



## **DEPARTMENT OF COMPUTER ENGINEERING**

### **AI-Lab Assignments Sample List 2025-26 (not limited to it)**

<b>Assignment No.</b>	<b>Sample Problem Statements</b>
<b>1</b>	1. Implement non-AI and AI technique to design any two-player game.
<b>2</b>	2. Implement Breadth First Search to solve Water-Jar problem. 3. Implement Breadth First Search to solve 8-Puzzle problem. 4. Implement Breadth First Search to solve Maze problem. 5. Implement Breadth First Search to solve Missionaries and Cannibals problem. 6. Implement Breadth First Search to solve 8-Queens problem. 7. Implement Depth First Search to solve Water-Jar problem. 8. Implement Depth First Search to solve 8-Puzzle problem. 9. Implement Depth First Search to solve Maze problem. 10. Implement Depth First Search to solve Missionaries and Cannibals problem. 11. Implement Depth First Search to solve 8-Queens problem. 12. Implement Depth Limited Search with depth=3 to solve Water-Jar problem. 13. Implement Depth Limited Search with depth=3 to solve 8-Puzzle problem. 14. Implement Depth Limited Search with depth=3 to solve Maze problem. 15. Implement Depth Limited Search with depth=3 to solve Missionaries and Cannibals problem. 16. Implement Depth Limited Search with depth=3 to solve 8-Queens problem.
<b>3</b>	17. Implement Best First Search to solve Water-Jar problem. 18. Implement Best First Search to solve 8-Puzzle problem. 19. Implement Best First Search to solve Maze problem. 20. Implement Best First Search to solve Missionaries and Cannibals problem. 21. Implement Best First Search to solve 8-Queens problem. 22. Implement Best First Search to solve the shortest route-finding problem.



## DEPARTMENT OF COMPUTER ENGINEERING

	23. Implement A* Search to solve Water-Jar problem. 24. Implement A* Search to solve 8-Puzzle problem. 25. Implement A* Search to solve Maze problem. 26. Implement A* Search to solve Missionaries and Cannibals problem. 27. Implement A* Search to solve 8-Queens problem. 28. Implement A* Search to solve the shortest route-finding problem
<b>4</b>	29. Implement an algorithm to solve 8-Queens Constraint satisfaction problem. 30. Implement an algorithm to solve any Cryptarithmic Constraint satisfaction problem. 31. Implement an algorithm to solve any Graph Colouring Constraint satisfaction problem.
<b>5</b>	32. Implement predicate logic using PROLOG for Family Tree. 33. Implement predicate logic using PROLOG for Bird classification. 34. Implement predicate logic using PROLOG for Vehicle Classification. 35. Implement predicate logic using PROLOG for Vegetable/Fruit Classification.
<b>6</b>	36. Implement an Expert system using PROLOG for Skillset-Job matching. 37. Implement an Expert system using PROLOG for Animal identification. 38. Implement an Expert system using PROLOG for Family Tree. 39. Implement an Expert system using PROLOG for Disease classification. 40. Implement an Expert system using PROLOG for Sentiment Analysis.

Subject Coordinator

**Dr. Radhika V. Kulkarni**