Cummins College of Engineering for Women

(An autonomous institute affiliated to Savitribai Phule Pune University)

Karve Nagar, Pune - 411 052.

Vision

To be globally renowned engineering institute for imparting holistic education and developing professional women leaders in engineering and technology

Structure and Syllabus

of

T. Y. BTech
(Information Technology)

2023 Pattern [R0]

List of Abbreviations

Abbreviation	Title
PC	Programme Core Course
BSC	Basic Science Course
ESC	Engineering Science Course
PE	Programme Elective Course
OE	Open Elective
VSEC	Vocational and Skill Enhancement Course
CC	Co-curricular Courses / Liberal Learning Course
IKS	Indian Knowledge System
VEC	Value Education Course
RM	Research Methodology
INTR	Internship
PROJ	Project
CEP	Community Engagement Project
Mm	Multidisciplinary Minor
AEC	Ability Enhancement course

MKSSS's Cummins College of Engineering for Women, Pune

(An Empowered Autonomous Institute, affiliated to Savitribai Phule Pune University)

Curriculum for UG Degree Course in BTech. Information Technology

(Academic Year: 2025-26 Onwards)

Third Year | Semester-V

Sr. No.	Course Code	Course Title	Teaching Scheme Hours /Week			Credits	Examination Scheme			Total Marks
			L	T	P		ISE	ESE	Pr/Or	
1	23PCIT501	Design and Analysis of Algorithms	3	0	0	3	50	50	0	100
2	23PCIT502	Theory of Computation	3	1	0	4	50	50	0	100
3	23PCIT503	Machine Learning	3	0	0	3	50	50	0	100
4	23PCIT504	Human Computer Interaction	2	0	0	2	25	25	0	50
5	23PEIT501	Programme Elective-I	3	0	0	3	50	50	0	100
6	23MmIT501	Multidisciplinary Minor Course 2	3	0	0	3	50	50	0	100
7	23PCIT501L	Design and Analysis of Algorithms Laboratory	0	0	2	1	25	0	25	50
8	23PCIT503L	Machine Learning Laboratory	0	0	2	1	25	0	25	50
9	23PEIT501L	Programme Elective-I Laboratory	0	0	2	1	25	0	25	50
10	23MmIT501L	Multidisciplinary Minor Course 2 Laboratory	0	0	2	1	50	0	0	50
		Total =	17	01	08	22	400	275	75	750

L=Lecture, T=Tutorial, P= Practical, Cr= Credits, ISE =In Semester Evaluation, ESE =End Semester Examination, Pr/Or = Practical/Oral

23PEIT501: Program Elective-I

A. Artificial Intelligence
B. Multimedia Techniques
C. Distributed Systems

23PEIT501L: Program Elective-I Laboratory

A. Artificial Intelligence Laboratory
B. Multimedia Techniques Laboratory
C. Distributed Systems Laboratory

23MmIT501 Multidisciplinary Minor Course 2

A. Green IT Infrastructure and Sustainability Metrics
B. Space Data Analytics

23MmIT501L Multidisciplinary Minor Course 2 Laboratory

A. Green IT Infrastructure and Sustainability Metrics Laboratory
B. Space Data Analytics Laboratory

APPROVED BY
Secretary Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

APPROVED BY
Chairman Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

23PCIT501 Design and Analysis of Algorithms

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Data structures

Course Objectives:
Familiarize students with
1. Algorithmic approaches for problem solving
2. Basics of computational complexity analysis
3. Various algorithm design strategies.
4. Different classes and solutions to problems such as P, NP etc.

Course Outcomes:
Students should be able to
1. Apply algorithmic techniques to solve computational problems.
2. Analyze the time and space complexity of algorithms for performance evaluation.
3. Solve optimization problems using appropriate algorithmic strategies.
4. Explain computational complexity classes of a given problem and various trends in algorithms

Unit I Introduction
Analysis of Algorithm, Efficiency- Analysis framework, asymptotic notations. Significance of asymptotic bounds in algorithm analysis. Examples of growth functions (e.g., constant, logarithmic, linear, quadratic). Trade-offs between time and space.
Overview of algorithmic strategies: Divide and Conquer, Greedy, Dynamic Programming, Backtracking, and Branch and Bound (high-level introduction).
Introduction to recurrences in algorithm analysis. Solving recurrences (Substitution, Recursion Tree, Master Theorem methods)

Unit II Divide and conquer method and Greedy strategy
Concept and general structure of divide-and-conquer algorithms. Advantages and limitations of the paradigm. Binary Search, Merge Sort, Quick Sort, Matrix Multiplication: Strassen's algorithm.
Concept of greedy choice and optimal substructure. Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Knapsack Problem (Fractional), Huffman Encoding, Job Sequencing

Unit III Dynamic Programming
Concept of overlapping subproblems and optimal substructure. Knapsack Problem (0/1), Matrix Chain Multiplication, Subset Sum Problem, Bellman-Ford algorithm, Floyd-Warshall algorithm, Optimal Binary Search Tree, Travelling Salesman Problem.

Unit IV Backtracking

Concept of backtracking as a method for solving constraint satisfaction problems. Pruning: Identifying and avoiding invalid or non-promising paths. N-Queens Problem, Subset Sum Problem, Graph coloring, Hamiltonian Cycle

Unit V Branch and bound

Concept and working of the branch and bound technique. Components of B&B: State space tree, bounding function, and branching strategies. LIFOBB, FIFOBB & LCBB. Concepts of upper bound and lower bound for pruning the search space. Travelling Salesman Problem**,** Knapsack Problem (0/1)

Unit VI Trends and classes in Algorithms

Computational complexity classes (P, NP, NP-Hard and NP-Complete). Satisfiability Problem (SAT). Introduction to Approximation algorithms, Heuristics, Randomized algorithms, Parallel algorithms (matrix multiplication, sorting).

