List of Elective Courses Offered by CSE

Course Code	Course Name	L	T	P	Credits
CSL420	Soft Computing	3	0	0	3
CSL421	Artificial Intelligence	3	0	2	4
CSL422	Machine Learning	3	0	2	4
CSL423	Distributed Computing Systems	3	0	2	4
CSL424	Real Time Systems	3	0	0	3
CSL425	Mobile Computing	3	0	0	3
CSL426	Cloud Computing	3	0	0	3
CSL427	Software Architecture	3	0	0	3
CSL428	Software Project Management	3	0	0	3
CSL429	Software Testing and Evaluation	3	0	0	3
CSL430	Advanced Computer Architecture	3	0	0	3
CSL431	Computer Graphics	3	0	0	3
CSL432	Human Computer Interaction	3	0	0	3
CSL433	Natural Language Processing	3	0	2	4
CSL435	Introduction to Parallel Computing	3	0	2	4
CSL436	Data Mining and Warehousing	3	0	2	4
CSL437	Bioinformatics	3	0	2	4
CSL438	Information Retrieval	3	0	2	4
CSL439	Business Intelligence	3	0	0	3
CSL440	Advanced Compilers	3	0	0	3
CSL441	Paradigms in Programming Languages	3	0	2	4
CSL444	Big Data Analytics	3	0	2	4
CSL445	Data Sciences	3	0	2	4
CSL446	Neural Networks and Deep Learning	3	0	0	3
CSL448	Quantum Computing	3	0	0	3
CSL449	Internet of Things: Programming, Protocols and Standards	2	0	2	3
CSL450	Computational Genomics and Proteomics	3	0	0	3
CSL451	Computational Finance	3	0	0	3
CSL452	Model-Driven Software Engineering	3	0	0	3
CSL453	Computational Logic Design Methodology	1	0	4	3
CSL454	Reinforcement Learning	3	0	2	4
CSL456	Natural Language Understanding and Generation	3	0	0	3
CSL457	Parallel Computer Architectures	3	0	0	3
CSL465	Cyber Security	2	1	2	4
CSL466	Blockchain Technology	2	1	0	3
CSL467	Industrial and Social Applications of Digital Twins	2	0	2	3
CSL468	Intelligent Systems	3	0	0	3

Electives from ECE:

Course Code	Course Name	L	T	P	Credits
ECL415	Digital Image Processing	3	0	0	3
ECL416	Image and Video Communication	3	0	2	4
ECL418	Neuro Fuzzy Technique	3	0	0	3
ECL430	Biomedical Engineering	3	0	0	3
ECL432	Wireless Sensor Networks	3	0	2	4

Course Code:	**CSL 420**	**Course Title:**	**Soft Computing**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**Data Structure**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

Students will be able to:

1. Identify appropriate learning technique for classification, pattern recognition, optimization problems, etc.
2. Apply and relate unified and mathematical basis, general principles of various soft computing techniques for decision making problems.
3. Design and analyze intelligent or humanistic systems using fuzzy logic and artificial neural network.

Course Contents:

Module 1:

Fuzzy Set Theory: Basic Definition and Terminology, Set Theoretic Operations, MF Formulation and Parameterization, MF of two dimension, Fuzzy Union, Intersection and Complement. Fuzzy Rules and Fuzzy Reasoning: Extension Principles and Fuzzy Relations, Fuzzy IF THEN Rules, Fuzzy Reasoning.

Module 2:

Fuzzy Inference System: Introduction, Mamdani Fuzzy Models, Other Variants, SugenoFuzzy Models, Tekamoto Fuzzy Models.

Module 3:

Fundamentals of Genetic Algorithms: Basic Concepts Creation, Offsprings Encoding, Fitness functions, Reproduction, Genetic Modelling: Inheritance Operators, Cross over, Inversion and detection, Mutation operator, Bitwise operators.

Module 4:

Introduction to ANN, Architecture, Back Propagation and feed Forward Networks, Offline Learning, Online Learning.

Module 5:
Supervised Learning of Neural Networks: Introduction, Perceptrons, Adaline, Back Propagation Multilayer Perceptrons, Back Propagation Learning Rules, Methods of Speeding. Radical Basis Function Networks, Functional Expansion Networks.

Module 6:
Unsupervised Learning: Competitive Learning Networks, Kohonen self-organising networks, Hebbian Learning, The Hopfield Network

Text Books:

1. Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence; Roger Jang, Tsai Sun, Eiji Mizutani, PHI.
2. Soft Computing and Its Applications : R.A. Aliev, R.R. Aliev
3. Neural Network: A Comprehensive Foundation; Simon Haykin, PHI.

Reference Books :

1. Elements of artificial Neural Networks; Kishan Mehtrotra, S. Ranka, Penram International Publishing (India).
2. Fuzzy Logic with Engineering Applications; Timothy Ross, McGraw-Hill.
3. Neural Networks and Fuzzy Systems: Bar Kosko , PHI.

Course Code:	**CSL 421**	**Course Title:**	**Artificial Intelligence**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Formulate problems so that exploratory search can be applied.
2. Apply and implement various search techniques to solve real world problems
3. Apply algorithms for designing games and solving constraint satisfaction problem
4. Represent knowledge using formal logic and apply algorithms to deuce conclusion
5. Design and develop practical algorithms for solving planning and uncertainty problems

Course Contents:

Module 1:
Introduction: What is AI? , History, Overview, Intelligent Agents, Performance Measure, Rationality, Structure of Agents, Problem-solving agents, Problem Formulation, Uninformed Search Strategies

Module2:
Informed (Heuristic) Search and Exploration, Greedy best first search, A* search, Memory bounded heuristic search, Heuristic functions, inventing admissible heuristic functions, Local Search algorithms, Hill-climbing, Simulated Annealing, Genetic Algorithms, Online search

Module 3:
Constraint Satisfaction Problems, Backtracking Search, variable and value ordering, constraint propagation, intelligent backtracking, local search for CSPs, Adversarial Search, Games, The minimax algorithm, Alpha-Beta pruning, Imperfect Real-Time Decisions, Games that include an Element of Chance.

Module 4:
Knowledge Based Agents, Logic, Propositional Logic, Inference, Equivalence, Validity and Satisfiability, Resolution, Forward and Backward Chaining, DPLL algorithm, Local search algorithms, First Order Logic, Models for first order logic, Symbols and Interpretations, Terms, Atomic sentences, complex sentences, Quantifiers, Inference in FOL, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution

Module 5:
Planning, Language of planning problems, planning with state-space search, forward and backward state-space search, Heuristics for state-space search, partial order planning, planning graphs, planning with propositional logic

Module 6:
Uncertainty, Handing uncertain knowledge, rational decisions, basics of probability, axioms of probability, inference using full joint distributions, independence, Baye's Rule and conditional independence, Bayesian networks, Semantics of Bayesian networks, Exact and Approximate inference in Bayesian Networks

Text:

1. Artificial Intelligence a Modern Approach : Russel and Norvig , Pearson Education, 2nd
2. Artificial Intelligence – A Practical Approach : Patterson , Tata McGraw Hill, 3rd

Course Code:	**CSL 422**	**Course Title:**	**Machine Learning**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:
Student will be able to:
1. Understand the working of various supervised and unsupervised machine learning models
2. Apply classification and regression models to solve real world problems.
3. Apply unsupervised learning to solve real world problems.
4. Evaluate the performance of various machine learning models.

Course Contents:
Module 1:
Introduction to ML, Linear Regression, Logistic Regression, Inductive Classification

Module 2:

Decision Trees: Representing concepts as decision trees. Recursive induction of decision trees. Picking the best splitting attribute: entropy and information gain. Searching for simple trees and computational complexity. Occam's razor. Overfitting, noisy data, and pruning.

Module 3
Artificial Neural Networks: Neurons and biological motivation. Linear threshold units.
Perceptrons: representational limitation and gradient descent training. Multilayer networks and Backpropagation.

Module 4:
SVM, Multiclass & Ordinal Classification, Kernel Methods, Bayesian Learning: Naive Bayes learning algorithm. Parameter smoothing. Bayes nets and Markov nets for representing dependencies. Hidden Markov Model, and Bayesian networks.

Module 5:
Clustering and unsupervised learning: Clustering. Hierarchical Aglomerative Clustering. k-means partitional clustering. Expectation maximization (EM) for soft clustering. Semi-supervised learning with EM using labelled and unlabelled data.

Module 6:
Evaluating hypothesis: Training and test splits, k-fold cross validation, confusion matrix, Estimating hypothesis accuracy, sample and true error.

Text Books:
1. Machine Learning, Tom Mitchell, McGraw Hill, 1997.
2. Ethem Alpaydin, Introduction to Machine Learning, PHI, 2016.

Reference Books:
1. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.
2. Christopher Bishop. Pattern Recognition and Machine Learning. 2e, 2006.
3. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification, Wiley, New York, 2001.

Course Code:	**CSL 423**	**Course Title:**	**Distributed Computing Systems**			
Category:	**ELECTIVE**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-Requisite (if Any) :	**NONE**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:
On successful completion of the course, students shall be able to:

1. Identify various advantages and challenges in developing the distributed algorithms for concepts like synchronization, mutual exclusion, deadlock detection, etc.
2. Evaluate various distributed algorithms' performance using various performance measures.
3. Design and develop distributed applications using low-level and high-level distributed programming primitives.
4. Identify various types of faults and fault handling mechanisms in order to build fault tolerant systems.

Course Contents:
Module 1: Introduction and Distributed Communication

Introduction and motivation to Distributed Systems, Characteristics, Applications, Challenges, Architecture types, Fundamental models.

Inter-process and inter-node communication using Sockets – connection oriented and connection-less, Remote Procedure Calls, Remote Method Invocation

Module 2 Distributed File System

Distributed File System Design and Implementation, Case Studies of NFS, Andrew File Systems, HDFS, Distributed Resource Management.

Module 3 Clock Synchronization and Theoretical Foundations

Clock Synchronization Techniques, Network Time Protocol, Logical Clocks, Vector Clocks.Causally Ordered Broadcast and Unicast, Termination Detection – Ring based and Dijkstra Scholten algorithms, Leader Election – Ring based, Franklin'salgorithm and Bully Algorithm

Module 4 Distributed Mutual Exclusion and Deadlock Detection

Distributed Mutual Exclusion – Token based algorithms – Lamport's, Ricart-Agarwala, Maekawa's algorithms, Non-Token based Algorithms – Suzuki Kasami, Raymond's algorithms, comparison of different algorithms.

Distributed Deadlock Detection, Resource and Communication Deadlocks – Centralized technique, Distributed technique - edge chasing and path pushing algorithms, Hierarchical technique, Recovery from Deadlocks.

Module 5 Fault Tolerance and Recovery

Fault Tolerance, Handling Crash faults –Two phase commit protocol, Non-blocking three phase commit protocol, Birman-Joseph Atomic Broadcast Protocol, Voting techniques for fault tolerance

Recovery – forward and backward recovery, undo-redo logs, Coordinated and Uncoordinated Checkpointing and Recovery algos (1 week)

Module 6 Agreement protocols

LSP Oral Messages, Agreement using Signed Messages

Text Books:

1. "Advanced concepts in Operating Systems", Mukesh Singhal and Niranjan Shivratri, McGraw Hill Education (India), 2017.

