(Affiliated to University of Mumbai)	***End Semester Examination (R-24/R-25) SH 2026 Answer Key with Marking Schema***
Branch:AI & DS	Course: Artificial Intelligence
Year/ Semester: TE-V	Course code:AIDSC501
Time: 03 hours	Marks: 80

Note: 1. All questions are compulsory. 2. Figures to right indicate full marks. 3. Assume suitable data wherever necessary.		Marks	CO	BL
Q. 1	All questions are compulsory.	**30**		
A.		05	CO1	2

Basis	AI	Machine Learning	Deep Learning
Meaning	Broad field of making machines perform tasks requiring intelligence.	Subset of AI; systems learn patterns from data.	Subset of ML using multi-layer neural networks.
Approach	Rules, search, reasoning, planning, learning, etc.	Learns a model from examples/data.	Learns hierarchical features from data.
Compute	Varies by technique.	Moderate to high, depending on model.	Often high, commonly uses GPUs.
Data / features	May or may not require training data.	Usually requires data and feature preparation.	Often benefits from large datasets; can learn features automatically.

1 mark each for five valid comparison points

<table>
<tr><td>B.</td><td>

Performance measure Correct diagnosis; patient safety; suitable treatment recommendation; low diagnostic error/cost; timely results. 1Mark
Environment Patients, diseases/conditions, hospital/clinic, laboratory, medical records and clinical guidelines. 1Mark
Actuators Display diagnosis and recommendations; request tests; generate alerts/referrals; update records. 1Mark
Sensors Symptoms and patient history; examination findings; lab/imaging results; vital signs; electronic health records. 1Mark
Complete PEAS identification / clarity Clearly labels all four PEAS components and relates them to diagnosis. 1Mark

</td><td>05</td><td>CO1</td><td>2</td></tr>
<tr><td>C.</td><td>

Local maximum (1 mark): the search reaches a state better than its immediate neighbours but not the global optimum.
Plateau (1 mark): neighbouring states have the same evaluation value, so progress may stop or become aimless.
Ridge (1 mark): the best direction may require a sequence of sideways/indirect moves that simple local moves cannot follow easily.
Random restart (1 mark): run hill climbing again from a newly selected random initial state.
Effect (1 mark): different starting points can escape a previous local maximum/plateau and increase the chance of finding a better or global solution; it does not guarantee it.

</td><td>05</td><td>CO2</td><td>3</td></tr>
<tr><td>D.</td><td>
<table>
<tr><th>Component</th><th>Role</th><th>Marks</th></tr>
<tr><td>Knowledge Base (KB)</td><td>Stores domain facts, rules, and relationships.</td><td>1.5</td></tr>
<tr><td>Inference Engine</td><td>Applies logical rules to facts in the KB to infer new facts or conclusions; may use forward or backward chaining.</td><td>1.5</td></tr>
<tr><td>Perception Subsystem</td><td>Receives and interprets percepts from sensors or inputs and converts them into information usable by the agent or KB.</td><td>1</td></tr>
<tr><td>Interaction</td><td>Perception supplies facts; the inference engine reasons over the KB; resulting conclusions support decisions and actions.</td><td>1</td></tr>
</table>
</td><td>05</td><td>CO3</td><td>2</td></tr>
</table>

<table>
<tr>
<td>E.</td>
<td>
<table>
<tr><th>Forward Chaining</th><th>Backward Chaining</th><th>Marks</th></tr>
<tr><td>Approach: Data-driven; starts from known facts and fires applicable rules.</td><td>Approach: Goal-driven; starts from a query or goal and seeks supporting facts and rules.</td><td>1</td></tr>
<tr><td>Limitation: May generate many irrelevant intermediate facts and waste time when the goal is narrow.</td><td>Limitation: May explore many subgoals or become inefficient when many alternative rules exist.</td><td>1</td></tr>
<tr><td>Memory usage: Can consume memory as inferred facts accumulate.</td><td>Dependency: Requires suitable rules and facts to support the goal; missing facts can cause failure.</td><td>1</td></tr>
<tr><td>Termination: May not stop early unless a goal is reached or no rules apply.</td><td>Limitation: Repeated subgoals and cycles require loop handling and may cause repeated checks.</td><td>1</td></tr>
<tr><td>Conclusion: Suitable when available facts are used to derive multiple possible conclusions.</td><td>Conclusion: Suitable when checking whether a specific goal can be established.</td><td>1</td></tr>
</table>
</td>
<td>05</td><td>CO3</td><td>2</td>
</tr>
<tr>
<td>F.</td>
<td>
Bellman Expectation Equation for state-value $V\pi(s)$ (5 marks)
For a policy π:

$V\pi(s) = \Sigma_a \pi(a|s) \Sigma_{s'} P(s'|s,a) [R(s,a,s') + \gamma V\pi(s')]$

• Correct equation and notation: 2 marks.
• Transition probability $P(s'|s,a)$: probability that action a in state s leads to next state s′; it weights each possible next-state outcome: 1 mark.
• Discount factor γ, where $0 \le \gamma \le 1$: controls the importance of future rewards relative to immediate rewards: 1 mark.
• Policy probability $\pi(a|s)$ and reward/value interpretation explained: 1 mark.
</td>
<td>05</td><td>CO4</td><td>2</td>
</tr>
<tr><td colspan="5"></td></tr>
</table>

Q.2 A	SectionMarksCriteriaPart 1: Tree Formulation & Left Subtree Evaluation3 Marks • Clear labeling of nodes/structure and initial bounds ($\alpha = -\infty$, □ = $+\infty$) (1 mark) • Accurate evaluation of node □ returning value *4* (1 mark) • Correct detection of $\alpha \geq \beta$ at node □ ($6 \geq 4$) and pruning of leaf 2 (1 mark) Part 2: Middle Subtree Cutoff & Right Subtree Evaluation4 Marks • Update of root $\alpha = 4$ and propagation down to □ and □ (1 mark) • Evaluation of □ returning *2*, causing $\beta = 2 \leq$ □ $= 4$ at node □ (1 mark) • Correct pruning of entire subtrees □ and □ (leaves *9,5,3,1*) (1 mark) • Evaluation of node □ returning *5* and setting $\beta = 5$ at node □ (1 mark) Part 3: Final Cutoff, Root Value & Summary3 Marks • Detection of cutoff at node □ ($7 \geq 5$) and pruning of right leaf 5 (1 mark) • Correct identification of all pruned nodes/leaves (1 mark) • Correct final Minimax value of the root node ($= 5$) (1 mark)	10	CO2	3
OR				
Q.2B	• Problem representation / chromosome: encode each candidate solution as a chromosome (binary string, numbers, permutation, etc.) — 1 mark. • Initial population: generate a set of candidate chromosomes, often randomly — 1 mark. • Fitness evaluation: compute a fitness score showing how well each candidate solves the problem — 2 marks. • Selection: choose parents with preference for fitter candidates (e.g., roulette-wheel or tournament selection) — 2 marks. • Crossover: combine parts of two parents to create offspring — 1.5 marks. • Mutation: randomly alter one or more genes to maintain diversity and explore new solutions — 1.5 marks. • Replacement/termination: form the next generation; repeat evaluation and genetic operators until a stopping criterion is met; return best solution — 1 mark. Workflow: Initialize population → Evaluate fitness → Selection → Crossover → Mutation → New population/replacement → termination check → best solution (otherwise repeat).	10	CO2	3

<table>
<tr><td>Q.3 A</td><td>Let R = it rains; W = grass is wet; P = picnic is cancelled; H = it is humid.

<table><tr><th>Statement</th><th>Logic form</th><th>CNF clause</th></tr><tr><td>If it rains, grass becomes wet</td><td>R → W</td><td>¬R ∨ W</td></tr><tr><td>If grass is wet, picnic is cancelled</td><td>W → P</td><td>¬W ∨ P</td></tr><tr><td>Either raining or humid</td><td>R ∨ H</td><td>R ∨ H</td></tr><tr><td>It is not humid</td><td>¬H</td><td>¬H</td></tr></table>Goal: prove P by refutation. Add negation of goal ¬P to the clause set.