Textbooks :

1. S. Sridhar, "Design and Analysis of Algorithms", 2nd ed., Oxford University Press, 2023.
2. E. Horowitz, S. Sahni, and S. Rajasekaran, "Computer Algorithms", 2nd ed., Silicon Press, 2019.

Reference books :

1. S. Sahni, "Data Structures, Algorithms, and Applications in C++", 2nd ed., Silicon Press, 2016.
2. G. Brassard and P. Bratle, "Fundamentals of Algorithms", Pearson, 2016.
3. P. S. Pacheco, M. Malensek, "An Introduction to Parallel Programming", 2nd ed., Morgan Kaufmann, 2021.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", 4th ed., MIT Press, 2010.
5. R. K. Shukla, "Analysis and Design of Algorithms: A Beginner's Approach", 1st ed., Wiley India, 2015.
6. B. Schmidt, J. González-Domínguez, C. Hundt, and M. Schlarb, "Parallel Programming Concepts and Practice", 1st ed., Springer, 2020.

23PCIT502 Theory of Computation

Teaching Scheme:
Lectures: 3 hours/week
Tutorial: 1 hour / week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 4

Prerequisites: Discrete Mathematics

Course Objectives:
Familiarize students with
1. Abstract computing models.
2. Types and applications of formal grammar.
3. Construction of turing machines
4. Application of Theory of Computation in System Programming

Course Outcomes:
Students should be able to
1. Construct abstract computing models for engineering problems
2. Apply the concepts of formal grammar for formal language representation
3. Construct Push-down Automata, Post Machines for engineering problems
4. Solve problems using Turing Machine

Unit I Finite Automata

Finite state machine, finite automaton model (FA), deterministic finite automaton (DFA) and non-deterministic finite automaton (NFA), NFA to DFA conversion, NFA with ϵ transitions - Equivalence between NFA with and without ϵ transitions,NFA with ϵ transitions to DFA conversion, Finite Automata with output- Moore and Mealy machines.

Unit II Regular Expression

Identity rules, constructing RE for FA, constructing FA for a RE, Pumping lemma of regular sets, closure properties of regular sets. Closure properties of RLs, lexical analyzer as an application, Introduction to Lex tool

Unit III Regular Grammar

Derivation trees, right most and leftmost derivation of strings, Chomsky hierarchy right linear and left linear regular grammars, equivalence between regular grammar and FA, inter conversion, parsing techniques, top-down parsing, bottom-up parsing, recursive descent parser as an application of regular grammar, Introduction to YACC tool

Unit IV Context Free Grammar

Ambiguity in context free grammar, minimization of context free grammar, Chomsky normal form, Greibach normal form, normalization of CFG

Unit V Push-down automata and Post machine

Push down automata- definition, model, acceptance of context free language, acceptance by final state and acceptance by empty state. Equivalence of context free language and PDA, inter conversion
Post Machine- Definition and construction of Post machine

Unit VI Turing Machine

Definition and construction of turing machines, computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of turing machines, universal turing machine, decidability/undecidability of problems, halting problem, correspondence problem, turing reducibility, modularized programming concept as an application of turing machines

Textbooks

1. Daniel I.A. Cohen, "Introduction to Computer Theory" Wiley-India, ISBN: 978-81-265-1334-5
2. Vivek Kulkarni, "Theory of Computation", Oxford University Press, ISBN-13: 978-0-19-808458-7.

Reference Books

1. John C. Martin, "Introduction to language and theory of computation", Tata McGraw Hill, Third edition,ISBN 0-07-049939-X
2. Hopcroft Ulman, "Introduction To Automata Theory, Languages And Computations", Pearson Education Asia, 2nd Edition
3. E V Krishnamurthy, "Introduction to Theory of Computer Science", EWP Second 2nd Edition.
4. John J Donovan, "Systems Programming", Tata McGraw-Hill Edition 1991, ISBN 0-07-460482-1
5. D.M. Dhamdhere, "Systems Programming and Operating Systems", Tata McGraw-Hill, ISBN-13:978-0-07-463579-7

23PCIT503 Machine Learning

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Linear Algebra and Univariate Calculus; Probability

Course Objectives:
Familiarize students with:
1. Concept of dataset
2. Applications of Machine Learning
3. Supervised and unsupervised Machine Learning algorithms
4. Evaluation metrics in Machine Learning

Course Outcomes:
Students will be able to:
1. Choose appropriate Machine Learning techniques for solving real-world problems.
2. Explain Machine Learning algorithms and techniques.
3. Apply Machine Learning algorithms to solve problems.
4. Evaluate various Machine Learning models.

Unit I Introduction to Machine Learning
Introduction: Concept of Machine Learning, examples of Machine Learning applications, training versus testing, positive and negative class, cross validation
Types of learning: Supervised, unsupervised and semi-supervised learning, incremental learning, reinforcement learning
Dataset preparation, dimensionality reduction

Unit II Linear Models: Classification
Binary and multi-class classification, evaluating classification models using contingency table / confusion matrix - accuracy, f1 score, ROC, AUC
Perceptron: Neurons, learning rate, threshold
Support Vector Machine: Hard margin, soft margin, kernel trick for non-linear data

Unit III Linear Models: Regression
Univariate Regression: Concept, evaluating regression models, constructing line of regression, assessment parameters - sum of squared error, mean squared error and root mean squared error.
Multivariate Regression: Concept
Polynomial curve fitting: Test-train curves, degree of polynomial
Theory of generalization: Overfitting and underfitting, Bias-variance dilemma, regularization

Unit IV Distance based Models and Rule based Models

Distance based models: Concept, Euclidean, Manhattan distance measures.

Distance based algorithms: Nearest neighbor algorithm, k-means, hierarchical clustering, silhouettes.

Rule based Models: Frequent item sets, confidence and support, association rule mining.

