COURSE NAME

GENERIC ELECTIVE-PYTHON PROGRAMMING

Credits: 2
Course Code:G23DS1T

Semester:III
No of Lecture Hours:30

Course Objective:

- To offer an easy syntax compared to Perl and the Unix/Linux "Shell Languages" and it is easier to learn and maintain.

Course Outcome:

CO1: **Implement** the structure and components of a Python program.
CO2: **Express** how to write loops and decision statements in Python.
CO3: **Interpret** how to write functions and pass arguments in Python.
CO4:**Explain** how to build Python packages module for reusability
CO5: **Create** files and GUI programs

Course Content	Hours Allotted	Pedagogy
Module I: • **Introduction to Python:** Python, Features of Python, Execution of a Python Program, Memory Management in Python, Comparisons between C and Python, Comparisons between Java and Python. • **Writing Our First Python Program:** Writing Our First Python Program, Executing a Python Program - Using Python's Command, Line Window, and Using Python's IDLE Graphics Window • **Data types in Python, Operators in python**	**(6)** 2 1 3	PowerPoint presentations / Lectures
Module II: Title: • **Control Statements:** The if Statement, A Word on Indentation, The if… else Statement, The if … elif … else Statement, The while Loop, The for Loop, Infinite Loops, Nested Loops, The else Suite, The break Statement, The continue Statement, The pass Statement, The assert Statement, The return Statement. • **Lists and Tuples:** List, Creating Lists using range() Function, Updating the Elements of a List, Concatenation of Two Lists, Repetition of Lists, Membership in Lists, Aliasing and Cloning Lists, Methods to Process Lists, Finding Biggest and Smallest Elements in a List, Sorting the List Elements, Number of Occurrences of an Element in the List.	**(6)** 2 4	Case Studies / Review of research articles

Tuples: Creating Tuples, Accessing the Tuple Elements, Basic Operations on Tuples, Functions to Process Tuples, Nested Tuples, Inserting Elements in a Tuple, Modifying Elements of a Tuple, Deleting Elements from a Tuple. **Dictionaries:** Operations on Dictionaries, Dictionary Methods, Using for Loop with Dictionaries, Converting Lists into Dictionary, Converting Strings into Dictionary		
Module III: • **Arrays:** Arrays in Python, Advantages of Arrays, Creating an Array, Importing the Array Module, Processing the Arrays, Types of Arrays, Working with Arrays using NumPy, Creating Arrays using array(), linespace, logspace,arrange(),zeros()and ones() Functions. • **Strings -** Sequences, Strings, Strings and Operators, String-Only Operators, Built-in F functions, String Built-in Methods, Special Features of Strings	**(6)** 3 3	Assignments
Module IV: • **Functions-** Introduction, Calling Functions, Creating Functions, Passing Functions, Formal Arguments, Variable Length Arguments, Functional Programming, Variable Scope, Recursion, Generators • **Classes and Objects:** Creating a Class, The Self Variable, Constructor, Types of Variables, Namespaces, Types of Methods, Instance Methods, Class Methods, Static Methods • **Exceptions-** Errors in a Python Program, Compile-Time Errors, Runtime Errors, Exceptions, Exception Handling, Types of Exceptions, The except Block, The assert Statement	**(6)** 2 2 2	Lectures
Module V: • **Files in Python:** Files, Types of Files in Python, opening a file, Closing a file 3 • **Graphical User Interface:** GUI in python, using tkinter, sample programs using Tkinter	**(6)** 3 3	Assignments / Lectures

Learning Resources	
1.	Textbook: Rao, Dr. R. Nageswara. Core Python Programming. 2nd Edition. Dream tech Press
2.	Reference books: Chun, Wesley J. **Core Python Programming**. 2nd Edition. USA: Pearson Education

PYTHON PROGRAMMING LAB

(GE Inter-Disciplinary)

Credits: 1 **Semester: III**

Subject Code: G23DS1P **No. of Practical Hours:30**

Objective: To develop applications using object oriented concepts of Python.

Outcome: Students will be able to demonstrate object oriented concepts of Python.

LIST OF PRACTICALS:

1. Temperature conversion.
2. Display Fibonacci series up to a limit.
3. Find factorial of a number.
4. Sum and product of all the items in the list.
5. Print the multiplication table of a number.
6. Print largest and smallest of items in the list.
7. Remove duplicates from a list.
8. Add, subtract and multiply two matrices.
9. Transpose a matrix.
10. Second smallest and largest number in a list.
11. Count the number of elements in a list within a range.
12. Get the frequency of the elements in a list.
13. Illustrate functions.
14. Check whether a string is palindrome or not.
15. Demonstrate pass by reference and pass by value.
16. Demonstrate on functional programming.
17. Demonstrate on scope of variables.
18. Demonstrate on module
19. Read and write to a file.
20. Python program on exception.
21. Display Invalid Mark exception if mark is greater than 100.
22. Append data to a file.
23. Program on recursion.
24. Program to print roots of a quadratic equation.
25. Program on break and continue.