ARTIFICIAL INTELLIGENCE

(Common to CSE & IT)

Course Code: 19CT1112	**L**	**T**	**P**	**C**
	2	**0**	**2**	**3**

Course Outcomes: At the end of the course, a student will be able to:

CO1: Demonstrate various AI applications, languages and Intelligent Agents.

CO2: Solve problems using uninformed and informed search strategies.

CO3: Apply propositional logic techniques for knowledge representation.

CO4: Classify the real world tasks using Back Propagation algorithm.

CO5: Build expert systems for real world applications.

UNIT I: **(6 Lectures)**

INTRODUCTION: Definition of AI, Goals of AI, Turing Test, History and foundations of AI, Branches of AI, Applications of AI, categorization and components of AI, AI Programming languages.

INTELLIGENT AGENTS: Introduction, Intelligent Systems, the Concept of rationality, types of Agents, Environments and its properties, PEAS.

Learning Outcomes: At the end of the unit, student will be able to

1. classify various AI Applications. (L2)
2. list the AI Languages. (L1)
3. explain various types of Agents. (L2)

UNIT II: **(8 Lectures)**

PROBLEM SOLVING AND SEARCHING:

Introduction to Problem Solving, Problem Formulation, State Space Representation, Problem Formulation of real-world problems, Production System, Problem Characteristics, Solving problems by searching.

UNINFORMED & INFORMED SEARCH STRATEGIES:

Breadth-First Search, Depth First Search, Uniform Cost Search, Depth-Limited Search, Iterative Deepening Search, Bidirectional Search, Comparing Uniform Search Strategies, Hill Climbing, Best First Search, A* Search, AO* Search.

Learning Outcomes: At the end of the unit, student will be able to

1. demonstrate the state space search and control strategies techniques. (L2)
2. describe Characteristics of a Problem. (L2)
3. apply uninformed search techniques to problems. (L3)
4. apply informed search techniques to problems. (L3)

UNIT III: **(6 Lectures)**

ADVERSARIAL SEARCH STRATEGIES: Introduction, Optimal Strategies, The Minimax Algorithm, Alpha-Beta Pruning, Constraint Satisfaction Problem, Cryptarithmetic Problem.

KNOWLEDGE AND REASONING: Knowledge Representation Issues, Predicate Logic - Resolution, Unification, Representation Knowledge Using Rules-Inference in First-Order Logic, Forward and Backward Reasoning.

Learning Outcomes: At the end of the unit, student will be able to

1. develop game playing strategies using AI techniques. (L3)
2. solve Cryptarithmetic Problems. (L3)
3. demonstrate logic techniques using Predicate Logic. (L2)
4. apply forward and backward reasoning to infer knowledge. (L3)

UNIT IV: **(6 Lectures)**

PLANNING: Introduction, Planning Problem, the language of Planning Problems.

ARTIFICIAL NEURAL NETWORK: Introduction to Artificial Neural Networks, Basic Models of Artificial Neural Networks, First Artificial Neurons: McCulloch–Pitts Model, Neural Network Architecture, Single-Layer Feed Forward ANN, Multilayer Feed Forward ANN, Activation Functions, Supervised Learning, Delta Learning Rule, Back propagation Algorithm.

Learning Outcomes: At the end of the unit, student will be able to

1. describe language of planning Problems. (L3)
2. explain various Activation Functions.(L2)
3. illustrate different artificial neural network architecture.(L2)
4. simplify real world problems using a Back propagation algorithm.(L4)

UNIT V: **(6 Lectures)**

NATURAL LANGUAGE PROCESSING: Introduction, Exponential, Natural Language for Communication, Syntactic Analysis, Augmented Grammar, Semantic Interpretation.

EXPERT SYSTEM: Introduction, Need and Justification of Expert System, Knowledge Representation, Knowledge Acquisition and Variation, Utilisation and Functionality, Basics of Prolog.

Learning Outcomes: At the end of the unit, student will be able to

1. demonstrate phases of natural language for communication. (L2)
2. explain the need and justification of an expert system. (L2)
3. build an expert system suitable for solving particular problems. (L3)

TEXT BOOKS:

1. Dr.Nilakshi Jain, *Artificial Intelligence : Making a System Intelligent,* Wiley Publications,1st Edition,2019.

REFERENCES:

1. Saroj Kaushik, *Artificial Intelligence*, Cengage Learning India,1st Edition,2011.
2. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Pearson Publications, 2020.

WEB REFERENCES:

1. https://ai.google/
2. https://www.coursera.org/learn/neural-networks-deep-learning#syllabus
3. https://swayam.gov.in/nd1_noc19_me71/preview
4. https://pypi.org/project/experta/

LIST OF PROGRAMS:

Write a program using python to:

1. a) Generate a Calendar for the given month and year?

   b). Implement a Simple Calculator program?

2. Design of Intelligent systems.
   (Suggested exercise: to control the VACUUM Cleaner moves)

3. Implement the production system and derive a solution for the real world AI problem.

   (Suggested exercise: Write a program to solve the following problem: You have two jugs, a 4-gallon and a 3-gallon. Neither of the jugs has markings on them. There is a pump that can be used to fill the jugs with water. How can you get exactly two gallons of water in the 4-gallon jug?).

4. Implement A* algorithm.
   (Suggested exercise: to find the shortest path).

5. Implement the Constraint Specific Problem.
   (Suggested exercise: a crossword puzzle).

6. Implement the alpha-beta pruning.
   (Suggested exercise: for a tic toc game).

7. Design a planning system using STRIPS.
   (Suggested exercise: an elevator problem to move a passenger from the 1st floor to the 4th floor in a building).

8. Design an expert system. [Hint: use PyPi package]
   (Suggested exercise: for detecting infant is diabetic or not by considering the following parameters:
   a. age = [int from 0 to 5]
   b. glycemie= [int 2/l]
   c. Signs, like shakiness, hunger, sweating, sweating, headache, diabetic_parents, pale,
      i. urination,thirst, blurred_vision, dry_mouth, smelling_breath, shortness_of_breath =
      ii. [Boolean: True/False]
   d. Knowledge Engine:

      def concerned_person(self)

      def hyper_glycemy(self, glycemie)

      def hypo_glycemy(self, glycemie)

      def has_signs_low_sugar(self, age)

      def protocole_risk_low(self)

```
def protocole_alert_low(self)

def has_diabetic_parents(self)

def has_signs_high_sugar(self, **_)

def protocole_risk_high(self)

def protocole_alert_high(self) )
```

9. Implement a single neural network and test for different logic gates.

10. using nltk library:
    a. extract all the words with and without punctuations from a given input sentence
    b. print their parts of speech.
    c. remove stop words.