<table>
<tr><td>I</td><td>Course Code</td><td>CS 243005</td></tr>
<tr><td>II</td><td>Course Title</td><td>Compiler Design</td></tr>
<tr><td>III</td><td>Credit Structure</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>3</td><td>4.5</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Theory of computation, Data structures, Programming</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>Learning Outcomes: after completing this course, the students will be able to
Understand the functionality of each phase of compiler design and and how they contribute to transforming high-level source code into executable machine code;
Design and implement lexical analyzers (lexers) and parsers for a given programming language using tools such as Lex/Flex and Yacc/Bison
Perform semantic analysis, including symbol table management and type checking, to ensure the correctness of programs.
Generate intermediate code representations and understand how these intermediate codes bridge the gap between high-level language constructs and target machine code.
Apply various optimization techniques to enhance the efficiency of generated code.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Introduction to Compiler Design: Overview of Compilers; Definition and purpose; Phases of compilation; Structure of a compiler; Compiler vs. Interpreter; Applications of Compilers

Lexical Analysis: Introduction - Role of the lexical analyzer; Tokens, lexemes, and patterns; Regular Expressions- Definition and examples; Converting regular expressions to finite automata; Deterministic Finite Automata (DFA) and Non-Deterministic Finite Automata (NFA); Lexical Analyzer Generators- Introduction to tools like Lex/Flex

Syntax Analysis: Introduction- Role of the parser, Context-Free Grammars (CFG), Parse trees and derivation trees; Parsing Techniques- Top-down Parsing such as Recursive Descent Parsing and Predictive Parsing for LL(1) Grammar; Bottom-up Parsing such as Shift-Reduce Parsing, LR Parsing - LR(0), SLR, LALR, and CLR; Parser Generators- Introduction to tools like Yacc/Bison

Syntax-Directed Translation: Introduction- Syntax-Directed Definitions (SDD), Syntax-Directed Translation Schemes (SDTS); Intermediate Code Generation- Three-address code, Types of intermediate code representations;

Semantic Analysis: Introduction- Role of semantic analyzer, Symbol Table Management; Type Checking- Types and Type Systems, Type Conversions and Coercions; Error Detection and Reporting;

Optimization: Introduction- Importance of optimization, Types of Optimization; Local Optimization Techniques such as Common Subexpression Elimination, Dead Code Elimination; Global Optimization Techniques such as Loop Optimization, Instruction</td></tr>
</table>

		Selection and Scheduling Code Generation and further optimization: Introduction- Role of code generator, Target Code Generation; Instruction Set Architectures (ISA)- Basic concepts, Mapping intermediate code to target machine code; Code Emission Techniques; Peephole Optimization and Register Allocation; Introduction to Linking and Loading; Static vs. Dynamic Linking;
VII	**Text / Reference books:**	1. Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Addison Wesley Publisher 2. Engineering a Compiler by Keith D. Cooper and Linda Torczon, Morgan Kaufmann Publisher Data Flow Analysis: Theory and Practice Hardcover by Uday Khedker, Amitabha Sanyal, and Bageshri Karkare, CRC press

