DISCIPLINE SPECIFIC ELECTIVES (DSE) COURSES OFFERED BY THE DEPARTMENT

DISCIPLINE SPECIFIC ELECTIVES (DSE-1)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Artificial Intelligence and Machine Learning ELDSE-1A	4	3	-0	1	Course Admission Eligibility	Basic knowledge of Python language

Learning Objectives

The Learning Objectives of this course are as follows:
Artificial Intelligence and Machine Learning has emerged as one of the most rapidly growing technology sector in today's time. This fascinating technology area which deals with designing 'machines which can think' is finding widespread application in almost every industrial and domestic sector. Advancement in the field of AI and ML has also led to complete revolution in the other technology areas including Robotics, embedded systems and Internet of Things. AI and ML is considered to be one of the major contributor to the paradigm shift in technology which has taken place over the past few decades, which is very similar in scale to past events such as the industrial revolution, the computer age, and the smart phone revolution.
This course will give an opportunity to gain expertise in one of the most fascinating areas of science and technology through a well-structured classroom program that covers almost all the topics related to designing machines which can replicate human intelligence and its applications in industry, defence, healthcare, agriculture and many other areas. This course will give the students a rigorous, advanced and professional graduate-level foundation in Artificial Intelligence and Machine Learning.

Learning outcomes

The Learning Outcomes of this course are as follows:

- Build intelligent agents for search and games
- Solve AI problems through programming with Python
- Learning optimization and inference algorithms for model learning

103

- Design and develop programs for an agent to learn and act ir structured environment
- To study different supervised and unsupervised learning algorithms.
- To understand the application development process using ML

.

SYLLABUS OF ELDSE-1A **Total Hours- Theory: 45 Hours, Practicals: 30 Hours**

UNIT – I (11 Hours)

Introduction: Concept of AI, history, current status, scope, Modeling Techniques: Turing Test Approach, Cognitive Modeling Approach, Rational Agent Approach and Laws of Thought Approach, AI System Architecture: Concept of Agent & Environment, Types of Agents: Reactive Agent, Model based Reflex Agent, Omniscient Agent, Goal Based Agent, Utility based Agent and Learning Agent, Knowledge based Agents and Knowledge Representation Techniques. Types of Environment, PEAS representation of Intelligent Agents, Problem Solving Agents, AI Problem Formulation, State space representation

UNIT – II (11 Hours)

Search Algorithms: Uninformed Search Algorithms: Breadth first search, Depth First Search, Depth Limited Search, Uniform Cost Search and Bidirectional Search, Heuristic Search Algorithms: concept of Heuristic Function, Greedy Best First Search, A* search algorithm, Game Search Algorithms: Minimax Search Algorithm and Alpha-Beta Pruning.

Simple AI problems (such as Water Jug Problem, Maze Problem, 8-Tile Puzzle problem, Traveling Salesman Problem).

UNIT – III (11 Hours)

Probabilistic Reasoning Model: Probability, conditional probability, Bayes Rule, Bayesian Networks- representation, construction and inference, Temporal model: concept of Transition probability, Markov Model and Hidden Markov model.

Markov Decision Process Model: MDP formulation, Elements of MDP Model, concept of Sequential Decision Processing, Example of MDP Problem: Agent in a grid world

UNIT – IV (12 Hours)

Machine Learning: Types of Machine Learning: Supervised Learning, Unsupervised Learning and Reinforcement Learning. Supervised Learning Vs. Unsupervised Learning **Supervised Learning Techniques:** Regression Analysis, Linear Regression, Classification Algorithm, Logistic Regression, K-NN Algorithm, Classification Vs. Regression, Linear Regression Vs. Logistic Regression, Decision Tree Classification Algorithm, Random Forest Algorithm, Clustering in Machine Learning, Hierarchical Clustering in Machine Learning, K-Means Clustering Algorithm

Practical component (if any) – Artificial Intelligence and Machine Learning
(Python software)

Learning outcomes

The Learning Outcomes of this course are as follows:

- Implement various search algorithms
- Implement Bayesian network
- Demonstrate classification and clustering
- Make a small project

LIST OF PRACTICALS (Total Practical Hours- 30 Hours)

1. Write a program to solve the given search tree using Breadth First Search
2. Write a program to solve the given search tree using Depth First Search and Depth Limited Search
3. Write a program to solve the given search tree using Uniform Cost Search
4. Write a program to solve the given search tree using Greedy Best First Search
5. Write a program to solve the given game search tree using Minimax Search
6. Program for construction and inference of a Bayesian network
7. Write a Program to perform Regression on given data sets
8. Write a Program to demonstrate Classification
9. Write a Program to demonstrate Clustering
10. Mini Project work

Note: Students shall sincerely work towards completing all the above listed practicals for this course. In any circumstance, the completed number of practicals shall not be less than eight.