Unit V Tree based Models and Probabilistic Models

Tree based Models: Building decision tree using Entropy, ID3, random forest,

Probabilistic Models: Bayes' theorem, independence assumption, naïve Bayes classification algorithm

Unit VI Artificial Neural Network

Limitations of perceptron, single layer perceptron, activation layers, artificial neural network and multilayer neural network, error in output, backpropagation

Textbooks:

1. Ethem Alpaydin, "Introduction to Machine Learning", PHI 4th Edition – 2020.
2. Peter Flach, "Machine Learning: The Art and Science of Algorithms that Make Sense of Data", Cambridge University Press – 2015.

Reference Books:

1. Charu C. Aggarwal, "Neural Networks and Deep Learning", Springer 2nd Edition - 2023.
2. Kevin Murphy, "Machine Learning – A Probabilistic Perspective", MIT Press – 2022.
3. Nikhil Buduma, "Fundamentals of Deep Learning – Designing Next Generation Machine Intelligence Algorithms", O'Reily – 1st Edition – 2017.
4. C. M. Bishop, "Pattern Recognition and Machine Learning", Springer 1st Edition – 2013.
5. Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 3rd Edition – 2011.

Other Resources

1. UCI Machine Learning Repository https://archive.ics.uci.edu
2. WEKA Collection of datasets https://waikato.github.io/weka-wiki/datasets/
3. Kaggle datasets https://www.kaggle.com/datasets

23PCIT504 Human Computer Interaction

Teaching Scheme:
Lectures: 2 hours/week

Examination Scheme:
In-Semester: 25 marks
End-Semester: 25 marks
Credits: 2

Course Objectives:
Familiarize students with
1. Concepts of human-computer-interaction
2. Concept of user centric approach
3. Applications of human-computer-interaction to real life use cases
4. Design of effective human-computer-interactions

Course Outcomes:
Students should be able to
1. Apply interaction design principles for effective user interaction
2. Identify user requirements to propose suitable solutions
3. Use prototyping and design methods for user interface development
4. Identify usability issues in interface design

Unit I Introduction to HCI
Importance of HCI, Goals of HCI and interdisciplinary nature, the psychology of everyday things, principles of user-centered HCI design and conceptual models, usability, case studies of good and bad interfaces.

Unit II Users and Interaction
Human capabilities: Perception, cognition, memory, and action, thinking: reasoning and problem solving, human emotions and psychology, individual differences, models of interaction, ergonomics, interaction styles, WIMP interface, interactivity, context of interaction, paradigms of interactions.

Unit III Models, Design Rules, Guidelines and Evaluation Techniques
Cognitive models: GOMS model, hierarchical task analysis model, principles that support usability, design standards, design guidelines, golden rules, goals of evaluation, evaluation criteria, evaluation through expert analysis, design heuristics, heuristics evaluation through user participation.

Unit IV Design process and applications
Interaction design process, user focus, scenarios, navigation design, screen design and layout, prototyping techniques, wire-framing, model-view-controller framework, website designing, designing for mobiles, conversational interfaces (Voice, Chatbots), Brain-Computer Interfaces, case studies and real-world applications like smartphone UI/UX, AR/VR interfaces

Textbooks

1. Helen Sharp, Yvonne Rogers, and Jenny Preece,"Interaction Design: Beyond Human-Computer Interaction", 6th Edition, Wiley publication, 2023
2. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human-Computer Interaction" 5th Edition, Pearson, 2023

Reference Books

1. Don Norman, "The Design of Everyday Things", MIT Press, 2023
2. Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, "About Face: The Essentials of Interaction Design", 5th Edition, Wiley, 2023
3. Ben Shneiderman, "Human-Centered AI", Oxford University Press, 2022

23PEIT501A Artificial Intelligence

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Discrete Mathematics and Data Structures.

Course Objectives:
Familiarize students with
1. The fundamental concepts, history, and goals of AI and explore the role of intelligent agents in AI systems
2. Knowledge of search strategies and problem-solving techniques
3. Skills required in representing and reasoning with knowledge
4. Advanced AI techniques

Course Outcomes:
Students will be able to:
1. Explain the fundamental concepts of AI.
2. Apply problem-solving and searching techniques for various applications.
3. Use knowledge representation methods and reasoning techniques to solve problems.
4. Explain applications and ethical considerations of AI.

Unit I: Introduction to Artificial Intelligence
Overview of AI, applications and goals of AI, intelligent agents: types of agents, environments, and agent architectures, introduction to AI techniques: symbolic AI, evolutionary algorithms

Unit II: Problem Solving by Uninformed Search
Problem Solving as State-Space Search, Uninformed Search Strategies: breadth-first search, depth-first search, uniform cost search, bidirectional search, comparing uninformed search strategies

Unit III: Problem Solving by Informed Search
Heuristic Search Strategies: greedy search, A* algorithm, adversarial search: minimax algorithm, alpha-beta pruning, constraint satisfaction problems : introduction, backtracking, and local search

Unit IV: Knowledge Representation
Knowledge Representation: propositional and first-order logic, inference in logic: resolution, unification, ontologies and semantic web basics, applications of knowledge representation: knowledge representation in real world AI systems

Unit V: Planning and Reasoning under Uncertainty
Introduction to Planning, problem solving by planning: STRIPS, partial order planning.
Reasoning under uncertainty: Bayesian Networks, probabilistic reasoning, applications of planning and reasoning in AI systems

Unit VI: AI Applications and Ethical Considerations

AI in Practice: case studies in Natural Language Processing, Robotics, and Decision Support Systems, Ethical and societal issues in AI: privacy, bias, transparency, accountability
Future Directions in AI: Generative AI (GenAI), hybrid approaches

Textbooks

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2021.
2. Elaine Rich and Kevin Knight, Artificial Intelligence, 3rd Edition, McGraw Hill, 2017.

