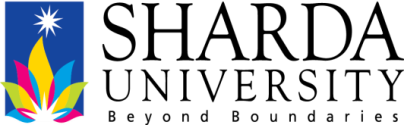
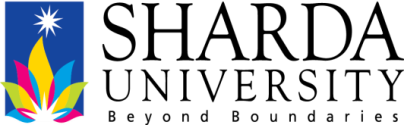
****

**Department of Computer Science & Engineering**

**CSP411: ARTIFICIAL INTELLIGENCE**

|  |  |  |
| --- | --- | --- |
| 1 | Course number | **CSE411/CSP312** |
| 2 | Course Title | **ARTIFICIAL INTELLIGENCE** |
| 3 | Credits | 4 |
| 4 | Contact Hours | 3-0-2 |
| 5 | Course Objective | The objective of the course is to introduce basic fundamental concepts in Artificial Intelligence (AI), with a practical approach in understanding them. To visualize the scope of AI and its role in futuristic development. |
| 6 | Course Outcomes | On successful completion of this module students will be able to   * distinguish between AI and non-AI solution, * apply AI techniques in problem solving, * analyse the best search technique and implement it in real-life applications * explore the scope of AI in various application domains |
| 7 | Outline syllabus | |
| 7.01 | CSE428.A | **INTRODUCTION TO AI** |
| 7.02 | CSE428.A1 | Foundation of AI, Goals of AI, History and AI course line |
| 7.03 | CSE428.A2 | Introduction to Intelligent Agents; Environment; Structure of Agent |
| 7.04 | CSE428.A3 | AI Solutions Vs Conventional Solutions; a philosophical approach; a practical approach |
| 7.05 | CSE428.B | **PROBLEM SOLVING AGENTS** |
| 7.06 | CSE428.B1 | Problem solving using Search Techniques; Problems; Solutions; Optimality |
| 7.07 | CSE428.B2 | Informed Search Strategies; Greedy Best-First; A\* Search; Heuristic Functions |
| 7.08 | CSE428.B3 | Uninformed Search Strategies; BFS; DFS; DLS; UCS; IDFS; BDS |
| 7.09 | CSE428.C | **KNOWLEDGE & REASONING** |
| 7.10 | CSE428.C1 | Knowledge-Based Agents; Logic; First-Order Logic; Syntax-Semantics in FOL; Simple usage; |
| 7.11 | CSE428.C2 | Inference Procedure; Inference in FOL; Reduction; Inference Rules; |
| 7.12 | CSE428.C3 | Forward Chaining; Backward Chaining; Resolution |
| 7.13 | CSE428.D | **LEARNING** |
| 7.14 | CSE428.D1 | Common Sense Vs Learning; Components; Representations; Feedback |
| 7.15 | CSE428.D2 | Learning Types: Supervised; Unsupervised; Reinforcement Learnings |
| 7.16 | CSE428.D3 | Artificial Neural Networks: Introduction, types of networks; Single Layer and Multi-Layer n/w. |
| 7.17 | CSE428.E | **APPLICATIONS** |
| 7.18 | CSE428.E1 | AI Present & Future; application case studies on NLP, Image Processing; |
| 7.19 | CSE428.E2 | Robotics – Hardware; Vision; Navigation based case studies; |
| 7.20 | CSE428.E3 | Ambient Intelligence case studies; |
| 8 | Course Evaluation | |
| 8.1 | Course work: 30 marks | |
| 8.11 | Attendance | 100% |
| 8.12 | Homework | Assignments (4) |
| 8.13 | Quizzes | 5 |
| 8.14 | Projects | Optional |
| 8.15 | Presentations |  |
| 8.16 | Any other | Posters (optional) |
| 8.2 | MTE | One, 20 marks |
| 8.3 | End-term examination: 50 marks | |
| 9 | References | |
| 9.1 | Text book\* | 1. Rich E& Knight K, Artificial Intelligence, Tata McGraw Hill, Edition 3. |
| 9.2 | other references | 1. Russell S &Norvig P, *Artificial Intelligence: A Modern Approach*, Prentice Hall 2. Dan W. Patterson, Artificial Intelligence & Expert Systems, Pearson Education with Prentice Hall India. Indian Edition. |

