St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – III of the following departments under Faculty of Data Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2026-27.

DEPARTMENT OF DATA SCIENCE

Category III
BSc. Artificial Intelligence and Machine Learning (Hons.)

Major Course-5: Principles of Data Structures and Algorithm Design

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Prerequisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Principles of Data Structures and Algorithm Design	4	0	0	10 + 2 from a recognized board in any stream	Nil

Course Outcomes:
Upon successful completion of the course, students will be able to:

1. Understand the concepts of abstract data types and various data structures.
2. Analyze the advantages and limitations of linear and non-linear data structures.
3. Implement fundamental data structures and their operations using suitable algorithms.
4. Apply searching and sorting techniques to solve computational problems efficiently.
5. Develop problem-solving skills using appropriate data structures and algorithmic approaches.

Learning Outcomes:

At the end of the course, learners will be able to:

1. Explain the concepts of abstract data types, arrays, linked lists, searching, and sorting techniques.
2. Implement stack and queue data structures and apply them to solve practical problems such as expression conversion and recursion.
3. Analyze and implement tree data structures including binary trees, binary search trees, AVL trees, heaps, and B-trees.
4. Represent graphs using different techniques and apply graph traversal and shortest/minimum spanning tree algorithms to solve real-world problems.

UNIT	TOPICS / SUBTOPICS	HOURS
1	**Introduction to Data Structures, Arrays & Linked List**: Introduction: Data, Data Types, Abstract Data Types (Primitive), User-Defined Data Types (Non-Primitive), Data Structures: Definition, Classification of Data Structures and details of each Classifications Array: Definition, Mapping, Sparse Matrix Linked List: Comparison of Array and Linked List, Types of Linked Lists, Representation of Linked Lists Operations on Doubly Linked Lists (Algorithm and Explanation), Creation, Traversal, Insertion (Front, In Between [After and Before], At End), Deletion (From Beginning, From Between, From End) Searching: Introduction to Searching, Searching Techniques (Sequential Search, Binary Search) Sorting: Introduction to Sorting, Sorting Technique (Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort)	**15**
2	**Stack & Queues:** Stack: Introduction (Idea of the Stack), Operations of the Stack (Algorithm and Explanation), Implementation of the Stack (Using linked list), Applications of the Stack, Reverse and Polish Conversion: Infix to Postfix using manually and stack for parenthesis and Non-parenthesis (with Algorithm), Recursion (Definition) Queue: Introduction (Idea of the Queue), Types of Queue, Operations of Simple and Circular Queue	**15**
3	**Tree:** Introduction, Terminology, Binary Tree: Definition, Representation of Binary Tree, Operation on Binary Tree, Creation, Insertion, Deletion Traversal (Pre-Order, In-Order and Post- Order), Conversion from (Pre, In or Post) into Binary Tree, Types of Binary Tree, Full Binary Tree, Complete Binary Tree, Binary Search Tree, Expression Tree, Threaded Binary Tree, Heap Tree, Height Balanced Tree (AVL Tree), B-Tree	**15**
4	**Graph**: Introduction, Basic Terminology, Representation of Graph, Adjacency Matrix (Array), Adjacency Linked, Traversal of Graph, Breadth First Traversal (Algorithm and Tracing), Depth First Traversal (Algorithm and Tracing), Application of Graph, Spanning Tree, Minimum Spanning Tree (BFS and DFS), Prim's Algorithm, Kruskal's Algorithm, Shortest Path Algorithm, Dijkstra's Algorithm	**15**

Textbook: Data and File Structures using C, Publisher: Oxford, Author: Reema Thareja

