations: Write a program to count the occurrences of each word in a given string.

Session 5 (2 Hours): Collections (Intro)

Topics:

- Lists (basic overview)
- Dictionaries (CRUD operations)
- Storing & Retrieving Data with Dictionaries

Programs / Activities:

1. Implement a program to insert, delete, and update elements in a list.
2. Implement a program to input a list of integer/floating point values and sort list and descendingly sort second half of sorted list

---

Python Basics – Program Practice Sheet

Session 1: Introduction to Python & Variables

Class Programs:

1. Write a Python program to declare variables, perform arithmetic operations, and display results.
2. Write a Python program to calculate the area and perimeter of a rectangle using variables and operators.

Student Exercises:

- Write a Python program to swap two numbers using a temporary variable.
- Write a Python program to convert temperature from Celsius to Fahrenheit.

Session 2: Data Types, Input/Output & Control Flow

Class Programs:

1. Create a program to check if a number is even or odd using if-else.
2. Write a Python program to print the first n Fibonacci numbers using a for loop.
3. Demonstrate the use of break, continue, and pass in loops.

Student Exercises:

- Write a Python program to find the largest of three numbers.
- Write a Python program to calculate the sum of digits of a number using a while loop.
- Write a Python program to print the multiplication table of a number using a for loop.

Session 3: Functions

Class Programs:

1. Implement a program to generate prime numbers up to n using a generator function.
2. Write a Python program to calculate the factorial of a number using recursion.

Student Exercises:

- Write a Python function to find the square and cube of a number.
- Write a Python function that takes a string and returns it reversed.

Session 4: Strings

Class Programs:

1. Implement a program that accepts a string and counts the number of vowels and consonants.
2. Write a program to count the occurrences of each word in a given string.

Student Exercises:

- Write a Python program to check if a string is a palindrome.
- Write a Python program to remove all spaces from a string.
- Write a Python program to find the longest word in a sentence.

Session 5: Collections (Intro)

Class Programs:

1. Create a program to store student details in a dictionary and retrieve details based on user input.
2. Write a Python program to find the maximum and minimum in a list.

Student Exercises:

- Create a dictionary with employee details and perform CRUD operations.
- Write a Python program to count the frequency of elements in a list.
- Write a Python program to merge two dictionaries.
