

UNIT I

**Chapter 1 : Introduction and Syntax of Python Program
1-1 to 1-17**

Syllabus : Features of Python - Interactive, Object -oriented, Interpreted, platform independent. Python building blocks - Identifiers, Keywords, Indentation, Variables, Comments. Python environment setup - Installation and working of IDE. Running Simple Python scripts to display 'welcome' message. Python Data Types : Numbers, String, Tuples, Lists, Dictionary. Declaration and use of data types.

| | | |
|-------|---|------|
| 1.1 | Introduction to Python..... | 1-1 |
| 1.2 | Features of Python | 1-2 |
| 1.2.1 | Python's "Everything is Object" Concept OR Python is Dynamic Typed Language | 1-2 |
| 1.2.2 | Python's Smart Memory Management..... | 1-3 |
| 1.3 | Python Building Blocks | 1-4 |
| 1.3.1 | Python Identifiers..... | 1-4 |
| 1.3.2 | Python Keywords..... | 1-4 |
| 1.3.3 | Python Indentation..... | 1-5 |
| 1.3.4 | Python Variables | 1-5 |
| 1.3.5 | Comments in Python | 1-6 |
| 1.4 | Python Environment Setup - Installation and Working of IDE | 1-7 |
| 1.5 | Running Simple Python Scripts to Display 'Welcome' Message..... | 1-11 |
| 1.6 | Python Data Types..... | 1-13 |
| 1.6.1 | Python Numbers..... | 1-13 |
| 1.6.2 | Python String..... | 1-14 |
| 1.6.3 | Python List | 1-15 |
| 1.6.4 | Python Tuple | 1-16 |
| 1.6.5 | Python Dictionary | 1-17 |

UNIT II

**Chapter 2 : Python Operators and Control Flow Statements
2-1 to 2-24**

Syllabus : Basic Operators : Arithmetic, Comparison/ Relational, Assignment, Logical, Bitwise, Membership, Identity operators, Python Operator Precedence. Control Flow. Conditional Statements (if, if ... else, nested if). Looping in python (while loop, for loop, nested loops). Loop manipulation using continue, pass, break, else.

| | | |
|-------|---------------------------|-----|
| 2.1 | Basic Operators..... | 2-1 |
| 2.1.1 | Arithmetic Operators..... | 2-1 |

| | | |
|-------|--|------|
| 2.1.2 | Comparison/Relational Operators..... | 2-2 |
| 2.1.3 | Assignment Operator..... | 2-3 |
| 2.1.4 | Logical Operators | 2-4 |
| 2.1.5 | Bitwise Operators | 2-5 |
| 2.1.6 | Membership Operators..... | 2-6 |
| 2.1.7 | Identity Operators..... | 2-7 |
| 2.1.8 | Python Operator Precedence | 2-7 |
| 2.1.9 | range() Function | 2-8 |
| 2.2 | Control Flow Statements | 2-9 |
| 2.3 | Conditional Statements OR Decision Making Statements..... | 2-10 |
| 2.3.1 | The if Statement | 2-10 |
| 2.3.2 | The if - else statement | 2-11 |
| 2.3.3 | The if - elif - else Statement OR The else - if Statement..... | 2-12 |
| 2.3.4 | The nested - if Statement | 2-13 |
| 2.3.5 | The Short-hand if-else Statement | 2-14 |
| 2.4 | Looping in Python..... | 2-15 |
| 2.4.1 | The while Loop | 2-15 |
| 2.4.2 | The for Loop..... | 2-16 |
| 2.5 | Loop Manipulation | 2-17 |
| 2.5.1 | Loop Manipulation using Keyword "continue"..... | 2-17 |
| 2.5.2 | Loop Manipulation using Keyword "break" | 2-19 |
| 2.5.3 | Loop Manipulation using Keyword "pass" | 2-20 |
| 2.5.4 | Loop Manipulation using Keyword "else" | 2-20 |
| 2.6 | Nested Loops | 2-22 |

