SYLLABI

(Semester –5)

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF TECHNOLOGY&ENGINEERING DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CS341: ARTIFICIAL INTELLIGENCE

Credit and Hours:

Teaching Scheme	Theory	Practical	Total	Credit
Hours/week	3	2	5	4
Marks	100	50	150	

Pre-requisite courses:

- Data Structures & Algorithms

Outline of the Course:

Sr No.	Title of the unit	Minimum number of Hours
1.	Introduction to AI, Problems, Problems Space and Search, Heuristic Techniques	10
2.	Logic and Programming Languages in AI	05
3.	Knowledge Representation and Rules	04
4.	Reasoning	06
5.	Weak Slot-And-Filler Structure and Game Playing and Planning	10
6.	NLP and Text Analytics and Neural Networks	06
7.	Expert Systems and Optimization Techniques and AI & ML Tools	04

Total Hours (Theory): 45

Total Hours (Lab): 30

Total Hours: 75

Detailed Syllabus:

1. Introduction to AI **10 Hours** **20 %**

What is AI, Applications of AI, characteristics, advantages and disadvantages.

Problems, Problems Space and Search, Heuristic Search Techniques

Defining The Problems as a State Space Search, Production Systems, Problem Characteristics, Production System Characteristics, Issues In The Design Of Search Programs,

Heuristic Search Techniques:

Hill Climbing, A*, AO*, Simulated Annealing, Branch and Bound, Nearest Neighbour, Blind Search Techniques: DFS, BFS, Best First Search, Control Strategies.

2 Logic and Programming Languages in AI **05 Hours** **15 %**

Logic:

Propositional Logic, Predicate Logic and Fuzzy Logic, Monotonic and non-Monotonic

Programming Languages:

Introduction to Prolog: Syntax & Numeric Function, Basic List Manipulation Functions In Prolog, Functions, Predicates & Conditional, Input, Output & Local Variables, Iteration & Recursion, Property Lists & Arrays. GUI Version of Prolog.

Python Programming: Syntax, Data Type, Libraries : NumPy, Numba, NumExpr, SciPy, AstroPy, Pandas, SymPy, MatplotLib, Jupyter, Ipython

3. Knowledge Representation **04Hours** **10 %**

Knowledge Representation:

Knowledge Representations And Mappings, Approaches To Knowledge Representation.

Representing Knowledge using Rules

Procedural Versus Declarative Knowledge, Logic Programming,

	Forward Versus Backward Reasoning.		
4	**Reasoning** **Symbolic Reasoning Under Uncertainty and Statistical Reasoning:** Introduction To Non-monotonic Reasoning, Logics For Nonmonotonic Reasoning **Statistical Reasoning** Probability And Bays’ Theorem, Certainty Factors And Rule-Base Systems, Bayesian Networks, Dempster-Shafer Theory	**06Hours**	**10 %**
5.	**Weak Slot-And-Filler Structure and Game Playing and Planning** **Weak Slot-And-Filler Structure:** Semantic Nets, Frames, Ontology, OWL, Reasoner **Game Playing and Planning:** Introduction: Games as Search Problems, Perfect Decisions in Two-Person Games, Imperfect Decisions, Alpha-Beta Pruning, Games That Include anElement of Chance, State-of-the-Art Game Programs : Chess, Checkers or Draughts, Othello, Backgammon, Go The Blocks World, Components Of A Planning System, Goal Stack Planning, Nonlinear Planning Using Constraint Posting, Hierarchical Planning, Reactive Systems.	**10 Hours**	**20 %**
6.	**NLP and Text Analytics and Neural Networks** **NLP and Text Analytics:** Introduction, Syntactic Processing, Semantic Analysis, Semantic Analysis, Discourse And Pragmatic Processing, Text Analytics, Text pre-processing, Bag of Words, Word Cloud, Machine Translation, sentiment analysis **Neural Networks** Introduction: Simple Perceptron, Hopfield Network, Learning In Neural Network, Application Of Neural Networks, Recurrent Networks, Deep Neural Network, Convolution Network, Restricted Boltzmann machine, Transfer learning	**06 Hours**	**15 %**

7. Expert Systems and Optimization Techniques and AI & ML Tools **04 Hours** **10 %**

Expert Systems:

An Introduction To Expert System, Explanation Facilities, Expert System Developments Process, Knowledge Acquisition.

