Course: BTech   **Semester:** 5

Prerequisite: Data structure, automata, and languages, Mathematics

Course Objective: This course provides a broad introduction to Artificial Intelligence. AI techniques for search and knowledge representation also Apply knowledge of AI planning and machine learning techniques to real-world problems.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:

1. Analyze real-world problems and apply appropriate AI techniques to solve them.
2. Create AI systems using heuristic search and knowledge representation techniques.
3. Implement core AI algorithms to solve problems and understand their functionalities.
4. Apply programming skills to build functional AI applications.
5. Analyze complex data and design neural network architectures for pattern recognition and problem-solving.

List of Practical

1.	Develop an AI-based medical diagnosis system using expert systems architecture and knowledge representation techniques.
2.	Build an intelligent agent for optimizing e-commerce inventory management using search algorithms like hill climbing and best-first search.
3.	Implement a constraint satisfaction algorithm to solve scheduling problems in healthcare facilities
4.	Create a recommendation system for personalized learning using means-end analysis and heuristic search techniques.
5.	Develop a problem-solving agent for optimizing resource allocation in logistics using A* and AO* algorithms.
6.	Develop a fuzzy logic-based system for predicting stock market trends considering uncertain market conditions.
7.	Write a program to implement BFS (Water Jug problem or any AI search problem). Write a program to implement DFS (Water Jug problem or any AI search problem).
8.	Define a predicate brother(X,Y) which holds iff X and Y are brothers. Define a predicate cousin(X,Y) which holds iff X and Y are cousins.

Define a predicate grandson(X,Y) which holds iff X is a grandson of Y.

Define a predicate descendent(X,Y) which holds iff X is a descendent of Y.

Consider the following genealogical tree:

father(a,b).

father(a,c).

father(b,d).

father(b,e).

father(c,f).

Say which answers, and in which order, are generated by your definitions for the following queries in Prolog:

?- brother(X,Y).

?- cousin(X,Y).

?- grandson(X,Y).

?- descendent(X,Y).

9.	Write a program to implement Tic-Tac-Toe game using python.
10.	Create a spell-checking application utilizing natural language processing (NLP) techniques, including syntactic and semantic analysis.
11.	Design a neural network architecture for pattern recognition in medical imaging for disease diagnosis.