23CS21SC - PYTHON PROGRAMMING

(Skill Enhancement Course)

(Common to CSE, CSE (DS), CSE (AI&ML), AI&DS, IT, and ECE)

Course Category:	Skill Enhancement course	**Credits:**	2
Course Type:	Practical	**Lecture-Tutorial-Practical:**	1-0-2
Prerequisite:	Familiarity with fundamental programming concepts such as variables, data types, operators, control structures (loops, conditionals), functions, and basic I/O operations.	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives:	• To introduce core programming concepts of Python programming language. • To demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries • To implement Functions, Modules and Regular Expressions in Python Programming and to create practical and contemporary applications using these		

<table>
<tr><td rowspan="7"><b>Course Outcomes Course Content</b></td><td colspan="2">Upon successful completion of the course, the students will be able to:</td></tr>
<tr><td>CO1</td><td>Apply Python programming concepts to solve a variety of computational problems (L3)</td></tr>
<tr><td>CO2</td><td>Classify data structures of Python (L4).</td></tr>
<tr><td>CO3</td><td>Exhibit competence in implementing and manipulating fundamental data structures such as lists, tuples, sets, dictionaries (L3)</td></tr>
<tr><td>CO4</td><td>Understand the principles of object-oriented programming (OOP) in Python, including classes, objects, inheritance, polymorphism, and encapsulation, and apply them to design and implement Python programs (L3)</td></tr>
<tr><td>CO5</td><td>Become proficient in using commonly used Python libraries and frameworks such as JSON, XML, NumPy, pandas (L2)</td></tr>
<tr><td colspan="2"><b><u>UNIT-I</u></b>
History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter Notebook. <b>Parts of Python Programming Language:</b> Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language. <b>Control Flow Statements:</b> if statement, if-else statement, if...elif…else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement.
<b>Sample Experiments:</b>
1. Write a program to find the largest element among three Numbers.
2. Write a Program to display all prime numbers within an interval
3. Write a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.
i) Arithmetic Operators ii) Relational Operators iii) Assignment Operatorsiv) Logical Operators v) Bit wise Operators vi) Ternary Operator vii) Membership Operators viii) Identity Operators</td></tr>
</table>

5. Write a program to add and multiply complex numbers
6. Write a program to print multiplication table of a given number.

UNIT-II

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command LineArguments.

Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.

Sample Experiments:

7. Write a program to define a function with multiple return values.
8. Write a program to define a function using default arguments.
9. Write a program to find the length of the string without using any library functions.
10. Write a program to check if the substring is present in a given string or not.
11. Write a program to perform the given operations on a list:
    i. Addition ii. Insertion iii. Slicing
12. Write a program to perform any 5 built-in functions by taking any list.

UNIT-III

Dictionaries: Creating Dictionary, Accessing and Modifying key: value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement.

Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozen set.

Sample Experiments:

13. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples, and print the concatenated tuples.
14. Write a program to count the number of vowels in a string (No control flow allowed).
15. Write a program to check if a given key exists in a dictionary or not.
16. Write a program to add a new key-value pair to an existing dictionary.
17. Write a program to sum all the items in a given dictionary.

UNIT-IV

Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.

Sample Experiments:

	18. Write a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper-case words from source must be lowered. 19. Python program to print each line of a file in reverse order. 20. Python program to compute the number of characters, words and lines in a file. 21. Write a program to create, display, append, insert and reverse the order of the items in the array. 22. Write a program to add, transpose and multiply two matrices. 23. Write a Python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square. **UNIT-V** **Introduction to Data Science:** Functional Programming, JSON and XML in Python, NumPy with Python, Pandas. **Sample Experiments:** 24. Python program to check whether a JSON string contains complex object or not. 25. Python Program to demonstrate NumPy arrays creation using array () function. 26. Python program to demonstrate use of ndim, shape, size, dtype. 27. Python program to demonstrate basic slicing, integer and Boolean indexing. 28. Python program to find min, max, sum, cumulative sum of array 29. Create a dictionary with at least five keys and each key represent value as a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows: a) Apply head () function to the pandas data frame b) Perform various data selection operations on Data Frame 30. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attributes with scatter and plot operations in matplotlib
Text Books & References Books	**REFERENCE BOOKS:** 1. Gowrishankar S, Veena A., Introduction to Python Programming, CRC Press. 2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2ndEdition, Pearson, 2024 3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.
E-Resources	1. https://www.coursera.org/learn/python-for-applied-data-science-ai 2. https://www.coursera.org/learn/python?specialization=python#syllabus