Year & Sem: P2S2	Course Code: IT407	Course Name: Introduction to Python Programming Language	No. of Credits: 3	L-T-P: 1-1-1

COURSE DESCRIPTION:
The purpose of the course is to provide the Basic programming methodology and writing programs in python

LEARNING OUTCOMES:
A student receives a basic knowledge about fundamentals of programming methodology.
- To understand the various steps in Program development.
- To understand the basic concepts in Python Programming Language.
- To learn how to write programs in interactive mode and script mode
- To get knowledge how to use operators
- To learn to write programs using conditional, loop statements etc..
- To make the student understand simple sorting and searching methods.
- To get basic knowledge on Functions

PREREQUISITE: Algorithms and Flow Charts (PUC – II, Semester -1)
Students should be able to learn before some basic fundamental programming language (but not mandatory). Some mathematical maturity also will be expected; students should have some idea of what constitutes how to write one. Some knowledge of basic probability will also be helpful.

COURSE CONTENT:
Unit-1:
Getting Started: Introduction to Python - an integrated high level language, interactive mode and script mode. Data types –Number (Integer - boolean, decimal, octal, hexadecimal; Floating point; Complex), none, Sequence (String, Tuples, List) Sets, Mapping.
Variables, Expressions and Statements: Values, Variables and keywords; Operators and Operands in Python: (Arithmetic, relational and logical operators), operator precedence, Expressions and Statements (Assignment statement); Taking input (using raw_input() and input()) and displaying output (print statement); Putting Comments.

Unit-2:
Predefine Functions: Importing Modules (entire module or selected objects), invoking built in functions, functions from math module (for example, ceil, floor, fabs, exp, log, log10, pow, sqrt, cos, sin, tan, degrees, radians), using random() and randint() functions of random module to generate random numbers, composition.
Conditional constructs and looping: if else statement while, for (range function), break, continue, else, pass, nested if, nested loops, use of compound expression in conditional and looping construct.

Unit-3:
Functions: Defining functions, invoking functions, passing parameters (default parameter values, keyword arguments), scope of variables, void functions and functions returning values, flow of execution

Strings: Creating, initialising and accessing the elements; string operators: +, *, in, not in, range slice [n:m]; comparing strings using relational operators; String functions & methods: len, capitalize, find, isalnum, isalpha, isdigit, lower, islower, isupper, upper, lstrip, rstrip, isspace, istitile, partition, replace, join, split, count, decode, encode, swapcase, String constants, Regular Expressions and Pattern Matching

Unit-4:
Lists: Concept of mutable lists, creating, initializing and accessing the elements, traversing, appending, updating and deleting elements, composition, lists as arguments, List operations –

joining, slicing, + , * , in , not in, List functions and methods - len( ), insert( ), append( ), extend( ), sort( ), remove( ), reverse( ), pop( ), list( ), count( ), extend( ), index( ), cmp( ), max( ), min( )

Unit-5:
Dictionaries: Concept of key-value pair, creating, initialising and accessing the elements in a dictionary, traversing, appending updating and deleting elements; Dictionary Functions and methods-cmp( ), len( ), clear( ), get( ), has_key( ), items( ), key( ), update( ), values( ), pop( ), fromkeys( ), dict( )

Unit-6:
Tuples: Immutable concept, creating, initialising and accessing elements in a tuple, Tuple assignment, Tuple slices, Tuple indexing, Tuple Functions- cmp(), len(), max(), min(), tuple(), index(), count(), sum(), any(), all(), sorted(), reversed()
Opening and reading files: Open text file-opening a text file-syntax-Different modes of opening a file-The file object attributes-Reading from text file-Introducing the Read it program-Example-Reading characters from a line-Reading a line from file-Problem set

COURSE MATERIALS:
Content: Available videos and Reading material in RGUKT Content Server

Python PROGRAMMING LAB:
Objectives:
- To write programs in C to solve the problems.
- To implement linear data structures such as lists, stacks, queues.
- To implement simple searching and sorting methods**.**

Recommended Systems/Software Requirements:
- Intel based desktop PC
- ANSI Python Application with Supporting Editors

Experiments:
Week-1:
1. To evaluate algebraic exp(ax+b)/(ax-b)
2. To Evaluate algebraic exp 2.5logx+cos32+x*x-y*y+sqrt(2*x*y)

3. To evaluate algebraic exp x power5 +10 x power 4+8 x power3+4x+2

Week – 2:
4. To swap 2 no using 3rd variable without using 3rd variable