Structure of Post Graduate

(ME Computer Science and Engineering)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
PATIALA, PUNJAB, INDIA

COURSE SCHEME & SYLLABUS

(2025)

M.E. (COMPUTER SCIENCE & ENGINEERING)

ME- COMPUTER SCIENCE AND ENGINEERING (2025)

SEMESTER I						
S. NO.	**CODE**	**TITLE**	**L**	**T**	**P**	**Cr**
1	PCS108	Advanced Data Structures	3	0	2	4
2	PCS206	Machine Learning	3	0	2	4
3	PCS111	Software Project Management	3	0	2	4
4	PCS113	Linear Algebra and Applied Statistical Methods	3	0	2	4
5	PCS114	Data Science	3	0	2	4
6	PHU004	Research Methodology, Ethics and IPR	2	0	0	2
		TOTAL	**17**	**0**	**10**	**22**
SEMESTER II						
1	PCS220	Multimedia Processing	3	0	2	4
2	PCS222	Deep Learning	3	0	2	4
3	PCS109	Advanced Algorithms	3	0	2	4
4	PCS223	Advanced Computer Networks	3	0	2	4
5		ELECTIVE-I	3	0	2	4
6		ELECTIVE-II	3	0	2	4
		TOTAL	**18**	**0**	**12**	**24**
SEMESTER III						
1	PCS397	Dissertation/Internship Interim Report	-	-	-	4
2	PCS391	Seminar	-	-	-	4
		TOTAL				**8**
SEMESTER IV						
1	PCS392	Project Semester/Dissertation	-	-	-	16
		TOTAL	**-**	**-**	**-**	**16**
		GRAND TOTAL - FOUR SEMESTER CREDITS				**70**

ELECTIVE I						
1	PCS212	Secure Coding	3	0	2	4
2	PCS217	Continuous Delivery and DevOps	3	0	2	4
3	PCS221	Cloud Infrastructure and Services	3	0	2	4
4	PCS224	Natural Language Processing	3	0	2	4
5	PCS225	Quantum Computing	3	0	2	4
ELECTIVE II						
1	PCS210	Block Chain Technologies	3	0	2	4
2	PCS226	Source Code Management	3	0	2	4
3	PCS252	GPU Computing	3	0	2	4
4	PCS227	Generative AI	3	0	2	4
5	PCS228	Quantum Machine Learning	3	0	2	4

PCS108: ADVANCED DATA STRUCTURES

L	T	P	Cr
3	0	2	4.0

Course Objective: To learn the advanced concepts of data structure and their implementation. The course has the main ingredients required for a computer science graduate and has all the necessary topics for assessment of data structures.

Introduction: Overview of various linear and non-linear data structures.

Complexity Analysis: Introduction to asymptotic complexity, Complexity of recursive algorithms, Amortized complexity, Complexity analysis of various sorting and searching techniques, Sorting in linear time.

Tree Structures: AVL Trees, Red-Black Trees, Splay Trees, B-trees, B+ Trees, Fibonacci heaps, Data Structures for Disjoint Sets, Self-Adjusting Data Structures, Temporal data structures, Succinct data structures, Dictionaries and cuckoo hashing.

Data Structures for Graphs and Related Algorithms: Representation, Type of graphs, Paths and circuits, Euler graphs, Hamiltonian paths and circuits, Cut-sets, Connectivity and separability, Planar graphs, Isomorphism, Graph colouring, Covering and partitioning, Depth- and breadth-first traversals, Minimum spanning tree: Prim's and Kruskal's algorithms, Shortest-path Algorithms: Dijkstra's, Bellman–Ford Algorithm and Floyd's algorithm, Topological sort, Max flow: Ford-Fulkerson algorithm, Max flow-min cut. Design the application using trees and graphs to optimize resource allocation, improve efficiency, and enhance sustainability.

Laboratory Work: To implement the data structures and related algorithms given above in a high level programming language.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Implement basic data structures and analyse them to solve fundamental problems.
2. Analyse the algorithms associated with advanced data structures, focusing on advanced tree structures.
3. Implement data structures for graphs and applying related algorithms to solve complex computational problems.
4. Explore the properties of graphs and employ them to model a variety of real-world problems.

