



SRI VENKATESWARA UNIVERSITY

I Year Syllabus for 4-Year UG Honours in B.Sc. (Data Science) as Major in consonance with Curriculum framework w.e.f. AY 2025-26

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
I	I	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation Lab	2	1
		2	Problem Solving Using C	3	3
			Problem Solving Using C Lab	2	1
	II	3	Python Programming and Data Structures	3	3
			Python programming and Data Structures lab	2	1
		4	Statistical Foundations for Data Science	3	3
			Statistical Foundations for Data Science lab	2	1

SEMESTER-II		
COURSE 3: PYTHON PROGRAMMING AND DATA STRUCTURES		
Theory	Credits: 3	3 hrs/week

Course Objectives

1. To introduce the fundamentals of Python programming, including environment setup, syntax, and core concepts.
2. To develop problem-solving skills using control flow, functions, and modules.
3. To provide knowledge of Python data structures, file handling, and exception handling for effective programming.
4. To impart object-oriented programming concepts and GUI development skills for building applications.

Course Outcomes (COs)

After successful completion of the course, students will be able to:

1. Explain the basic features, syntax, data types, and operators of Python programming.
2. Apply control flow constructs, functions, and modules to develop structured Python programs.
3. Demonstrate the use of sequences, sets, and dictionaries for effective data handling and manipulation.
4. Implement file handling techniques and apply exception handling mechanisms for robust applications.
5. Develop object-oriented and GUI-based applications using Python.

Unit 1. Basics of Python Programming:

Introduction to Python, Features of Python, Programming Modes - Interactive Mode & Script Mode, Identifiers, Naming Conventions, Keywords (Reserved Words), Built-in Data Types, Literals - Integer, Float, Complex, Boolean, String, Variables, Operators, Expressions, Assignment Statements, Input/Output Statements, Python Syntax (Lines, Comments, Indentation)

Operators & Operands, Classification of Operators - Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Assignment, Augmented Assignment, Identity Operators, Expressions & Precedence Rules.

Unit 2. Control Flow, Functions & Modules:

Control Flow - if Statement, if-else, if-elif-else. Iterative Statements – while, for, Nested Loops, Loop Control Statements – break, continue, pass; else with loops

Need for Functions, Defining & Invoking User-defined Functions, Return Statement, Function Input/Output Cases, Scope of Variables - Local, Global, Nested Functions, Function Arguments - Required, Positional, Default, Variable-length, main() Function, Documentation Strings, Recursive Functions, Anonymous Functions (Lambda), Library Functions

Modules - Import, from..import, Creating & Using Modules, Namespaces

Unit 3. Sequence, Set, Mapping Types:

Strings- Representation, Indexing, Slicing, Immutability, String Operators, Traversal, Accumulation, Formatting & Methods

Lists - Overview, Indexing, Slicing, Methods, Mutability, List Operations - Add, Update, Delete, Search, Copy, Traverse, Comprehension

Tuples - Operations, Immutability, Tuple Assignment, Arrays & Operations
Sets - Overview, Methods, Mathematical Operations, Frozenset, Comprehension

Dictionaries - Overview, Methods, Operations, Traversal, Comparison

Unit 4. File Handling, Exception Handling & Object Oriented Programming:

File Handling - Types, Paths, Basic Operations on Files - Open/Close, Read/Write, CSV Files, OS/Pathlib

Error & Exception Handling - Syntax Errors, Built-in Exceptions, Catching and Handling Exceptions: try-except, raise, User-defined Exceptions, Assertions

OOP Concepts: Classes, Objects, Attributes, Methods, Constructor and Destructors

Encapsulation: Private and Public Members

Inheritance: Single, Multilevel, Multiple, Method Overriding

Unit 5: Abstract Data Structures and GUI Programming

Abstract Data Structures (ADTs): Concepts and Importance

Linked List: Definition, Types- Singly, Doubly, Circular; Node Structure, Insertion, Deletion, Traversal (Single Linked list implementation only)

Stacks: LIFO Principle, Implementation using List, Applications

Queues: FIFO Principle, Implementation using List, Priority Queues

GUI Programming with Tkinter: Widgets (Label, Button, Entry, Menu, Listbox, Canvas etc.), Event Handling, Building Simple GUI Apps

Textbooks:

1. Python Programming-An Object Oriented approach, Anita Goel, Universities Press
2. Python Programming using Problem Solving Approach Reema Thareja Oxford University Press 2020
3. Exploring Python, Budd T A, McGraw-Hill Education, 1st Edition, 2011.

Reference Book:

1. Python: The Complete Reference, Martin C. Brown, Mc Graw-Hill, 2018
2. Fundamentals of Python, Kenneth A. Lambert. (2019), First Programs, 2nd Edition, CENGAGE Publication.

Activities:

Outcome: Explain the basic features, syntax, data types, and operators of Python programming.

Activity: Conduct a "Python Basics Lab" where students write small programs to demonstrate literals, variables, data types, and operators (e.g., swapping numbers, simple calculator).

