

Weekly Teaching Schedule for IT-323 Python Programming

Course Title: Python Programming **Course Code:** IT-323 **Credit Hours:** 3(2-1)

Course Objectives: The purpose of this course is to provide a practical introduction to basic procedural programming skills using Python programming language from scratch.

Course Learning Outcomes (CLOs)

At the end of the course, you should be able to:

1. Understand the Basics of Python Programming

- Explain fundamental programming concepts.
- Set up and use Python development environments.
- Work with input, output, and variables of different data types.

2. Implement Control Structures and Loops

- Use decision-making structures (`if`, `if-else`, `if-elif-else`).
- Apply `for` and `while` loops for repetition.
- Utilize Boolean logic, relational, and arithmetic operators.

3. Work with Data Structures and String Manipulation

- Perform operations on strings (formatting, slicing, searching).
- Create and manipulate lists, tuples, dictionaries, and sets.
- Utilize list comprehension for efficient coding.

4. Develop Modular and Efficient Programs

- Define and call user-defined functions with appropriate arguments.
- Use lambda functions and store functions in modules.
- Work with NumPy arrays for numerical computations.

5. Handle Errors and Improve Code Reliability

- Identify and resolve programming errors.
- Implement exception handling (`try-except`).

Weekly Schedule of Classes

Week #1 Introduction (1 week)

- **Lecture #1:** Understanding computer programming, how a computer program works
- **Lecture #2:** Installing and using a Python development environment (Anaconda + Spyder, Jupyter, Google Colab, etc.)
- **Lecture #3:** Modes of using Python (Interactive and Scripting).

Week #2 Input, Processing, Output (3 weeks)

- **Lecture #1:** Understanding the importance of Python built-in functions and library functions,
- **Lecture #2:** Output statements for phrases,
- **Lecture #3:** Declaring and initializing variables of different data types,

Week #3

- **Lecture #1:** Concept of mutability, Introduction to errors and exceptions, Input statements,
- **Lecture #2:** Output statements for printing values, Dealing with variables of different data types,
- **Lecture #3:** Statements for arithmetic expressions and assignments, Basic memory concepts in Python,

Week #4

- **Lecture #1:** Importing libraries/modules and importing from libraries/modules, using library functions,
- **Lecture #2:** The math module.
- **Lecture #3:** Python's built-in data structures, Identity and Membership operators. (3 weeks)

Week #5 Decision (Selection Structures) and Boolean Logic (2 weeks)

- **Lecture #1:** Understanding of the flow control structures in computer programs, Relational operators and Boolean expressions
- **Lecture #2:** Single-Selection structure (using if statement), Double-Selection structure (using if-else statement), Multiple-Selection structure (using ifelif-else statement),
- **Lecture #3:** Multiple-Selection structure: (using multiple nested if-else statements),

Week #6

- **Lecture #1:** Logical or Boolean operators
- **Lecture #2:** Planning a computer program
- **Lecture #3:** Representations of algorithms: (pseudo-code, program source-code and flowchart).

Week #7 Count-Controlled Repetition Structures (1 week)

Lecture #1: Count-Controlled repetition using for **Lecture #2:** Count-Controlled repetition using while loops
Lecture #3: Generating sequences and sums

Week #8 Sentinel-Controlled Repetition Structures (1 week)

- **Lecture #1:** Sentinel-Controlled repetition using while Loop
- **Lecture #2:** break and continue statement
- **Lecture #3:** Sentinel-Controlled repetition using for loop

Week #9 Introduction to Strings (2 weeks)

- **Lecture #1:** String Manipulations
- **Lecture #2:** Formatting strings: Adjusting case
- **Lecture #3:** Formatting strings: Adding and removing spaces

Week #10

- **Lecture #1:** Formatting strings: Finding and replacing substrings
- **Lecture #2:** Formatting strings: Splitting
- **Lecture #3:** Formatting strings: Partitioning strings

Week #11 User-Defined Functions (2 weeks)

- **Lecture #1:** Defining and calling user-defined functions, Value-returning functions,
- **Lecture #2:** Local variables,
- **Lecture #3:** Global variables and constants

Week #12

- **Lecture #1:** Insights into passing arguments to functions, Default arguments, Flexible arguments,
- **Lecture #2:** Anonymous (lambda) function
- **Lecture #3:** Storing functions in modules

Week #13: Lists and Tuples (2 weeks)

- **Lecture #1:** Creating 1D and 2D lists/tuples, Indexing
- **Lecture #2:** List: Slicing, Operations
- **Lecture #3:** Tuple: Slicing, Operations, Deep copy versus shallow copy,

Week #14

- **Lecture #1:** Lists/tuples as function arguments
- **Lecture #2:** Explicit traversing in lists
- **Lecture #3:** List comprehension

Week #15: Introduction to Miscellaneous Topics (1 week)

- **Lecture #1:** NumPy Arrays
- **Lecture #2:** Catching Exceptions (try and except)
- **Lecture #3:** Creating Dictionaries and Sets.