UNIT III

Chapter 3 : Data Structures in Python 3-1 to 3-38

Syllabus : **Lists :** (a) Defining lists, accessing values in list, deleting values in list, updating lists. (b) Basic List Operations (c) Built – in List functions. **Tuples :** (a) Accessing values in Tuples, deleting values in Tuples, and updating Tuples (b) Basic Tuple operations. (c) Built – in Tuple functions. **Sets :** (a) Accessing values in Set, deleting values in Set and updating Sets. (b) Basic Set operations. (c) Built – in Set functions **Dictionaries :** (a) Accessing values in Dictionary, deleting values in Dictionary and updating Dictionary. (b) Basic Dictionary operations. (c) Built – in Dictionaries functions.

| | | |
|-------|---|-----|
| 3.1 | List..... | 3-1 |
| 3.1.1 | Declaring List OR Creating List OR Defining List..... | 3-1 |



| | | | | | |
|---------|---|------|---------|--|------|
| 3.1.2 | Accessing List Elements using Index Numbers..... | 3-2 | 3.3.4.4 | Set Union | 3-27 |
| 3.1.3 | Negative List Indexing | 3-3 | 3.3.4.5 | Set Intersection | 3-27 |
| 3.1.4 | Iterating through List using for-loop..... | 3-3 | 3.3.4.6 | Set Difference..... | 3-28 |
| 3.1.5 | Slicing Lists (Using Slicing Operator (:) on List) | 3-4 | 3.3.4.7 | Set Symmetric_Difference | 3-28 |
| 3.1.6 | Basic List Operations..... | 3-5 | 3.3.5 | Additional Set Functions | 3-29 |
| 3.1.6.1 | Adding element : append(), insert() and extend() Functions..... | 3-5 | 3.3.6 | Deleting Set..... | 3-31 |
| 3.1.6.2 | Finding Length of List : using __len__ () Function.... | 3-8 | 3.4 | Dictionaries | 3-31 |
| 3.1.6.3 | Updating Element : Using Assignment Operator (=) to over-write Value | 3-8 | 3.4.1 | Creating Dictionary OR Declaring Dictionary OR Defining Dictionary | 3-31 |
| 3.1.6.4 | Delete or Remove Element from List | 3-9 | 3.4.2 | Accessing Dictionary Elements using Index Numbers..... | 3-31 |
| 3.1.6.5 | Testing Membership Operators on List | 3-10 | 3.4.3 | Iterating through List using for-loop..... | 3-32 |
| 3.1.7 | Built-in List Functions | 3-11 | 3.4.4 | The keys() and values() Functions..... | 3-33 |
| 3.1.8 | Nested List (Multi-Dimensional List)..... | 3-12 | 3.4.5 | Basic Dictionary Operations | 3-33 |
| 3.1.9 | Deleting List..... | 3-15 | 3.4.5.1 | Change or Update Elements in Dictionary : update() Function | 3-33 |
| 3.2 | Tuples | 3-16 | 3.4.5.2 | Finding Length of List : using __len__ () Function | 3-34 |
| 3.2.1 | Declaring Tuple OR Declaring Tuple | 3-16 | 3.4.5.3 | Updating Dictionary : Using Assignment Operator (=) to over-write Value | 3-35 |
| 3.2.2 | Accessing Tuple Elements using Index Numbers | 3-17 | 3.4.5.4 | Remove Element : pop(), popitem() and clear() Functions..... | 3-35 |
| 3.2.3 | Negative Tuple indexing | 3-18 | 3.4.6 | Built-in Dictionary Functions | 3-36 |
| 3.2.4 | Iterating through Tuple using for-loop | 3-18 | 3.4.7 | Deleting Dictionary | 3-37 |
| 3.2.5 | Slicing Tuple..... | 3-19 | | | |
| 3.2.6 | Basic Tuple Operations | 3-20 | | | |
| 3.2.6.1 | Tuples are Immutable..... | 3-20 | | | |
| 3.2.6.2 | Finding Length of List : use of __len__ () Function | 3-20 | | | |
| 3.2.6.3 | Testing Membership Operators on Tuple..... | 3-21 | | | |
| 3.2.6.4 | Concatenation of Tuple..... | 3-21 | | | |
| 3.2.6.5 | Multiplying Tuple..... | 3-21 | | | |
| 3.2.7 | Built-in Tuple Functions | 3-22 | | | |
| 3.2.8 | Deleting Tuple | 3-22 | | | |
| 3.3 | Set..... | 3-23 | | | |
| 3.3.1 | Declaring Set OR Defining Set OR Creating Set..... | 3-23 | | | |
| 3.3.2 | 'Set' is not Subscriptable OR No indexing to 'Set' Elements OR 'Set' Object does not Support Indexing | 3-24 | | | |
| 3.3.3 | Iterating through Set using for-loop..... | 3-24 | | | |
| 3.3.4 | Basic Set Operations..... | 3-25 | | | |
| 3.3.4.1 | Adding Element : update() and add() Functions..... | 3-25 | | | |
| 3.3.4.2 | Finding Length of Set : using __len__ () Function | 3-25 | | | |
| 3.3.4.3 | Removing Element : discard(), remove(), pop() and clear() Functions..... | 3-25 | | | |