Chapter-4 (4.1, 4.2, 4.3) – Introduction to Data Structures, Chapter-5 (5.1, 5.2, 5.3, 5.6.5, 5.16) – Array and Searching, Chapter-8 (8.2, 8.7) – Linked List, Chapter-9 (9.1, 9.3, 9.4, 9.5, 9.7, 9.8, 9.11, 9.12, 9.13, 9.14, 9.16 *Only Definition+, 9.17 *Definition and 9.17.1+) – Stack & Queues, Chapter-10 (10.1, 10.2, 10.4 *excluding 10.4.4+) – Tree, Chapter-11 (11.1, 11.2.2, 11.2.3, 11.3, 11.4 *Definition and 11.4.2+, 11.6 *Definition and 11.6.2+) – Tree, Chapter-12 (12.1 *Definition and 12.1.1, 12.1.2+) – Tree, Chapter-13 (13.1, 13.4, 13.5, 13.7 *excluding 13.7.5+) – Graph, Chapter-14 (14.1, 14.2, 14.3, 14.4, 14.5, 14.6) – Sorting

Reference Books:

1. Data Structures and Algorithms in C++ (Publisher: Dreamtech By B. M. Harwani),
2. Magnifying Data Structures (Publisher: PHI By: Arpita Gopal)
3. Data Structures using C & C++ (Publisher: Wiley-India By : Rajesh K. Shukla)
4. Introduction to Data Structures in C (Publisher: Pearson Education By: Ashok N. Kamthane)
5. Data Structures Using C (Publisher: Pearson Education By : A. K. Sharma)

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – III of the following departments under Faculty of Data Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2026-27.

DEPARTMENT OF DATA SCIENCE

Category III
BSc. Artificial Intelligence and Machine Learning (Hons.)

Major Course-6: Python Programming

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Prerequisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Python Programming	2	0	4	10 + 2 from a recognized board in any stream	Nil

Course Outcomes:
Upon successful completion of the course, students will be able to:

1. To establish a foundational understanding of Python's interpreted ecosystem, memory management model (variables as object assignments), and structural paradigms in comparison to compiled languages like C++.
2. To develop algorithmic thinking and problem-solving skills using core control structures, procedural abstraction, functional modularity, and visual logic design via the Turtle module.
3. To master the selection, implementation, and manipulation of Python's built-in data structures—including strings, lists, tuples, dictionaries, and sets—to handle complex data organization.
4. To instill Object-Oriented Programming (OOP) principles in Python, emphasizing encapsulation, inheritance, object referencing, and the "everything is an object" paradigm for scalable software development.

Learning Outcomes:

At the end of the course, learners will be able to:

1. Analyze and translate C++ procedural/compiled logic into idiomatic Python code, utilizing proper naming conventions, interpreted workflows, and standard debugging environments (IDLE and Jupyter Notebooks).
2. Design and modularize complex programs by breaking them down into reusable functions with appropriate scope handling, mathematical reasoning, and conditional control flow.
3. Evaluate and manipulate mutable and immutable data structures using sequence slicing, comprehensions, and built-in methods to optimize data retrieval and storage logic.

4. Construct robust software architectures using object-oriented principles, defining custom classes, overriding methods, managing object aliases, and leveraging multi-level inheritance for effective code reuse.