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243006</td></tr>
<tr><td>II</td><td>Course Title</td><td>Information Security</td></tr>
<tr><td>III</td><td>Credit Structure</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>IV</td><td>Prerequisites:</td><td>Computer Networks</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>Learning Outcomes: after completing this course, the students will be able to
Learn the design principles of Block ciphers, stream ciphers, hash functions, MAC’s, public key encryption, and digital signatures.
To protect data and respond to threats that occur over the Internet.
An ability to apply security principles and practices to the environment, hardware, software, and human aspects of a system.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Introduction: Core Information Security Principles, CIA (Confidentiality, Integrity, Availability), Classical Cryptography, Modern Cryptography
Symmetric key cryptography: Block ciphers, stream ciphers, hash functions, MAC, authenticated encryption.
Asymmetric Key cryptography: Number theory and Finite fields, DHKE, RSA, Elliptic Curve Cryptography, Digital signatures, Public Key Infrastructure and Digital certificates.
Network Security: TCP/IP threats, The IPSEC protocol, The SSL and TLS protocols, Firewalls and Virtual Private Networks (VPNs), Electronic mail security, Worms, DDoS attacks, BGB and security considerations.
Intrusion Detection and Prevention: Intrusion detection and Prevention techniques, Anti-malware software, Security information management, Network session analysis, System integrity validation</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>Fundamentals of Information Systems Security By David Kim, Michael G. Solomon, Jones & Bartlett Learning
Katz, Lindell: Theory of modern cryptography, CRC Press, 3rd Edition.
Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management; Hossein Bidgoli, John Wiley & Sons
Douglas R. Stinson, Maura B. Paterson (2018). Cryptography: theory and practice 4/E Chapman and Hall/CRC.
P.W. SINGER, A. FRIEDMAN (2014), Cybersecurity: What Everyone Needs to Know, OUP, 1st Edition.</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243007</td></tr>
<tr><td>II</td><td>Course Title</td><td>Distributed Systems</td></tr>
<tr><td>III</td><td>Credit Structure</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>IV</td><td>Prerequisites:</td><td>Operating Systems, Data Structure and Algorithms, Computer Networking</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>Upon successful completion of this course, students will be able to understand the core concepts in distributed computing, such as logical clocks, consistent cuts, consensus, replication and fault tolerance, etc., as well as the common techniques to implement them, so as to build correct and performant distributed systems and applications.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Introduction to Distributed Systems, Basic Algorithms in Message Passing System, Leader Election in Rings, Distributed Minimum Spanning Tree, Models of Distributed Computation, Causality & Logical Time, Global State and Snapshot Recording Algorithms,
Distributed Mutual Exclusion Algorithms, Distributed Shared Memory, Consensus and Agreement Algorithms, Impossibility of distributed consensus with one faulty process, Fault-tolerance, PAXOS and Friends, Distributed Data Analytics: MapReduce, Distributed Machine Learning, Blockchain Consensus: Introduction to Blockchain Consensus, blockchain, Bitcoin blockchain consensus, The CAP Theorem, Advanced Topics: Edge Computing</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>1. Kshemkalyani, Ajay D., and Mukesh Singhal. Distributed computing: principles, algorithms, and systems. Cambridge University Press, 2011.
2. Attiya, Hagit, and Jennifer Welch. Distributed computing: fundamentals, simulations, and advanced topics. Vol. 19. John Wiley & Sons, 2004.
3. Lynch, Nancy A. Distributed algorithms. Elsevier, 1996.</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 245001</td></tr>
<tr><td>II</td><td>Course Title</td><td>Computer Vision</td></tr>
<tr><td>III</td><td>Credit Structure</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>0</td><td>3</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Programming and data structures, basic algebra and machine learning techniques</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
Understand the functionality of different phases of computer vision and image processing.
Understand the concepts related to feature extraction, semantic segmentation, object detection, and pattern recognition.
Apply computer vision techniques to various application domains including Biometrics, Medical diagnosis, document processing, mining of visual content, advanced rendering, etc.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Digital Image Formation and low-level processing:
Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.
Feature Extraction:
Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, HOG, GLOH, Scale-Space Analysis- Image Pyramids.
Image Segmentation:
Region Growing, Edge Based approaches to segmentation, and Texture Segmentation
Image geometry: Single view geometry, Multi view geometry, Applications of image geometry.
Machine Learning for Computer Vision:
Convolutional Neural Networks, Semantic segmentation, Object detection, Generative Adversarial Networks and Transformers.</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2022.
Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.
Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243008</td></tr>
<tr><td>II</td><td>Course Title</td><td>Mobile Computing</td></tr>
<tr><td>III</td><td>Credit Structure</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>0</td><td>3</td></tr>
</table>
</td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Computer Networks, Operating Systems</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
• Describe the working principles of various cellular technologies, as well as appreciate the associated challenges.
• Compare/contrast various aspects of Mobile and Pervasive Computing, viz. localization, sensing, security and privacy concerns, application development, etc.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>• Overview of Wireless Systems: Infrastructure-based vs Ad-hoc, Wireless LANs, Cellular systems, Sensor networks, Bluetooth, WiFi, WiMAX.
• Mobile Edge Computing (MEC): Overview, MEC architecture and applications, Workload balancing and offloading, Green MEC.
• Mobility and Handoff Management: Link layer mobility mechanisms (location management protocols), Network layer mobility mechanisms (Macro and Micro mobility protocols), Handoff management protocols, Mobile IP.
• Cellular Networks: LTE and 5G overview, 5G Architecture, RAN and dynamic CRAN, Mobility management and Network slicing in 5G.
• Communication technologies such as Lora, Sigfox, UWB.
• Smartphone-based platform architectures and applications.
• Energy management with continuous and multi-modal data streaming.
• Localization and Location-based services (LBS).
• Selected case studies (Activity monitoring, UAV, Connected vehicles)</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>1. I. STOJMENOVIC (2002), Handbook of Wireless Networks and Mobile and Pervasive Computing, Wiley.
2. 2. S. LOKE (2006), Context-aware Pervasive Systems: Architectures for a New Breed of Applications, CRC Press.
3. A. OSSEIRAN, J.F. MONSERRAT, P. MARSCH, (Eds.), 5G Mobile and Wireless Communications Technology, Cambridge University Press, 2016.
4. Selected papers and material from top conferences and journals such as MobiSys, SenSys, MobiCom, UbiComp, TMC, etc.</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 245002</td></tr>
<tr><td>II</td><td>Course Title</td><td>Operations Research</td></tr>
<tr><td>III</td><td>Credit Structure</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>0</td><td>3</td></tr>
</table>
</td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Basics of algebra and calculus; Basics of probability and statistics
Familiarity with programming languages</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
• Develop mathematical models to represent and solve real-world optimization problems, including linear programming, integer programming, and network optimization
• Utilize appropriate algorithms and methods, such as the Simplex method, Branch and Bound, and heuristic approaches, to find optimal solutions to various types of optimization problems
• Analyze how changes in problem parameters impact the optimal solution and interpret the results to make informed decisions
• Employ software tools and programming languages to implement and solve optimization models.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Introduction: Overview of Operations Research (OR) and its applications, Modeling and solution techniques;
Linear Programming: Formulation of Linear Programming Problems; Graphical Method for LP problems with two variables; Simplex algorithm; Sensitivity Analysis and Interpretation; Duality Theory: Primal and Dual Problems; Dual Simplex Method; applications;
Integer Programming: Basics and Applications; Formulation of Integer Programming Problems; Branch and Bound Method; Branch and Cut Method; Cutting Planes Method; Mixed-Integer Programming;
Network Optimization: Shortest Path Problem: Algorithms (Dijkstra’s, Bellman-Ford); Maximum Flow Problem: Ford-Fulkerson Algorithm; Minimum Cost Flow Problem; Transportation Problem and Assignment Problem: Formulation and Solution;
Nonlinear Programming: Introduction to Nonlinear Programming Problems; Unconstrained Optimization: Gradient Descent, Newton’s Method; Constrained Optimization: Lagrange Multipliers; KKT Conditions and their Application; Quadratic Programming: Basics and Applications;
Dynamic Programming: Introduction to Dynamic Programming; Deterministic Dynamic Programming: Multistage Decision Processes; Stochastic Dynamic Programming: Markov Decision Processes (MDPs); Applications: Inventory Control, Resource Allocation
Simulation and Heuristic Methods: Basics of Monte Carlo Simulations; Heuristic Methods: Genetic Algorithms, Simulated Annealing; Metaheuristics: Ant Colony Optimization, Particle Swarm Optimization;</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>1. Introduction to Operations Research by FREDERICK S. HILLIER and GERALD J. LIEBERMAN. McGrowHill (MGH) publisher
2. Operations Research – an introduction by Hamdy A Taha; Person publisher</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 245003</td></tr>
<tr><td>II</td><td>Course Title</td><td>Deep learning</td></tr>
<tr><td>III</td><td>Credit Structure</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>0</td><td>3</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Machine Learning</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
Student will be able to understand the applicability of Advanced machine learning technique in different discipline
Students will be equipped to create and evaluate innovative problem-solving approaches and machine learning techniques tailored to address data analysis challenges arising in emerging applications.
Students will be capable of scaling machine learning techniques to large datasets by utilizing novel data structures and emerging computational tools, even after the course has concluded</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Module 1: Introduction to Machine learning and Deep learning; Application in Different Domain; Image processing; Computer vision, anomaly detection; Data Creation. Feed Forward Network; Back Propagation algorithms.