UNIT IV

Chapter 4 : Python Functions, Modules and Packages

4-1 to 4-57

Syllabus : Use of Python built : in functions (e.g. type/ data conversion functions, math functions etc.). User defined functions : Function definition, Function calling, function arguments and parameter passing, Return statement, Scope of Variables: Global variable and Local Variable. Modules : Writing modules, importing modules, importing objects from modules, Python built – in modules (e.g. Numeric and mathematical module, Functional Programming Module) Namespace and Scoping. Python Packages: Introduction, Writing Python packages, Using standard (e.g. math, scipy, Numpy, matplotlib, pandas etc.) and user defined packages.

| | | |
|-------|------------------------------|-----|
| 4.1 | User Defined Functions | 4-1 |
| 4.1.1 | Function Definition..... | 4-1 |
| 4.1.2 | Function Calling..... | 4-2 |



| | | |
|---------|---|------|
| 4.1.3 | Function Arguments and Parameter Passing..... | 4-3 |
| 4.1.4 | Return Statement | 4-6 |
| 4.1.5 | Returning Multiple Values..... | 4-8 |
| 4.1.6 | Scope of Variables | 4-10 |
| 4.1.7 | Function with Default Arguments/Parameters OR Functions with Optional Arguments/Parameters..... | 4-12 |
| 4.1.8 | Keyword Arguments OR kwargs in Python | 4-14 |
| 4.1.9 | Nested Function is allowed in Python OR Inner Functions | 4-15 |
| 4.1.10 | Python Function's Nature : Pass by Value or Pass by Reference..... | 4-15 |
| 4.1.11 | Variable Length Arguments | 4-16 |
| 4.2 | Python Built-in Functions | 4-19 |
| 4.2.1 | Datatype and Object Related Functions..... | 4-20 |
| 4.2.2 | Data Conversion Functions..... | 4-23 |
| 4.2.3 | Input-Output Functions | 4-28 |
| 4.2.4 | Files and Expression Related | 4-29 |
| 4.2.5 | Class and Inheritance Related..... | 4-29 |
| 4.2.6 | Mathematical Functions..... | 4-30 |
| 4.2.7 | for built-in Help | 4-31 |
| 4.3 | Python Modules..... | 4-33 |
| 4.3.1 | Writing Modules..... | 4-33 |
| 4.3.2 | Importing Modules | 4-34 |
| 4.3.2.1 | Simple import Statement | 4-34 |
| 4.3.2.2 | Import with Renaming..... | 4-34 |
| 4.3.2.3 | The from...import Statement OR Importing Objects from Modules OR Importing Individual Object..... | 4-35 |
| 4.3.2.4 | Import everything using * | 4-35 |
| 4.3.2.5 | Importing Multiple Modules | 4-35 |
| 4.3.3 | Python built-in Modules | 4-35 |
| 4.3.3.1 | Numeric and Mathematical Modules..... | 4-36 |
| 4.3.3.2 | Functional Programming Module | 4-45 |
| 4.3.5 | Namespace and Scoping..... | 4-49 |
| 4.4 | Python Packages..... | 4-50 |
| 4.4.1 | Writing Python Packages..... | 4-50 |
| 4.4.2 | Using Standard and user-defined Packages..... | 4-54 |