UNIT	TOPICS / SUBTOPICS	HOURS
1	**Introduction to Python and its ecosystem:** Why Python is the industry standard for AI and ML. Translation from C++ to Python paradigms. Transitioning from compiled to interpreted environments. Introduction to the IDLE and Jupyter Notebook interfaces. Variables as object assignments. Variable naming conventions and snake_case vs camelCase. Reserved keywords in Python. Numbers and numeric types. Arithmetic operators. Comparative operators. Logical operators and Boolean expressions. Basic introduction to the string data type. Introduction to the Turtle module for visual logic. Drawing geometric shapes with Turtle.	15
2	**Decision Making and Functions:** Branching with if-else and if-elif-else statements. Nested decision-making structures. Logical operators in control flow. Efficiency and readability of different decision-making structures. Introduction to functions. Defining and calling functions. Function arguments and parameters. Understanding the difference between printing and returning values. Modularizing code for reusability. Variable scope within functions. Local vs global scope. Global keyword and its implications. Abstraction through functions.	15
3	**Various Data Structures in Python:** In-depth string manipulation. String indexing and slicing. Immutable sequence properties. String methods including find, count, replace, split, and join. List manipulation and mutability. List methods for appending, popping, and sorting. Introducing the Tuple data structure for immutable sequences. Introduction to Dictionaries as key-value pairs. Mapping and searching in Dictionaries. Sets and set operations for uniqueness and membership testing. Sequence unpacking and nesting data structures.	15
4	**OOP in Python:** Foundations of Object-Oriented Programming. Understanding variables as object assignments rather than equivalences. Identifying Python object types and the "everything is an object" philosophy. Defining classes and creating instances. Class attributes vs instance attributes. The self keyword and constructor methods (init). Implementing methods for object behavior. Principles of Encapsulation. Introduction to Inheritance and code reuse. Method overriding and super(). Multi-level inheritance. Understanding object references and aliasing	15

Practical Lists:

1. Setting Up Python Environment and Exploring IDLE & Jupyter Notebook
2. Working with Variables, Data Types, and Arithmetic Operations in Python
3. Logical, Comparison Operators and Building Simple Boolean Expressions
4. Introduction to Strings and Basic String Manipulation Tasks
5. Drawing Geometric Shapes Using the Turtle Module
6. Implementing Decision-Making Programs Using if, if-else, and if-elif Structures
7. Building Mini Logic-Based Applications Using Nested Conditions
8. Writing and Calling Functions with Parameters and Return Values
9. Exploring Variable Scope, Local vs Global Variables, and Code Modularization
10. String Indexing, Slicing, and Advanced String Methods Practice
11. List Operations: Creating, Modifying, Sorting, and Nesting Lists
12. Working with Tuples, Dictionaries, and Set Operations
13. Building a Menu-Driven Program Using Dictionaries and Control Flow
14. Mini Project: Student Record Management System Using Core Data Structures
15. Introduction to Classes and Creating Basic Python Objects
16. Implementing Class Attributes, Instance Attributes, and Constructors
17. Encapsulation and Designing Methods for Object Behavior
18. Implementing Inheritance, Method Overriding, and Using super()
19. Multi-Level Inheritance and Understanding Object References and Aliasing
20. Capstone Project: Building a Modular Object-Oriented Application Integrating Data Structures and Functions

Textbook:

Bell, Ana. Get Programming: Learn to Code with Python. Manning Publications, 2018.

Reference Books:

1. Python Crash Course: A Hands-On, Project-Based Introduction to Programming by Eric Matthes (No Starch Press)
2. Fluent Python: Clear, Concise, and Effective Programming by Luciano Ramalho (O'Reilly Media)
3. Introduction to Computation and Programming Using Python: With Application to Computational Science and Understanding Data by John V. Guttag (MIT Press)

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – III of the following departments under Faculty of Data Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2026-27.

DEPARTMENT OF DATA SCIENCE

Category III
BSc. Artificial Intelligence and Machine Learning (Hons.)

Major Course-7: Data Structures Laboratory and Command Line Environment

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Prerequisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Data Structures Laboratory and Command Line	0	0	8	10 + 2 from a recognized board in any stream	Nil

UNIT	TOPICS / SUBTOPICS	HOURS
1	**Command Line Environment:** **MS-DOS & Linux Practical Assignments:** 1. Execute basic commands in MS-DOS such as dir, cls, date, and time and observe system information 2. Create and navigate directories using md, cd, and dir by forming a folder structure for Data Structures programs 3. Create, rename, copy, and delete files using DOS commands like copy, rename, and del 4. Move files between directories and organize Data Structures programs into appropriate folders using move 5. Create a simple batch file (.bat) to display messages and list files in a director 6. Execute basic Linux commands such as pwd, ls, whoami, and clear in terminal 7. Create and navigate directories in Linux using mkdir, cd, and rmdir for organizing program files	**30**

