Third Year B.C.A. (Under Science) Semester VI

Course Code: BCA602 **Course Title: Python Programming**
Total Contact Hours: 48 hrs. **Total Credits: 04** **Total Marks: 100**
(60 Lectures)

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives:

- To introduce various concepts of programming to the students using Python.
- Students should be able to apply the problem solving skills using Python

Unit No.	Contents	No. of Lectures
Unit 1	**Introduction to Python Scripting** • Why Scripting is Useful in Computational Science • Classification of Programming Languages • Productive Pairs of Programming Languages • Gluing Existing Applications • Scripting Yields Shorter Code, Efficiency • Type-Specification (Declaration) of Variables • Flexible Function Interfaces • Interactive Computing • Creating Code at Run Time • Nested Heterogeneous Data Structures • GUI Programming • Mixed Language Programming • When to Choose a Dynamically Typed Language Why Python? Script or Program? • Application of Python • Concept (immutable)	**04**

Unit 2	**Basic Python** • Python identifiers and reserved words • Lines and indentation, multi-line statements • Comments • Input/output with print and input functions, • Command line arguments and processing command line arguments • Standard data types - basic, none, Boolean (true & False), numbers • Python strings • Data type conversion • Python basic operators (Arithmetic, comparison, assignment, bitwise logical) • Python membership operators (in & not in) • Python identity operators (is & is not) • Operator precedence • Control Statements, Python loops, Iterating by subsequence index, loop control statements (break, continue, pass) □ Mathematical functions and constants (import math), Random number functions	**06**
Unit 3	**Python strings** • Concept, escape characters • String special operations • String formatting operator • Single quotes, Double quotes, Triple quotes • Raw String, Unicode strings, Built-in String methods. • Python Lists - concept, creating and accessing elements, updating & deleting lists, basic list operations, reverse • Indexing, slicing and Matrices • built-in List functions • Functional programming tools - filter(), map(), and reduce() • Using Lists as stacks and Queues, List comprehensions	**06**
Unit 4	**Python tuples and sets** • Creating & deleting tuples • Accessing values in a tuple • Updating tuples, delete tuple elements • Basic tuple operations • Indexing, slicing and Matrices, built- in tuple functions. • Sets - Concept, operations.	**06**

Unit 5	**Python Dictionary** • Concept (mutable) • Creating and accessing values in a dictionary • Updating dictionary, delete dictionary elements • Properties of dictionary keys • built-in dictionary functions and methods.	**04**
Unit 6	**Functions** • Defining a function (def) • Calling a function • Function arguments - Pass by value, Keyword Arguments, default arguments • Scope of variable - basic rules • Documentation Strings • Variable Number of Arguments • Call by Reference • Order of arguments (positional, extra & keyword) • Anonymous functions • Recursion • Treatment of Input and Output Arguments • Unpacking argument lists • Lambda forms • Function Objects • function ducktyping & polymorphism	**08**
	□ Generators (functions and expressions) and iterators, list comprehensions	
Unit 7	**Files and Directories** • Creating files • Operations on files (open, close, read, write) • File object attributes, file positions, Listing Files in a Directory • Testing File Types • Removing Files and Directories • Copying and Renaming Files • Splitting Pathnames • Creating and Moving to Directories • Traversing Directory Trees • Illustrative programs: word count, copy file	**06**

Unit 8	**Python Classes / Objects** • Object oriented programming and classes in Python - creating classes, instance objects, accessing members • Data hiding (the double underscore prefix) • Built-in class attributes • Garbage collection : the constructor • Overloading methods and operators • Inheritance - implementing a subclass, overriding methods • Recursive calls to methods • Class variables, class methods, and static methods	**08**
Unit 9	**Python Exceptions** • Exception handling : assert statement • Except clause - with no exceptions and multiple exceptions □ Try - finally, raising exceptions, user-defined exceptions.	**02**

Reference Books:

1. Introducing Python- Modern Computing in Simple Packages – Bill Lubanovic, O„Reilly Publication
2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress
3. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries, et al., Pragmatic Bookshelf, 2/E 2014
4. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication

   Learning with Python ", Green Tea Press, 2002
5. E-Books : python_tutorial. pdf, python_book_01.pdf
6. Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller
7. A Beginner‟s Python Tutorial: http://en.wikibooks.org/wiki/A Beginner%27s Python Tutorial.