UNIT V**Chapter 5 : Object Oriented Programming in Python****5-1 to 5-28**

Syllabus : Creating Classes and Objects. Method Overloading and Overriding. Data Hiding. Data abstraction. Inheritance and composition classes. Customization via inheritance specializing inherited methods.

| | | |
|---------|--|------|
| 5.1 | Creating Classes and Objects | 5-1 |
| 5.1.1 | Creating Class..... | 5-1 |
| 5.1.2 | Declaring Object..... | 5-2 |
| 5.1.3 | The 'Self' Parameter..... | 5-3 |
| 5.1.4 | Create Instant Attributes of Object OR Externally Modifying Object's Properties | 5-5 |
| 5.1.5 | Deleting an Attribute | 5-7 |
| 5.1.6 | Defining Constructor OR The <code>__init__()</code> Method | 5-7 |
| 5.1.6.1 | Default Constructor..... | 5-8 |
| 5.1.6.2 | Parameterized Constructor | 5-9 |
| 5.1.6.3 | Constructor Overloading is not allowed in Python | 5-9 |
| 5.1.6.4 | Constructor with Default Arguments | 5-10 |
| 5.1.7 | Defining destructor OR The <code>__del__()</code> Method..... | 5-11 |
| 5.1.8 | "Can Manually Call Constructor and Destructor..?" | 5-11 |
| 5.1.9 | Deleting an Object..... | 5-12 |
| 5.1.10 | The Pass Statement..... | 5-13 |
| 5.2 | Data Hiding | 5-13 |
| 5.3 | Inheritance and Composition Classes..... | 5-15 |
| 5.3.1 | Single Inheritance..... | 5-16 |
| 5.3.2 | Constructor and Destructor in Inheritance..... | 5-16 |
| 5.3.3 | Parameterized Constructors in Base Class and Derived Class | 5-18 |
| 5.3.4 | Multiple Inheritance | 5-18 |
| 5.3.5 | Multilevel Inheritance..... | 5-20 |
| 5.4 | Data Abstraction | 5-21 |
| 5.5 | Method Overloading and Overriding | 5-23 |
| 5.5.1 | Method Overloading | 5-23 |
| 5.5.2 | Method Overriding..... | 5-25 |
| 5.5.3 | Use of Super to Access Members of Base Class | 5-26 |
| 5.6 | Customization via Inheritance Specializing Inherited Methods | 5-27 |

**UNIT VI****Chapter 6 : File I/O Handling and Exception Handling****6-1 to 6-28**

Syllabus : I/O Operations : Reading keyboard input, Printing to screen. File Handling : Opening file in different modes, accessing file contents using standard library functions, Reading and writing files, closing a file, Renaming and deleting files, Directories in Python, File and directory related standard functions. Exception Handling : Introduction, Exception handling - 'try: except:' statement, 'raise' statement, User defined exceptions.