Reference Books:

1. "Distributed Systems: Concepts & Design", Coulouris, George, Jean Dollimore, and Tim Kindberg, 5th Edition , AWL Press. Pearson Education, 2011.
2. "Modern Operating Systems", Andrew S. Tanenbaum, 5th Edition, Pearson Education, 2022.

Course Code:	**CSL 424**	**Course Title:**	**Real Time Systems**			
Category :	**ELECTIVE**		**L**	**T**	**P**	**C**

		Credit Assigned :	**3**	**0**	**0**	**3**
Pre-Requisite (if Any) :	**Operating System**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Able to apply the knowledge of general operating system to real time operating systems where the concepts of operating system is governed by the timing constraints on the tasks.
2. Able to design the real time task model for any given real time application.
3. Able to evaluate the performance of various real time task scheduling algorithms, resource access control protocols etc for Uniprocessor systems as well as multiprocessor systems.
4. Able to compare various open source RTOS platforms.
5. Able to work on small sensor based systems where they can apply the real time task scheduling, synchronization, communication etc.

Course Contents:

Module 1:

Introduction to RTS, WCET notion, Types of RTS, Task Types, Jobs – Periodic, Sporadic, Aperiodic, Applications of RTS, Predictability, Reference Model, Types of schedulers, Cyclic and Priority based Schedulers.

Module 2:

Cyclic, priority based schedulers – static/dynamic – RM, EDF, LST, Optimality of EDF, Non-optimality of EDF, Scheduling with precedence constraints, Multiprocessor scheduling – static and dynamic systems, Problems of Predictability in multi-processor systems, Predictability of preemptive priority based scheduling in uniprocessor systems, Performance Measure of validation techniques.

Module 3:

Cyclic scheduling, frame size constraints, Job Slicing, Aperiodic job scheduling using Slack stealing, Sporadic job scheduling, Practical considerations, Disadv of cyclic scheduling.

Module 4:

Priority Based Sched, Static-Dynamic Systems, Fixed, Variable Priorities, Schedulable Utilization, Schedulable Utilization of EDF, Schedulability Test of EDF, Unpredictability of Dynamic Priority in Overload, Liu-Layland Theorem, Optimality of RM in Simply-Periodic Systems, Concept of Critical Instants, Time Demand Analysis, Practical factors – Non- preemption, self-suspension, context switch time, Limited priority levels, Mapping techniques, Impact on schedulability, Tick Scheduling.

Module 5:

Aperiodic jobs in Priority based systems, Polling Server, Combining with background server, Polling Server Parameters, Deferrable Server (DS), Combining with Background Server, Deferrable Server parameters, Disadv of DS, Simple Sporadic Server Rules, Combining Background time, Proof of Simple Sporadic Server as a periodic task, Constant Utilization Server

for Deadline Driven systems, Total Bandwidth Server, Starvation free CU/Background Server, Preemptive Weighted Fair Queuing Server, Scheduling of Sporadic Jobs in Fixed Priority and Dynamic Priority Systems.

Module 6:

Resource Control, Model, Priority Inversion, Uncontrolled Priority Inversion, Anomalies, NPCS, Blocking Time, Disadvantages of NPCS, Priority Inheritance Protocol, Deadlocks due to Priority Inheritance Protocol, Priority Ceiling Protocol, DeadlockAvoidance,Analysis of Priority Ceiling Protocol, Blocking time, context switches, Stack Sharing Priority Ceiling Protocol, example, Priority Ceiling Protocol in Dynamic Priority Systems, Preemption Levels, Fixed Preemption Level Systems like EDF, Basic Preemption Ceiling Protocol, Multiple units of resources, Priority ceiling, Preemption ceiling and stack based preemption ceiling protocols for multiple unit resources.

Text Books:

1. Real-Time Systems :Jane W.S. Liu, Pearson Education
2. Real Time Systems : C.M. Krishna & Kang G. Shin : McGraw Hill

Course Code:	**CSL 425**	**Course Title:**	**Mobile Computing**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**Computer Networks**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Recognize the basic concepts of Mobile computing
2. Analyse security, energy efficiency, mobility, scalability, and their unique characteristics in wireless networks.
3. Classify the components of Mobile telecommunication system and associated protocols.
4. Apply knowledge of TCP/IP extensions for mobile networking platforms and application development.

Course Contents:

Module 1: INTRODUCTION

Mobile Computing – Mobile Computing Vs wireless Networking – Mobile Computing Applications – Characteristics of Mobile computing – Structure of Mobile Computing Application. MAC Protocols – Wireless MAC Issues – Fixed Assignment Schemes – Random Assignment Schemes – Reservation Based Schemes.

Module 2: MOBILE INTERNET PROTOCOL AND TRANSPORT LAYER

Overview of Mobile IP – Features of Mobile IP – Key Mechanism in Mobile IP – route Optimization. Overview of TCP/IP – Architecture of TCP/IP- Adaptation of TCP Window – Improvement in TCP Performance.

Module 3: MOBILE TELECOMMUNICATION SYSTEM

Global System for Mobile Communication (GSM) – General Packet Radio Service (GPRS) – Universal Mobile Telecommunication System (UMTS), 2G, 3G, 4G.

Module 4: MOBILE AD-HOC NETWORKS

Ad-Hoc Basic Concepts – Characteristics – Applications – Design Issues – Routing Essential of Traditional Routing Protocols –Popular Routing Protocols – Vehicular Ad Hoc networks (VANET) – MANET Vs VANET – Security.

Module 5: MOBILE PLATFORMS AND APPLICATIONS

Mobile Device Operating Systems – Special Constraints & Requirements – Commercial Mobile Operating Systems – Software Development Kit: iOS, Android, BlackBerry, Windows Phone – M-Commerce – Structure – Pros & Cons – Mobile Payment System– Security Issues.

Text Books:

1. Prasant Kumar Pattnaik, Rajib Mall, "Fundamentals of Mobile Computing", PHI Learning Pvt. Ltd, New Delhi – 2012.

Reference Books:

1. Jochen H. Schller, "Mobile Communications", Second Edition, Pearson Education, New Delhi, 2007.
2. DharMa Prakash Agarval, Qing and An Zeng, "Introduction to Wireless and Mobile systems", Momson Asia Pvt Ltd, 2005.
3. Uwe Hansmann, Lomar Merk, Martin S. Nicklons and Momas Stober, "Principles of Mobile Computing", Springer, 2003.
4. WilliaM.C.Y.Lee,"Mobile Cellular Telecommunications-Analog and Digital Systems", Second Edition,Tata Mc Graw Hill Edition, 2006.
5. C.K.Toh, "AdHoc Mobile Wireless Networks", First Edition, Pearson Education, 2002.

Course Code :	**CSL426**	**Course Title :**	**Cloud Computing**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Understanding of different layers of cloud computing, infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Introduction of practical IaaS, PasS, SaaS, including Amazon ECS, GAE, Force.com, Microsoft Azure, etc.
2. Gain knowledge of cloud storage system design issues, including directory management, data placement, and consistency issues. Practical cloud storage system solutions including GFS, Big Table, HDFS, etc.
3. Student should have good knowledge of authentication, authorization and secure access in the cloud. Introduction to cloud security. Secure computation in the cloud.
4. Insights into the virtualization technologies: Hypervisor, emulation, and application VM. Platform virtualization, storage virtualization, and network virtualization.

Course Contents:

Module 1:
Cloud system architectures, Cloud programming frameworks and what is "infrastructure-as-a service", “platform-as-a-service” and “software-as-a-service”, Cloud computing delivery models — public, private and hybrid clouds, cloud-in-a-box.

Module 2:
Big data concepts, storage and management, Security, scalability, privacy, lock-in, and other risks (and mitigations) for individuals and companies, Introduction to Hadoop, Map-reduce.

Module 3:
Virtualization, clustering and resource management, HPC in cloud computing Cloud applications considerations for updates, backups, disaster recovery and fault tolerance, Data center networks and Energy use in data centers.

Module 4:
Cloud Service Providers: Cloud Platform, Cloud Storage, Cloud Connect, Cloud Print, App Engine on cloud , Web Services on cloud , Elastic Compute Cloud, Simple Storage Service, Simple Queue creation on cloud , service, cloud Assessment and Planning Toolkit, cloud features integration , Service Cloud: Knowledge as a Service

Module 5:
Case Study: Design of a cloud system, Utility Pricing Details, Service Level Agreement
Lab: Video streaming on cloud , File hosting services, Secure storage of personal data, Backup solutions for systems, sites, and software, storage management and privacy

Text Books:

1. Resse G, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud,O’ Reilly.

Reference Books:

1. Buyya R., Broberg J., Goscinski A. M., Cloud Computing — Principles and Paradigms, Wiley

Course Code:	**CSL 427**	**Course Title:**	**Software Architecture**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

On completion of this course students will be able to:

1. Design and understand software architecture for large scale software systems.
2. Recognize major software architectural styles, design patterns, and frameworks
3. Describe a software architecture using various documentation approaches and architectural description languages
4. Develop architectural alternatives for a problem and select among them
5. Use well-understood paradigms for designing new systems

Course Contents:

Module 1:

Software process and the role of modeling and analysis, software architecture and software design.

Software Modeling and Analysis: Analysis modeling and best practices, traditional best practice diagrams such as DFDs and ERDs

Module 2:

Software Architecture: architectural styles, architectural patterns, analysis of architectures, formal descriptions of software architectures , Architectural description languages and tools

Module 3:

Software Design: design best practices, design patterns, design case studies, component technology, object oriented frameworks, distributed objects, interoperability standards, case studies., software quality

Module 4:

UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns, documenting software architecture, reconstructing software architecture.

Module 5:

Middleware components, programming models, implementation, systems qualities Moving from qualities to architecture and views Components and COTS, Economics- Driven Architecture, Software product line, Software architecture future.

Module 6:

Issues in Software Architecture: Scalability and interoperability issues, web application architectures, case studies.

