Syllabus for 4 Year B. Tech Course in Computer Science and Engineering (Artificial Intelligence)

Sixth Semester

Operations Research & Optimization Techniques (TIU-UMA-T302)
Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3

Course Outcome

CO1	To analyze any real-life system with limited constraints and depict it in a linear programming form and hence find its solution.
CO2	To study a variety of problems such as assignment, transportation, travelling salesman etc. and understand how real-life problems are depicted and solved mathematically
CO3	To understand basic concepts of game theory and strategies involved.
CO4	To understand different queuing situations and find the optimal solutions using models for different situations.
CO5	To provide idea of project scheduling and network models for service and manufacturing systems, and apply operations research techniques and algorithms to solve these network problems.

Course Content

Module-1:
Linear Programming: Convex Sets, Graphical Method, Simplex Method
Module-2:
Transportation and Assignment Formulation of Transportation Problem, Initial Feasible Solution Methods, Optimality Test, Degeneracy in TP; Assignment Problem, Hungarian Method, Travelling Salesman Problem.
Module-3:
Game Theory: Two Person Zero Sum Game, Pure and Mixed Strategies, Algebraic Solution Procedure, Graphical Solution.
Module-4:
Introduction to Queuing Models: Elements of Queuing Model, Pure Birth Death Model.
Module-5:
Project Scheduling and Network Analysis: PERT and CPM review, crashing of an activity, Crash-cost slope, Time-cost trade, Solution of network problems using Simplex technique. Time estimates for PERT, Probability of completion of a project within a scheduled time.

Recommended Books:

Main Reading

1. Linear Programming and Game Theory by Ghosh and Chakraborty
2. Operations Research: An Introduction by Hamdy A. Taha
3. Operations Research: Theory and Applications by J K Sharma
4. Operations Research by S D Sharma
5. Operations Research by Kanti Swarup

Computer Networks (TIU-UCS-T304)

Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3
Prerequisite Course: Data Structures and Algorithms (TIU-UCS-T201)

Course Outcome

CO1	Describe the general principles of data communication, the concept of the layered approach
CO2	Describe how computer networks are organized with the concept of layered approach
CO3	Design logical sub-address blocks with a given address block and network topology
CO4	Understanding of simple LAN with hubs, bridges, and switches
CO5	Describe how routing protocols work

Course Content

Module-1:
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

Module-2:
Data link layer: Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channels. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

Module-3:
Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

Module-4:
Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

Module-5:
Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

Recommended Books:

Main Reading

1. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

Supplementary Reading

1. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
2. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education

Compiler Design (TIU-UCS-T320)

Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3

Course Outcome

CO1	Understand fundamentals of language parser and identify the relationships among different phases of compiler
CO2	Illustrate the use of different types of parsers and their constructions, production rules and language semantics.
CO3	Inherited and synthesized attributes with their evaluations, run time storage allocation
CO4	Describe techniques for intermediate code generation and code optimization

Course Content

Module-1: Compiler Structure
Analysis-synthesis model of compilation, various phases of a compiler, tool-based approach to compiler construction.

Module-2: Lexical analysis
Interface with input, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, and implementation. Regular definition, Transition diagrams, LEX.

Module-3: Syntax analysis
Context free grammar, ambiguity, associativity, precedence, top-down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, Bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.

Module-4: Syntax directed definitions
Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top-down evaluation of attributes, L- and S-attributed definitions.

Module-5: Type checking
Type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

Module-6: Run time system
Storage organization, activation tree, activation record, parameter passing Symbol table, dynamic storage allocation. Intermediate code generation: Intermediate representations, translation of declarations, assignments Intermediate Code generation for control flow, Boolean expressions and procedure calls, implementation issues.

Module-7: Code generation and instruction selection
Issues, basic blocks and flow graphs, register allocation, code generation DAG representation of programs, code generation from DAGs, peep-hole optimization, code generator generators, specifications of machine.

Module-8: Code optimization
Source of optimizations, and optimization of basic blocks, loops, global dataflow analysis, and solution to iterative data flow equations. Code improving transformations, dealing with aliases, data flow analysis of structured flow graphs.

Recommended Books:

Main Reading

1. Aho, Ullman,Sethi and Lam, Principles of Compiler Design, Pearson Education
2. Holub, Compiler Design in C, PHI

Supplementary Reading

1. Andrew L. Appel,Modern Compiler Implementation in C, Foundation Books, Delhi
2. Dick Gruneet. Al., Modern Compiler Design, Wiley Dreamtech
3. S. Chattopadhyay, Compiler Design, PHI
4. S. Pal: Systems Programming, Oxford University Press

Software Engineering (TIU-UCS-T314)

Contact Hours/Week: 3–0–0 (L–T–P)

Credit: Theory–3

Course Outcome

CO1	To understand software Engineering layered architecture and the process frame work, specially related to embedding feedback and intelligence in the product
CO2	To analyze software process models such as the waterfall, spiral, evolutionary models and agile method for software development
CO3	To design and understand requirement specifications, project planning, scheduling, cost estimation, risk management
CO4	To describe data models, object models, context models and behavioral models
CO5	To learn coding style and testing issues
CO6	To know about the quality checking mechanism for software process and product.

