Module (Course Syllabus) Catalogue

2023-2024

College/ Institute	Erbil Technical Engineering College	
Department	Information System Engineering	
Module Name	Artificial Intelligent	
Module Code	ARI702	
Degree	Technical Diploma ☐ Bachelor ☒ High Diploma ☐ Master ☐ PhD ☐	
Semester	7	
Qualification	PhD In Computer Engineering	
Scientific Title	Lecturer	
ECTS (Credits)	6	
Module type	Prerequisite ☐ Core ☒ Assist. ☐	
Weekly hours		
Weekly hours (Theory)	(2)hr Class	(84)Total hrs Workload
Weekly hours (Practical)	(2)hr Class	(78)Total hrs Workload
Number of Weeks	14	
Lecturer (Theory)	Shahab Wahhab Kareem	
E-Mail & Mobile NO.	Shahab.kareem@epu.edu.iq	
Lecturer (Practical)	Soran bakhtiyar	
E-Mail & Mobile NO.		
Websites		

Course Book

Course Description	Principles of knowledge-based search techniques, automatic deduction, knowledge representation using predicate logic, machine learning, probabilistic reasoning. Applications in tasks such as problem solving, data mining, game playing, natural language understanding, computer vision, speech recognition, and robotics.
Course objectives	• **Scientific goal: to understand the principles that make intelligent behavior possible in natural or artificial systems analyze natural and artificial agents··, formulate and test hypotheses about what it takes to construct intelligent agents··, design, build, and experiment with computational systems that perform tasks that require intelligence** • **Engineering goal: design useful, intelligent artifacts. Analogy between studying flying machines and thinking machines.**
Student's obligation	
Required Learning Materials	

	Task		Weight (Marks)	Due Week	Relevant Learning Outcome
Evaluation	Paper Review				
	Assignments	Homework	5	5	
		Class Activity	2	7	
		Report	5	8	
		Seminar			
		Essay			
		Project	5	11	
	Quiz		8	4	
	Lab.		10	6	
	Midterm Exam		25	7	

	Final Exam	40	12	
	Total	100		
Specific learning outcome:	1- Uninformed Search Methods – Be able to formulate problem solving tasks as searching a state space graph, problem representation in terms of states, goal test, operators, state-space graph search formulation, closed world assumption, expanding a node, frontier list, partial solution path, solution path, search tree, breadth-first search, depth-first search, chronological backtracking, uniform-cost search, iterative-deepening search, bidirectional search, completeness, optimality, admissibility, time and space complexity, detecting repeated states, explored list. 2- Informed Search Methods – Understand heuristic functions, evaluation functions, best-first search, greedy best-first search, beam search, algorithm A, algorithm A*, admissible heuristic, consistent heuristic, better informed heuristic, devising heuristics. 3- Local Search Methods – Local search problem formulation, operators, neighborhood, move set, hill-climbing algorithm, local optima problem, hill-climbing with random restarts, stochastic hill-climbing (simulated annealing) algorithm, escaping local optima, Boltzman's equation, cooling schedule, genetic algorithms, crossover, mutation, fitness function, proportional fitness selection, population, crowding. 4- Game Playing – Zero-sum games, perfect information games, deterministic vs. stochastic games, game playing as search, search tree, branching factor, ply, minimax principle, minimax algorithm, static evaluation function, alpha-beta pruning, cutoff, alpha-beta pruning algorithm, best case and worst case of alpha-beta vs. minimax, iterative-deepening with alpha-beta, horizon effect, quiescence search, representing non-deterministic games, chance nodes, expectimax value, Monte Carlo tree search. 5- Constraint Satisfaction - Problem formulation in terms of variables, domains and constraints, constraint graph, depth-first search, backtracking with consistency checking, most constrained variable heuristic, most constraining variable heuristic, least constraining value heuristic, minconflicts heuristic, min-conflicts algorithm, forward checking algorithm, arc consistency algorithm (AC-3). 6- Unsupervised Learning – Inductive learning problem, unsupervised learning problem, feature space, feature, attribute, examples, labels, classes, training set, testing set, classification problems, inductive bias, preference bias, hierarchical agglomerative clustering algorithm, single linkage, complete linkage, average linkage, dendrogram, k-means clustering algorithm, cluster center, distortion cluster quality.			
Course References:	• Artificial Intelligence: A Modern Approach Stuart Russell and Peter Norvig 3rd Edition, 2009 – Recommended but not required. – Older editions are also useable---but you will have to search the text for the relevant sections – • Alternate Book: Computational Intelligence: A Logical Approach by David Poole and Alan Mackworth.			