Text Books:

1. M. Shaw, “Software Architecture Perspectives on an Emerging Discipline”, PHI
2. Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice”, Pearson Education Asia
3. R. Taylor, N. Medvidovic, E. Dashofy, “Software Architecture – Foundations, Theory, and Practice”, Wiley India
4. Jan Bosch, “Design and Use of Software Architectures”, Addision-Wesley-Pearson Education
5. Christine Hofmeister, Robert Nord, Dilip Soni, “Aoolied Software Architecture”, Addision-Wesley Pearson Education
6. Dikel, D.Met Al, “Software Architecture: Organizational Principles and Pattern”, Prentice Hall

Course Code:	CSL 428	Course Title:	Software Project Management			
Category :	ELECTIVE	Credit Assigned :	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any) :	NONE	Type of Course :	Computer Science and Engineering			

Course Outcomes:

1. Match organizational needs to the most effective software development model
2. Understanding the basic concepts and issues of software project management
3. Effectively planning the Software projects and employ mechanisms for tracking the software projects
4. Implementing project plans through managing people, communication and change
5. Developing skills for tracking and controlling software deliverables and address real-world management challenges

Course Contents:

Module 1:

Overview of Software Project Management, The Project Life Cycle, Software Development Life Cycle Models, Life Cycles and Metrics, Process Maturity: SEI CMM

Module 2:

Estimation Techniques of IT, Project Scoping, Project Planning, Project Control, Project Phase-Out, Risk Management, Configuration Management

Module 3:

People Management, Team Dynamics, Net Present Value, Project Portfolio Management Software Quality Assurance, Project Leadership

Text:

1. R.K. Wysocki et al. : Effective Project Management: Traditional, Agile, Extreme, 5th Edition, Wiley India, 2011.
2. C. Jones : Applied Software Measurement, Assuring Productivity and Quality, McGraw Hill

Reference:

1. D. I. Cleland : Project Management, Strategic Design and Implementation, 3rd edition, McGraw-Hill.

Course Code	CSL 429	Course Title:	Software Testing and Evaluation
Category:	Elective	Credit Assigned	L T P C 3 0 0 3

Pre-Requisite (if Any)	Software Engineering	Type of Course	Computer Science and Engineering

Course Outcomes:

1. Design test cases suitable for a software development for different domains.
2. Identify suitable tests to be carried out and Prepare test planning based on the document.
3. Document test plans and test cases designed
4. Use automatic testing tools to develop and validate a test plan technology

Course Contents:

Module 1:

INTRODUCTION: Testing as an Engineering Activity ,Testing as a Process ,Testing Maturity Model, Testing axioms ,Basic definitions, Software Testing Principles ,The Tester‘s Role in a Software Development Organization ,Origins of Defects ,Cost of defects, Defect Classes , The Defect Repository and Test Design ,Defect Examples- Developer/Tester Support of Developing a Defect Repository

Module 2:

TEST CASE DESIGN STRATEGIES: Using Black Box Approach to Test Case Design Boundary Value Analysis, Equivalence Class Partitioning, State based testing, Cause-effect graphing, Compatibility testing ,user documentation testing, domain testing – Random Testing ,Requirements based testing – Using White Box Approach to Test design ,Test Adequacy Criteria ,static testing vs. structural testing ,code functional testing ,Coverage and Control Flow Graphs ,Covering Code Logic, Paths ,code complexity testing, Additional White box testing approaches, Evaluating Test Adequacy Criteria.

Module 3:

LEVELS OF TESTING: Unit Test, Unit Test Planning ,Designing the Unit Tests ,The Test Harness ,Running the Unit tests and Recording results, Integration tests , Designing Integration Tests, Integration Test Planning, Scenario testing, Defect bash elimination System Testing, Acceptance testing ,Performance testing ,Regression Testing ,Internationalization testing ,Ad-hoc testing, Alpha, Beta Tests ,Testing OO systems Usability and Accessibility testing ,Configuration testing ,Compatibility testing, Testing the documentation – Website testing.

Module 4:

TEST MANAGEMENT: Organization structures for testing teams, testing services, Test Planning, Test Plan Components, Test Plan Attachments ,Locating Test Items – test management – test process – Reporting Test Results, Introducing the test specialist, Skills needed by a test specialist, Building a Testing Group, The Structure of Testing Group- .The Technical Training Program.

Module 5:

TEST AUTOMATION: Software test automation, skills needed for automation, scope of automation, design and architecture for automation, requirements for a test tool challenges in automation, Test metrics and measurements, project, progress and productivity metrics.

TEXT BOOK:

1. Ilene Burnstein, ―Practical Software Testing‖, Springer International Edition, 2003.

2. Edward Kit,‖ Software Testing in the Real World – Improving the Process‖, Pearson Education,
3. Boris Beizer,‖ Software Testing Techniques‖ – 2nd Edition, Van Nostrand Reinhold, New York, 1990.
4. Aditya P. Mathur, ―Foundations of Software Testing _ Fundamental Algorithms and Techniques‖, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008

Course Code:	**CSL 430**	**Course Title:**	**Advanced Computer Architecture**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Make the students aware about various trends in computer design, architecture of advanced processors.
2. Realization about issues related to instruction level, thread level, data level parallelism in multi/many core systems, memory organization & optimization techniques.
3. Understand the design issues with shared/distributed memory systems, multi / many core / GPGPU architecture

Course Contents:

Module 1:

Classes of computers, Trends in technology, power and costs, dependability, quantitative principles of computer design, Introduction to computing models.

Module 2:

Principles of scalable performance, performance metrics and measures, speedup, performance laws, advanced processor technology, superscalar and VLIW processors, Verified memory, cache memory organizations, shared memory organizations. Memory hierarchy, cache performance, protection and examples of virtual memory, cache coherence.

Module 3:

Pipeline and superscalar techniques, linear pipeline processors, reservation and latency analysis, collision free scheduling, pipeline schedule optimization, instruction pipeline design, arithmetic pipeline design, super scalar and super pipeline design.

Module 4:

Multiprocessors and multi-computers, Brief overview of SIMD, MIMD, vector architectures and multi-core architectures.

Module 5:

Elementary theory about dependence analysis, techniques for extraction of parallelism, branch prediction, dynamic scheduling, multiple issue and speculation, limits on instruction level parallelism, Thread level parallelism

Text Books:

1. Computer Architecture : A Quantitative Approach : Hennessy and Patterson : Morgan Kaufmann: 4th
2. Advanced Computer Architecture, Kai Hwang , McGraw Hill
3. Advanced Computer Architectures: A design space approach, Sima D, Fountain T. And Kacsuk P, Pearson Education

Course Code:	**CSL 431**	**Course Title:**	**Computer Graphics**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Student will able to understand the structure of modern computer graphics systems and the basic principles of implementing computer graphics primitives
2. Student will be able to analyze and implement key algorithms for modeling and rendering graphical data
3. Student will be able to develop design and problem solving skills with application to computer graphics
4. Student will acquire experience in constructing interactive computer graphics programs

Course Contents:

Module 1:

Introduction of Graphics Systems: Use of Computer graphics, Video Display Devices, Refresh Cathode-Ray Tubes, Raster and Random Scan Displays, Color CRT Monitors, Direct View Storage Tubes, Flat Panel Displays, Three-Dimensional Viewing Devices, Stereoscopic & Virtual Reality Systems, Raster and Random Scan Systems, Different Input and Hard Copy Devices, Graphics Software.

Module 2:

Output Primitives: Points and Lines, Line Drawing Algorithms (DDA & Bresenham's), Circle and Ellipse Generating Algorithms, Conic Sections.

Module 3:

2D Geometric Transformations: Different types of transformations and their matrix representations, Homogeneous Coordinates, Composite Transformations, transformations between Coordinate Systems, Affine transformations, Window-to-Viewport Coordinate transformation, Clipping-Point, Line, Polygon, Curve and Text Clipping.

Module 4:

3D Concepts and Object Representation: Three Dimensional Display Methods, Polygon Surfaces, Curved Lines & Surfaces, Qudric Surfaces, Spline Representations, Cubic Spline interpolation methods, Bezier Curves and Surfaces.

Module 5:
3D Transformations and Viewing: Translation, Rotation, Scaling, Reflection, Shears, Composite Transformations, Projections- Parallel and Perspective, Projection Transformations, Clipping.

Module 6:
Visible Surface Detection Methods: Classification of Visible Surface Detection Algorithms, Back Face Detection, Depth Buffer Method, A-Buffer Method, Scan-Line Method, Depth Sorting Method, BSP-Tree Method & Area Subdivision Method. Illumination Models and Surface Rendering: Light Sources, Basic Illumination Models, Polygon- Rendering Methods.

Text Books:

1. 1. D. Hearn & M.P. Baker - Computer Graphics, 2/e , Pearson Education, New Delhi, 2005

Reference Books:

1. W.M. Newman.- Principle of Interactive Computer Graphics, Mc Graw Hill Publication, New Delhi, 1995.
2. S. Harrington -Computer Graphics- A Programming Approach, McGraw Hill Publication, New Delhi, 1994.
3. J.D. Foley - A Fundamental of Computer Graphics Addition Wesley, London, 1993.

Course Code:	**CSL432**	**Course Title:**	**Human Computer Interaction**			
Category:	**ELECTIVE**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any)	**NONE**	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

1. Recognize the requirements of designing the user centered and highly usable software systems.
2. Contribute and apply the advancement of Human-Computer Interaction theory and practice.
3. Evaluate methods, quality factors, and data analysis techniques.
4. Analyze interaction problems from a technical, cognitive, and functional perspective.

Course Contents:

Module-1
Introduction to Interaction Design: User Experience, Process of Interaction Design, Goals of Interaction Design and User Experience, Understanding and Conceptualizing Interaction, Conceptual Models, Interface Metaphors, Interaction Types, Paradigms and Frameworks, Cognitive Aspects, Cognition, Cognitive Framework. Social Interaction, Emotional Interaction.

Module 2
Interfaces: Types of Interfaces, Natural User Interfaces, Data Gathering and Key Issues, Data Recording, Interviews, Questionnaires, Observation, Choosing and Combining Techniques, Data

Analysis, Interpretation and Presentation of Qualitative and Quantitative Analysis, Simple Analysis, Tools and Theoretical Frameworks, Presenting the Findings.

Module 3
Process of Interaction Design: Introduction, Establishing Requirements, Data Gathering for Requirements, Task Description, Task Analysis, Design, Prototyping and Construction, Conceptual Design and Physical Design, Using Scenarios, Prototypes in Design.

Module 4
Design Process: Interaction Design Basics, Design Rules, Software Lifecycle, Universal Design and Multimodal Interaction, Design for Diversity, GOMS.

Module 5
Evaluation: Introduction, Goals of Evaluation, Evaluation through Expert Analysis and User Participation, Evaluation Framework, Observing and Testing Users, Case Studies.

TEXT BOOKS:

1. Sharp, H., Rogers, Y., and Preece, J, "Interaction Design: Beyond Human Computer Interaction", Third Edition, John Wiley & Sons, Inc., 2011.
2. Alan Dix, Janet E. Finlay, Gregory D. Abowd and Russell Beale, "Human Computer Interaction", Pearson Education, Third Edition, 2004.

REFERENCE BOOKS:

1. Wilbert O. Galitz, "The Essential Guide to User Interface Design: An Introduction to Gui Design Principles and Techniques", Third Edition, John Wiley Sons, 2002.
2. Benyon, D., Turner, P., and Turner, S, "Designing Interactive Systems: People, Activities, Contexts, and Technologies", Addison-Wesley, 2005.

Course Code:	**CSL 433**	**Course Title:**	**Natural Language Processing**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Overview:

1. Analyse computational treatments of words, sounds, sentences, meanings, and conversations.
2. Analyse Knowledge-based and statistical approaches to language processing for syntax (language structures), semantics (language meaning), and pragmatics/discourse.
3. Develop models for Text Processing and part of Speech Tagging with Hidden Markov Model.
4. Develop models for information extraction, machine translation, automatic summarization, question answering, and interactive dialog systems.

Course Contents:

Module 1:
Introduction: NLP tasks in syntax, semantics, and pragmatics. Applications such as information extraction, question answering, and machine translation. The problem of ambiguity. The role of machine learning. Brief history of the field. Chomsky hierarchy, regular languages, and their

limitations. Finite-state automata. Practical regular expressions for finding and counting language phenomena. A little morphology.

Module 2:
The role of language models. Simple N-gram models. Estimating parameters and smoothing. Evaluating language models. Key algorithmic tool: dynamic programming, first a simple example, then its use in optimal alignment of sequences. String edit operations, edit distance, and examples of use in spelling correction, and machine translation. Constituency, CFG definition, use and limitations. Chomsky Normal Form. Top-down parsing, bottom-up parsing, and the problems with each. The desirability of combining evidence from both directions.

