Gyanmanjari Innovative University

Syllabus
Gyanmanjari Institute of Technology
Semester-4 (B. Tech.)

Subject: Artificial Intelligence: Foundations and Real-World Applications-BET1CE14310

Type of course: Professional Core

Prerequisite: Basic knowledge of calculus, linear algebra, probability, and programming (preferably Python). Familiarity with logic and fundamental algorithms is beneficial.

Rationale:

This course offers a thorough introduction to Artificial Intelligence, combining strong theoretical concepts with practical implementation. Students will learn key AI topics such as state- space problem-solving, heuristic search, knowledge representation, logic-based reasoning, game theory, automated planning, and expert systems. The course emphasizes hands-on coding, including implementing search algorithms, developing game-playing AIs, building rule-based expert systems, and programming in Prolog. Through case studies and project-based learning, students will gain the skills required to design and evaluate intelligent systems, preparing them for advanced studies and AI-related careers.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P	C	SEE	CCE	
4	0	2	5	100	50	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; CCE-Continuous and Comprehensive Evaluation.

Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	**Introduction to AI and Problem-Solving through Search** ***Theory Topics:*** AI Foundations: Definition, History, and Evolution of AI. Intelligent Agents and Environments. Problem-Solving Fundamentals: Problem Formulation, State-Space Representation. Uninformed Search Strategies: Breadth-First Search (BFS), Depth-First Search (DFS), Depth-Limited Search (DLS), Iterative Deepening DFS (IDDFS). Informed (Heuristic) Search Strategies: Greedy Best-First Search, A* Search Algorithm. Constraint Satisfaction Problems (CSPs): Formulation and Solving Techniques. ***Practical:*** 1. Implement BFS to Find the Shortest Path in a Grid Maze. 2. Implement DFS to Explore All Paths in a Maze. 3. Implement Depth-Limited Search (DLS) for a Tree. 4. Implement Iterative Deepening DFS (IDDFS). 5. Implement Greedy Best-First Search with a Custom Heuristic. 6. Implement A Search Algorithm* 7. Compare BFS, DFS, and A on the Same Problem*. 8. Solve N-Queens Using Backtracking (CSP) 9. Implement a Simple Graph Coloring Solver 10. Solve a Real-World CSP: Sudoku Solver	18	20%

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Search Strategy Simulation** Algorithm selection, implementation and performance analysis will be tested.	20	
2	**Active Learning Assignment AI Detective Lab** Students act as AI engineers to build a digital detective system that solves a mystery using search algorithms and CSP techniques. They apply uninformed and informed search to find suspects and clues, then use CSP methods to validate the final solution.		10
	Total	20	10

Examination Style:

Search Strategy Simulation (20 Marks)
Students will be given a complex scenario like "Design an optimal delivery route system for a logistics company that considers traffic, delivery windows, and vehicle capacity constraints." They must select the most appropriate search algorithm, implement it, and justify their choice with performance analysis comparing against alternative approaches.

AI Detective Lab (10 Marks)
Students act as AI engineers building a digital detective system that solves a multi-stage mystery using core AI concepts. They first explore AI foundations by tracing how early problem-solving evolved into modern search-based reasoning, then model the detective as an intelligent agent navigating an environment of suspects, clues, and evidence. The mystery world is represented as a state space, where students apply uninformed search (BFS, DFS, DLS, IDDFS) to uncover basic clues, followed by informed search (Greedy Best-First, A*) using custom heuristics to identify the answers efficiently. Producing a complete AI-driven case solution supported by algorithm comparisons, visualizations, and a final justification report in pdf format and have to upload it on GMIU web portal.

Knowledge Representation & Reasoning

Theory Topics:

Knowledge Representation Paradigms: Semantic Networks, Frames, Scripts. Logic and Automated Reasoning: Propositional Logic, First-Order, Predicate Logic. Inference Mechanisms: Unification, Forward Chaining, Backward Chaining. Resolution in Predicate Logic. Knowledge Engineering techniques.