Reference Books

1. David L. Poole and Alan K. Mackworth, Artificial Intelligence: Foundations of Computational Agents, 2nd Edition, Cambridge University Press, 2017.
2. Peter Flach, Artificial Intelligence: A New Synthesis, 2nd Edition, Morgan Kaufmann, 2019.
3. Kevin Warwick, Artificial Intelligence: The Basics, 2nd Edition, Routledge, 2019.

23PEIT501B Multimedia Techniques

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester:50 marks
End-Semester: 50 marks
Credits: 3

Course Objectives:
Familiarize students with
1. Foundational knowledge about multimedia elements, standards, and system architecture
2. Appropriate tools and techniques for processing multimedia data
3. Advanced multimedia concepts like AR, VR
4. Multimedia security challenges

Course Outcomes:
Students will be able to:
1. Apply core concepts of multimedia to integrate various multimedia elements.
2. Apply processing techniques to enhance various multimedia elements.
3. Explain the principles of animation and 3D modeling techniques in multimedia production.
4. Explain emerging multimedia trends.

Unit I: Introduction to Multimedia

Overview of multimedia: definitions, elements (text, images, audio, video, animation), applications of multimedia in various fields, multimedia system architecture, multimedia software tools fundamentals of data types, standards, and formats

Unit II: Text and Audio Processing

Text Processing: representation, fonts, styles, and encoding, audio processing: basics of sound, audio formats (WAV, MP3, AAC), and digitization, audio processing techniques: sampling, quantization, filtering, and compression, speech synthesis, and recognition

Unit III: Image Processing

Fundamentals of pixel-based image, representation and image formats, image processing techniques: enhancement, manipulation, and transformation, basics of image compression, and image file standards

Unit IV: Video Processing

Video formats, standards, frame rate, and resolution concepts video digitization, sampling, and compression, basics of video editing, processing, and streaming

Unit V: Animation Techniques

Types of animation: 2D, 3D, and stop motion, principles of animation, tools for creating and manipulating animations, basics of character rigging, motion capture, and keyframing, 3D modeling: introduction to 3D graphics, popular tools, and techniques for creating and rendering 3D content.

Unit VI: Emerging Trends in Multimedia Technology

Augmented Reality (AR) and Virtual Reality (VR), future trends in multimedia, content protection techniques, watermarking and secure distribution methods.

Textbooks

1. Ranjan Parekh, "Principles of Multimedia", 2nd Edition, McGraw Hill Education, 2018.
2. Tay Vaughan, "Multimedia: Making It Work", 9th Edition, McGraw Hill Education, 2018.

Reference Books

1. Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, "Fundamentals of Multimedia", 2nd Edition, Springer, 2021.
2. P. Godse, Multimedia Technologies, 1st Edition, Technical Publications, 2020.
3. K. Andleigh, K. Thakkar, "Multimedia Systems and Design", Reprint Edition, Pearson Education India, 2019.

23PEIT501C Distributed Systems

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Computer Networks, Operating Systems, Database Management Systems

Course Objectives:
Familiarize students with
1. Fundamental knowledge of distributed systems architectures and models.
2. Process Communication and synchronization in a distributed environment.
3. Methods of fault tolerance and replication for distributed systems.
4. Distributed File Systems and naming services.

Course Outcomes:
Students should be able to
1. Apply basic concepts of distributed systems for communication.
2. Apply various synchronization and mutual exclusion algorithms.
3. Explain techniques for fault tolerance, resource and process management.
4. Explain concepts of distributed file system and naming services for distributed environments.

Unit I Introduction to Distributed Systems
Characterization of distributed systems: Issues, goals, and types of distributed systems, distributed system models, hardware concepts, software concepts, middleware: models of middleware, services offered by middleware, client server model.

Unit II Communication
Layered protocols, inter process communication (IPC): MPI, remote procedure call (RPC), remote object invocation, remote method invocation (RMI), message-oriented communication, stream oriented communication, group communication.

Unit III Synchronization
Clock synchronization, logical clocks, election algorithms, mutual exclusion, distributed mutual exclusion - classification of mutual exclusion algorithms, requirements of mutual exclusion algorithms, and performance measure.
Election algorithms- non-token-based algorithm, token-based algorithm.

Unit IV Resource and Process Management
Desirable features of global scheduling algorithm, task assignment approach, load balancing approach, load sharing approach, introduction to process management, process migration, threads, virtualization, clients, servers, code migration.

Unit V Replication and Fault Tolerance
Introduction to replication and consistency, data-centric and client- centric consistency models, replica management, fault tolerance: introduction, process resilience, reliable client-server and group communication, recovery, distributed commit, checkpoints

Unit VI Distributed File Systems and Name Services

Introduction and features of DFS, file models, file accessing models, file-caching schemes, file replication, introduction to name services and domain name system, directory services.

Textbooks

1. Andrew S. Tanenbaum and Maarten Van Steen, "Distributed Systems: Principles and Paradigms", 2nd edition, Pearson Education, 2015.
2. George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems: Concepts and Design", 5th Edition, Pearson Education, 2012.

Reference books

1. S. Tanenbaum and M. V. Steen, "Distributed Systems: Principles and Paradigms", Second Edition, Prentice Hall, 2006.
2. M. L. Liu, "Distributed Computing Principles and Applications", Pearson Addison Wesley, 2004.
3. Sunita Mahajan, Seema Shah, "Distributed Computing", Oxford University Press, 2nd Edition, ISBN-13: 978-0198093480, 2013.
4. Abhijit Belapurkar, Anirban Chakrabarti, Harigopal Ponnapalli, Niranjan Varadarajan, Srinivas Padmanabhuni, Srikanth Sunder rajan, "Distributed System Security: Issues, Processes and solutions", Willey online Library, ISBN: 978-0-470-51988-2, 2009.
5. Robert Love, "Linux System Programming", 2nd Edition, O'Reilly, 2013.