Module 3:
Lexical syntax. Hidden Markov Models (Forward and Viterbi algorithms and EM training). Probabilistic language modeling and its applications. Markov models. N-grams. Estimating the probability of a word, and smoothing. Generative models of language. Their application to building an automatically-trained email spam filter, and automatically determining the language. The concept of parts-of-speech, examples, usage. The Penn Treebank and Brown Corpus. Probabilistic (weighted) finite state automata.

Module 4:
Spelling correction, Information Retrieval, Machine translation, Vector representation of words, Bag of words, CBOW, skip gram models, Recurrent Neural networks, Parts of speech tagging, Word sense disambiguation, Text summarization, Question answering systems.

Text Books:

1. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995.
2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
3. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
4. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Reference Books:

1. Radford, Andrew et. al., Linguistics, An Introduction, Cambridge University Press, 1999.
2. Journals: Computational Linguistics, Natural Language Engineering, Machine Learning, Machine Translation, Artificial Intelligence.
3. Conferences : Annual Meeting of the Association of Computational Linguistics (ACL), Computational Linguistics (COLING), European ACL (EACL), Empirical Methods in NLP (EMNLP), Annual Meeting of the Special Interest Group in Information Retrieval (SIGIR), Human Language Technology (HLT).

Course Code:	**CSL435**	**Course Title:**	**Introduction to Parallel Computing**			
Category:	**Elective**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-requisite (If any):	**NONE**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

On successful completion of the course, students shall be able to:

1. Identify the dependency in computation.
2. Analyze and decompose the computation into parts for parallel computation.
3. Apply OpenMP, MPI and GPU programming for the development of parallel programs.
4. Simulate on parallel computing architectures and do performance evaluation amongst the parallel algorithms.

Course Contents:

Module 1:

Parallel Computers and architectures, Dependency analysis, loop transformation techniques

Module 2:

Programming with Shared Memory: OpenMP and Message-Passing Computing

Module 3:

Embarrassingly Parallel Computations, Partitioning and Divide-and-Conquer Strategies, Pipelined
Computations

Module 4:

Synchronous Computations, Load Balancing and Termination Detection, Introduction to GPU Programming

Module 5:

Sorting Algorithms, Numerical Algorithms, Image Processing algorithms, Searching and Optimization

Text Books:

1. "Introduction to Parallel Computing", Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, 2nd edition, Addison-Wesley, 2003.
2. "Parallel Programming in C with MPI and OpenMP", Michael J. Quinn, McGraw Hill, 2003.
3. "CUDA by Example: An Introduction to General-Purpose GPU Programming", Jason Sanders, Edward Kandrot, 1st edition, Addison-Wesley Professional, 2010.

Reference Books:

1. "Designing and Building Parallel Programs", Ian Foster, 1st edition, Pearson, 2019.
2. "An Introduction to Parallel Computing: Design and Analysis of Algorithms", Ananth Grama, 2nd edition, Pearson, 2004.
3. "Parallel Programming – Techniques and applications Using Networked Workstations and Parallel Computers", Barry Wilkinson and Michael Allen, 2nd edition, Pearson, 2004.
4. "Multi-Core Programming - Increasing Performance through Software Multi-Threading", Shameem Akhter and Jason Roberts, Intel Press, 2006.

Course Code:	**CSL 436**	**Course Title:**	**Data Mining and Warehousing**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**

Pre-Requisite (if Any) :	NONE	Type of Course :	Computer Science and Engineering

Course Outcomes:

1. Identify the scope and necessity of Data Mining & Warehousing for the society.
2. Describe the designing of Data Warehousing so that it can be able to solve the root problems.
3. To understand various tools of Data Mining and their techniques to solve the real time problems.
4. To develop ability to design various algorithms based on data mining tools.
5. To develop further interest in research and design of new Data Mining techniques.

Course Contents:

1. Introduction to Data warehousing - Application of Data warehousing and mining, Data warehouse development life cycle, Data warehouse analysis, CUBE, ROLL UP and STAR queries.
2. Data Warehouse Design - Massive denormalisation, STAR schema design , Data warehouse Architecture, OLAP, ROLAP and MOLAP , concepts of Fact and dimension table
3. Space Management in Data warehouse - Schemas for storing data in warehouse using different storage structures, B-tree index, hash index, clusters, Bitmap index functional index, domain index, Data partitions.
4. Performance and Tuning - Query optimization, memory management, process management. I/o management for Data warehouse.
5. Data Mining Tools –Association rules, a priori algorithm, Fp-trees algorithm, constraints and solution.
6. Cluster analysis- paradigms, DBSCAN, cluster algorithms.
7. Mining tools- decision trees and applications.

Text Books:

1. Jiawei Han, Micheline Kamber, "Data mining- Concepts & Techniques", Morgan Kaufmann
2. Michale Corey, Michale Abbey; Oracle 8i Data Warehousing; Tata McGraw Hill.
3. Navathe and Elmasry ; Fundamentals of Database Systems; Addison Wesley, 2000
4. Arun Pujari; Data Mining; Orient Longman, 2003

Course Code:	CSL 437	Course Title:	Bioinformatics			
Category:	ELECTIVE	Credit Assigned:	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any) :	NONE	Type of Course:	Computer Science and Engineering			

Course Outcomes:

On successful completion of the course, students shall be able to:

1. Implement basic string based computational methods and algorithms to understand the cell and biological systems.
2. Analyze and apply algorithms and programming techniques like dynamic programming, hashing, and suffix trees.
3. Demonstrate data handling and analysis and apply data compression.

4. Develop multidisciplinary approach to the systematic analysis of complex biological phenomena including evolution and gene expression.
5. Apply computational concepts of data science and machine learning to systems biology problems, implement solutions, verify results and contribute to the society.

Course Contents:

Module 1: Basics of biology

Central Dogma of molecular biology, Understanding DNA, RNA and Protein Data, Computer representation of data.

Module 2: String Matching

Sequences: Problem statement, Edit distance and substitution matrices, Global and local alignments, KMP Algorithm, Boyre Moore, suffix trees.

Module 3: Compression algorithms

Burrow Wheeler Transform (BWT), Lampel Ziv WelchTransform (LZW) and implementation.

Module 4: Phylogenetic trees

Introduction to Evolution, Phylogeny -Molecular Evolution, Phylogeny Example, Sankoff and Finch Algorithms.

Module 5: Introduction to Data Analytics

Biological Databases,Types of data, Data Visualization, Prediction from biological data, Examples of K-means (here) and Hierarchical Clustering.

Module 6: Modelling biological systems, Conditional Probability/ Bayes Theorem, Hidden Markov models.

Miscellaneous topics: Pathways and networks, Microarrays, Biomedical images, Genetic Algorithms and applications.

Text Books:

1. "An Introduction to Bioinformatics Algorithms", Jones, Pevzner, MIT Press, 2004.
2. "Algorithms on Strings, Trees and Sequences", Gusfield, Cambridge University Press, 1997.

Reference Books:

1. "An Introduction to Systems Biology: Design Principles of Biological Circuits" Uri Alon, 1st edition, Chapman & Hall/CRC Press, 2006

Course Code:	**CSL 438**	**Course Title:**	**Information Retrieval**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-Requisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Understanding the basics of Information retrieval like what is a corpus, what is precision and recall of an IR system

2. Understanding the data structures like Inverted Indices used in Information retrieval systems
3. Understanding the basics of web search
4. Understanding the different techniques for compression of an index including the dictionary and its posting list
5. Understanding the different components of an Information retrieval system
6. Developing the ability of develop a complete IR system from scratch

Course Contents:

Module 1:

Boolean retrieval, the term vocabulary and postings lists, Dictionaries and tolerant retrieval, Introduction to index-construction and index-compression

Module 2:

Scoring, term weighting and the vector space model Computing scores in a complete search system, Evaluation in information retrieval, Introduction to Relevance feedback and query expansion.

Module 3:

Probabilistic information retrieval, review of basic probability theory, the probability ranking principle, the binary independence model. Language models for information retrieval, Language modeling versus other approaches to IR, Text classification and Naive Bayes, Bayesian Network approaches to IR.

Module 4:

Vector space classification, Support vector machines and machine learning on documents, Fl at clustering, Hierarchical clustering, Matrix decomposition and latent semantic indexing.

Module 5:

Introduction to Web search basics, Web crawling and indexes, Link analysis. Typical Assignments: Based on techniques studied, implementation of those techniques, study of research papers.

Text Books:

1. An Introduction to Information Retrieval: Christopher D. Manning, Prabhakar Raghavan,Hinrich Schütze, Cambridge University Press, Cambridge, England, 2009
2. Information Retrieval: Implementing and evaluating search engines: Stefan Büttcher, Charles L. A. Clarke, Gordon V. Cormack, MIT Press, 2010

Reference Books:

1. Information Retrieval: Algorithms and Heuristics: David A. Grossman, Ophir Frieder, Springer.

Information Retrieval: Data Structures and Algorithms by Frakes, Pearson.

Course Code:	**CSL 439**	**Course Title:**	**Business Intelligence**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**

Pre-Requisite (if Any) :	NONE	Type of Course :	Computer Science and Engineering

Course Outcomes:

1. Apply the principles of Business Intelligence in various business application.
2. Apply different data integration techniques and develop multidimensional data model.
3. Implement Data Warehouse methodology and BI project life cycle.
4. Design an enterprise dashboard that depicts the key performance indicators which helps in decision making

Course Contents:

Module 1:

Introduction to Business Intelligence What is Business Intelligence, Why do we need Business Intelligence, EIS, MIS, DSS & BI, Information Pyramid – Data, Information, Knowledge & Intelligence. Basis for Operational, Tactical & Strategic Decision Making, OLTP Vs. OLAP, Requirements Gathering in BI through Business Questions, BI in various Domains and Functional Area

Module 2:

Principles of Dimensional Modelling Foundation for Fact based decision making, The STAR and SNOWFLAKE schema, Pros & Cons of the STAR/SNOWFLAKE Schema Dimensional Model, Slowly Changing Dimension tables, Fact-less Fact Tables, Aggregation Strategy, Time Dimension

Module 3:

Business Intelligence System Architecture Need for Enterprise Class Business Intelligence Infrastructure, The BI Ecosystem, Building Blocks of a N-Tier BI System – Servers & Communication Protocols, The Central Repository – Metadata, Information Consumption User Interfaces – Desktop Vs. Web Vs. Mobile, Open Architecture, Scalability, Performance in BI – In Memory Analytics

Module 4:

BI Project Lifecycle Typical BI Project Lifecycle, Requirements Gathering & Analysis – Functional & Non- Functional Requirements, Reports & Dashboards Design – Mock-up and Storyboarding, Testing in a BI Project, BI Project Deployment, Post Production Support

Module 5:

Enterprise Class BI Tool First Level of Abstraction of the Data Warehouse in MicroStrategy, Building the Schema Objects – Attributes, Facts, Transformation & Hierarchies, Building Reusable Application Objects – Metrics, Filters, Prompts, Five Styles of BI, Building Reports – Grids & Graphs, Report Manipulation over the Web – Pivoting, Sorting, Drilling, Exporting etc., Setting up Report Distribution, Report Project

Text Books:

1. Turban E., Sharda R., Delen D., King D., Business Intelligence, Pearson Education.

Reference Books:

1. Sabherwal R. and Becerra-Fernandez I., Business Intelligence, Wiley.
2. Kimball R., Ross M., The Kimball Group Reader: Relentlessly Practical Tools for Data

3. Warehousing and Business Intelligence, Wiley and Sons (2010).

Course Code:	CSL 440	Course Title:	Advanced Compilers			
Category :	ELECTIVE	Credit Assigned :	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any) :	Compilers (CSL 304)	Type of Course :	Computer Science and Engineering			

Course Outcomes:

1. Appreciation of parsing and code generation techniques
2. Understanding of optimizations problems and issues, data flow analysis framework and mathematical modeling.
3. Appreciation of role of machine specific issues in compiler construction, the choice of instructions, the availability of registers etc.
4. Ability to combine different optimization techniques to achieve the overall objective of program efficiency.
5. Appreciation of optimization techniques for multi-processor machines and parallelizing optimization schemes.