	• *Artificial Intelligence: Foundations of Computational Agents*, second edition, Cambridge University Press 2017

Course topics (Theory)	Week	Learning Outcome
introduction to artificial intelligence and the role of agents.	1	describe what an intelligent agent is identify the goals of Artificial Intelligence classify the inputs and the outputs of various agents
dimensions of complexity. applications domains.	2	
introduction to knowledge representation.	3	characterize simplifying assumptions made in building AI systems determine what simplifying assumptions particular AI systems are making suggest what assumptions to lift to build a more intelligent system than an existing one
agent architecture and control	4	
hierarchical control.	5	Chapter 3 presents the search techniques covered in the lectures as well as other ideas not covered.
searching and graphs.	6	
uninformed search strategies.	7	define a directed graph represent a problem as a state-space graph
bounded search, iterative deepening, branch and bound.	8	
refinements to search strategies, including loop checking, multiple-path pruning, bidirectional search, and dynamic programming.	9	explain how a generic searching algorithm works demonstrate how depth-first search will work on a graph demonstrate how breadth-first search will work on a graph predict the space and time requirements for depth-first and breadth-first searches
heuristic search, including best-first search and A* search.	10	devise an useful heuristic function for a problem demonstrate how best-first and A ∗ search will work on a graph predict the space and time requirements for best-first and A ∗ search
Constraint satisfaction problems and consistency algorithms (arc consistency).	11	explain how cycle checking and multiple-path pruning can improve efficiency of search algorithms explain the complexity of cycle checking and multiple-path pruning for different search algorithms justify why the monotone restriction is useful for A ∗ search predict whether forward, backward, bidirectional or island-driven search is better for a particular

local search, randomized algorithms and genetic algorithms for solving CSPs.		problem demonstrate how dynamic programming works for a particular problem
propositional reasoning and definite clauses. bottom-up proof procedure.	12	show how constraint satisfaction problems can be solved with generate-and-test show how constraint satisfaction problems can be solved with search explain and trace arc-consistency of a constraint graph show how domain splitting can solve constraint problems
top-down proof procedure.	13	explain the model of deterministic planning represent a problem using the STRIPs representation of actions.
action semantics and representations.	14	Explain the components and the architecture of a learning problem Explain why a learner needs a bias Identify the sources of error for a prediction
Practical Topics	**Week**	**Learning Outcome**
Introduction to python **Classes and Functions**	1,2	*Download anaconda* *Identify the classes and function*
Files and exceptions	3,4	*Read , write file*
Understanding Object-Oriented Basics	5,6	*Identify the Object-Oriented Basics*
Object-Oriented basics	7,8	*Object-Oriented Basics*
Machin Learning	9,10	Implement the Classification algorithms
Machin Learning	11,12	Implement the gaming
Machin Learning	13,14	Implement the clustering

Questions Example Design

Give some real-world applications of AI.

There are various real-world applications of AI, and some of them are given below:

- **Google Search Engine:** When we start writing something on the google search engine, we immediately get the relevant recommendations from google, and this is **because of different AI technologies**.

- **Ridesharing Applications:** Different ride-sharing applications such as Uber uses AI and machine learning to determine the type of ride, minimize the time once the car is hailed by the user, price of the ride, etc.
- **Spam Filters in Email:** The AI is also used for email spam filtering so that you can get the important and relevant emails only in your inbox. As per the studies, Gmail successfully filters 99.9% of spam mails.
- **Social Networking:** Different social networking sites such as Facebook, Instagram, Pinterest, etc., use the AI technology for different purposes such as face recognition and friend suggestions, when you upload a photograph on Facebook, understanding the contextual meaning of an emoji in Instagram, and so on.
- **Product recommendations:** When we search for a product on Amazon, we get the recommendation for similar products, and this is because of different ML algorithms. Similarly, on Netflix, we get personalized recommendations for movies and web series.

Extra notes:

External Evaluator

Dr. roojwan