23MmIT501A Green IT Infrastructure and Sustainability Metrics

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Essentials of Green Computing

Course Objectives:

Familiarize students with

1. Energy characteristics of storage systems and methods to enhance their efficiency.
2. Green networking principles and analyze energy-efficient protocols and standards.
3. Sustainability assessment tools and IT maturity models
4. Holistic thinking for eco-conscious cloud architecture and systems

Course Outcomes:

Students should be able to

1. Explain the energy-saving methods for IT storage technologies.
2. Analyze networking components, protocols, and standards for optimizing energy efficiency.
3. Evaluate IT systems using sustainability metrics and LCA tools.
4. Assess the impact of green IT regulations, standards, and frameworks.

Unit I Green Data Storage

Outline 17 SDG, Storage energy profiles (HDD, SSD, tape), Power-saving techniques: caching, transitioning, RAID, SAN and NAS, Virtualization and cloud-based storage optimization, data management for green storage

Unit II Green Networks and Communications

Green networking goals and components,Protocols: IEEE 802.3az, EMAN, energy-aware routing, Case studies in green network redesign

Unit III Sustainable Information Systems and Metrics

Sustainability hierarchies: product level, organisation level, city levels, Life Cycle Assessment (LCA) tools: SimaPro, openLCA Metrics: PUE, DCiE, carbon calculators

Unit IV Regulatory Frameworks for Green IT

Laws: RoHS, REACh, WEEE, GHG mandates, Standards: ENERGY STAR, Blue Angel, ISO 50001, NGO and industry roles (Green Grid, Greenpeace)

Unit V Green Cloud Computing and Architecture

PaaS, IaaS, SaaS from energy use perspective, Virtualization, energy resource management, dynamic provisioning, carbon metering, Green brokers and carbon directories

Unit VI Sustainable IT Planning and Maturity Models

Capability maturity models (SICT, G-Readiness), Strategy development for sustainability Dashboards for IT energy monitoring and compliance

Textbooks

1. San Murugesan and G.R. Gangadharan "Harnessing Green IT: Principles and Practices " ,published by CRC Press
2. Bud E. Smith, "Green Computing: Tools and Techniques for Saving Energy, Money, and Resources", CRC Press

Reference Books

1. Mohammad S. Obaidat, Alagan Anpalagan and Isaac Woungang, "Handbook of Green Information and Communication Systems",Elsevier, 2013
2. Rafeal Mechlore, "Green Computing: Sustainability in Software" Independently Published, 2023

Other References

1. Online Documentation for
    - SimaPro: https://simapro.com
    - openLCA: https://www.openlca.org
2. Government and NGO websites
    - EU WEEE Directive
    - Greenpeace Reports

23MmIT501B Space Data Analytics

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 Marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Engineering Physics, Engineering Mathematics, Principles of Space Technology
Course Objectives:
Familiarize students with
1. Fundamentals of digital image processing
2. Photogrammetry and GNSS
3. Data Processing and Dissemination techniques
4. Applications of Remote sensing (RS) and Geographical Information System (GIS)

Course Outcomes:
Students will be able to:
1. Apply fundamentals of digital image processing for space applications
2. Explain Photogrammetry and GNSS
3. Use data Processing and artificial intelligence techniques for space applications
4. Discuss applications of RS and GIS

Unit I Fundamentals of Digital Image Processing

Remote Sensing Images: Histogram, Image Statistics, Image Display, Colour cube, Look-up-Table, Colour Composites, FCC generation. Image Correction, Image Enhancement, Spatial Enhancement, Image Transformations, Image Classification, Advanced Classification Technique

Unit II Fundamentals of Photogrammetry

Concepts of Photogrammetry, Stereo Photogrammetry, Digital Elevation Model (DEM): Digital Terrain Model (DTM), Digital Surface Model (DSM), nDSM, bare earth DEM, Structures of DTM (Contours, Grid, and TIN), DEM interpolation techniques, derivatives and 3D visualization, Ortho-photo.

Unit III GNSS

Basics of Geodesy, Satellite Navigation and Augmentation systems, Principles and components of GNSS, Differential Global Positioning System (DGPS), Errors in observations and corrections; Overview of GPS Aided GEO Augmented Navigation (GAGAN), Indian Regional Navigation Satellite System (IRNSS) (NavIC)

Unit IV Geographical Information System (GIS)

Introduction to GIS, Spatial and Non-spatial data models, Spatial and Non-spatial queries. in DBMS, Spatial Data Analysis, Network Analysis and Spatial Interpolation Techniques, Geo-data visualization and analysis

Unit V Data Processing and Dissemination techniques

Big Geo-data, Artificial Intelligence, Machine Learning, Reinforcement Learning, Deep Learning,CNN, RNN, Cloud Based Platforms, Earth Engine, Data Cube, Analysis Ready Data (ARD), Geo-Portals, Multidimensional data analytics and visualization.

Unit VI Terrestrial Applications of RS and GIS

Water Resources Assessment and Monitoring, Applications of RS and GIS, Space and Planetary Exploration

Text Books

1. Rémi Cresson, "Deep Learning for Remote Sensing Images with Open-Source Software", CRC Press, ISBN: 9781000093612 (2020)
2. Thomas M. Lillesand, Ralph W. Kiefer, J W Chipman, "Remote Sensing and Image Interpretation", 7th Edition, Wiley, ISBN 978-1-118-34328-9 (2015)

Reference Books

1. R C Olsen, "Remote Sensing from Air and Space," SPIE Press, Washington (2007)
2. Robert A Schowengerdt, "Remote Sensing Models and methods for Image Processing, Elsevier (2006)

Web Sources

1. https://www.nrsc.gov.in/Knowledge_EBooks?language_content_entity=en

23PCIT501L Design and Analysis of Algorithms Laboratory

Teaching Scheme:
Practical : 2 hours / week

Examination Scheme:
In-Semester: 25 marks
Practical: 25 marks
Credits:1

Prerequisites: Data Structures

Course Objectives:
Familiarize students with
1. Basics of computational complexities.
2. The space and time requirements of the algorithms.
3. The various algorithmic design techniques.
4. The categorization of the given problem for finding an appropriate solution

Course Outcomes:
Students should be able to
1. Implement algorithmic solutions to solve computational problems using appropriate strategies.
2. Develop program by applying specific algorithmic paradigms.
3. Analyze the time and space complexity of implemented algorithms.
4. Test and validate algorithms with different inputs and edge cases.