Course Contents:

Module 1:

Review of compiler fundamentals – lexical analysis, parsing, semantic analysis and intermediate code generation, error recovery, run time storage management, code generation.

Module 2:

Code optimization – Peephole optimization, control flow analysis, data flow analysis, dependence analysis, redundancy elimination, loop optimization, procedural and inter procedural optimization, instruction scheduling.

Module 3:

Compiling for High performance architectures, Compiling for scalar pipeline, compiling for vector pipeline, super scaler and VLIW processors, compiling for multiple issue processors, compiling for memory hierarchy. Parallelization and Vectorization, Dependence and dependence testing.

Module 4:

Loop Normalization, Induction variable Exposure, Enhancing Fine Grained Parallelism, Loop Interchange, Scalar Expansion, Scalar and Array Renaming, Node splitting, Index-set splitting, Loop skewing.

Text Books:

1. Optimizing Compiler for Modern Architecture: A dependence based approach , Randy Allen, Kennedy
2. Advanced Compiler Design and implementation : Steven S. Muchnick

Reference Books:

1. Engineering & Compiler : Keith D. Cooper & Linda Torczon: Morgan Kaufmann

Course Code	CSL 441	Course Title:	Paradigms in Programming Languages

Category:	**ELECTIVE**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Pre-Requisite (if Any)	**NONE**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

1. Evaluate various programming language paradigms to illustrate their usefulness.
2. Analyze differences between various strategies for implementation of programming languages and realize their effects on the efficiency of the programs.
3. Design and implement algorithms for implementing various features of programming languages like dynamic memory management schemes, supporting variety of data types, exception handling mechanisms, etc.
4. Evaluate various programming languages so as to choose an appropriate programming language for the problem to be solved.

Course Contents:

Module 1:
Introduction to Programming Languages:
Evolution of languages, language paradigms, Brief concepts of machine Code, assembly code, assemblers, High Level Languages, Compilation and Interpretation, Hybrid strategies, Bootstrapping, T-Diagrams, Self-Compiling Compilers.

Module 2:
Data Types:
Various data Types, Numeric types, implementation of various data types like int, float, boolean, char, enum, subranges, Basics of IEEE 754 floating point standards, Type checking, Type equivalence, type coercion, conversion, concept of symbol table, Records (Structures) and Variants (Unions), Arrays, Strings, Sets, Pointers and Recursive Types.

Module 3:
Names, Scopes, and Bindings:
Names, Bindings, Lifetime, Var and value model, Scope, Static and Dynamic Scoping, Aliases, Intern and Extern Static variables in C, Separate compilation, Heap Management – First fit, best fit implementations, Buddy system, Fibbonacci heaps, Garbage Collection - Reference Count, Mark and Sweep, Tracing collection without recursion, Integration of compaction with garbage collection, Generational Collection and Conservative Collection.

Module 4:
Control Flow:
Expression evaluation, Assignment Statements, Short Circuit Evaluation of Expression Variables, Selection Statements, Case Statements, Jump Table, Iteration, Enumerated loops, While loop, C for loop, do-while loop.

Module 5:
Subroutines and Control Abstraction
Review of Stack Layout, Calling Sequences, Access to local variables, Static link, non-local references, Caller and callee responsibilities, Leaf routine optimizations, Register windows, inline function calls, Tail recursion, Tree Recursion, Thinking Recursively, Stack Smashing due to lack of bound checks, Parameter passing, Conformant Arrays, Passing of Closures, Call by Name, Variable number of arguments, Exception Handling.

Module 6:
Data Abstraction and Object Orientation
Modular Programming, Generic Subroutines and Modules, Classes, Constructors and Destructors, Implementation issues, Operator Overloading, Templates, Implementation issues for Generic Templates, Representation of an object, Inheritance, Protected Specifier, Dynamic method binding and its implementation, Abstract Classes and Multiple Inheritance.

Text:

1. Michael L. Scott, "Programming Language Pragmatics" (Fourth Edition), Elsevier.

<table>
<tr><td>Course Code</td><td>CSL 444</td><td>Course Title:</td><td>Big Data Analytics</td></tr>
<tr><td>Category:</td><td>ELECTIVE</td><td>Credit Assigned</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>NONE</td><td>Type of Course</td><td>Computer Science and Engineering</td></tr>
<tr><td colspan="4">Course Outcomes:
1. Work with big data platform and explore the big data analytics techniques business applications.
2. Design efficient algorithms for mining the data from large volumes.
3. Analyze the HADOOP and Map Reduce technologies associated with big data analytics.
4. Understand the fundamentals of various big data analytics techniques.
5. Build a complete business data analytics solution

Course Contents:

MODULE - I Introduction: Introduction to big data : Introduction to Big Data Platform ,Challenges of Conventional Systems - Intelligent data analysis ,Nature of Data ,Analytic Processes and Tools ,Analysis vs Reporting

MODULE - II Mining data streams : Introduction To Streams Concepts, Stream Data Model and Architecture ,Stream Computing ,Sampling Data in a Stream ,Filtering Streams ,Counting Distinct Elements in a Stream ,Estimating Moments ,Counting Oneness in a Window, Decaying Window Real time Analytics Platform(RTAP) Applications ,Case Studies ,Real Time Sentiment Analysis-Stock Market Prediction

MODULE - III Hadoop: History of Hadoop- the Hadoop Distributed File System ,Components of Hadoop Analyzing the Data with Hadoop-,Scaling Out- Hadoop Streaming- Design of HDFS,HDFS Basics- Developing a Map Reduce Application-How Map Reduce Works-Anatomy of a Map Reduce Job run-Failures-Job Scheduling, Shuffle and Sort, Task execution - Map Reduce Types and Formats- Map Reduce Features Hadoop environment.

MODULE - IV Frameworks: Applications on Big Data Using Pig and Hive, Data processing operators in Pig, Hive services, HiveQL, Querying Data in Hive ,fundamentals of HBase and ZooKeeper - IBM InfoSphere BigInsights and Streams.</td></tr>
</table>

MODULE - V **Predictive Analytics** Simple linear regression, Multiple linear regression Interpretation 5 of regression coefficients. Visualizations ,Visual data analysis techniques interaction techniques - Systems and applications

Text:

1. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGrawHill Publishing, 2012
2. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley& sons, 2012
3. Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons, 2007.
4. Tom White "Hadoop: The Definitive Guide" Third Edition, O'reilly Media, 2012.
5. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.

Course Code:	**CSL 445**	**Course Title:**	**Data Sciences**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-Requisite (if Any) :	**Database Management System**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Students will demonstrate skill in data management and proficiency with statistical analysis of data.
2. Students will develop the ability to build and assess data-based models.
3. Students will execute statistical analyses with professional statistical software.
4. Students will apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

Course Contents:

Module 1:

Basic of Data Science: What is Data Science? Big Data and Data Science hype, Why now? – Datafication, Current landscape of perspectives, Skill sets required. Statistical Inference: Populations and samples, Statistical modeling, probability distributions, fitting a model. Business Analytics, Data, Information, Understanding Business Analytics

Module 2:

Python for data Science: Python data types, Python Lists, Conditional Statements, Functions packages, Numpy, matplotlip, control flow and pandas

Module 3:

Introduction to R and Data Manipulation: Data types and function, Variables in R, Scalars, Vectors, Matrices, List, Data frames, functions in R, Factors, Data manipulation: Data sorting, Find and remove duplicates record, Cleaning data, Recoding data, Merging data

Module 4:

Data Import and Exploratory Data Analysis: Data Import: Reading Data, Writing Data in R, Web Scraping, Exploratory Data Analysis: Box plot, Histogram, Pie graph, Line chart

Module 5:
Statistics, Linear, Logistic Regression: Basics of Statistics, Inferencial statistics, Probability, Hypothesis, Standard deviation, Outliers, Correlation, Linear & Logistic Regression,

Module 6:
Clustering techniques, Regression & Classification: Introduction to Data Mining, Understanding Machine Learning, Supervised and Unsupervised Machine Learning, K-means clustering

Text Books:

1. An Introduction to Data Science by Jeffrey S. Saltz and Jeffrey M. Stanton, Sage Publication, 2018.
2. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinny, O'Reilly Media, 2017.
3. Practical Data Science with R, by Nina Zumel, Jim Porzak, John Mount, Publisher: Dreamtech, 2014.

Reference Books:

1. Learning Python, 5th Edition by Mark Lutz, O'Reilly Media, 2017.
2. An Introduction to Data Science by Jeffrey S. Saltz and Jeffrey M. Stanton, Sage Publication, 2018.
3. Data Science by John D. Kelleher and Brendan Tierney, MIT Press, 2018.
4. Principles of Data Science by Sinan Ozdemir, 2018.

Course Code:	**CSL 446**	**Course Title:**	**Neural Networks and Deep Learning**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**Machine Learning**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Ability to understand the fundamental concepts of deep neural networks and underlying error convergence algorithms such as backpropagation and gradient descent.
2. Acquire the depth knowledge of various deep models, architectural aspects, comparative analysis and their applications.
3. Ability to implement, analyze, optimize deep models or the computer vision and natural language processing tasks.
4. Ability to apply the advancements of deep models for various problems in different domain.

Course Contents:

Course Contents:

Module 1:
Neural network working, Introduction of deep neural network. Backpropagation and Gradient Descent algorithms in deep networks. Role of vectorization in various operations in deep learning. Comparisons of shallow and deeper networks.

Module 2:
Bias variance trade-off, overfitting and its remedies, regularization: L1, L2, and dropout. Hyperparameter and its tuning in converging deep networks. Exponentially weighted moving averages, Gradient descent

optimization algorithms and its importance such as momentum, RMSProp, Adam's algorithm, Nesterov, Adagrad, Adadelta, and Adamax Introduction and importance of domain knowledge in deep learning.

Module 3:
Convolutional neural networks, its architecture, deep CNN, parameter sharing, and applications. Generalized parameters computation. Introduction and parameters computation of several pretrained CNN architectures such as LeNet, AlexNet, Inception and its successors, ResNet architectures. Depth-wise pooling: 1x1 convolution.

Module 4:
Recurrent neural networks and its architectural variants such as LSTM, GRU. Architectural aspects and applications. Comparative analysis of RNN, LSTM, GRU.
Word representation: character and word level embedding, CBoW and skip gram model, Word2Vec, negative sampling, GloVe, Attention Model.
Hybridization of deep models.

Module 5:
Applications/Case studies of deep learning models
Computer vision: Introduction to object detection, YOLO, regression formulation, process, network design, and loss function.
Natural language processing: Sentiment Analysis.