	8. Create, copy, move, and remove files in Linux using touch, cp, mv, and rm, 9. View and edit file contents using commands such as cat, more, less, and a text editor like nano or vi 10. Use wildcard characters (*, ?) and search utilities like find and grep to locate files and text content	
2	**Searching and Sorting & Link List:** Searching and Sorting Programs: 1. Write a program to implement sequential search 2. Write a program to implement binary search 3. Write a program to implement bubble sort 4. Write a program to implement selection sort 5. Write a program to implement merge sort 6. Write a program to implement quick sort 7. Write a program to implement insertion sort Singly Link List Operations: 1. Write program to implement operations using Singly link list (Insert at first, Insert at Last, Insert at specified location [Before or After the Node], Delete from first, Delete from last, Delete any specified node, Traversal, Sorting, Splitting, Merging, Counting Operations [Total no. of nodes, even and odd no. of nodes]) Doubly Link List Operations: 2. Write program to implement operations using Doubly link list (Insert at first, Insert at Last, Insert at specified location [Before or After the Node], Delete from first, Delete from last, Delete any specified node, Traversal, Sorting, Splitting, Merging)	**30**
3	**Stack Operations & Expressions:** Stack Assignments: 1. Write a program to implement operations in stack Using Linked List (PUSH, POP, PEEP, CHANGE) 2. Write a program to implement Evaluation of given postfix expression 3. Write a program to implement conversion of infix expression into postfix expression(parentheses and non-parentheses) 4. Write a program to implement recursion 5. Write a program to reverse the string using the stack	**30**
4	**Queue, Tree and Graph:** Assignments: 1. Write a program to implement Simple Queue operations using Linked List (ENQUEUE, DEQUEUE, Traversal [display]) 2. Write a program to implement Circular Queue operations Using Linked List (ENQUEUE, DEQUEUE, Traversal [display]) 3. Write a program to implement operations on Binary Search Tree using Linked List (Creation, Insertion, Traversal [In-order, Pre-order, Post-order]) 4. Write a program to implement DFS and BFS traversal Of a graph	**30**

Textbook: Data and File Structures using C, Publisher: Oxford, Author: Reema Thareja

Chapter-4 (4.1, 4.2, 4.3) – Introduction to Data Structures, Chapter-5 (5.1, 5.2, 5.3, 5.6.5, 5.16) – Array and Searching, Chapter-8 (8.2, 8.7) – Linked List, Chapter-9 (9.1, 9.3, 9.4, 9.5, 9.7, 9.8, 9.11, 9.12, 9.13, 9.14, 9.16 *Only Definition+, 9.17 *Definition and 9.17.1+) – Stack & Queues, Chapter-10 (10.1, 10.2, 10.4 *excluding 10.4.4+) – Tree, Chapter-11 (11.1, 11.2.2, 11.2.3, 11.3, 11.4 *Definition and 11.4.2+, 11.6 *Definition and 11.6.2+) – Tree, Chapter-12 (12.1 *Definition and 12.1.1, 12.1.2+) – Tree, Chapter-13 (13.1, 13.4, 13.5, 13.7 *excluding 13.7.5+) – Graph, Chapter-14 (14.1, 14.2, 14.3, 14.4, 14.5, 14.6) – Sorting

Reference Books:

1. Data Structures and Algorithms in C++ (Publisher: Dreamtech By B. M. Harwani),
2. Magnifying Data Structures (Publisher: PHI By: Arpita Gopal)
3. Data Structures using C & C++ (Publisher: Wiley-India By : Rajesh K. Shukla)
4. Introduction to Data Structures in C (Publisher: Pearson Education By: Ashok N. Kamthane)
5. Data Structures Using C (Publisher: Pearson Education By : A. K. Sharma)