Text Books:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, The MIT Press, 4th edition (2022).
2. Y. Langsam, M. J. Augenstein, and A. O. Tenenbaum, Data Structures Using C and C++, Pearson Education India, 2nd edition (2015).

Reference Books:

1. Peter Brass, Advanced Data Structures, Cambridge University Press, 1st edition (2008).
2. J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education India, 1st edition (2013).
3. Horwitz E., Sahni S., Rajasekaran S., Fundamentals of Computers Algorithms, Universities Press, 2nd edition, (2008).

PCS206: MACHINE LEARNING

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: This course provides an advanced understanding of machine learning techniques, focusing on their application to real-world problems. It incorporates the principles of ethical AI, sustainable development, and the potential societal impact of machine learning technologies. The course includes both theoretical foundations and practical applications, emphasizing the role of machine learning in achieving global sustainability goals.

Introduction to Machine Learning and SDGs: Overview of machine learning (ML) types: supervised, unsupervised, and reinforcement learning, Sustainable AI practices and ethics, SDGs and their relevance to ML applications Applications in healthcare, education, and environmental sustainability,

Ethical AI and Fairness in Machine Learning: Reducing bias in ML models, Fairness, transparency, and inclusivity in AI applications, Ethical implications of AI in decision-making

Supervised Learning: Regression and Classification, Linear regression, logistic regression, and decision trees, Applications of supervised learning in real-world problems: healthcare, finance, and education, Ensuring fairness and minimizing bias in ML algorithms

Unsupervised Learning: Clustering and Association**,** K-means, hierarchical clustering, and principal component analysis (PCA), Applications of clustering in smart cities, sustainable development, and resource optimization, Bias detection in unsupervised learning

Machine Learning in Healthcare and Education: Predictive models for disease outbreaks, personalized healthcare, and educational advancements, AI-powered educational platforms and adaptive learning systems, Addressing inequalities in healthcare and education through ML

Machine Learning for Sustainability: Using ML for climate change prediction, energy optimization, and waste management, Role of machine learning in responsible consumption and production, Sustainable solutions for industries and infrastructure through ML

Advanced Topics and Future Trends: Deep learning, neural networks, and reinforcement learning, The role of AI in Industry 4.0 and Smart Cities, Future trends in AI for achieving SDGs.

Laboratory Work: Hands-on implementation of machine learning algorithms, Projects related to healthcare, education, climate change, social impact assessment and sustainability, Case studies of successful ML applications for SDGs.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Explain basic principles, techniques, and applications of machine learning, with a focus on sustainable and ethical AI practices.
2. Apply regression, classification, and clustering techniques to real-world problems, ensuring ethical considerations and minimizing biases in model predictions.

3. Build machine learning models that address global challenges, including healthcare disparities, educational inequality, and environmental sustainability.
4. Evaluate and compare different machine learning models and algorithms for their ability to promote fairness, inclusivity, and transparency, adhering to ethical guidelines.
5. Design and implement machine learning solutions that contribute to the achievement of the Sustainable Development Goals, focusing on sectors such as healthcare, education, and environmental sustainability.

Text Books:

1. Mitchell T.M., Machine Learning, McGraw Hill, 2nd edition, (1997).
2. Alpaydin E., Introduction to Machine Learning, MIT Press, 2nd edition, (2010).

Reference Books:

1. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag, 2nd edition, (2006).
2. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press, 1st edition, (2009).

PCS111: SOFTWARE PROJECT MANAGEMENT

L	T	P	Cr
3	0	2	4.0

Course Objectives: The objective of this course is to provide a foundational understanding of software project management and project planning and to apply these methodologies effectively in real-world scenarios.

Project Planning: Characteristics of a software project, Software scope and feasibility, resources, the SPM plan, Project Management Knowledge Areas.