Research direction in deep learning. Advanced convolutional: tiled convolution, dilated convolution, transpose or deconvolution, networks in networks. Introduction to bidirectional, multidimensional, stacked sequence models.

Text Books:
1. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville
2. Deep Learning: A Practitioner's Approach, by Adam Gibson and Josh Patterson

Reference Books:
1. Neural Networks and Deep Learning by Michael Nielsen.

Course Code	**CSL448**	**Course Title:**	**Quantum Computing**
Category:	**ELECTIVE**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **0** | **3**
Pre-Requisite (if Any)	**NONE**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:
1. To provide the students an introduction to quantum computation
2. To offer background material related to the algebra of complex vector spaces and quantum mechanics
3. To design application using quantum computing

Course Contents:
Module 1:

Introduction to Quantum Computation: Quantum bits, Bloch sphere representation of a qubit, multiple qubits.

Module 2:
Background Mathematics and Physics: Hilber space, Probabilities and measurements, entanglement, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis.

Module 3:
Quantum Circuits: single qubit gates, multiple qubit gates, design of quantum circuits.

Module 4:
Quantum Information and Cryptography: Comparison between classical and quantum information theory. Bell states. Quantum teleportation. Quantum Cryptography, no cloning theorem.

Module 5:
Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor factorization, Grover search.

Module 6:
Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation.

Text Books:

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press., 2002.
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information , Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific., 2004.
3. Pittenger A. O., An Introduction to Quantum Computing Algorithms, 2000

Course Code :	**CSL449**	**Course Title :**	**Internet of Things: Programming, Protocols and Standards**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			2	**0**	**2**	**3**
Pre-Requisite (if Any) :	**Computer Networks**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:
On completion of this course students will be able to:
1. Identify IoT enabling technologies, components and architectures
2. Analyze various IoT specific protocols and standards
3. Interface I/O devices, sensors & communication modules, and analyze data from various sources
4. Design and Implement IoT based applications on embedded platform

Course Contents:

Module 1:

Introduction to Internet of Things: Definition and Characteristics, Architecture, Physical and Logical Design, IoT Enabling Technologies, IoT Levels and Deployment Templates, Domain Specific IoTs examples.

Module 2:

IoT Hardware: Microcontrollers, Microprocessors, SoC, Interfacing I/O devices, Sensors & Communication Modules; Open-source Electronics Platform such as Arduino, Raspberry Pi, Programming on Arduino and Raspberry PI.

Module 3:

IoT Protocols and Standards: MQTT and SMQTT, CoAP, XMPP, AMQP; IEEE 802.15.4: ZigBee, Bluetooth and BLE, 6LoWPAN and RPL; RF Protocols: RFID, NFC.

Module 4:

IoT Platforms Design Methodology: IoT Physical devices and endpoints, Integration of IoT applications with Cloud computing.

Module 5:

Case Studies Illustrating IoT Design: Home Automation, Cities, Environment, Agriculture, and Productivity Applications.

Text Books:

1. Internet of Things, A Hands-on Approach, Arshdeep Bahga, Vijay Madisetti, University Press, 2015.
2. The Internet of Things, Oliver Hersent, David Boswarthick, and Omar Elloumy, Wiley, 2015.

Reference Books:

1. The Internet of Things - How Smart TVs, Smart Cars, Smart Homes, Smart Cities are changing the World, Michael Miller, 2015, Pearson.
2. Internet of Things Architecture and Design Principles, Raj Kamal, McGraw Hill Education (India) Pvt Ltd, 2017.

List of Lab Assignments / Experiments

1. Programming simple Arduino based IoT projects.
2. Programming Raspberry PI using Python.
3. Integration of IoT applications with the Cloud.

Course Code:	**CSL450**	**Course Title:**	**Computational Genomics and Proteomics**			
Category:	**ELECTIVE**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any):	**Data Structures, Algorithm**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

On completion of this course students will be able to:

1. Analyse the fundamental concepts of Genomics and proteomics.
2. Apply genomics and proteomics algorithms to process biological data.
3. Analysis and Comparison of the difference between various biological data processing algorithms.
4. Design and Evaluate real-life-based data analytics projects and applications.

Course Contents:

Module 1:

Introduction, main biological sequences: DNA, RNA, and protein, central dogma process, DNA: DNA composition: nucleotides, helical structure, intron, exon. Protein: amino acid, Synthesis, protein structure classification, binding site, the active site, and subcellular localization.

Module 2:

Biological sequencing by synthesis, base calling, error in sequencing, Biological sequence reads, working with sequencing reads, positional analyzing the reads, read alignment and its problem, puzzled genome, Biological sequence algorithms: sliding window algorithm, Improving on sliding window algorithm, Biological Sequence programming with Python.

Module 3:

Approximate matching and the pigeonhole principle. Hamming, and Levenshtein distance. Algorithms for approximate Levenshtein distance: Baeza-Yates–Gonnet algorithm, Phonetic coding, Dynamic Programming: The Needleman-Wunsch algorithm and Smith-Waterman algorithm, Wagner-Fischer Algorithm. Pre-processing, and indexing. Hash tables for indexing, methods for hash indexing, Indexing through grouping and ordering, k-mers and k-mer indexes.

Module 4:

Rules of assembly, De novo assembly, Overlaps, and overlap graphs. Algorithms for assembly: Shortest common superstring and the greedy version. Third law of assembly.

Module 5:

Repetitive strings and difficulty in congregation. De-Bruijn graphs and Eulerian walks. How real assemblers work. The future of assembly. Data variability and replication, Data transforms, Clustering, Dimension reduction, Pre-processing, and normalization.

Module 6:

Null and alternative hypotheses analysis: K-mer's algorithm for pattern search, bin creation, extraction of a suitable candidate, filtering criteria, and probabilistic elongation. Machine learning for biological sequences.

Text Book:

1. A Primer for Computational Biology by Shawn T. O'Neil.
2. Introduction to Computational Biology by Michael S. Waterman.

Reference Books:

1. Bioinformatics with Python Cookbook: Learn how to use modern Python bioinformatics libraries and applications to do cutting-edge research in computational biology by Tiago Antao.

2. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology by Dan Gusfield.
3. Machine Learning Methods for Computational Biology by Limin Li.

Data Repository:

1. https://www.ncbi.nlm.nih.gov/gene
2. https://www.ncbi.nlm.nih.gov/nucleotide/
3. https://www.uniprot.org/

Course Code:	**CSL451**	**Course Title:**	**Computational Finance**			
Category:	**ELECTIVE**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite (If any):	**NONE**	**Type of Course:**	**Computer Science Engineering**			

Course Outcomes:

On successful completion of the course, students shall be able to:

1. Understand the structure and functioning of financial markets and derivatives.
2. Derive and use of Martingales in different settings
3. Do time series analysis of the incoming data along with Black-Sholes
4. Demonstrate the use of Reinforcement Learning as an online algorithm for optimization

Course Content:

Module 1 - Financial Markets and Derivatives- Introduction to Financial Markets: Bonds, Stocks, and Derivatives; Organization of Financial Markets: Structure, Participants, and Regulatory Framework; Margins and Transaction Costs: Understanding the cost of trading; Derivatives and Pricing: Options, Futures, and Swaps, Binomial Option Pricing Model, Black-Scholes Model and Option Greeks (Delta, Gamma, Vega, Theta, Rho)

Module 2 - Martingale Theory- Conditional expectations, Martingales, Fair and Unfair games, Convergence Theorem, Doob's Theorem, Ito Calculus, Martingale inequalities, and convergence.

Module 3 - Specialized Martingales – Bounded Martingales, Uniform Martingales, Uniform Integrability, Law of iterated logarithms, Radon-Nikodym Theorem, Supermartingles, and submartingales.

Module 4 - Time Series Analysis, ARIMA, Volatility Modeling: GARCH Models for Volatility Estimation, Technical Analysis.

Module 5 - Reinforcement Learning, Multi-Armed Bandits, Temporal Difference Learning, Q Learning, Q+ Learning, SARSA, Special Topics – Sigma-Algebra, Volatility Modeling, Stochastic Differential Equations

Text Books:

1. "Probability with Martingales", David Williams, Publisher: Cambridge University, 1991.
2. "Introduction to the Economics and Mathematics of Financial Markets", Jaksa Cvitanic and Fernando Zapatero, MIT Press, 2004.
3. "Time Series Analysis: Forecasting and Control", George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung, Fifth Edition, Wiley, 2015.

Reference Books:

1. "Quantitative Trading: How to Build Your Own Algorithmic Trading Business", Ernest P. Chan, 2nd Edition, John Wiley & Sons, September 2021
2. "Reinforcement Learning", Richard Sutton and Andrew Barto, 2nd Edition, MIT Press, 2018

<table>
<tr><td>Course Code</td><td>CSL452</td><td>Course Title</td><td colspan="4">Model-Driven Software Engineering</td></tr>
<tr><td>Category</td><td>ELECTIVE</td><td>Credit Assigned:</td><td colspan="4"><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>0</td><td>3</td></tr></table></td></tr>
<tr><td>Prerequisite (if Any)</td><td>NONE</td><td>Type of Course</td><td colspan="4">Computer Science and Engineering</td></tr>
<tr><td colspan="6">Course Outcomes:
On successful completion of the course, students shall be able to:
1. Understand model-driven software engineering principles
2. Apply modeling languages and constraints to define model-driven architectures
3. Analyze model to model transformations, classification and types of transformations
4. Evaluate model to text transformations, code generation, and transformation languages</td></tr>
<tr><td colspan="6">Course Contents:

Module 1: Model-Driven Software Engineering Principles
Overview of Model-Driven Software Engineering methodology, Tool support, Automating software development, Use cases of Model-Driven Software Engineering

Module 2: Model-Driven Architectures
Definitions and Assumptions, Modeling Levels, General purpose and domain specific languages, Architecture-driven modernization, Business process modeling

Module 3: Modeling Languages
Anatomy of Modeling Languages, Multi view modeling and language extensibility, UML and UML Extensibility, Overview of domain specific languages, Defining modeling constraints

Module 4: Model to Model Transformations
Model transformations and their classification, Exogenous and Out-place transformations, Endogenous and In-place transformations, Bi-directional model transformations

Module 5: Model to Text Transformations
Basics of Model driven code generation, code generation through programming languages, code generation through model to model transformations, Template based transformation language

Text Books:
● Marco Brambilla, Jordi Cabot, Manuel Wimmer “Model-Driven Software Engineering in Practice” Morgan and Claypool 2nd Edition
● Jack Herrington “Code Generation in Action” Manning

Web References: https://www-users.cs.york.ac.uk/dkolovos/research/recommended-reading/</td></tr>
</table>

Course Code:	CSL453	Course Title:	Computational Logic Design Methodology			
Category:	ELECTIVE	Credit Assigned	L	T	P	C
			1	0	4	3
Pre-Requisite (if Any)	Data Structure, DSA, Design and Analysis of Algorithms	Type of Course	Computer Science and Engineering			

Course Outcomes:

On successful completion of the course, students shall be able to:

1. Apply the learnt knowledge of data structure and algorithm course in problem formulation and its solution.
2. Analyze the computational and space complexity of the proposed feasible & optimal solution for various problems.
3. Optimize the solution by considering the tradeoff between time and space complexity and selection of an underlying data structure for formulating the solution.

