

Branch Name:	MCA
Program Code:	CS201
Course Name:	Introduction to Artificial Intelligence Practical
Course Code:	4CS2010203P
Pre-requisite Course:	Basic programming concepts, data structures, discrete mathematics, fundamentals of algorithms and complexity, probability theory

Course Objective:

1. Introduce the core principles and historical development of AI.
2. Familiarize students with intelligent agent design and problem-solving strategies such as search algorithms.
3. Equip students with knowledge of knowledge representation, reasoning, and planning techniques.
4. Develop understanding of Prolog, logical Programming learning.
5. Expose students to applications of AI in areas like Search, Games and Problem Solving.
6. Encourage ethical awareness and critical thinking regarding the societal impact of AI technologies.

Teaching and Examination Scheme:

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
-	-	4	2	-	-	25	25	50

Suggested Course Practical List:

To perform the following Lab work:

S/W Required: GNU Prolog, SWI-Prolog, MATLAB, Python, Code block C/C++

1. Define family relationships (parent, child, sibling, grandparent) using facts and rules.
2. Identify all ancestors of a person using recursive rules.
3. Create a knowledge base of animals and classify them as birds, mammals, etc., and query their properties.
4. Implement a simple expert system to identify diseases based on symptoms.
5. Define arithmetic operations using recursive logic (e.g., addition, factorial).
6. Represent and solve a basic logic puzzle (e.g., who owns the zebra?).
7. Create a list-based program to find the maximum of a list.
8. Use lists to reverse elements and query the reversed list.
9. Implement a simple database of books and authors and query based on genre or author.
10. Model and reason about traffic lights (e.g., when is it safe to cross?).
11. Represent family tree data and infer cousin or uncle relationships.
12. Define a rule-based planning system to plan a simple sequence of tasks (e.g., preparing tea).
13. Simulate an AI agent's decision-making using facts and rules.
14. Model a simple library management system and allow checking book availability.
15. Create a timetable conflict checker for classrooms using constraints.

16. Use lists and recursion to calculate the length of a list.
17. Model a set of rules for granting university admission based on criteria.
18. Represent propositional logic formulae and test satisfiability (manually).
19. Simulate a basic flight route planner using simple search rules.
20. Define and simulate the game of Tic-Tac-Toe with basic moves.
21. Write a knowledge base for different animals and identify which can fly (Flying Penguin example).
22. Define and implement rules for sorting a list of numbers using recursion.
23. Represent and evaluate first-order logic statements (e.g., all humans are mortal, Socrates is a human).
24. Simulate a basic constraint logic programming example like Sudoku (3x3).
25. Model a simple chatbot using pattern matching and rules for basic conversation.
26. Implement Breadth-First Search to find a path between two cities in a connected graph.
27. Model a maze and solve it using Depth-First Search in PROLOG.
28. Simulate a simple two-player game (e.g., Nim game) with Minimax strategy.
29. Develop an A search algorithm for finding the shortest route between locations with given heuristics.

City	Heuristic (h) to Goal
A	10
B	6
C	4
D	7
E	2
Goal	0

30. Implement Alpha-Beta Pruning in a simple game tree (like a reduced Tic-Tac-Toe tree).
31. Simulate uncertain knowledge by assigning confidence levels to rules and use rule selection based on thresholds.
32. Model the Tweety example using exceptions and rules to handle uncertainty (e.g., birds generally fly unless they are penguins).
33. Implement a rule-based system where multiple causes can lead to a symptom, and the program ranks possible causes.
34. Design a simplified Bayesian-like reasoning system using rule weights or scoring system (simulated in PROLOG).
35. Build a basic expert system for weather prediction using rule-based inference with uncertain facts (e.g., "if clouds and humidity are high, rain is likely").

Textbooks:

1. Introduction to Artificial Intelligence, By: Wolfgang Ertel, 2nd Edition, Springer

References Books:

1. Artificial Intelligence A Modern Approach, By: Stuart Russell and Peter Norving, 3rd Edition, Pearson Publication
2. First Course In Artificial Intelligence, By Deepak Khemani, Mc Graw Hill
3. Artificial Intelligence, By Elaine Rich and Kevin Knight, 2nd Edition, Mc Graw Hill
4. Artificial Intelligence: Making a System Intelligent, By: Dr. Nilakshi Jain, As per AICTE, Wiley Publication
5. Introduction to Artificial Intelligence, By: Eugene Charniak and Drew McDermott, Addison-Wesley Series in Computer Science
6. Artificial Intelligence: An Engineering Approach, By: Robert J. Schalkoff, Mc Graw Hill,
7. Programming in Prolog: Using the ISO Standard, By: William F. Clocksin, Christopher S. Mellish, 5th Edition, Springer

List of Useful websites / MOOCs

1. Learners are advised to opt for NPTEL and SWAYAM courses that are relevant to this course

Course Learning Outcomes (CLO): On completion of this course, the students will be able to:

CLO	Description	Bloom's Taxonomy Level
CLO1	Apply formal logic systems to represent and reason with knowledge.	Remembering Understanding Applying
CLO2	Analyze the inherent limitations of classical logic in AI	Remembering Analyze
CLO3	Develop logic program using PROLOG	Applying
CLO4	Implement search strategies and game-playing algorithms for effective problem-solving in AI	Applying
CLO5	Apply probabilistic reasoning and Bayesian networks to handle uncertainty	Applying

Mapping of CLOs with Pos & PSOs

Course Learning Outcomes	Program Out comes (POs)												Program Specific Outcomes (PSOs)	
	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CLO1	M	M	H	L		L		L	L			L	L	L
CLO2	H	M	M	L		M		L		M		L		
CLO3	M	L	H	H	M	M	L	M			M	H	M	H
CLO4	H	H	H	M	M	H	M	H	M		M	H	H	M
CLO5	H	M	M	M		H	H	H	M		M	H	H	M

H: High, M: Medium, L: Low