<table><tr><th>Resolution step</th><th>Parent clauses</th><th>Resolvent</th></tr><tr><td>1</td><td>(R ∨ H), (¬H)</td><td>R</td></tr><tr><td>2</td><td>(¬R ∨ W), R</td><td>W</td></tr><tr><td>3</td><td>(¬W ∨ P), W</td><td>P</td></tr><tr><td>4</td><td>P, (¬P)</td><td>□ (empty clause)</td></tr></table>Since assuming ¬P leads to the empty clause, the assumption is contradictory. Therefore P is entailed: the picnic is cancelled.

Suggested allocation: symbols and translations (3); add negated goal (1); valid resolution steps (4); contradiction and conclusion (2).</td><td>10</td><td>CO3</td><td>4</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>Q.3B</td><td><table><tr><th>Point</th><th>Expected Content</th><th>Marks</th></tr><tr><td>Definition / Operation</td><td>Uses explicit facts and rules in a knowledge base with an inference mechanism to derive conclusions or actions.</td><td>2</td></tr><tr><td>Advantages</td><td>Explainability and traceable rules; explicit domain knowledge; consistency; ability to encode expert guidelines; easier auditing and updating of individual rules.</td><td>3</td></tr><tr><td>Critical-Domain Relevance</td><td>In medical diagnosis or compliance, decisions can be linked to rules or guidelines, supporting review, accountability, and policy checking.</td><td>2</td></tr><tr><td>Limitations</td><td>Knowledge acquisition bottleneck; costly expert maintenance; brittle or incomplete rules; difficulty handling uncertainty and noisy data without additional methods; limited adaptability.</td><td>2</td></tr><tr><td>Balanced Conclusion</td><td>Useful where transparency and rule compliance matter, but can be combined with statistical or machine learning methods rather than universally replacing them.</td><td>1</td></tr></table></td><td>10</td><td>CO3</td><td>4</td></tr>
</table>

Q.4 A	Definition, States & Actions (3 Marks) ● Definition (1 mark): An MDP models sequential decision-making where outcomes are stochastic and the next state depends on the current state and action. ● States □ (1 mark): All possible situations or configurations the agent can occupy. ● Actions □ (1 mark): The set of all valid decisions available to the agent in each state. Transition Function & Reward (4 Marks) ● Transition Function $P(s' \mid s, a)$ (2 marks): The conditional probability of reaching next state □′ after taking action □ in state □. ● Reward □(□, □, □′) or □(□, □) (2 marks): The immediate scalar feedback (positive reinforcement or penalty) returned by the environment after a state transition. Interaction Loop & Objective (3 Marks) ● Agent–Environment Loop (1.5 marks): The cyclical process where the agent observes state → takes action under a policy → environment returns next state + reward → repeat. ● Objective (1.5 marks): Find an optimal policy π that maximizes the expected cumulative (discounted) reward over time	10	CO4	3
	OR			
Q.4B	Required event: School Off O = T, Rain R = T, Humid H = F, Cloud C = T. The network factorizes as P(H) P(C) P(R|H,C) P(W|R) P(O|R). Since W (Wet Grass) is not part of the requested event, sum over its possible values: Σw P(W=w|R)=1. Therefore: P(O, R, ¬H, C) = P(¬H) × P(C) × P(R | ¬H, C) × P(O | R) = (1 − 0.2) × 0.6 × 0.7 × 0.7 = 0.8 × 0.6 × 0.7 × 0.7 = 0.2352. Answer: 0.2352 (23.52%). Suggested allocation: identify complements P(¬H)=0.8 (1); correct factorization / omit or marginalize W (3); substitute CPT values (3); arithmetic and final answer (3).	10	CO4	3