A. **Suggested List of Laboratory Assignments**

1. Write a program to implement a Bubble sort algorithm using Brute Force method.
2. Write a program to implement a Merge sort algorithm using the Divide and Conquer method.
3. Write a program to implement a minimum spanning tree algorithm – Prim's using Greedy method.
4. Write a program to implement a Bellman Ford algorithm using Dynamic Programming approach.
5. Write a recursive program to find the solution for N-Queens problem using Backtracking approach
6. Write a program to find the solution for 0/1 Knapsack problem using Branch and Bound approach.
7. Implement a parallel program to compute the sum of elements in an array using OpenMP.

B. Additional List of Laboratory Assignments

1. Write a program to implement a Binary search algorithm using the Divide and Conquer method.
2. Write a program to implement a Quick sort algorithm using the Divide and Conquer method.

3. Write a program to implement a minimum spanning tree algorithm – Kruskal using Greedy method.
4. Write a program to implement a fractional knapsack algorithm using Greedy method.
5. Write a program to implement an Optimal Binary Search Tree algorithm using Dynamic Programming approach.
6. Write a program to find the solution for Traveling salesperson problem using Dynamic Programming approach.
7. Write a program to find the solution for Traveling salesperson problem using Branch and Bound approach.
8. Implement a parallel program to find the maximum element in an array using OpenMP.
9. Implement matrix multiplication using OpenMP parallel for directives.
10. Write a program to compute the sum of an array using MPI with scatter, compute, and reduce operations.
11. Implement matrix-vector multiplication using MPI.
12. Write a program to implement a simple message passing system using MPI between two processes
13. Write a program to add two vectors using CUDA C in Google Colab.
14. Implement matrix multiplication using CUDA C in Google Colab.

Textbooks :

1. Horowitz and Sahani, Fundamentals of computer Algorithms, Galgotia, ISBN 81-7371-612-9, 2008.
2. R. C. T. Lee, SS Tseng, R C Chang, Y T Tsai, Introduction to Design and Analysis of Algorithms, A Strategic approach, Tata McGraw Hill, ISBN-13: 978-1-25-902582-2. ISBN-10: 1-25-902582-9, 2012.
3. Anany Levitin, Introduction to the Design & Analysis of Algorithm, Pearson, ISBN 81-7758-835-4, 2017.

Reference books :

1. Chandra, Rohit; Dagum, Leo; Kohr, David; Maydan, Dror; McDonald, Jeff; and Menon, Ramesh. Parallel Programming in OpenMP, Morgan Kaufmann, ISBN 1-55860-671-8, 2000.
2. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, ISBN 978-0071232654, 2017.
3. Sanders, Jason; Kandrot, Edward. CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley, ISBN 978-0-13-138768-3, 2010.
4. David R. Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang, Hands-On GPU Programming with CUDA C and C++, Morgan Kaufmann, ISBN 978-0128119860, 2021.

23PCIT503L Machine Learning Laboratory

Teaching Scheme: **Examination Scheme:**
Practical: 2 hours/week **In-semester Exam:** 25 marks
Oral: 25 marks
Credit: 1

Course Objectives:
Familiarize students with:
1. Programming of Machine Learning algorithms
2. Libraries for Machine Learning
3. Usage of large datasets
4. Evaluation metrics for Machine Learning techniques

Course Outcomes:
Students will be able to:
1. Implement Machine Learning algorithms.
2. Compare the performance of various Machine Learning algorithms.
3. Apply Machine Learning algorithms to large datasets.
4. Evaluate different Machine Learning models.

Suggested list of assignments:
1. Use WEKA explorer to classify the Iris dataset. Study various options in WEKA.
2. Implement any two classification algorithms to classify test instances. Use Iris (or any other suitable) dataset from Kaggle. Predict the class of a test instance and compare accuracy, error rate, precision, recall and f1 score.
   a. SVM
   b. kNN
   c. Naive Bayes
3. Implement regression for predicting a target value of a test instance. Use Advertisement (or any other suitable) dataset from Kaggle. Calculate sum of squared error, mean squared error and root mean squared error.
4. Cluster the instances using following algorithms and calculate their silhouette values. Use Diabetes (or any other suitable) dataset from Kaggle.
   a. K-means clustering
   b. Hierarchical clustering

Open ended assignment:
Develop an application to find frequent item item-sets. Provide a facility to run the program for various measures (e.g. support, confidence). Use any suitable dataset available online.

Additional list of assignments:

1. Implement regression by calculating the means, co-variance, and variance. Calculate the slope and intercept and derive the equation of line of regression. Use Advertisement dataset.
2. Cluster instances from Diabetes (or any other suitable) dataset by calculating distances. Use Euclidean distance measure.
3. Implement Naive Bayes classification algorithm by calculating probabilities. Use Weather-nominal (or any other suitable) dataset.
4. Find frequent item item-sets from Store (or any other suitable) data by applying generate and prune steps.
5. Implement multivariate regression or predicting a target value for test instances. Use any suitable dataset from Kaggle. Calculate sum of squared error, mean squared error and root mean squared error.

Textbooks:

1. Andrea Muller and Sarah Guido, "Introduction to Machine Learning with Python", O'Reily – 2017.
2. Michael Bowles, "Machine Learning in Python", Wiley – 2018.

