

**PREREQUISITES:** A course on “Programming for Problem Solving”.

**COURSE OBJECTIVES:**

- Learn Syntax and Semantics and create Functions in Python.
- Handle Strings and Files in Python.
- Understand the Lists, Dictionaries and Recursion in Python.
- Implement Object Oriented Programming concepts in Python.
- Build and introduction to GUI based programs in Python.

**COURSE OUTCOMES:**

- CO1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
- CO2: Demonstrate proficiency in handling Strings and File Systems.
- CO3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and Recursion.
- CO4: Interpret the concepts of Object-Oriented Programming as used in Python.
- CO5: Implement exemplary applications related to GUI based programs in Python.

**UNIT- I**

**Introduction to Python,** Comments, Variables, Reading Input from the Keyboard, Performing Calculations, Operators. Type conversions, Expressions, Data types. loops: if, if-else, else-if Statements, Nested Decision Structures, Comparing Strings. Repetition Structures: Introduction, while loop, for loop, calculating a Running Total, Input Validation Loops, Nested Loops. Data types and Expressions: Strings, Assignment and Comments, Numeric Data Types and Character Sets, Expressions, Functions and Modules.

**UNIT- II**

**Control Statements:** Definite Iteration, Formatting Text for Output, Selection, Conditional Iteration. **File and Exceptions:** Introduction to File Input and Output, Using Loops to Process Files, Processing Records, Exceptions. **Functions:** Introduction, Defining and Calling a Void Function, designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning **Functions**-Generating Random Numbers, the math Module, Storing Functions in Modules.

**UNIT – III**

**Strings and Text Files:** Accessing Characters and Substrings in a String, Strings and Number System, String Methods, Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings. Text Files, Data Encryption, Lists, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List Methods and Useful Built-in Functions, Copying Lists, Processing Lists, Two-Dimensional Lists, Tuples Sequences, Tuples. Dictionaries and Sets: Dictionaries, Sets, Serializing Objects. Recursion: Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms.

## **UNIT – IV**

**Design with Classes:** Classes and Objects, Classes and Functions, Classes and Methods, Working with Instances, Inheritance and Polymorphism. Object-Oriented Programming: Procedural and Object-Oriented Programming, Classes, techniques for Designing Classes.

## **UNIT – V**

**Graphical User Interfaces:** Behavior of terminal based programs and GUI-based programs, Coding simple GUI-based programs, other useful GUI resources. GUI Programming: Graphical User Interfaces, Using the tkinter Module, display text with Label Widgets, Organizing Widgets with Frames, Button Widgets and Info Dialog Boxes, Getting Input with Entry Widget, Using Labels as Output Fields, Radio Buttons, Check Buttons. Simple Graphics and Image Processing: Overview of Turtle Graphics, Two dimensional Shapes, Colors and RGB System, Image Processing.

### **TEXT BOOKS:**

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning.
2. Allen B. Downey, Think Python First Edition, by, Orielly publishing

### **REFERENCE BOOKS:**

1. John V. Guttag, Introduction to Computation and Programming Using Python, The MIT Press.
2. James Payne, Beginning Python using Python 2.6 and Python 3, Wrox publishing
3. Paul Gries, Practical Programming: An Introduction to Computer Science using Python 3, The Pragmatic Bookshelf, 2nd edition (4 Oct. 2013)
4. Charles Dierach, Introduction to Computer Science using Python