Practical:

11. Build a Semantic Network for Family Relationships
12. Implement a Frame-Based System for Vehicle Classification
13. Create a Simple Propositional Logic Knowledge Base
14. Build a First-Order Logic Knowledge Base
15. Implement Unification Algorithm
16. Build a Forward Chaining Inference Engine
17. Implement a Backward Chaining Inference Engine.

2	18	20%

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Knowledge Base Architect:** Students will be building a university course management system and explaining their knowledge representation choices	20	
2	**Active Learning Assignment Logic Puzzle Solver:** To help students understand knowledge representation through hands-on, inquiry-based activities that simulate real-world reasoning Applications.		10
	Total	20	10

Examination Style:

Knowledge Base Architect Challenge (20 Marks)

	Students will design and implement a knowledge base for a university course management system. They must represent entities, relationships, and write reasoning procedures to answer complex queries about courses, prerequisites, schedules, and faculty assignments. **Logic Puzzle Solver (10 Marks)** By the end of this activity, students will be able to: Understand key knowledge representation concepts, Apply logical reasoning techniques, Implement inference mechanisms, and Use different representation schemes for problem-solving. 1) Choose a Logic Puzzle (e.g., Wumpus World, Zebra Puzzle, and Family Relationships). 2) Design Knowledge Representation: Identify entities, properties, and relationships. Choose appropriate representation scheme. 3) Implement Reasoning: Apply inference mechanisms to solve the puzzle. 4) Document the Process: Explain the representation choices and reasoning steps in pdf and upload it on GMIU Web Portal.		
3	**Advanced AI: Game Playing, Planning, and Expert Systems** ***Theory Topics:*** Adversarial Search & Game Theory: Minimax Algorithm, Alpha-Beta Pruning. Automated Planning: STRIPS representation, Goal Stack Planning, Hierarchical Planning. Expert Systems (ES): Architecture (Knowledge Base, Inference Engine, UI), Development Lifecycle. Knowledge Engineering: Techniques for acquiring and representing expert knowledge. Real-world Applications: Case studies in medical diagnosis, fault detection, and financial analysis. ***Practical:*** 18. Build a Tic-Tac-Toe AI using the minimax algorithm. 19. Implement a basic goal-stack planner for the "Blocks World" problem. 20. Create a basic Movie Recommendation Expert System. 21. Design and develop a prototype expert system for plant disease diagnosis. 22. Extend game playing to more complex games like Connect Four. 23. Implement hierarchical planning for a logistics domain. 24. Paper-Rock-Scissors AI with Simple Strategy	18	25%

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	**AI Game & Plan Showcase** Students will implement advanced game AI and planning systems for complex scenarios.	20	
2	**Active Learning Assignment Expert System Prototype** Simulate real-world expert system development where students design rule-based systems, populate knowledge bases, and implement Inference engines for domain-specific problem-solving.		10
	Total	20	10

Examination Style:

AI Game & Plan Showcase (20 Marks)
A comprehensive project where students: Implement an AI for Connect Four using Minimax with Alpha-Beta pruning, Develop a planning system for warehouse logistics, Analyze algorithm performance and decision quality, Submit complete code with detailed documentation.

Expert System Prototype (10 Marks)
Students will individually design and implement a small rule-based Expert system. They must: Choose a specific domain, Develop the knowledge base with at least 20 rules, Implement forward or backward chaining, Test the system with sample cases, Document the design in pdf and Upload on GMIU portal.