Course Contents:

Module 1: Introduction

Algorithm & Data structure Overview, Growth of a function, Feasible & Optimal Solution, Computational & space complexity, mathematical overview. Elementary data structures: array, stack, queues, linked list, trees, graphs, priority queue, sets, union-find, etc. Recursive & Iterative formulation, algorithm design paradigm overview.

Module 2: Linear DS

Two sum Problem & its extension to three sum, k sum problem. First missing positive number, sliding window maximum problem, two pointer techniques, slow and fast pointer, remove nth node from end of list, trapping rain water, valid palindromes, linked list cycle detection, palindrome list, happy numbers, valid parenthesis, reverse polish notation

Module 3: Tree and Graph

Maximum depth of Binary tree, Max path sum, number of islands, kth smallest in BST, longest increasing path in a matrix, topological sort, connected components

Module 4: DAC & Greedy Approach

Sorted Array problems: Median, merge k sorted lists, maximum subarray, kth smallest/largest element in sorted/unsorted array, searching in 2D matrix, top k frequent items, Longest Substring with At Least K Repeating Characters, majority element, Sorted Array problems: duplicates removal, rotation, shifting, etc. Tiling and skyline problem.

Activity selection, Huffman coding, Egyptian fraction, Minimum product subset of an array, Partitioning array problems, first fit / best fit algorithm, Spanning trees, coin change

Module 5: Dynamic programming & Backtracking

Maximal subarray, Maximum product subarray, Subset sum, Pascal's Triangle, coin change, common/increasing substring & subsequence, palindrome substring and subsequence, largest rectangle histogram & Maximum sum rectangle in a 2D matrix.

Palindrome Partitioning, Word Wrap Problem, Word break, Shortest path-based problems

Text Books:

1. Algorithms by Jeff Erickson (freely available online)
2. Algorithms Illuminated by Tim Roughgarden
3. Competitive Programming 4: The Lower Bound of Programming Contests in the 2020s by Steven Halim and Felix Halim

References:

1. Introduction to Algorithms: Cormen T.H. et.al: Prentice Hall of India
2. Computer Algorithms: Horowitz, Sahani, Rajsekharan , Galgotia Publications Pvt.Ltd

3. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests Antti Laaksonen

Course Code	**CSL454**	**Course Title:**	**Reinforcement Learning**
Category:	**ELECTIVE**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Pre-Requisite (if Any)	**ML (CSL 422)**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

1. To incorporate and analyse various elements & characteristics of reinforcement learning
2. To Formulate decision problems and set up, run, and analyse computational experiments
3. To Apply reinforcement learning for various real-life problems

Course Contents:

Module 1:

Course logistics and overview. Origin and history of Reinforcement Learning research. Its connections with other related fields and with different branches of machine learning. Brush up of Probability concepts - Axioms of probability, concepts of random variables, PMF, PDFs, CDFs, Expectation. Concepts of joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence.

Module 2:

Introduction to RL terminology, Markov property, Markov chains, Markov reward process (MRP). Introduction to and proof of Bellman equations for MRPs along with proof of existence of solution to Bellman equations in MRP. Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.

Module 3:

Overview of dynamic programming for MDP, definition and formulation of planning in MDPs, principle of optimality, iterative policy evaluation, policy iteration, value iteration, Banach fixed point theorem, proof of contraction mapping property of Bellman expectation and optimality operators, proof of convergence of policy evaluation and value iteration algorithms, DP extensions.

Module 4:

Overview of Monte Carlo methods for model free RL, First visit and every visit Monte Carlo, Monte Carlo control, On policy and off policy learning, Importance sampling. Getting started with the function approximation methods, Revisiting risk minimization, gradient descent from Machine Learning, Gradient MC and Semi-gradient TD(0) algorithms, Eligibility trace for function approximation, Afterstates, Control with function approximation, Least squares, Experience replay in deep Q-Networks.

Module 5:

Getting started with policy gradient methods, Log-derivative trick, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods.

Text Books:

1. Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition.

2. Algorithms for Reinforcement Learning. C. Szepesvari. Morgan and Claypool Publishers, 2010

Reference Books:

1. Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn van Otterlo, Eds.
2. Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

Course Code:	**CSL 456**	**Course Title:**	**Natural Language Understanding and Generation**			
Category:	**ELECTIVE**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any) :	**NONE**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

On successful completion of the course, students shall be able to:

1. Convert text into vectors using various methods enabling text/speech analysis.
2. Implement a document retrieval system/search engine/similarity search/vector similarity.
3. Apply Deep learning for Natural language understanding and language generation.
4. Study case studies related to Natural Language Processing like sentiment analysis, question answering, etc.

Course Contents:

Module 1:

Basics of Natural language processing, Understanding and Generation, Vector Models & Text Pre-processing, Bag of Words, Count Vectorizer (Theory), Tokenization, Stop words, Stemming and Lemmatization, Vector Similarity, TF-IDF, Word-to-Index Mapping, NLP Tasks.

Module 2:

Introduction to Machine Learning and Deep Learning Models, Probabilistic Models (Introduction), Markov Chain, PCA, LSA/LDA, SVD, Probability Smoothing and Log-Probabilities, Language Models, and Sequence Models.

Module 3:

Neural Word Embeddings: Word2Vec, GloVe, FastText, Embedding for Language Models (ELMo), Contextualized word embeddings, Beam Search, Introduction to evaluation metrics: BLEU, GLUE Benchmarks.

Module 4:

Transformer model and its building blocks: Positional Encoding, Self-Attention, Multihead Attention, Masked Multi-head Attention, Bidirectional Encoder Representations from Transformers (BERT), Modifications and Advancements in BERT, Large Language Model, Megatron, ChatGPT, Pretraining and fine-tuning of the language models.

Module 5:

Case studies: Text Classification, Sentiment Analysis, Named Entity Recognition, Spam detection, Recommendation systems, Question answering system, Topic Modelling.

Text Books:

1. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", Daniel Jurafsky and James H. Martin, 2nd edition, Pearson Education India, 2013.
2. "Natural Language Processing with Transformers: Building Language Applications with Hugging Face", Lewis Tunstall, Leandro von Werra, Thomas Wolf., 1st edition, O'Reilly Media, 2022

Reference Book:

1. "Natural Language Processing with Python: Analysing Text with the Natural Language Toolkit", Steven Bird, Ewan Klein, Edward Loper., O'Reilly Media, 2011.
2. "Neural Network Methods for Natural Language Processing (Synthesis Lectures on Human Language Technologies)", Yoav Goldberg, Graeme Hirst, Morgan & Claypool 2017.
3. "Natural Language Processing in Action: Understanding, analyzing, and generating text with Python", Hobson Lane, Hannes Hapke, Cole Howard, 1st edition, Manning Publication, 2019.

Course Code:	**CSL 457**	**Course Title:**	**Parallel Computer Architecture**			
Category:	**ELECTIVE**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any):	Computer System Organization / Computer Architecture and Organization, Operating Systems	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

On successful completion of the course, students will be able to:

1. Understand various trends in parallel computer design, architecture of advanced / parallel processors.
2. Apply advance parallelism concepts at instruction level, thread level and data level in multi/many core systems.
3. Analyze the advanced concepts of cache memory organization and optimization techniques.
4. Evaluate the suitability of various parallel architectures (such as multi / many core / GPU architectures) with respect to its memory organization, processor organization, interconnection network etc.

Course Contents:

Module 1:

Introduction to parallel programming models and architecture: Need of Parallel Computing, Moore'law, Dennard Scaling, Amdalh's Law, Multicore revolution, Moore's Law for parallel architectures, performance metrics and evaluation, benchmarks, Gem Simulator, shared memory model, message passing model.

Module 2:

Processor Architectures: Pipelined Processor Architecture (Case study of 5 stage pipeline of MIPS processor: Datapath and control) and issues of hazards (RAW, WAR and WAW hazards) and overcoming methods, Branch Prediction techniques (1-bit predictor, 2-bit predictor, correlating predictor etc), Floating point pipelined datapath and control.

Module 3:

Superscalar Processor Architecture (Tomosulo Algorithm), VLIW Processors, Compiler optimization (loop unrolling), Dynamic Instruction Scheduling, ILP, ILP with Speculation, TLP, Hazard Detection and Recovery. Multiprocessors and multi-computers, Brief overview of SIMD, MIMD, vector architectures and multi-core architectures.

Module 4:

Memory Organization and Cache Memory Basics: Recap of Memory Organization: Memory Hierarchy-Cache and Virtual memory, Overview of Cache coherence, Coherence Protocols-Snooping, Directory based protocols, MSI, MESI, and Correctness of coherence protocols- Types of cache misses, update vs invalidate protocol, NUMA architecture

Module 5:

Cache Memory Optimization: AMAT, Optimization techniques to reduce the miss rate, miss penalty, hit time etc. (Six basic Cache Optimization Techniques and few more optimizations), Case studies: MIPS processor, ARM processor, Introduction to Reconfigurable Computing (FPGA), GPU Architecture (AMD/Nvedia/Intel).

Text Books:

1. "A Quantitative Approach: Hennessy and Patterson: Computer Architecture", John L. Hennessy and David A, 4th Edition , Morgan Kaufmann, 2006.
2. "Advance Computer Architecture: Parallelism, Scalability, Programmability", Kai Hwang and Naresh Jotani, 3rd Edition, McGraw Hill, 2015.
3. "Computer Organization and Design: The Hardware Software Interface", Hennessy and Patterson, 3rd Edition, Morgan Kaufmann, 2004.

Course Code:	**CSL 465**	**Course Title:**	**Cyber Security**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			2	**1**	**2**	**4**
Pre-Requisite (if Any) :	**None**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

- Able to employ, design and implement appropriate security methodology and policies to offer security to computers and digital information
- Identify & analyse Information Security threats and vulnerabilities in various systems

- Apply security metrics to real world scenarios
- Identify trade-offs and concessions in the design and development process of security system
- Enforce the standards and cyber laws to enhance security in the development of security system

Course Contents:

Module 1: Introduction to Cyber Security

Overview of Cyber Security, Internet Governance – Challenges and Constraints, Cyber Threats:- Cyber Warfare-Cyber Crime-Cyber terrorism-Cyber Espionage, Need for a Comprehensive Cyber Security Policy, Need for a Nodal Authority, Open Source/ Free/ Trial Tools: nmap, zenmap, Port Scanners, Network scanners, Metasploit, Manual Inspection, Search Tools for Profiling, Automated Web Crawling, General Countermeasures.

Module 2: Cyber Security Vulnerabilities and Cyber Security Safeguards

Cyber Security Vulnerabilities-Overview, Weak Authentication, Unprotected Broadband communications, Poor Cyber Security Awareness. Classes of Attack – Code injection, Time-of-check-to-time-of-use race conditions, Sybil attack, Distributed Denial of Service and BGP hijacking, SQL command injection, phishing, and cross-site scripting (XSS).
Cyber Security Safeguards- Overview, Access control, Audit, Authentication, Biometrics, Cyber Attacks and Mitigation Techniques, Deception, Denial of Service Filters, Security policy, Threat Management.

Module 3: Online Anonymity with Securing Web Application, Services and Servers: Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns, Challenges, Obfuscation and diversity methods, Differential Privacy (Dwork), Anonymous Networks, Tor Network, I2P Network, Freenet, Darknet, Anonymous OS – Tails, Secure File Sharing, VPN, Proxy Server, Connection Leak Testing, Secure Search Engine, Web Browser Privacy Configuration, Anonymous Payment.