Module 2: Convolutional Neural Networks (CNN): convolution, filters, pooling, stride, drop out, layers and applications; Fully Connected Layers; Popular CNN Architectures (LeNet, AlexNet, VGG, ResNet); Transfer Learning and Fine-Tuning; Data Augmentation and Regularization Techniques; Backpropagation Through Time (BPTT)

Module 3: Recurrent Neural Networks (RNN): unfolding, bidirectional networks, encoder, decoder and attention models; LSTM, GRU; Applications.

Module 4: Generative Adversial Network (GAN): Generator and Discriminator; Training Challenges and Solutions; Variants of GANs (DCGAN, WGAN, CycleGAN); Autoencoders and Variational Autoencoders (VAEs); Denoising and Sparse Autoencoders; Applications in Data Generation

Module 5: Word Embedding’s, GPT, Applications in Text Classification, Translation, and Summarization</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>Goodfellow I, Bengio Y and Courville A, Deep Learning, 1st Edition, MIT Press (2017).
Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems By Aurélien Géron Publisher: O'Reilly Media ISBN-13: 978-1492032649 ISBN-10: 1492032646
Computer and Machine Vision – Theory, Algorithms and Practicalities, E. R. Davies, Elsevier (Academic Press), 4th edition, 2013.</td></tr>
</table>