St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – I of the following departments under Faculty of Data Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF DATA SCIENCE

BSc. Artificial Intelligence and Machine Learning (Hons.)

Major Course-1: Introduction to Searching Strategies in AI

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		
Introduction to Searching Strategies in AI	4	0	0	10 + 2 from a recognized board in any stream	Nil

Course Objectives:

Learning Outcomes:

After completing this course, students will be able to:

- CO-1: Understand the fundamental principles of search in Artificial Intelligence and its significance in problem solving
- CO-2: Develop the ability to represent the problems in a structured way to enable systematic exploration of problems
- CO-3: Explore various search strategies for decision making and problem solving in both deterministic and non-deterministic environments
- CO-4: Apply appropriate search methods to model intelligent behavior in real world and abstract problem domains

UNIT	TOPICS / SUBTOPICS	HOURS
1	**Fundamentals of Artificial Intelligence**: Definition and History of AI: Exploration of AI's evolution, key milestones, and its impact on various industries. Applications and Scope of AI: Discussion on AI applications in fields like healthcare, finance, robotics, and more. Basic Problem-Solving Methods: Introduction to problem-solving paradigms, including algorithms and heuristics.	**15**
2	**Uninformed Search Strategies:** Problem Formulation in AI: Techniques to represent problems, including state spaces and goal definitions. Depth-First Search (DFS): Mechanics of DFS, its implementation, advantages, and limitations. Breadth-First Search (BFS): Mechanics of BFS, its implementation, advantages, and limitations. Depth-First Iterative Deepening (DFID): Combining DFS and BFS advantages to handle infinite depths.	**15**
3	**Informed Search Strategies:** Heuristic Functions: Concept of heuristics, designing admissible and consistent heuristics. Best-First Search: Introduction to best-first search and its applications. A Search Algorithm*: Detailed study of A*, its optimality, and practical applications.	**15**
4	**Iterative Deepening A***: Combining depth-first search's space efficiency with breadth-first search's completeness. Recursive Best-First Search (RBFS): Memory-efficient version of best-first search suitable for large problems. Bidirectional Search: Simultaneous forward and backward search to reduce search time. Algorithm AO*: Handling problems with AND-OR structures, applicable in decision-making scenarios.	**15**

Essential / Recommended Readings:

A first course in Artificial Intelligence by Deepak Khemani

Suggestive Readings:

Artificial Intelligence a modern Approach by Stuart Russelll and Peter Norvig

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – I of the following departments under Faculty of Data Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF DATA SCIENCE

BSc. Artificial Intelligence and Machine Learning (Hons.)

Major Course-2: Logic Building with C 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		
Logic Building with C Programming	2	0	4	10 + 2 from a recognized board in any stream	Nil

Course Objectives:

This course aims to:

1. Develop logical thinking and problem-solving abilities using structured programming principles.
2. Introduce the core concepts and syntax of the C programming language for implementing logic-based solutions.
3. Strengthen the ability to design and implement programs using modular and efficient logic with functions, arrays, and pointers.
4. Enhance programming skills to solve real-world logical problems through file handling and memory management in C.

Learning Outcomes:

After completing this course, students will be able to:

- CO-1: Apply logical reasoning to break down problems and implement solutions using control structures in C.
- CO-2:. Construct modular programs that demonstrate clear and effective logic using functions and arrays.

- CO-3: Use pointers and string manipulation to handle data logically and efficiently.
- CO-4 : Solve practical problems by applying file handling and dynamic memory concepts to build logic-driven programs.

UNIT	TOPICS / SUBTOPICS	HOURS
1	**Introduction to Programming Languages** • Machine level language, Assembly language, High-level language • Classification: Procedural vs. Non-Procedural Language, History of C • Basic Structure of C and Executing a C Program • Identifiers, Keywords, Data Types, Constants, Variables • Storage Classes, Type Casting, Comments **Operators & Expressions** • Types of Operators, Precedence & Associativity **Decision Making** • If, If-else, Nested If-else, Switch **Looping & Control Statements** • While, Do-while, For Break, Continue, Goto, Exit	15
2	**Practical Implementation of C Programming Fundamentals** **Understanding the Editor** **Getting Started with C: Writing and Executing Basic Programs"** • Basic I/O functions: printf(), scanf(), gets(), puts(), etc. **"Mastering Console I/O: Character-Based Input & Output"** • getchar(), putchar(), getch(), getche(), putch() **"Operators in Action: Exploring Expressions in C"**	30

- Programs using arithmetic, relational, logical, bitwise, and assignment operators

 "Decision Making in C: Implementing If-Else and Switch"

- Writing programs with if-else, nested if, and switch-case

Working with Operators and Expressions

Looping Through Logic: Implementing While, Do-While, and For Loops"

