CS-430/730 | COURSE OUTLINE

Artificial Intelligence

Course code	CS-430/730 (undergraduate section CS-430; graduate section CS-730)
Credit hours	3 (2 + 1): two lecture hours and one three-hour laboratory per week
Levels	BS (final year) | MS | PhD — one course, three levels of assessment
Prerequisites	Data structures and algorithms; discrete mathematics; probability; linear algebra; programming in Python.
Instructor	Dr. Asif Mehmood, Department of Computer Science
Contact	asifnazquaidian@gmail.com
Delivery	Lectures with active learning, weekly programming laboratory, problem sets, graduate research project
Textbook	Artificial Intelligence: Search, reasoning, uncertainty, decision making and learning machines, from foundations to foundation models (course textbook, October 2026), with the reference books below.

1 Course description

Artificial intelligence is the study of agents that perceive their environment and act to achieve goals. This course develops the field's core methods with mathematical rigour and working code. It begins with the agent framework and with search: uninformed and heuristic search, optimality proofs for A*, local and evolutionary optimisation, adversarial search from minimax and alpha–beta pruning to Monte Carlo tree search and AlphaZero, and constraint satisfaction. It then treats knowledge and reasoning: propositional and first-order logic, satisfiability, and fuzzy logic for approximate reasoning. The second half handles uncertainty and decisions: Bayesian networks and their inference, hidden Markov models and Kalman and particle filters, Markov decision processes, and reinforcement learning from temporal-difference learning to deep and policy-gradient methods. The course closes with the neural foundations of modern AI, transformers and large language models, reasoning and agentic systems, brain-inspired AI, and the safety, ethics and governance of AI. The coverage and level follow leading courses such as Stanford CS221, MIT 6.034 and Berkeley CS188 and the standard text of Russell and Norvig; graduate students add proofs, primary literature and a research project.

2 Course learning outcomes (CLOs)

Outcomes are written at the cognitive levels of Bloom's revised taxonomy and mapped to the programme learning outcomes (PLOs) of the host programme (for computing degrees, the Seoul Accord graduate at-

CLO	Students will be able to…	Bloom level	PLO
CLO1	Formulate real problems as search, constraint-satisfaction, logical, probabilistic or decision problems for an appropriate agent design.	C3 Apply	PLO1, PLO2
CLO2	Implement and analyse search, game-playing and constraint algorithms, including their completeness, optimality and complexity.	C4 Analyse	PLO2, PLO3
CLO3	Represent knowledge and reason with propositional, first-order and fuzzy logic, and with Bayesian networks and temporal models.	C3 Apply	PLO2
CLO4	Solve sequential decision problems with dynamic programming and reinforcement learning and evaluate the learned policies.	C4 Analyse	PLO2, PLO5
CLO5	Explain the architecture, training and limitations of transformer-based foundation models and evaluate AI systems for safety, fairness and regulatory compliance.	C5 Evaluate	PLO6, PLO8
CLO6	Design, implement and report an original AI study with a clear hypothesis, baselines and a reproducible evaluation.	C6 Create	PLO4, PLO9

tributes used by NCEAC).

3 Weekly schedule

Sixteen weeks: fourteen teaching weeks, a mid-term week and a final-examination week. Each teaching week has two lecture hours and one three-hour computational laboratory.

Week	Topic (textbook chapter)	Laboratory	Assessment / CLO
1	Intelligent Agents and the Foundations of AI (Ch. 1)	Agents in a grid world	Diagnostic quiz
2	Problem Solving by Search (Ch. 2)	Uninformed search on the 8-puzzle	Problem set 1 (CLO1, CLO2)
3	Informed Search and Heuristics (Ch. 3)	A* with different heuristics	Programming lab 1 (CLO2)
4	Local Search, Optimisation and Evolutionary Computation (Ch. 4)	Annealing and evolving on a rugged landscape	Problem set 2 (CLO2)
5	Adversarial Search and Games (Ch. 5)	Minimax, alpha–beta and MCTS for tic-tac-toe	Programming lab 2 (CLO2)
6	Constraint Satisfaction Problems (Ch. 6)	Backtracking with heuristics: Sudoku and n-queens	Quiz 1 (CLO1, CLO2)
7	Logic, Satisfiability and Knowledge Representation (Ch. 7)	A DPLL solver and the 3-SAT phase transition	Problem set 3 (CLO3); project proposal (graduate)
8	**Mid-term examination (Chapters 1–7)**	–	Mid-term (CLO1–CLO3)
9	Fuzzy Logic and Approximate Reasoning (Ch. 8)	A Mamdani fuzzy controller from scratch	Programming lab 3 (CLO3)