4	**Symbolic Programming with Prolog** ***Theory Topics:*** Introduction to Logic Programming: The Prolog paradigm. Prolog Basics: Facts, Rules, Queries, Variables, and Unification. Controlling Execution: Backtracking, Recursion, Cut (!). Data Structures: Lists, Trees, and their manipulation. Solving Problems with Prolog: Using Prolog for puzzles, symbolic reasoning, and implementing small expert systems.	18	20%

Practical:

25. Write Prolog programs to define family relationships and perform queries
26. Implement recursive procedures on lists (member, append, reverse).
27. Implement simple Prolog Calculator.
28. Implement Simple Prolog Database Query System
29. Solve logic puzzles (e.g., Sudoku, Zebra Puzzle) using Prolog's built-in
30. Implement tree traversal algorithms in Prolog backtracking. Prolog Puzzle: Who Owns the Fish?
31. Implement Prolog To-Do List Manager

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Prolog Logic Mastermind** Students solve complex symbolic reasoning problems using Prolog's full capabilities.	20	
2	**Debug the Prolog Program** Students develop critical thinking and debugging skills to identify and correct errors in Prolog code, and upload their corrected solutions on the GMIU portal Individually.		10
	Total	20	10

Examination Style:

Prolog Logic Mastermind (20 Marks)
Students are given a complex symbolic reasoning problem (e.g., course scheduling with multiple constraints, complex family relationship Inference). They must write a complete Prolog program demonstrating Mastery of facts, rules, recursion, lists, and backtracking control.

Debug the Prolog Program (10 Marks)

Provide a faulty Prolog program with intentional 10 mistakes (logical, syntactic, and semantic). Students work is to identify and correct: Incorrect unification patterns, Missing base cases in recursion, Improper use of cut operator, Logical inconsistencies in rules.

5	**AI System Integration and Future Trends** ***Theory Topics:*** AI System Architecture: Integrating multiple AI components (search, knowledge, planning) into a cohesive system. Introduction to Machine Learning: Linking symbolic AI with statistical learning (overview of Neural Networks, Supervised/Unsupervised Learning). AI Ethics and Trust: Bias in AI, Explainable AI (XAI), Fairness, Accountability. Current Trends: Overview of Natural Language Processing, Computer Vision, and Robotics. ***Practical:*** 32. Capstone Project: Integrate search, knowledge base, and planning components into unified AI system. 33. Use scikit-learn to train a simple classifier. 34. Analyze case studies for ethical issues and propose mitigations 35. Implement a simple NLP text processing component. 36. Create an AI system design document for a real-world problem. **Evaluation Method:** **Sr. No. | Evaluation Methods | SEE | CCE** 1 | **Capstone Project:** Comprehensive evaluation of integrated AI systems demonstrating full-stack AI development skills. | 20 | 2 | **AI Ethics Debate & Position Paper** Students engage in critical analysis of AI ethics through debate and written argumentation. | | 10 | **Total** | 20 | 10 ***Examination Style:*** **Capstone Project (20 Marks)** Students work in team to build an integrated AI system (e.g., smart course advisor, personalized learning planner). The project must incorporate: Search algorithms for recommendation, Knowledge representation for	18	15%

	domain modeling, Planning for goal achievement, Prolog components for rule-based reasoning, Ethical considerations documentation. **AI Ethics Debate & Position Paper (10 Marks)** Students participate in structured debates on AI ethics topics (e.g., "Autonomous Weapons Systems", "AI in Hiring Decisions"). They submit position papers arguing their stance with technical and ethical reasoning, demonstrating critical thinking about AI's societal impact.		

Suggested Specification Table:

Distribution of Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	10%	10%	20%	10%	10%	40%

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from the above table.

Course Outcome:

After learning the course, the students should be able to:	
CO1	Understand AI fundamentals, history, and intelligent agent types.
CO2	Formulate real-world problems as state-space or constraint satisfaction models and apply appropriate search strategies (uninformed, informed) to solve them efficiently.
CO3	Design knowledge representation schemes using logic and apply inference mechanisms for automated reasoning.
CO4	Develop AI agents for adversarial game playing using algorithms like Minimax and Alpha-Beta pruning, and create basic automated planning systems.
CO5	Build expert systems and implement logic-based programs in Prolog using recursion, backtracking, and list manipulation.