Evaluation Method:

- Lab performance checklist (execution of 3 mini tasks)
- Short quiz with multiple-choice and fill-in-the-blanks on syntax, data types, and operators

Outcome: Apply control flow constructs, functions, and modules to develop structured Python programs.

Activity: Group activity - "Python Problem Solving Challenge": Students solve real-life problems (e.g., finding prime numbers, grade calculator, menu-driven calculator) using control structures, functions, and importing standard modules.

Evaluation Method:

- Code submission with proper use of functions/modules (20%)
- Viva-voce to explain logic and flow of control (40%)
- Unit test with scenario-based programming questions (40%)

Outcome: Demonstrate the use of sequences, sets, and dictionaries for effective data handling and manipulation.

Activity: Hands-on mini project – "Student Data Manager": Students create a program using lists, tuples, sets, and dictionaries to store and manipulate student records (e.g., marks, courses, hobbies).

Evaluation Method:

- Practical demo of program with at least 5 data operations (add, search, delete, update, traverse)
- Evaluation rubric for correctness, efficiency, and use of appropriate data structure

Outcome: Implement file handling techniques and apply exception handling mechanisms for robust applications.

Activity: Individual assignment - "File-Based Address Book": Students create a program to store, update, and retrieve data from files, with exception handling for invalid inputs or missing files.

Evaluation Method:

- Assessment of program correctness (file read/write, append, delete, exception handling)
- Short quiz with error-tracing and debugging questions (given code with errors, students identify and correct)

Outcome: Develop object-oriented and GUI-based applications using Python.

Activity: Mini Project – "Student Information System with GUI": Students design a simple Tkinter-based application with classes/objects for handling student data, including basic GUI widgets (Entry, Button, Listbox).

Evaluation Method:

- Project demo and presentation (50%)
- Rubric-based evaluation for OOP concepts (classes, inheritance, encapsulation) and GUI design (widgets, event handling) (30%)
- Peer review/feedback on usability (20%)

SEMESTER-II
COURSE 3: PYTHON PROGRAMMING AND DATA STRUCTURES

Practical	Credits: 1	2 hrs/week
------------------	-------------------	-------------------

1. Basic Python Programs:
 - a. Write a program to display basic details (name, roll number, department) using print() and demonstrate different literal types (int, float, string, boolean, complex).
 - b. Write a program to perform arithmetic, relational, logical, bitwise, and assignment operations on given inputs.
2. Control Flow Practice
 - a. Write a program to find the largest of three numbers using if-elif-else.
 - b. Write a program to check whether a number is prime or not using loops.
 - c. Write a program to illustrate the use of loop control statements (break, continue, pass).
3. Functions and Recursion
 - a. Write a program to define a function to calculate factorial of a number (using recursion).
 - b. Write a program to demonstrate different types of function arguments (default, positional, keyword, variable-length).
4. Write a program to illustrate string slicing, concatenation, repetition, and built-in methods.
5. Write a program to create a list of numbers, perform insertion, deletion, searching, sorting, and list comprehension.
6. Write a program to demonstrate tuple packing, unpacking, and immutability.
7. Write a program to implement set operations (union, intersection, difference, subset, superset).
8. Write a program to create a dictionary of student roll numbers and marks, and perform add, update, delete, and traversal operations.
9. Write a program to read and display count of vowels, consonants, digits, and spaces of a text file.
10. Write a program to copy the contents of one file into another file.
11. Write a program to read and process student marks from a CSV file (calculate average, highest, lowest).
12. Write a program to demonstrate exception handling using try-except-finally.
13. Write a program to create a class Student with attributes and methods to display details.
14. Write a program to demonstrate single and multilevel inheritance.
15. Implement stack (LIFO) and queue (FIFO) using lists and linked lists.
16. Implement singly linked lists: node creation, insertion, deletion, traversal.
17. Write a Tkinter program with Label, Entry, and Button widgets to take user input and display it.
18. Write a Tkinter program to create a simple calculator application.

SRI VENKATESWARA UNIVERSITY::TIRUPATI
B.Sc. Data Science Honours
I year II Semester
Course 3: PYTHON PROGRAMMING AND DATA STRUCTURES

MODEL QUESTION PAPER

Time: **3 Hours**

Max. Marks: **75**

PART-A

Answer any **FIVE** of the following. Each Question Carries 3 marks.

(5X3=15)

1. Short answer question from unit-I
2. Short answer question from unit-I
3. Short answer question from unit-II
4. Short answer question from unit-II
5. Short answer question from unit-III
6. Short answer question from unit-III
7. Short answer question from unit-IV
8. Short answer question from unit-IV
9. Short answer question from unit-V
10. Short answer question from unit-V

PART-B

Answer any **FIVE** of the following. Each Question Carries 12 marks.

(5X12=60)

11. a) Essay question from Unit-I
OR
b) Essay question from Unit-I
12. a) Essay question from Unit-II
OR
b) Essay question from Unit-II
13. a) Essay question from Unit-III
OR
b) Essay question from Unit-III
14. a) Essay question from Unit-IV
OR
b) Essay question from Unit-IV
15. a) Essay question from Unit-V
OR
b) Essay question from Unit-V