Module 4: Intrusion Detection and Prevention

Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques, Anti-Malware software, Network/host based Intrusion detection/prevention Systems, Network Session Analysis, System Integrity Validation, Cloud Storage Encryption, Encrypt DNS Traffic and Email communication, Secure IM and video calls, Username/password Threats, Password Guessing and its Countermeasures, Eavesdropping attacks and its Countermeasures, Forms-based Authentication attacks and its countermeasures.

Module 5: Indian IT Act and Standards of Cyber Law

Cyber Security Regulations, Indian IT ACT, Adjudication under Indian IT ACT, IT Service Management Concept, IT Audit standards, ISO/IEC 27000 Series, COBIT, HIPPA, SOX, System audit, Information security audit, ISMS, SoA (Statement of Applicability), BCP (Business Continuity Plan), DR (Disaster Recovery), RA (Risk Analysis/Assessment).

Module 6: Cyber Forensics

Introduction to Cyber Forensics, Need of Cyber Forensics, Cyber Evidence, Forensic Ballistics and Photography, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, WIFI Security (War-driving), Network Forensics, Mobile Forensics, Cloud Forensics, Investigation Tools, eDiscovery, EDRM Model, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail

Recovery, Hands on Case Studies, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

Text Books:

1. Scott Augenbaum, The Secret to Cybersecurity: A Simple Plan to Protect Your Family and Business from Cybercrime, 2019.
2. Yuri Diogenes, Erdal Ozkaya, Cybersecurity: Attack and Defense Strategies: Infrastructure security with Red Team and Blue Team tactics, 2018
3. Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, Cybersecurity Essentials, 2018

Reference Books :

1. Michael Sikorski, Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, 2012.
2. Raef Meeuwisse, Cybersecurity for Beginners, 2017.
3. William Stallings, "Cryptography and Network Security", Pearson Education/PHI, 2006.

Course Code:	**CSL 466**	**Course Title:**	**Blockchain Technology**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			2	**1**	**0**	**3**
Pre-Requisite (if Any) :	**None**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

- Familiarise the functional/operational aspects of cryptocurrency Ecosystem.
- Analyse emerging abstract models for Blockchain Technology.
- Identify major research challenges and technical gaps existing between theory and practice in cryptocurrency domain
- Investigate and apply the concept of Blockchain Technology to solve the real world problems
- Design and develop new blockchain to address the need of new transactions to offer more security

Course Contents:

Module 1: Introduction

Introduction, Applications of Blockchain Technology, opportunity & challenge in Blockchain Technology, Market Growth of Blockchain Technology, The consensus problem - Asynchronous Byzantine Agreement - AAP protocol and its analysis - Nakamoto Consensus on permission-less, nameless, peer-to-peer network - Abstract Models for BLOCKCHAIN - GARAY model - RLA Model - Proof of Work (PoW) as random oracle - formal treatment of consistency, liveness and fairness - Proof of Stake (PoS) based Chains - Hybrid models (PoW + PoS),

Module 2: Blockchain Smart Contracts & Wallet

Blockchain Features, Public VS private Blockchain, Cryptography, Hashing and Transactions, Blockchain Structure, Mining and Consensus, How Blockchain Works, Centralization and

Decentralization, Smart Contracts, Deploying Smart Contracts, Key Properties of smart Contracts, Language for Smart Contracts, Wallets, Transactions, Public & Private keys.

Module 3: Bitcoin Case Study of Blockchain

Bitcoin introduction, Getting Bitcoin Wallet, Getting and Sending Bitcoins, Working of Mining Blockchain, Earning Bitcoin Programmatically, Acceptance of Bitcoin on Website Bitcoin, Blocks - Merkley Tree - hardness of mining - transaction verifiability - anonymity - forks - double spending - mathematical analysis of properties of Bitcoin.

Module 4: Blockchain based crypto currency and Ethereum

Understanding various Cryptocurrency, Concept of Opening Cryptocurrency Wallets, Buying Cryptocurrency Wallet, Withdrawal Cryptocurrency Wallets, Bitcoin with Cryptocurrency, Ethereum, Ethereum Virtual Machine (EVM), Wallets for Ethereum – Solidity, Ethereum, Ethereum Virtual Machine, blockchain applications using the Ethereum.

Module 5: Network Applications on Hyperledger

(Trends and Topics) - Zero Knowledge proofs and protocols in Blockchain - Succinct non interactive argument for Knowledge(SNARK) - pairing on Elliptic curves – Zcash, Fundamentals Of Hyperledger, Ethereum VS Hyperledger, Genesis Block Configuration, Creating Network Using Fabric

Text Books:

- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016.

Reference Books / Papers:

- Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015.
- J.A.Garay et al, The bitcoin backbone protocol - analysis and applications EUROCRYPT 2015 LNCS VOl 9057, (VOLII), pp 281-310.
- R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks , EUROCRYPT 2017, (eprint.iacr.org/2016/454).
- R.Pass et al, Fruitchain, a fair blockchain, PODC 2017 (eprint.iacr.org/2016/916).

Course Code:	**CSL 467**	**Course Title:**	**Industrial and Social Applications of Digital Twins**			
Category:	**ELECTIVE**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			2	**0**	**2**	**3**
Pre-requisite (If any):	**NONE**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

On successful completion of the course, students will be able to:

1. Understand the fundamental concepts and applications of Digital Twin technology.
2. Develop skills in modelling and simulating complex systems.

3. Apply Digital Twin solutions to industrial and social systems for decision-making.
4. Solve multidisciplinary problems through system dynamics and AI-based modelling.
5. Explore emerging trends and envision future advancements in Digital Twins.

Course Contents:

Module 1: Foundations of Digital Twins

Understanding the Concept of Digital Twins; Historical Evolution and Significance; Basics of Creating a Digital Twin- GML Structure, Domain Modelling; Types of Digital Twins, The Role of Digital Twins in Decision-Making and Control.

Module 2: Modelling and Simulation with Digital Twins

Basics of System Modelling; Navigating Uncertainty; Types of modelling techniques - Enterprise modelling, System Dynamics, AI-based modelling; Simulation techniques; Simulation Technologies- Vensim, AnyLogic, NetLogo; Assignment and mini case study, Discussion, Q&A and evaluation.

Module 3: Digital Twins for Business Systems

Types of Problems and their Complexities; Industrial Applications - Cyber Physical Systems, Supply Chain, Product and Customer Digital Twin; Industry Talk by invited speaker; Assignment and mini case study; Discussion, Q&A and evaluation.

Module 4: Role of Digital Twins for Social Systems

Digital Twins in Healthcare; Water Management, Agriculture, and Traffic Control; Instructor Guided Assignment and mini case study; Discussion, Q& A and evaluation

Module 5: Future Trends and Practical Applications

Exploring Upcoming Trends; Imagining the Future; Group Projects- Problem Definition, Practical Steps in Creating a Digital Twin, Teams work on applying digital twin concepts to a real-world problem, Presentation, and findings.

References:

1. Book: Digital Twins, Simulation, and the Metaverse: Driving Efficiency and Effectiveness in the Physical World through Simulation in the Virtual Worlds, Michael Grieves (Editor), Edward Y. Hua (Editor), Springer, 2024

   https://www.amazon.com/Digital-Twins-Simulation-Metaverse-Effectiveness/dp/3031691067

2. Book chapter: Enterprise Digital Twin: An Approach to Construct Digital Twin for Complex Enterprises, Souvik Barat, IGI Global, 2020

   https://www.igi-global.com/chapter/enterprise-digital-twin/256901

3. "Enabling technologies and tools for digital twin.", Qi, Qinglin, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, and Andrew YC Nee. *Journal of Manufacturing Systems* 58, 3-21, 2021.

4. “Human Digital Twin in the context of Industry 5.0.”, Wang, Baicun, Huiying Zhou, Xingyu Li, Geng Yang, Pai Zheng, Ci Song, Yixiu Yuan, Thorsten Wuest, Huayong Yang, and Lihui Wang., *Robotics and Computer-Integrated Manufacturing*, Elsevier, 85, 102626, 2024.
5. “Digital twin in industry: State-of-the-art”, Tao, Fei, He Zhang, Ang Liu, and Andrew YC Nee., 15, no. 4, 2405-2415, *IEEE Transactions on industrial informatics-IEEE explore* , 2018.
6. "Digital twin modeling.” Tao, Fei, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, and Ping Ji. *Journal of Manufacturing Systems* 64: 372-389, 2022.
7. "Advancements and challenges of digital twins in industry.” Tao, Fei, He Zhang, and Chenyuan Zhang. 4, no. 3: 169-177.*Nature Computational Science*, 2024

Course Code:	**CSL 468**	**Course Title:**	**Intelligent Systems**			
Category:	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Pre-Requisite (if Any):	**NONE**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

On successful completion of the course, students will be able to:

1. Analyse and implement various search techniques to solve real-world problems.
2. Create automated games and solve constraint satisfaction problems.
3. Apply algorithms to draw automated inferences using logical agents.
4. Understand AI ethics and apply practical algorithms for making decisions under uncertainty and for solving planning problems

Course Contents:

Module 1:

The foundation of AI, the state of the Art in AI, Agents, and environment, Good behaviour – the concept of rationality, Problem-solving agent, Search algorithms – Best First Search, Search data structures, Redundant paths, Measuring problem-solving performance, Uninformed Search Strategies - Breadth-first search, uniform-cost search, Depth-first search and the problem of memory, Depth-limited and iterative deepening search, Bidirectional search, Comparing uninformed search algorithms.

Module2:

Informed (Heuristic) Search Strategies - Greedy best-first search, A* search, Satisficing search: Inadmissible heuristics and weighted A*, Memory-bounded search, Recursive Best First Search, Heuristic Functions - The effect of heuristic accuracy on performance, Generating heuristics.

Module 3:

Search in Complex Environment - Local Search and Optimization Problems Adversarial Search and Games: Optimal decision in Games, the Minimax algorithm, Alpha-Beta pruning, Stochastic games, Limitations of Game Search Algorithms.

Module 4:

Knowledge Based Agents, Logic, Propositional Logic, Propositional Theorem Proving, Inference, Equivalence, Validity and Satisfiability, Resolution, Forward and Backward Chaining, DPLL algorithm, First Order Logic, Models for first-order logic, Symbols and Interpretations, Terms, Atomic sentences, complex sentences, Quantifiers, Inference in FOL, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

Module 5:

Constraint Satisfaction Problems, Backtracking Search, variable and value ordering, constraint propagation, intelligent backtracking, local search for CSPs, Automated Planning, Language of planning problems, planning with state-space search, forward and backward state-space search, Heuristics for state-space search.

Module 6:

Uncertain knowledge and reasoning - Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Bayes' Rule and Its Use, Applying Bayes' rule, Using Bayes' rule, Naive Bayes Models, Text classification with naive Bayes. The Semantics of Bayesian Networks. The Ethics of AI – Fairness and bias, trust and transparency, AI safety.

Text Books:

1. "Artificial Intelligence a Modern Approach", Stuart Russell and Peter Norvig, 4th edition, Pearson, 2020.

Reference Book:

1. "Artificial Intelligence: Foundations of Computational Agents", David L. Poole and Alan K. Mackworth, 3rd edition, Cambridge University Press, 2023.