Software Project Estimation: Size/scope estimation: Decomposition techniques, WBS, LOC, Function point (FP), FP vs LOC, Schedule estimation: GANTT Charts, Activity networks, PERT, CPM networks. Effort/ Cost estimation Models

Agile Project Management and Sustainability: Fundamentals of Agile Methodology, Agile Principles, Extreme Programming (XP), User Stories, Applying Agile in the Organization, Agile project management frameworks like Scrum and Kanban**.** Minimizing Waste (Lean Principles in Software), Incremental & Efficient Development, Sustainable Work Practices: Shorter Development Cycles – Less waiting time, faster value delivery, Continuous Monitoring & Optimization, Kanban boards to visualize and reduce work-in-progress waste, Implement Test-Driven Development (TDD) to avoid defects and reduce rework.

Measurement and Tracking Planning: Earned Value Analysis.

Quality Planning: Quality control, Quality assurance, Formal Technical Reviews, The SQA Plan, ISO and CMM standards.

Risk Management: Reactive vs proactive Risk strategies, Risk projection, Risk Refinement, Risk Monitoring, Monitoring and management, RMMM plan.

Team Management: Team structures: hierarchical, egoless, chief programmer, mixed, generalist team, specialist team, and hybrid team; Team software Process; Resource leveling, Building team skill set.

Configuration Management: Baselines, Configurable items, SCM repository, SCM process, version control, configuration audit, Git based configuration management.

Project Monitoring and Control: Audits and Reviews.

Laboratory work:
Implementation of software project management concepts using tools like MS Project, Rational Suite (RequisitePro, Purify, etc.), Advanced Cost Estimation Models.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):

After completion of this course, the students will be able to:

1. Apply basic software project planning, scope, feasibility concepts and project estimation techniques.
2. Demonstrate agile project management using UI sketching, user stories, and product backlog.
3. Understand the concept of team structure and team management.
4. Analyze the implementation of quality assurance, quality control, and risk management.

Text Books:

1. Schwalbe K., Introduction to Project Management, Cengage Learning 8th edition, (2015).
2. Jalote P., Software Project Management in Practice, Pearson 1st edition, (2016).

Reference Books:

1. Hughes B., Cotterell M., and Mall R., Software Project Management, Tata McGraw Hill 6th edition, (2017).
2. Stellman A., and Greene J., Applied Software Project Management, O'Reilly Media Inc. 1st edition, (2005).

PCS113: LINEAR ALGEBRA AND APPLIED STATISTICAL METHODS

L	T	P	Cr
3	0	2	4.0

Course Objectives: The course aims to introduce to the students, fundamental principles as well as advanced topics in statistics and sampling techniques. This course underscores the importance of statistical methods to perform scientific and engineering research

Linear Algebra: Introduction to matrices, rank of matrix, System of linear equations, Eigenvalues and Eigenvectors, diagonalization, vector spaces, subspace, span, linear independence/dependence, basis, dimension and linear transformation.

Basic Probability and Statistical Principles: Event, probability, axioms of probability, conditional probability, Bayes' rule, independence, random variables, probability mass function, probability density function, expectation, functions of random variables, normal distribution, central limit theorem.

Hypothesis tests: Introduction to sampling distributions (standard Normal, chi-square, F and t distributions) and their properties, introduction to hypothesis tests (difference between one tailed and two tailed tests), level of significance of test and power of test, one way ANOVA, two-way ANOVA with examples for sustainability studies such as applications in climate change research and energy efficiency.

Time Series Analysis: Least square method, Autoregressive models, moving average models, autoregressive moving average models and their applications for climate forecasting and energy consumption prediction.

Laboratory work:
Each laboratory experiment will consist of numerical exercises on one of the above topics. Laboratory experiments will be performed using Matlab/SPSS/Python.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Apply linear algebraic methods to perform computational task.
2. Compute probabilities of events with an understanding of random variables and distributions.
3. Design an experiment using hypothesis testing and ANOVA to derive statistical inferences.
4. Analyze time series data with different time series models.

Text Books:

1. Strang, G., Introduction to Linear Algebra, Wellesley-Cambridge Press, 6th Edition (2023)
2. Walpole R. E., Myers R. H., Myers S. L. and, Keying Y., Probability and Statistics for Engineers and Scientists, Pearson Education, 9th edition, (2021).
3. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M., Time Series Analysis: Forecasting and Control, Wiley, 5th edition, (2015).