Week	Topic (textbook chapter)	Laboratory	Assessment / CLO
10	Probabilistic Reasoning and Bayesian Networks (Ch. 9)	Inference in the burglary network	Problem set 4 (CLO3)
11	Reasoning over Time: Hidden Markov Models and Filters (Ch. 10)	HMM filtering and a particle filter	Programming lab 4 (CLO3)
12	Markov Decision Processes (Ch. 11)	Value iteration and policy iteration in the grid world	Problem set 5 (CLO4)
13	Reinforcement Learning (Ch. 12)	Q-learning and SARSA on cliff walking	Programming lab 5 (CLO4)
14	Neural Networks, Transformers and Large Language Models (Ch. 13)	Self-attention and a tiny language model from scratch	Quiz 2 (CLO4, CLO5)
15	Brain-Inspired AI, Safety, Ethics and Governance (Ch. 14)	A fairness audit	Project presentations (CLO6); policy brief (CLO5)
16	**Final examination (all chapters)**	–	Final (CLO1–CLO5); project report (CLO6)

4 Assessment

Undergraduate (BS) and graduate (MS/PhD) students attend the same lectures but are assessed differently: graduate students complete the advanced (MS) and research (PhD) exercises and a research project.

BS students

Problem sets (5)	15%
Quizzes (2)	10%
Programming labs (5)	20%
Mid-term examination	20%
Final examination	35%

MS and PhD students

Problem sets with MS/PhD problems	15%
Programming labs (5)	10%
Critical review of two research papers	10%
Mid-term examination	15%
Research project (proposal, presentation, report)	25%
Final examination	25%

Graduate research project

Graduate students choose a project in week 7 and present it in week 15. A project poses a question and answers it empirically with baselines, ablations and error bars, or theoretically with proofs. Examples: comparing A*, weighted A* and learned heuristics on the 15-puzzle; Monte Carlo tree search with a learned value function for a small board game; exact and approximate inference in a Bayesian network for clinical diagnosis; deep Q-learning versus policy gradients on a control task, with sample-efficiency curves; evaluating a small open language model for hallucination or bias with a reproducible benchmark; a fuzzy controller tuned by a genetic algorithm. Deliverables: a two-page proposal, a 15-minute presentation and a report in conference format (6–8 pages) with code. PhD reports must contain a clearly stated hypothesis and a comparison with the literature.

Grading rubric for laboratory reports and projects

Criterion	Excellent (4)	Satisfactory (2–3)	Needs work (0–1)
Model and method	Correct equations, algorithms and parameters, justified choices	Minor errors that do not change conclusions	Wrong model or unexplained method
Code and results	Runs, reproducible, figures labelled with units, baselines and error bars	Runs with small gaps	Does not run or results missing
Interpretation	Links results to theory, physiology or the literature	Describes results without depth	No interpretation
Communication	Clear structure, correct references	Understandable, some gaps	Unclear or unreferenced

5 Reading list

Course textbook

Artificial Intelligence: Search, reasoning, uncertainty, decision making and learning machines, from foundations to foundation models. Dr. Asif Mehmood, October 2026.

Reference books and papers

1. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021 (principal text).
2. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018 (free online).
3. D. Koller and N. Friedman, *Probabilistic Graphical Models.* MIT Press, 2009; S. J. D. Prince, *Computer Vision: Models, Learning, and Inference.* Cambridge University Press, 2012 (Chapters 2–4, 10–11, 19).
4. T. J. Ross, *Fuzzy Logic with Engineering Applications*, 2nd ed. Wiley, 2004 (Chapters 1–5, 13).
5. I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning.* MIT Press, 2016; S. J. D. Prince, *Understanding Deep Learning.* MIT Press, 2023.
6. D. Kriesel, *A Brief Introduction to Neural Networks*, 2007; V. B. Rao and H. V. Rao, *C++ Neural Networks and Fuzzy Logic*, 2nd ed., 1995.
7. Primary papers named in each chapter, including Turing (1950), Hart, Nilsson and Raphael (1968), Pearl (1988), Watkins and Dayan (1992), Silver *et al.* (2016, 2018), Vaswani *et al.* (2017) and Ouyang *et al.* (2022).

6 Course policies

- Attendance of at least 80% of lectures and laboratories is required to sit the final examination.

- Problem sets and labs are individual work. Ideas may be discussed, but code and text must be written and understood by the student. Use of AI assistants must be acknowledged with a short statement of how they were used; submitting generated work as one's own is academic misconduct.
- Late submissions lose 10% per day and are not accepted after five days without a documented reason.
- Plagiarism is handled according to the institution's academic-integrity policy and the HEC plagiarism policy.
- Students with a disability or special need should contact the instructor in the first two weeks so that reasonable adjustments can be made.