- Programs demonstrating loops and their use cases

3	**Control Structures, Arrays, Functions & Pointers**	**15**

User-Defined Functions

- Function Categories (No argument no return, etc.)
- Recursion, Nesting Functions, Scope & Lifetime
- Call byValue and Call by Reference

Arrays (One Dimentional Array and 2D Array

Structure in C

Introduction to Pointers

- Definition, Concept, Advantages
- Pointer Arithmetic, Pointers with Arrays & Functions

Memory Management

- Static and Dynamic Memory Management
- Function :-malloc(), calloc(), realloc(), free()

4	**Implementation of Control Structures, Arrays, Functions & Pointers**	**30**

Array Adventures: One & Two-Dimensional Arrays"

- Writing programs for array initialization, manipulation, and operations

 "Strings in C: Handling and Manipulating Character Arrays"

- Programs using string functions like strcpy(), strcmp(), strlen()

"Functions and Recursion: Building Modular Code"

- Implementing different function categories & recursion
- Pointer operations, pointer arithmetic, malloc(), calloc(), realloc(), free()
- **Structure Implementation**

"Pointer Magic: Understanding Memory and Dynamic Allocation"

Essential / Recommended Readings:

- Programming in ANSI C. (6th Ed.) – Balagurusami - Tata McGraw Hill Publication
- Programming In C (2nd Ed.) - Ashok N. Kamthane - Pearson Education

Suggestive Readings:

- The C Programming Language - DENNIS M. RITCHIE- AT&T Bell Laboratories Murray Hill, New Jersey • Let us C – (15th Ed.) - Yashwant Kanetkar - BPB Publications
- Programming in C - Reema Thareja - Oxford University Press

Practical List:

1. Find the Simple Interest. Inputs are principal amount, period in year and rate of interest.
2. Find the area and perimeter of square and rectangle. Input the side(s) through the keyboard.
3. Write a program to enter the temperature in Fahrenheit and convert it to Celsius. [C = ((F-32)*5)/9]
4. Write a program to store and interchange two numbers in variables a and b.
5. Write a program to enter two numbers and find the smallest out of them. Use conditional operator.
6. Write a program to enter text with gets() and display it using printf() statement also find the length of the text.
7. Write a program to check given year is a Leap year or not.
8. Write a C program to find the maximum from given three numbers (Using Nested IF).
9. Take marks from the user and print grade accordingly (>=75 marks – Distinction, <75 and >=60 marks – First, <60 and >=50 – Second, <50 and >=35 – Pass, <35 – Fail) using if … else if….else statement and also by using logical operators).
10. Write a program to accept number of seconds and display its corresponding hours, minutes and seconds.
11. Write a program to check whether the blood donor is eligible or not for donating blood. The conditions laid down are as under. Use if statement.
    a) Age should be above 18 yrs but not more than 55 yrs.
12. Write a program to calculate bill of a job work done as follows. Use if else statement.a) Rate of typing 3 Rs/pageb) Printing of 1st copy 5Rs/pages & later every copy 3Rs/page. The user should enter the number of pages and print out copies he/she wants.

13. Write a program to find sum of N numbers. (Using while loop)
14. Write a program to print 1,2,3,….N where N number scanned by user. (Using while loop)
15. Write a program to find factorial of given number.
16. Write a program to find reverse of a given number.
17. Write a program to find the sum of first 100 odd nos. and even nos.
18. Write a program to find maximum from given N inputs by user.
19. Write a program to find sum of the digits entered by the user.
20. Write a program to generate Fibonacci series up to N numbers.
21. Write a program to find GCD and LCM of given 2 numbers.
22. Write a program to check whether given number by the user is Palindrome or not.
23. Write a program to check whether the given number is Prime or not.
24. Write a C program to find x1+x2+x3+x4+ …. + xn.
25. Write a program to print following pyramid.

```
    *
   * *
  * * *
 * * * *
```

26. Write a program that accepts an integer N, if the integer N = 4, then print the pyramid:

```
    1
   121
  12321
 1234321
```

27. Write a program which will take 10 numbers from user and stored it in the array. It will print all the numbers, their sum and average of it.
28. Write a program to find binary of given number.
29. Write a program to sort and array.
30. Write a program to search an element from the array.
31. Write a program to find addition of two matrices of 3*3.
32. Take two strings from the user and check whether the string is palindrome or not.
33. Write a program to find sum, average of two numbers passed to user defined functions called sum(int,int) and average(int,int).
34. Write a program to print Fibonacci series using recursive UDF.
35. Write a program to find length of the given string (without including string.h).
36. Write a program which will accept two strings from the user and print the message that the strings are same or not.
37. Write a program that uses function digit(N,k) that return the value of the kth digit from the right of the number N. For eg. The function call digit (254693,2) should return 9.
38. Write a program to Swap Numbers in Cyclic Order Using Call by Reference.
39. Write a program to create an array that will store integer pointers. (Array of pointers)
40. Write a program to demonstrate an example of pointer to an array.