Reference Books:

1. Ian H. Witten, Eibe Frank, Mark A Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Elsevier 3rd Edition.

Other Resources:

1. UCI Machine Learning Repository https://archive.ics.uci.edu
2. WEKA Collection of datasets https://waikato.github.io/weka-wiki/datasets/
3. Kaggle datasets https://www.kaggle.com/datasets

23PEIT501LA Artificial Intelligence Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: 2 hours/week	**In-Semester:** 25 marks
	Oral: 25 marks
	Credit: 1

Prerequisites: Discrete Mathematics and Data Structures.

Course Objectives:
Familiarize students with
1. Foundational skills in implementing AI algorithms.
2. Designing and testing logical inference systems.
3. Planning techniques.
4. Probabilistic reasoning methods.

Course Outcomes:
Students will be able to:
1. Implement classical AI algorithms to solve domain specific problems.
2. Apply heuristic, adversarial and constraint satisfaction techniques to give AI-based solutions.
3. Apply logical reasoning and basic game strategies for intelligent decision-making.
4. Compare the performance of various algorithms.

Suggested List of Assignments (At least 5)

1. Implement a Simple Reflex Agent that reacts to basic inputs.
2. Implement BFS and DFS algorithms to find paths in a simple maze or graph. Compare their performance.
3. Develop a uniform-cost search algorithm to solve a grid-based pathfinding problem. Compare the cost of the path found with BFS and DFS results.
4. Implement the A* algorithm to find the shortest path in a grid with obstacles. Define start and goal points. Compare A*'s path with that of uniform-cost search to demonstrate efficiency using heuristics.
5. Develop a simple game (such as Tic-Tac-Toe) using the Minimax algorithm. Add alpha-beta pruning to improve efficiency, and observe performance gains in decision-making.
6. Implement a program to solve constraint satisfaction problem using any searching technique

Additional list of Assignments:

1. Create a set of logical statements representing a knowledge base and implement forward or backward chaining for inference.
2. Implement a simple STRIPS-based planner for a problem such as a robot moving objects between locations. Define initial states, goal states, and actions with preconditions and effects, to explore automated planning.
3. Use a Bayesian network library/tool to create a Bayesian network for a problem.

Textbooks

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2021.
2. Kevin Knight and Elaine Rich, Artificial Intelligence, 3rd Edition, McGraw Hill, 2017.

Reference Books

1. Kevin Warwick, Artificial Intelligence: The Basics, 2nd Edition, Routledge, 2019.
2. Nils J. Nilsson, Artificial Intelligence: A New Synthesis, 2nd Edition, Morgan Kaufmann, 2019.
3. Wolfgang Ertel, Introduction to Artificial Intelligence, 2nd Edition, Springer, 2020.

23PEIT501LB Multimedia Techniques Laboratory

Teaching Scheme:
Practical: 2 hours/week

Examination Scheme:
In-Semester: 25 marks
Oral: 25 marks
Credit: 1

Course Objectives:
Familiarize students with
1. Multimedia Processing Techniques
2. Practical Skills in Multimedia Editing and Enhancing
3. Basic animations and 3D models,
4. Basic multimedia security techniques

Course Outcomes:
Students will be able to:
1. Apply multimedia processing techniques to create and edit multimedia components
2. Apply image and video enhancement and manipulation skills
3. Use multimedia tools to design and develop 2D animation.
4. Implement basic multimedia security techniques.

Suggested List of Assignments: (At Least 5)
1. Use text processing software to apply different fonts, styles, and encoding.
2. Use audio editing software to perform basic audio processing, such as trimming, adjusting volume, applying filters, and exporting in different audio formats.
3. Apply basic image filters (e.g., blur, sharpen, color balance) and observe the effect on image quality and appearance.
4. Use video editing software to cut, join, and apply transitions to video clips. Export the edited video in multiple formats (e.g., MP4, AVI) and compare file sizes and quality.
5. Create a short 2D animation.
6. Use image editing software to add a visible text or logo watermark to an image.

Additional list of Assignments:

1. Use a VR application or tool (e.g., Unity with a VR add-on) to create a simple VR environment
2. Use an AR app or software (e.g., Unity with Vuforia) to create a simple AR scene.
3. Create a simple 3D model.
4. Compress an image using JPEG and PNG formats with varying compression levels.Analyze the visual quality and file size of each version and discuss how compression affects quality.
5. Record or edit a short video clip and export it in multiple frame rates and resolutions. Compare the visual differences and file sizes of each version to understand the relationship between quality and file size.

Textbooks

1. Ranjan Parekh, Principles of Multimedia, 3rd Edition, McGraw Hill Education, 2021.
2. Tay Vaughan, Multimedia: Making It Work, 9th Edition, McGraw Hill Education, 2018.

Reference Books

1. Prabat K. Andleigh and Kiran Thakrar, Multimedia Systems and Content-Based Image Retrieval, 1st Edition, PHI Learning, 2020.
2. Mark W. Purvis, Digital Multimedia, 3rd Edition, Routledge, 2019.
3. Isaac V. Kerlow, The Art of 3D Computer Animation and Effects, 4th Edition, Wiley India, 2017.

23PEIT501LC Distributed Systems Laboratory

Teaching Scheme:
Practical: 2 hours/week

Examination Scheme:
In-Semester: 25 marks
Oral: 25 marks
Credit: 1

Prerequisites: Computer Networks, Operating Systems, Database Management Systems

Course Objectives:
Familiarize students with
1. Design and Implementation methodology for distributed systems applications.
2. Applications of middleware technologies in distributed systems.
3. Methods of communication in a distributed environment.
4. Algorithms for synchronization and mutual exclusion.