| | | | | | |
|---------|---|-----|---------|---|-------------|
| 6.1 | I/O operations..... | 6-1 | 6.2.4 | Closing a File..... | 6-11 |
| 6.1.1 | Reading Input..... | 6-1 | 6.2.5 | Directories in Python | 6-11 |
| 6.1.2 | Printing to Screen..... | 6-2 | 6.2.6 | File and Directory Related Standard Functions..... | 6-11 |
| 6.1.3 | print() Function's Keyword Arguments | 6-2 | 6.2.6.1 | Getting Current Directory..... | 6-11 |
| 6.1.4 | Output Formatting OR Formatted String Output..... | 6-3 | 6.2.6.2 | Changing Directory..... | 6-12 |
| 6.1.5 | String Modulo Operator (%)..... | 6-3 | 6.2.6.3 | List all Directories and Files..... | 6-12 |
| 6.2 | File Handling | 6-4 | 6.2.6.4 | Creating New Directory..... | 6-12 |
| 6.2.1 | Opening File in Different Modes | 6-4 | 6.2.6.5 | Renaming a Directory or File | 6-13 |
| 6.2.2 | Accessing File Contents using Standard Library Functions..... | 6-5 | 6.2.6.6 | Removing a Directory of File..... | 6-13 |
| 6.2.3 | Reading and Writing Files..... | 6-6 | 6.2.6.7 | Checking Existence of Directory or File | 6-14 |
| 6.2.3.1 | Writing into File using write() Function | 6-6 | 6.2.6.8 | Checking Whether Directory of File | 6-14 |
| 6.2.3.2 | Writing into File using writelines() Function | 6-6 | 6.2.7 | Renaming and Deleting Files..... | 6-14 |
| 6.2.3.3 | Reading File using read() Function | 6-7 | 6.3 | Exception Handling..... | 6-15 |
| 6.2.3.4 | Reading from File using readline() Function..... | 6-7 | 6.3.1 | Introduction | 6-15 |
| 6.2.3.5 | Reading from File using readlines() Function..... | 6-8 | 6.3.2 | Exception Handling - <i>try: except</i> Statement | 6-17 |
| 6.2.3.6 | Reading File using for-loop..... | 6-8 | 6.3.2.1 | try:except Statement..... | 6-17 |
| 6.2.3.7 | File Pointer's Manipulation : seek() and tell() Functions | 6-8 | 6.3.2.2 | try with Multiple Except Blocks..... | 6-18 |
| | | | 6.3.2.3 | try: except:else Blocks..... | 6-19 |
| | | | 6.3.2.4 | except Block without any Exception Name OR Generic except:block | 6-20 |
| | | | 6.3.2.5 | except Block with Multiple Exceptions..... | 6-21 |
| | | | 6.3.2.6 | try: finally Blocks..... | 6-22 |
| | | | 6.3.2.7 | try:except:finally Blocks | 6-23 |
| | | | 6.3.3 | The Raise Statement OR Manually Raising Exception..... | 6-24 |
| | | | 6.3.4 | User Defined Exceptions | 6-25 |
| | | | | • List of Practicals..... | L-1 to L-19 |
| | | | | • Time Management Sheet | |
| | | | | • Model Question Papers | Q-1 to Q-4 |



List of Practicals

| Program No. | Program Statement | Page No. |
|-------------|---|----------|
| 1. | Install and configure Python IDE. | L-1 |
| 2. | Write simple Python program to display message on screen. | L-1 |
| 3. | Write simple Python program using operators : (a) Arithmetic Operators (b) Logical Operators (c) Bitwise Operators | L-1 |
| 4. | Write simple Python program to demonstrate use of conditional statements : (a) 'if' statement (b) 'if ... else' statement (c) Nested 'if' statement | L-3 |
| 5. | Write python program to demonstrate use of looping statements: (a) 'while' loop (b) 'for' loop (c) Nested loops | L-3 |
| 6. | Write python program to perform following operations on Lists : (a) Create list (b) Access list (c) Update list (Add item, Remove item) (d) Delete list | L-4 |
| 7. | Write python program to perform following operations on Tuples: (a) Create Tuple (b) Access Tuple (c) Update Tuple (d) Delete Tuple | L-6 |
| 8. | Write python program to perform following operations on Tuples: (a) Create Set (b) Access Set elements (c) Update Set (d) Delete Set | L-6 |
| 9. | Write python program to perform following operations on Dictionaries: (a) Create Dictionary (b) Access Dictionary elements (c) Update Dictionary (d) Delete Set (e) Looping through Dictionary | L-7 |
| 10. | (a) Write Python program to demonstrate math built- in functions (Any 2 programs) (b) Write Python program to demonstrate string built - in functions (Any 2 programs) | L-8 |



| Program No. | Program Statement | Page No. |
|-------------|---|----------|
| 11. | Develop user defined Python function for given problem: (a) Function with minimum 2 arguments (b) Function returning values | L-10 |
| 12. | Write Python program to demonstrate use of: (a) Built-in module (e.g. keyword, math, number, operator) (b) user defined module. | L-12 |
| 13. | Write Python program to demonstrate use of: (a) built-in packages (e.g. NumPy, Pandas) (b) user defined packages | L-14 |
| 14. | Write a program in Python to demonstrate following operations: (a) Method overloading (b) Method overriding | L-16 |
| 15. | Write a program in Python to demonstrate following operations: (a) Simple inheritance (b) Multiple inheritance | L-17 |
| 16. | Write a program in Python to handle user defined exception for given problem | L-19 |