Instructional Method:

The course delivery method will depend upon the requirement of content and needs of students. The teacher, in addition to conventional teaching methods by black board, may also use any tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 15% topics are suggested for flipped mode instruction. Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory.

The internal evaluation will be done on the basis of the Active Learning Assignment. Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in the laboratory.

Reference Books:

[1] Artificial Intelligence: A Modern Approach, 4th Edition by Stuart Russell and Peter Norvig, Pearson.

[2] Prolog Programming for Artificial Intelligence by Ivan Bratko, Pearson Education.

[3] Artificial Intelligence: Structures and Strategies for Complex Problem Solving by George F. Luger, Pearson.

[4] Introduction to Artificial Intelligence by Wolfgang Ertel, Springer.

[5] Building Expert Systems by Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat, Addison-Wesley.

Suggested Assessment Guidelines

Module-1: Introduction to AI & Problem Solving Through Search

Search Strategy Simulation (20 Marks)		
Criteria	**Description**	**Marks**
Problem Understanding	Correct formulation of the given scenario as a state-space or CSP model. Clear identification of states, actions, and constraints.	5
Algorithm Selection & Justification	Appropriate choice of search algorithm (BFS/DFS/A*/Greedy/CSP) with logical justification based on problem characteristics.	5
Implementation & Correctness	Efficient and error-free implementation. Algorithm produces correct outputs for given test cases.	5
Performance Analysis & Comparison	Clear comparison of algorithms in terms of nodes expanded, time/space complexity, and path optimality. Visualizations included if required.	5
	Total	20

Module-2: Knowledge Representation & Reasoning

Knowledge Base Architect (20 Marks)		
Criteria	**Description**	**Marks**
Knowledge Base Design	Use of appropriate KR scheme (semantic networks/frames/logic) with accurate representation of entities and relationships.	5
Inference Mechanism	Correct implementation of forward/backward chaining or resolution. Inference steps are logically valid and complete.	5
Query Handling	System correctly answers complex queries about the domain. Queries are handled efficiently and accurately.	5
Documentation & Justification	Clear explanation of design choices, reasoning process, and validation of results. Code is well-commented and structured.	5
	Total	20

Module-3: Advanced AI – Game Playing, Planning & Expert Systems

AI Game & Plan Showcase (20 Marks)		
Criteria	**Description**	**Marks**
Game AI Functionality	Minimax/Alpha-Beta algorithm correctly implemented. AI plays game optimally without errors.	5
Planning System Correctness	STRIPS/Goal Stack/Hierarchical planner works as intended and generates valid plans for given scenarios.	5
Performance & Analysis	Analysis of AI decision-making, pruning efficiency, plan optimality, and computational trade-offs.	5
Code Quality & Documentation	Clean, modular code with clear comments. Complete report with implementation details and testing results.	5
	Total	20

Module-4: Symbolic Programming using Prolog

Prolog Logic Mastermind (20 Marks)		
Criteria	**Description**	**Marks**
Program Correctness	Prolog program runs without errors and produces correct outputs for all queries.	5
Use of Prolog Features	Effective use of recursion, backtracking, lists, cuts, and unification as required.	5
Logical Reasoning	Program demonstrates sound logical reasoning and correctly solves the given symbolic problem.	5
Code Structure & Clarity	Well-organized code with meaningful predicates and clear logic flow. Comments explain key sections.	5
	Total	20

Module-5: AI System Integration & Future Trends

Capstone Project (20 Marks)		
Criteria	**Description**	**Marks**
System Integration	Successful integration of multiple AI components (search, KR, planning, Prolog) into a cohesive system.	5
Functionality & Workflow	System works end-to-end with clear input/output flow. All components function as intended.	5
Testing & Validation	Comprehensive testing with varied inputs. System handles edge cases and produces valid results.	5
Documentation & Ethical Consideration	Clear project documentation, including design choices, limitations, and ethical implications discussed.	5
	Total	20