Optimization Techniques and AI & ML Tools:

Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization(PSO), Honey Bee
AI , Machine Learning and Data Analytics Tools

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Understand the various searching techniques, constraint satisfaction problem and example problems- game playing techniques.
CO2	Apply these techniques in applications, which involve perception, reasoning and learning.
CO3	Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
CO4	Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
CO5	Demonstrate proficiency-developing applications in an 'AI language', expert system shell, or data-mining tool.
CO6	Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	1	1	1	1	1	1	-	-	2	2	2	3
CO2	2	2	-	-	-	-	-	1	2	1	1	1	2	-
CO3	3	1	1	-	-	1	1	-	-	-	-	-	2	-
CO4	3	2	3	3	-	-	-	1	2	2	2	2	3	3
CO5	3	2	2	2	3	2	2	-	-	-	-	-	2	-
CO6	2	1	1	-	-	3	-	-	-	-	2	2	3	3

Enter correlation levels 1, 2 or 3 as defined below:
1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)
If there is no correlation, put “-”

Recommended Study Material:

- **Text Books:**
  1. “Artificial Intelligence” -By Elaine Rich and Kevin Knight (2nd Edition) Tata Mcgraw-Hill.
  2. Stuart J. Russell and Peter Norvig, Artificial Intelligence 3e: A Modern Approach, 3rd Edition. Person
  3. Introduction to Prolog Programming By Carl Townsend
- **Reference Books:**
  1. “Artificial Intelligence and Expert System, Development”-By D.W.Rolston, Mcgraw-Hill International Edition.
  2. “Artificial Intelligence And Expert Systems ”By D.W.Patterson
  3. “PROLOG Programming For Artificial Intelligence” By Ivan Bratko, Addison-Wesley
  4. “Programming with PROLOG” –By Klocksin and Mellish.

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF TECHNOLOGY&ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CS350: OPERATING SYSTEM

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

Pre-requisite courses:

- Introduction to computer and computer architecture.

Outline of the course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction	02
2.	Process Management	04
3.	Inter process Communication	08
4.	Deadlock	06
5.	Memory Management	08
6.	Input Output Management	07
7.	File Systems	06
8.	Unix/Linux File System	04

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

Detailed Syllabus:

✓ **1. Introduction** **02 Hours** **05%**

1.1 What is an OS? Evolution of OS

1.2 OS Services

1.3 Types of OS

1.4 Concepts of OS

1.5 Different Views Of OS

✓ **2. Process Management** **04 Hours** **08%**

2.1 Process, Process Control Block, Process States,

2.2 Threads, Types of Threads and Dispatching, Concurrent Threads

✓ **3. Inter process Communication** **08 Hours** **15%**

3.1 Race Conditions, Critical Section, Co-operating Thread/Mutual Exclusion

3.2 Hardware Solution, Strict Alternation, Peterson's Solution

3.3 The Producer Consumer Problem, Semaphores, Event Counters, Monitors

3.4 Message Passing and Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem.

4. Deadlock **06 Hours** **15%**

4.1 Deadlock Problem, Deadlock Characterization

4.2 Deadlock Detection, Deadlock recovery

4.3 Deadlock avoidance: Banker's algorithm for single & multiple resources

4.4 Deadlock Prevention.

4.5 CPU Scheduling, Protection: Address space, Address Translation

5. Memory Management **08 Hours** **18%**

5.1 Paging: Principle of Operation, Page Allocation, H/W Support For Paging

5.2 Multiprogramming with Fixed partitions

5.3 Segmentation

5.4 Swapping

5.5 Virtual Memory: Concept, Performance Of Demand Paging, Page Replacement Algorithms, Thrashing and Working Sets

6. Input Output Management **07 Hours** **14%**

6.1 I/O Devices, Device Controllers, Direct Memory Access

6.2 Principles of Input/output S/W: Goals of The I/O S/W, Interrupt Handler, Device Driver, Device Independent

6.3 I/O Software Disks: RAID levels, Disks Arm Scheduling Algorithm, Error Handling.

7. File Systems **06 Hours** **15%**

7.1 File Naming, File Structure, File Types, File Access, File Attributes, File Operations, Memory Mapped Files

7.2 Directories: Hierarchical Directory System, Pathnames, Directory Operations,

7.3 File System Implementation, Contiguous Allocation, Linked List Allocation, Linked List Using Index, Inodes

8. Unix/Linux File System **04 Hours** **10%**

8.1 Buffer Cache, Inodes, The system calls: ialloc, ifree, namei, alloc and free

8.2 Mounting and Unmounting, files systems, Network File systems

8.3 EXT file system in Linux

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Visualize and understand Operating system functionality and working of OS. Understanding of functionality, services of operating system and differentiate between different types of OS.
CO2	Define thread and process. Visualize how processes and threads are managed by the operating systems. Simulate and analyze various process scheduling algorithms. Explain and analyze different Inter process communication techniques
CO3	Describe deadlock and classify detection, recovery, prevention and avoidance algorithms. Test scenarios to report deadlock
CO4	Compare and evaluate various memory management schemes. Simulate and analyze Memory management algorithms. Identify and describe the role of I/O devices.
CO5	Understand the file systems. Understand the secondary storage and simulate disk scheduling algorithm.
CO6	Understand the basic commands of Linux file systems

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	-	2	-	-	-	-	-	-	1	-	-
CO2	3	2	-	-	2	-	-	-	-	-	-	-	1	-
CO3	3	2	-	-	2	-	-	-	-	-	-	-	1	-

CO4	3	2	-	-	2	-	-	-	-	-	-	-	1	-
CO5	3	2	-	-	2	-	-	-	-	-	-	-	1	-
CO6	3	2	-	-	2	-	-	-	-	-	-	1	1	-

Enter correlation levels 1, 2 or 3 as defined below:
1: Slight (Low) 2: Moderate (Medium) 3: Substantial
(High) If there is no correlation, put "-"

Recommended Study Material:

- **Text Books:**

1. Modern Operating Systems -By Andrew S. Tanenbaum, Third Edition PHI
2. Operating System Concepts Avi Silberschatz, Peter Baer Galvin, Greg Gagne, Ninth Edition, Wiley

- **Reference Books:**

1. Operating Systems, D.M. Dhamdhare, TMH
2. Operating Systems Internals and Design Principles , William Stallings , Seventh Edition, Prentice Hall
3. Unix System Concepts & Applications, Sumitabha Das, TMH
4. Unix Shell Programming, Yashwant Kanitkar, BPB Publications

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF TECHNOLOGY&ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CS351: DESIGN & ANALYSIS OF ALGORITHMS

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

Pre-requisite courses:

- Computer Programming.

Outline of the course:

Sr. No.	Title of the unit	Minimum number of hours
1.	To derive time and space complexity of algorithm.	03
2.	Analysis of Algorithm	06
3.	Greedy Algorithm	07
4.	Divide and Conquer Algorithm	07
5.	Dynamic Programming	08
6.	Exploring Graphs	04
7.	Backtracking & Branch & Bound	05
8.	String Matching and Introduction to NP- Completeness	05

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

Detailed Syllabus:

1.	**Basics of Algorithms and Mathematics**	**03 Hours**	**05 %**
1.1	What is an algorithm?		
1.2	Performance Analysis, Model for Analysis - Random Access Machine (RAM), Primitive Operations		
1.3	Time Complexity and Space Complexity		
2.	**Analysis of Algorithm**	**06 Hours**	**14 %**
2.1	The efficiency of algorithm, average and worst case analysis, elementary operation		
2.2	Asymptotic Notation		
2.3	Analyzing control statement		
2.4	Analyzing Algorithm using Barometer		
2.5	Solving recurrence Equation		
2.6	Sorting Algorithm		
3.	**Greedy Algorithm**	**07 Hours**	**16 %**
3.1	General Characteristics of greedy algorithms		
3.2	Problem solving using Greedy algorithm		
3.3	Making change problem		
3.4	Graphs: Minimum Spanning trees (Kruskal’s algorithm, Prim’s algorithm		
3.5	Graphs: Shortest paths; The Knapsack Problem; Job Scheduling Problem		
4.	**Divide and Conquer Algorithm**	**07 Hours**	**16 %**
4.1	Multiplying large Integers Problem		
4.2	Binary Search		
4.3	Sorting (Merge Sort, Quick Sort)		
4.4	Matrix Multiplication		
4.5	Exponential		

5. Dynamic Programming **08 Hours** **18 %**

5.1 Introduction, The Principle of Optimality

5.2 Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient

5.3 Making Change Problem

5.4 Assembly Line-Scheduling

5.4 Knapsack Problem

5.5 Shortest Path

5.6 Matrix Chain Multiplication

5.7 Longest Common Subsequence

6. Exploring Graphs & Backtracking **04 Hours** **09 %**

6.1 An introduction using graphs and games,

6.2 Traversing Trees – Preconditioning Depth First Search-Undirected Graph; Directed Graph, Breath First Search, Applications of BFS & DFS

7. Backtracking & Branch & Bound **05 Hours** **12%**

7.1 Backtracking –The Knapsack Problem; The Eight queens problem, General Template

7.2 Brach and Bound –The Assignment Problem; The Knapsack Problem, The min-max principle

8. String Matching and Introduction to NP-Completeness **05 Hours** **10%**

8.1 The naïve string matching algorithm

8.2 The Rabin-Karp algorithm

8.3 The class P and NP Problems

8.4 Polynomial reduction

8.5 NP- Completeness Problem

8.6 NP-Hard problems

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Analyze the asymptotic performance of algorithms.
CO2	Derive time and space complexity of different sorting algorithms and compare them to choose application specific efficient algorithm.

CO3	Understand and analyze the problem to apply design technique from divide and conquer, dynamic programming, backtracking, branch and bound techniques and understand how the choice of algorithm design methods impact the performance of programs.
CO4	Understand and apply various graph algorithms for finding shorted path and minimum spanning tree.
CO5	Synthesize efficient algorithms in common engineering design situations.
CO6	Understand the notations of P, NP, NP-Complete and NP-Hard.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	-	-	-	-	-	-	-	-	-	1	-
CO2	2	2	-	-	-	-	-	-	-	-	-	2	2	-
CO3	3	3	3	3	2	-	-	-	-	-	-	2	2	-
CO4	2	3	3	1	-	-	-	-	-	-	-	-	2	-
CO5	1	-	1	-	-	-	-	-	-	-	-	2	1	1
CO6	3	1	-	-	-	-	-	-	-	-	-	-	1	-

Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)If
there is no correlation, put “-”

Recommended Study Material:

- **Text Books:**

  1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest and Clifford Stein, MIT Press

- **Reference Books:**

  1. Fundamental of Algorithms by Gills Brassard, Paul Bratley, Pentice Hall of India.
  2. Fundamental of Computer Algorithms by Ellis Horowitz, Sartazsahni and sanguthevar Rajasekarm, Computer Sci.P.
  3. Design & Analysis of Algorithms by P H Dave & H B Dave, Pearson Education.

- **Web Materials:**

1. http://www.stanford.edu/class/cs161/
2. http://www.itl.nist.gov/div897/sqg/dads/
3. http://highered.mcgraw-hill.com/sites/0073523402/