Essential/recommended readings

1. Stuart Russell and Peter Norvig, —Artificial Intelligence: A Modern Approach‖ , 3rd Edition, Prentice Hall
2. Elaine Rich and Kevin Knight, —Artificial Intelligence‖, Tata McGraw Hill
3. Trivedi, M.C., —A Classical Approach to Artificial Intelligence‖, Khanna Publishing House, Delhi.
4. Saroj Kaushik, —Artificial Intelligence , Cengage Learning India, 2011
5. Introduction to Machine Learning with Python, by Andreas C. Müller, Sarah Guido, O'Reilly Media, Inc., 2016

Suggestive readings

1. David Poole and Alan Mackworth, —Artificial Intelligence: Foundations for Computational Agents, Cambridge University Press 2010
2. Machine Learning by Tom. M. Mitchell, Tata McGraw Hill
3. Introduction to Machine Learning by Nils. J. Nillson

DISCIPLINE SPECIFIC ELECTIVES (DSE-2)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Algorithm Design and Analysis ELDSE-1B	4	3	0	1	Course Admission Eligibility	Basic Knowledge of Python language

Learning Objectives

The Learning Objectives of this course are as follows:

- To develop the understanding of usage of basic data structures like stack, queue, linked list, trees
- To introduce the students to design and analyse algorithms
- To highlight the differences between various problem-solving techniques for an efficient algorithm design
- To provide an understanding of algorithm design through a survey of the common algorithm design paradigms of Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Optimization
- To develop proficiency in Problem Solving and Programming
- To provide an understanding of time and space complexities of algorithms designed to solve computational problems
- To familiarize with various Searching and Sorting techniques

Learning outcomes

The Learning Outcomes of this course are as follows:

- Implement data structures like Stacks, Queues. Linked List, trees
- Use an appropriate algorithm using the algorithm design techniques, namely,
- Iterative, Divide and Conquer, Greedy, Dynamic Programming for a series of
- computational problems
- Apply various Searching and Sorting techniques
- Solve computational problems with an understanding of time and space complexities of algorithms

SYLLABUS OF ELDSE-1B Total Hours- Theory: 45 Hours, Practicals: 30 Hours

UNIT – I (11 Hours)

Data Structures: Stacks, array implementation of stack, operation on stacks, application of stacks-conversion of infix expression to prefix and postfix, evaluation

of expression; Queues, array implementation of queues, operation on queues, Linked List and its implementation of stack and queue.

UNIT – II (11 Hours)
Trees: Introduction to trees, Binary search tree, preorder, postorder and inorder traversal (recursive)
Searching Techniques: Linear and Binary Search, Hashing techniques

UNIT – III (12 Hours)
Algorithm Design Techniques: Iterative techniques-Insertion Sort, Divide and Conquer-Merge Sort, Dynamic Programming-Weighted Interval Scheduling, 0-1 Knapsack Problem

UNIT – IV (11 Hours)
Greedy Algorithm- Interval Scheduling, Fractional Knapsack problem, Dijkstra's shortest path problem. Comparison between Dynamic programming and Greedy algorithm
Sorting Techniques: Quick Sort, Heap sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Time and Space complexity

Practical component (if any) – Algorithm Design and Analysis
(Python/MATLAB software)

Learning outcomes

The Learning Outcomes of this course are as follows:

- Implement Data Structures
- Develop algorithms and write programs in Python language
- Write programs based on Algorithm design techniques
- Implement various Sorting techniques
- Prepare a Technical Report on the experiments carried

LIST OF PRACTICALS (Total Practical Hours – 30 Hours)

1. Program to create a stack and perform Pop, Push, traverse operations on the stack using Linear Linked List
2. Program to create a linear queue using Linked List and implement insertion, deletion and display of the queue elements
3. Program to create a Binary Tree to perform traversals (Preorder, Postorder, Inorder) using the concept of recursion.
4. Program to solve the Interval Scheduling problem
5. Program to solve the Weighted Interval Scheduling problem
6. Program to solve the 0-1 Knapsack problem
7. Program to implement Insertion Sort
8. Program to implement Merge Sort
9. Program to implement Heap Sort
10. Program to implement Quick Sort

11. Program to implement Bucket Sort
12. Program to implement Radix Sort
13. Program to implement Binary Search

Note: Students shall sincerely work towards completing all the above listed practicals for this course. In any circumstance, the completed number of practicals shall not be less than eleven.