Course Content

Module-1:
Introduction to software engineering: Software and software engineering, phases in software development, software development process models, role of management in software development, role of metrics and measurement.

Module-2:
Software requirement specifications: Role of SRS, problem analysis, requirement specification, validation, metrics, monitoring and control.
Planning a software project: Cost estimation, project scheduling, staffing, personal planning, team structures, SCM, quality assurance plans, project-monitoring plans, risk management, Knowledge driven approach and development.

Module-3:
System design: Design objectives, design principles, module level concepts, design methodology, structured design, design specifications, verification metrics, monitoring and control.
Detailed design: Module specification, detailed design and process design language, verification.

Module-4:
Coding: Programming practice, verification, and metrics.
Testing: Testing fundamentals, functional testing, structural testing, testing process, comparison of different V & V techniques.

Module-5:
Software quality; Garvin's quality dimensions, McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction to Capability Maturity Models (CMM and CMMI); Introduction to software reliability, reliability models and estimation.

Recommended Books:

Main Reading:

1. Software Engineering, Ian Sommerville

Supplementary Reading:

1. R. Mall, "Fundamentals of Software Engineering", Prentice Hall of India
2. R. S. Pressman, "Software Engineering: a Practitioner's Approach", Tata McGraw Hill
3. D. Bell, "Software Engineering for Students", Pearson

Artificial Intelligence (TIU-UCS-T350)

Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3

Course Outcome

CO1	To know the fundamental concepts Artificial Intelligence such as knowledge representation, problem solving and expert systems
CO2	To know the use of AI to solve communication problems using Natural Language Processing
CO3	To develop knowledge of decision making and learning methods
CO4	To develop new facts from existing knowledge base using resolution and unification
CO5	To demonstrate the way of writing Facts and Rules in order to solve some problems based on rules and to develop systems for question-answer

Course Content

Module-1: Basics of AI

Introduction: Overview of Artificial intelligence- Problems of AI, AI technique, Tic - Tac - Toe problem., Intelligent Agents: Agents & environment, nature of environment, structure of agents, goal-based agents, utility-based agents, learning agents., Learning: Forms of learning, inductive learning, learning decision trees, explanation-based learning, learning using relevant information, neural net learning & genetic learning.

Module 2: Different types of searching algorithms, Problem Solving

Problems, Problem Space & search: Defining the problem as state space search, production system, constraint satisfaction problems, issues in the design of search programs, Search techniques: Solving problems by searching: Problem solving agents, searching for solutions; uniform search strategies: breadth first search, depth first search, depth limited search, bidirectional search, comparing uniform search strategies, Heuristic search strategies: Greedy best-first search, A* search, memory bounded heuristic search: local search

algorithms & optimization problems: Hill climbing search, simulated annealing search, local beam search, genetic algorithms; constraint satisfaction problems, local search for constraint satisfaction problems, Adversarial search: Games, optimal decisions & strategies in games, the minimax search procedure, alpha-beta pruning, additional refinements, iterative deepening.

Module 3: Knowledge & Reasoning

Knowledge representation issues, representation & mapping, approaches to knowledge representation, issues in knowledge representation, Using predicate logic: Representing simple fact in logic, representing instant & ISA relationship, computable functions & predicates, resolution, natural deduction, Representing knowledge using rules: Procedural vs declarative knowledge, logic programming, forward vs backward reasoning, matching, control knowledge, Probabilistic reasoning: Representing knowledge in an uncertain domain, the semantics of Bayesian networks, Fuzzy sets, and fuzzy logics, belief propagation. Markov processes, and Hidden Markov models.

Module 4: Different fields of AI, Natural Language Processing

Introduction, Syntactic processing, semantic analysis, discourse, and pragmatic processing. Expert Systems: Representing and using domain knowledge, expert system shells, and knowledge acquisition. Basic knowledge of programming language like Prolog.

Recommended Books:

Main Reading

1. Artificial Intelligent e: Elaine Rich, Kevin Knight, Mc-Graw Hill.
2. Introduction to AI & Expert System: Dan W. Patterson, PHI.
3. Artificial Intelligent by Luger (Pearson Education)
4. Russel & Norvig, Artificial Intelligent e: A Modern Approach, Pearson Education