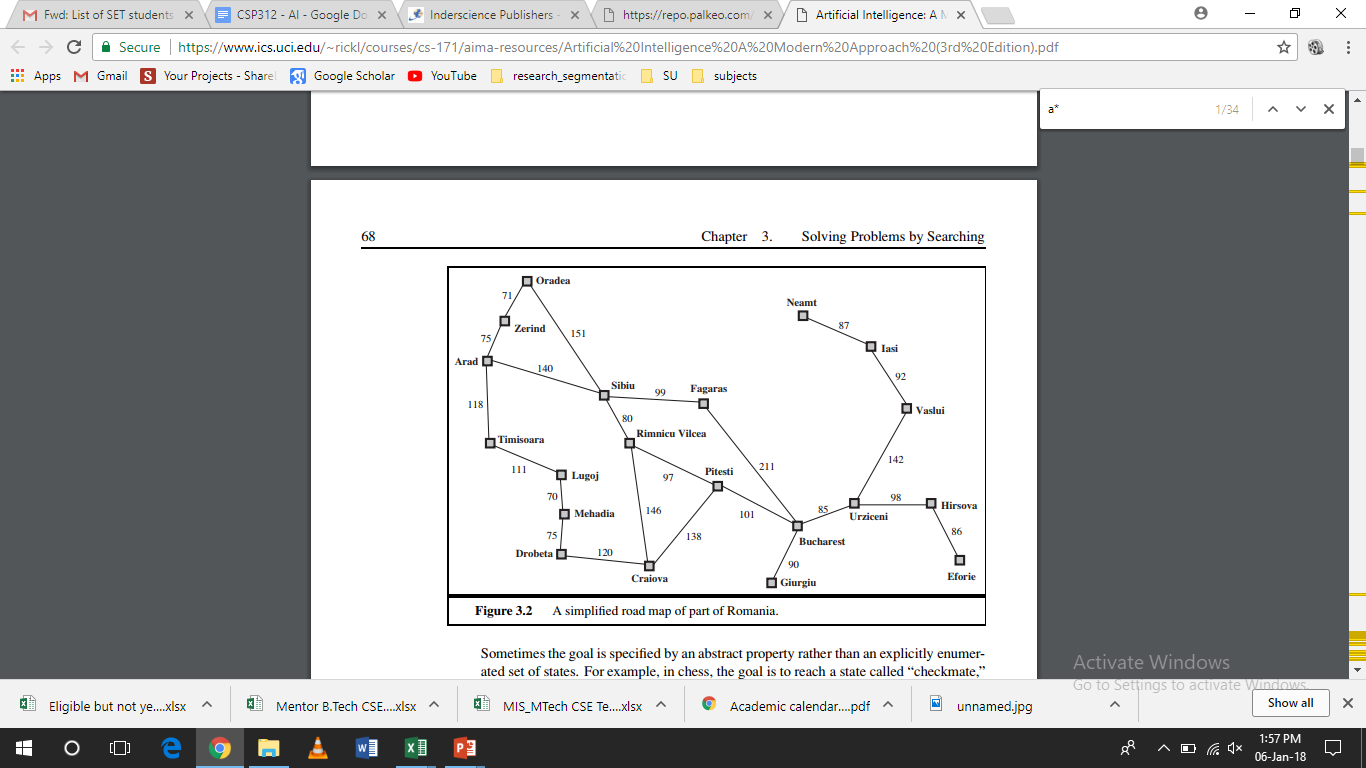
****

**Department of Computer Science & Engineering**

**CSP312: ARTIFICIAL INTELLIGENCE LAB**

**List of Experiments**

1. Implementation of Water Jug Problem.
2. Refer following figure as map with distance details, Write a program in your preferred language to generate path from ARAD to BUCHREST, analyze result obtained by
3. Depth First Search
4. Breadth First Search
5. Uniform Cost Search



1. Write a program in your preferred language to generate steps to solve Tower of Hanoi problem.
2. Write a program in your preferred language to solve the 8 puzzle Problem-using A\* algorithm.
3. Introduction to Lisp, and basic programming in Lisp like following:
   1. Write a LISP function to compute sum of squares.
   2. Write a LISP function to compute difference of squares. (if x > y return x2 – y2, Otherwise y2 – x2).
   3. Write a Recursive LISP function which takes one argument as a list and return last element of the list. (Do not use last predicate.)
   4. Write a Recursive LISP function which takes one argument as a list and return list except last element of the list. (Do not use butlast.)
   5. Write a Recursive LISP function which takes one argument as a list and return reverse of the list. (Do not use reverse predicate).
   6. Write a Recursive LISP function which takes two arguments first an atom second a list returns a list after removing first occurrence of that atom within the list.
   7. Write a Recursive LISP function which appends two lists together.
   8. Write a recursive LISP function which takes 2 lists as arguments and returns a list containing alternate elements from each list.
4. Advance programming in Lisp like following:
5. Write a function that compute the factorial of a number.(factorial of 0 is 1, and factorial of n is n\*(n-1)\*...1.Factorial is defined only for integers greater than or equal to 0.)
6. Write a function that evaluate a fully parenthesized infix arithmetic expression. For examples, (infix (1+ (2\*3))) should return 7.
7. Write a function that performs a depth first traversal of binary tree. The function should return a list containing the tree nodes in the order they were visited.
8. Write a LISP program for water jug problem.
9. Write a LISP program that determines whether an integer is prime.
10. Write PROLOG program to Program to categorize animal characteristics.
11. Write PROLOG program to solver for the linear equation A\*X + B = 0. Let the predicate linear (A, B, X) return the root X of the equation.
12. Write a PROLOG program that answers questions about family members and relationships includes predicates and rules which define sister, brother, father, mother, grandchild, grandfather and uncle. The program should be able to answer queries such as the following:

father(x, Amit)

grandson(x, y)

uncle (sumit, puneet)

mother (anita, x)

**Signature of Course Coordinator Signature of Lab Coordinator**

**Signature of Head of the Department**