Essential/recommended readings

1. M.T.Goodrich, R.Tamassia, M.H.Goldwasser, Data Structures & Algorithms, Wiley
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, Prentice Hall India. Third edition (2015).
3. J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education India, First Edition (2013).
4. S. Lipschutz, Data Structures with C, Schaum's Outlines Series, Tata McGraw Hill
5. A.M.Tenenbaum, Y.Langsam, M.J. Augenstein, Data Structures using C, Pearson/PHI

Suggestive readings

1. Sarabasse and A.V. Gleder, Computer Algorithm-Introduction to Design and Analysis, Pearson Education, Third Edition (1999).

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVES (DSE-3)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Mathematics Foundation for Computing ELDSE-1C	4	3	0	1	Course Admission Eligibility	Basic Knowledge of Python language

Learning Objectives

The Learning Objectives of this course are as follows:

- The aims is to introduce to students of electronics new mathematics such as Boolean algebra, relations, and graph theory which though look abstract concepts can be used effectively to design and analyze electronic circuits.
- To apply mathematical techniques for real world and engineering problems and expose students to some front-line techniques used in industry and academics.

Learning outcomes

The Learning Outcomes of this course are as follows:

- Formulate recurrence relations to solve problems involving an unknown sequence. Student should see the significance in light of the Forbenius method they learn.
- Use Boolean algebra to design and analyze digital switching circuitry, such as found in personal computers, pocket calculators, CD players, cellular telephones, and a host of other electronic products.
- Appreciate circuit analysis in terms of topology.

SYLLABUS OF ELDSE-1C Total Hours- Theory: 45 Hours, Practicals: 30 Hours

UNIT – I (12 Hours)

Elementary Combinatorics: Basic counting principles, Permutations and Combinations (with and without repetitions), Binomial theorem, Multinomial theorem, Counting subsets, Set-partitions, Stirling numbers Principle of Inclusion and Exclusion, Derangements, Inversion formula.

Generating functions: Algebra of formal power series, Generating function models, Calculating generating functions, Exponential generating functions.

UNIT – II (10 Hours)

Recurrence Relations: Recurrence Relations, generating functions, iteration and induction, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Divide and conquer relations, Solution of recurrence relations, Solutions by generating functions.

UNIT – III (11 Hours)

Boolean Algebras and Switching Circuits: Axioms of Boolean Algebra, De Morgan's law, Simplification of Boolean Expressions, Representation theorem, Boolean polynomials, Boolean polynomial functions, Disjunctive normal form and conjunctive normal form, Minimal forms of Boolean polynomial,3, 4 and 5 variable Karnaugh diagrams, Quine-McCluskey method, Switching circuits and applications of switching circuits.

UNIT – IV (12 Hours)

Graph Theory: Introduction to Graph Theory with emphasis on DC circuit analysis, Representing circuital network as a graph, identification of branches, nodes, Tree branch/ twig. Formulation of incidence matrix. Usage of incidence matrix to solve for node voltage in two loop DC circuits with voltage and/ or current sources.

Practical component (if any) – Mathematics Foundation for Computing
(Python software)

Learning outcomes

The Learning Outcomes of this course are as follows:

- Implement python programs to calculate permutation and combinations.
- Write python programs based on Boolean Algebra and Minimize Karnaugh diagrams
- Should be able to do node analysis using incidence matrix/ Graph Theory.

LIST OF PRACTICALS (Total Practical Hours – 30 Hours)

1. Write a program that generates all the permutations of a given set of digits (with or without repetitions).
2. Write a program to generate Fibonacci Series using recursion.
3. Write a program to implement binary search using recursion.
4. Write a Program to accept the truth values of variables x and y, and print the truth table of the following logical operations:
    a. Conjunction
    b. Disjunction
    c. NAND
    d. NOR
    e. Exclusive OR
    f. Exclusive NOR
    g. Negation

5. Determine node voltages of given two loop circuits using given incidence matrix.

Essential/recommended readings

1. V. Krishnamurthy, Combinatorics, Theory and Application, Affiliated East-West Press 1985.
2. C.L. Liu & Mahopatra, Elements of Discrete mathematics, 2nd Sub Edition 1985, Tata McGraw Hill
3. G. Langholz, A. Kandel and J. Mott, Foundations of Digital Logic Design, World Scientific, Singapore, 1998.
4. Kenneth H. Rosen. Discrete Mathematics and Its Application. McGraw-Hill Education, Pennsylvania, U.S.A, 2011.
5. M.O. Albertson and J.P. Hutchinson, Discrete Mathematics with Algorithms, John Wiley and Sons (USA, 1988).

Suggestive readings

1. T.H. Coremen, C.E. Leiserson, R. L. Rivest, Introduction to Algorithms, Prentice Hall India (3rd edition 2009)

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.