**GE8151 PROBLEM SOLVING AND PYTHON PROGRAMMING L T P C 3 0 0 3**

**OBJECTIVES:**

 To know the basics of algorithmic problem solving

 To read and write simple Python programs.

 To develop Python programs with conditionals and loops.

 To define Python functions and call them.

 To use Python data structures –- lists, tuples, dictionaries.

 To do input/output with files in Python.

**UNIT I ALGORITHMIC PROBLEM SOLVING 9**

Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

**UNIT II DATA, EXPRESSIONS, STATEMENTS 9**

Python interpreter and interactive mode; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

**UNIT III CONTROL FLOW, FUNCTIONS 9**

Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

**UNIT IV LISTS, TUPLES, DICTIONARIES 9**

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, mergesort, histogram.

**UNIT V FILES, MODULES, PACKAGES 9**

Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file.

**TOTAL : 45 PERIODS OUTCOMES:**

**Upon completion of the course, students will be able to**

 Develop algorithmic solutions to simple computational problems

 Read, write, execute by hand simple Python programs.

 Structure simple Python programs for solving problems.

 Decompose a Python program into functions.

 Represent compound data using Python lists, tuples, dictionaries.

 Read and write data from/to files in Python Programs.

**TEXT BOOKS:**

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist‘‘, 2nd edition, Updated for Python 3, Shroff/O‘Reilly Publishers, 2016 (http://greenteapress.com/wp/think-python/)

2. Guido van Rossum and Fred L. Drake Jr, ―An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.

**REFERENCES:**

1. John V Guttag, ―Introduction to Computation and Programming Using Python‘‘, Revised and expanded Edition, MIT Press , 2013

2. Robert Sedgewick, Kevin Wayne, Robert Dondero, ―Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.

3. Timothy A. Budd, ―Exploring Python‖, Mc-Graw Hill Education (India) Private Ltd.,, 2015.

4. Kenneth A. Lambert, ―Fundamentals of Python: First Programs‖, CENGAGE Learning, 2012.

5. Charles Dierbach, ―Introduction to Computer Science using Python: A Computational Problem-Solving Focus, Wiley India Edition, 2013.

6. Paul Gries, Jennifer Campbell and Jason Montojo, ―Practical Programming: An Introduction to Computer Science using Python 3‖, Second edition, Pragmatic Programmers, LLC, 2013.

**GE8161 PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY L T P C 0 0 4 2 OBJECTIVES:**

 To write, test, and debug simple Python programs.

 To implement Python programs with conditionals and loops.

 Use functions for structuring Python programs.

 Represent compound data using Python lists, tuples, dictionaries.

 Read and write data from/to files in Python.

**LIST OF PROGRAMS:**

1. Compute the GCD of two numbers.

2. Find the square root of a number (Newton‘s method)

3. Exponentiation (power of a number)

16

4. Find the maximum of a list of numbers

5. Linear search and Binary search

6. Selection sort, Insertion sort

7. Merge sort

8. First n prime numbers

9. Multiply matrices

10. Programs that take command line arguments (word count)

11. Find the most frequent words in a text read from a file

12. Simulate elliptical orbits in Pygame

13. Simulate bouncing ball using Pygame

**PLATFORM NEEDED** Python 3 interpreter for Windows/Linux **TOTAL: 60 PERIODS OUTCOMES: Upon completion of the course, students will be able to:**

 Write, test, and debug simple Python programs.

 Implement Python programs with conditionals and loops.

 Develop Python programs step-wise by defining functions and calling them.

 Use Python lists, tuples, dictionaries for representing compound data.

 Read and write data from/to files in Python.