<table>
<tr><td>Q.5 A</td><td>• Definition: STRIPS represents planning states as predicates/facts and actions using preconditions, add effects and delete effects — 2 marks.
• Preconditions: conditions that must be true before an action can execute — 2 marks.
• Add list: facts made true after the action — 2 marks.
• Delete list: facts made false after the action — 2 marks.
• Correct example with all three parts and explanation of state update — 2 marks.
Example: action Move(A,B)
Preconditions: At(Robot,A) ∧ Connected(A,B)
Add list: At(Robot,B)
Delete list: At(Robot,A)
If the preconditions hold, executing Move removes At(Robot,A) and adds At(Robot,B); Connected(A,B) remains true.</td><td>10</td><td>CO5</td><td>2</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>Q.5B</td><td><table>
<tr><th>Part</th><th>Expected Answer</th><th>Marks</th></tr>
<tr><td>Dynamic Programming</td><td>Uses a known Markov Decision Process (MDP) model, including transition probabilities and rewards, along with Bellman backups to calculate state values and policies.</td><td>2</td></tr>
<tr><td>Policy Evaluation</td><td>For a fixed policy \(\pi\), estimates \(V^\pi\) by repeatedly applying the Bellman expectation update until values converge or a specified threshold is reached: \(\displaystyle V(s)\leftarrow\sum_a\pi(a\mid s)\sum_{s'}P(s'\mid s,a)[R+\gamma V(s')]\).</td><td>2.5</td></tr>
<tr><td>Policy Improvement</td><td>For each state, selects the action with the highest expected return using the current value estimates: \(\displaystyle \pi_{\text{new}}(s)=\arg\max_a\sum_{s'}P(s'\mid s,a)[R+\gamma V(s')]\).</td><td>2.5</td></tr>
<tr><td>Iteration / Termination</td><td>Repeats policy evaluation and policy improvement until the policy no longer changes, indicating a stable policy.</td><td>1</td></tr>
<tr><td>Flowchart</td><td>Draws a correct flowchart showing initialization, policy evaluation, policy improvement, a policy-stability decision, and a loop back when the policy is not stable.</td><td>2</td></tr>
</table></td><td>10</td><td>CO5</td><td>2</td></tr>
<tr><td colspan="5"></td></tr>
</table>

Q.6 A	Hardware Architecture — Sensors & Actuators (3 Marks) ● Sensors (1.5 marks): Depth/RGB cameras, LiDAR, wheel encoders, proximity/ultrasonic sensors, and gripper force/tactile sensors to gather environmental and contact data. ● Actuators (1.5 marks): Electric wheel motors for mobile base locomotion, articulated multi-axis robotic arm for reaching, and an adaptive gripper (mechanical/suction) for item grasping. Perception, Localization & Path Planning (4 Marks) ● Perception (1.5 marks): Computer vision and object detection models analyze camera feeds to recognize target packages, read barcodes, and identify obstacles. ● Localization & Mapping (1 mark): Fuses LiDAR, camera, and encoder data (SLAM) to build warehouse maps and estimate the robot's real-time pose (□, □, □). ● Path Planning & Navigation (1.5 marks): Uses global planners (□* / Dijkstra) on grid maps to compute shortest collision-free routes to target racks, with real-time local replanning if pathways are blocked. Decision-Making, System Integration & Safety (3 Marks) ● Decision-Making & Picking (1.5 marks): Orchestrates picking workflows—sequences target approach, aligns end-effector, applies optimal grasping force, and handles failure recovery. ● System Integration & Safety (1.5 marks): Executes a closed feedback loop:. Integrates safety stops for nearby workers and automated emergency handling	10	CO6	3
OR				

Q.6B	Data Preparation & Text Cleaning (3 Marks) ● Data and Labels (1 mark): Collect customer reviews categorized into positive, negative, and neutral, then split the dataset into training and test subsets. ● Text Cleaning & Normalization (2 marks): Lowercase text, strip HTML/noise/punctuation, tokenize, and apply lemmatization or stemming while deliberately preserving sentiment negations (e.g., "not"). Feature Representation & Model Training (4 Marks) ● Numerical Representation (2 marks): Transform text into feature vectors via Bag-of-Words or TF-IDF, fitting strictly on training data and transforming test data with the fitted vectorizer. ● Model Training (2 marks): Train a multi-class classification model (e.g., Multinomial Naive Bayes or Logistic Regression) using the extracted features and training labels. Testing, Evaluation & Pipeline Workflow (3 Marks) ● Evaluation & Testing (2 marks): Predict sentiment labels on unseen test data and evaluate performance using accuracy, precision, recall, macro F1-score, and a confusion matrix. ● Pipeline Trace (1 mark): Raw Review → Cleaning/Tokenization → Feature Vectorization → Model Inference → Predicted Class.	10	CO6	3
********** All the Best**********				