Course Outcomes:
Students should be able to
1. Implement middleware technologies that support distributed applications
2. Execute various communication protocols in distributed environment
3. Implement algorithms for distributed mutual exclusion and synchronization
4. Develop interoperable communication system using distributed object paradigm

Suggested list of assignments:
1. Establish client server communication using Socket Programming (TCP and UDP) for any distributed application.
2. Implement a middleware technology for any distributed application using Remote Procedure Call (RPC).
3. Implement a middleware technology for any distributed application using Remote Method invocation (RMI).
4. Implement any one token-based election algorithm and evaluate the same.
5. Implement any one non-token-based election algorithm and evaluate the same.
6. Develop any distributed application using Message queuing system in Publish-Subscribe paradigm.

Additional list of assignments:
1. Implement using RMI and JDBC (My SQL/ACCESS) Student Database with following operations: a. Insert, b. Display, c. Delete, d. Modify, e. Result.
2. Develop an interoperable communication system using distributed object concepts.
3. Implement Message Passing Interface (MPI) for any distributed application.
4. Implement any quorum-based leader election algorithm and evaluate the same.
5. Implement any one coordinator selection algorithm (bully/ring).

Textbooks

1. Andrew S. Tanenbaum and Maarten Van Steen, "Distributed Systems: Principles and Paradigms", 2nd edition, Pearson Education, 2015.
2. George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems: Concepts and Design", 5th Edition, Pearson Education, 2012.

Reference books

1. S. Tanenbaum and M. V. Steen, "Distributed Systems: Principles and Paradigms", Second Edition, Prentice Hall, 2006.
2. M. L. Liu, "Distributed Computing Principles and Applications", Pearson Addison Wesley, 2004.
3. Sunita Mahajan, Seema Shah, "Distributed Computing", Oxford University Press, 2nd Edition, ISBN-13: 978-0198093480, 2013.
4. Abhijit Belapurkar, Anirban Chakrabarti, Harigopal Ponnapalli, Niranjan Varadarajan, Srinivas Padmanabhuni, Srikanth Sunder rajan, "Distributed System Security: Issues, Processes and solutions", Willey online Library, ISBN: 978-0-470-51988-2, 2009.
5. Robert Love, "Linux System Programming", 2nd Edition, O'Reilly, 2013.

Other References:

1. MIT 6.824: Distributed Systems (YouTube + Course Site): https://www.youtube.com/@6.824
2. RMI: https://www.youtube.com/watch?v=NmGytKDhCx4
3. RPC: https://www.youtube.com/watch?v=gr7oaiUsxSU
4. MPI: https://www.youtube.com/watch?v=tm8M5H1OZmw

23MmIT501LA Green IT Infrastructure and Sustainability Metrics Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: 2 hours/week	**In-Semester:** 50 marks
	Credit: 1

Prerequisites: Essentials of Green Computing

Course Objectives:

Familiarize students with

1. To implement and simulate energy-efficient IT systems in storage and networking.
2. To apply LCA tools for evaluating IT devices' environmental impact.
3. To design dashboards and estimate sustainability metrics like PUE and emissions.
4. Applying theoretical concepts through case-based evaluations

Course Outcomes:

Students should be able to

1. Evaluate energy performance of RAID and cloud storage through simulation
2. Analyze network topologies using energy-aware protocols through simulation
3. Understand the impact of life cycle assessments of IT devices using tools
4. Evaluate key metrics for IT infrastructure

Suggested list of assignments

1. Configure RAID levels with OpenMediaVault and analyze energy usage
2. Simulate IEEE 802.3az using Packet Tracer or GNS3
3. Perform simplified LCA using openLCA on IT devices
4. Design an Excel/Sheets-based dashboard with PUE, CO_2 estimates
5. Case Study – Analyze Google/Facebook data center and present energy optimization strategies
6. Estimate cloud deployment energy profile using AWS/GCP console (free tier)

Textbooks

1. San Murugesan and G.R. Gangadharan "Harnessing Green IT: Principles and Practices " ,published by CRC Press
2. Bud E. Smith, "Green Computing: Tools and Techniques for Saving Energy, Money, and Resources", CRC Press

Other Resources

1. Cisco NetAcad Platform
2. GNS3 documentation (https://docs.gns3.com/)
3. openLCA: https://www.openlca.org
4. OpenMediaVault: https://www.openmediavault.org
5. Cisco Packet Tracer: Free via NetAcad
6. AWS & GCP Free Tiers: Access via student accounts or sandbox environments

23MmIT501LB Space Data Analytics Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: 2 hours/week	**In-Semester:** 50 Marks
	Credit: 1

Prerequisites: Engineering Physics, Engineering Mathematics, Principles of Space Technology

Course Objectives:

Familiarize students with

1. Digital Image Processing techniques
2. Methods of Photogrammetry and GNSS
3. Data Processing and Dissemination techniques
4. Space applications using space data analytics

Course Outcomes:

Students will be able to:

1. Apply digital image processing techniques for space applications
2. Use tools for Photogrammetry and GNSS
3. Implement data processing and dissemination techniques for space applications
4. Develop prototype for space technology application in team

Suggested List of Assignments:

1. Process satellite/remote sensed images using digital image processing techniques.
2. Use any simulation tool for Photogrammetry and GNSS
3. Perform data analytics on space data by building appropriate AI model
4. Choose any Space technology application. Define the complete problem statement with user requirements and expected outcomes. Develop a prototype for the chosen space application

Note: Students can form groups of maximum 4 to develop the application.

Textbooks

1. Rémi Cresson, "Deep Learning for Remote Sensing Images with Open-Source Software", CRC Press, ISBN: 9781000093612 (2020)
2. Thomas M. Lillesand, Ralph W. Kiefer, J W Chipman, "Remote Sensing and Image Interpretation", 7th Edition, Wiley, ISBN 978-1-118-34328-9 (2015)

Reference Books

1. R C Olsen, "Remote Sensing from Air and Space," SPIE Press, Washington (2007)
2. Robert A Schowengerdt, "Remote Sensing Models and methods for Image Processing, Elsevier (2006)

Web Sources

1. https://www.nrsc.gov.in/Knowledge_EBooks?language_content_entity=en