Reference Books:

1. Lay, D. C., Lay, S. R., & McDonald, J. J., Linear Algebra and Its Applications, Pearson, 10th edition, (2016).
2. Hogg, R. V., Tanis, E. A., & Zimmerman, D. L., Probability and Statistical Inference, Pearson, 10th edition, (2018).

PCS114: DATA SCIENCE

L	T	P	Cr
3	0	2	4.0

Course Objective: The course on Data Science aims to deliver in-depth knowledge of various data science techniques.

Introduction to data science: Overview of Data Science, Types of data- Record, Temporal, Spatial-Temporal, Graph, Unstructured, semi and structured data, Raw and Processed data, Data Analysis- Data collection, integration, cleansing, extraction, modeling, prediction, visualization, privacy and security, Data Wrangling, Business Analytics, Business Intelligence, Applications of data science and data manipulations, Preparation of data for machine learning models.

Basics of Python programming for data science: Data Types, Data Structures- List, Tuple, Dictionaries, Flow Control, Functions, Overview of Python libraries in context to data science- Scientific computing with Python (SciPy), Numpy arrays, Keras, Tensorflow, Matplotlib, Pandas, Use of Tableau for plotting various charts for data visualization.

Association Rule Mining: Introduction, Association rules, Support and confidence, Itemsets, Frequent itemsets, Closed and maximal Itemsets, Association rule mining methods- Naïve, Apriori, Direct Hashing & Pruning (DHP), Dynamic Itemset Counting, FP- Growth, Performance evaluation of algorithms.

Data Analysis and Visualisation: Exploratory Data Analysis, Data Analysis Pipeline: Data pre-processing- Attribute values, Attribute transformation, Sampling, Dimensionality Reduction, Eigenfaces, Multidimensional Scaling, Non-linear Methods, Graph-based Semi-supervised Learning, Representation Learning Feature subset selection, Distance and Similarity calculation. Visualization- Traditional Visualization, Multivariate Data Visualization, Text Data Visualization, Network Data Visualization, Temporal Data Visualization, and visualization Case Studies, designing dashboards using Tableau.

Big Data Analytics: Introduction to Big Data, Elements of Big Data, Big Data Architecture, Big Data Warehouses, Big Data Analytics, Functional vs. Procedural Programming Models for Big Data, Hadoop Echo System, Data Ingestion, MapReduce & Sqoop, Basics of Impala and Hive, Apache Flume and HBase, Apache Pig, Basics of Apache Spark, Spark SQL, Graph databases (Neo4J), NoSQL Databases (MongoDB).

Course Learning Outcomes (CLO):

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

1. Apply the fundamental knowledge of various key concepts of data science and business analytics.
2. Demonstrate the use of various Python libraries to solve real-world problems.
3. Implement various association rule mining techniques for market-based analysis.
4. Design and analyze various visualization plots using data analytics and visualization techniques.
5. Apply the concepts of big data analysis to extract knowledge.

Text Books

1. Grus J., Data Science from Scratch: First Principles with Python, O'Reilly Media (2019) 2nd ed.
2. VanderPlas J., Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media (2016) 1st ed.

Reference Books

1. Thangaraj M., Big Data Analytics: Concepts, Techniques, Tools and Technologies, PHI Learning (2022) 1st ed.
2. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons Inc (2015) 1st ed.

PHU004: RESEARCH METHODOLOGY, ETHICS AND IPR

L	T	P	Cr
2	0	0	2.0

Course Objectives: The course aims to equip the students to analyse research related information. And also it sensitize the students to ethical research practices. To equip them to write technical reports and research paper and the process of patent filing. Also to create awareness about the consequences of IPR Infringement

Unit 1: Meaning of Research Problem, Sources of Research Problem, Criteria and Characteristics of good Research Problem, Errors in selecting a research Problem, scope and objectives of research problem.

Unit 2: Effective Literature studies, approaches and analysis.

Unit 3: Effective Technical Writing, How to write report and Research paper; developing a research proposal.

Unit 4: Non Parametric Tests: When to use a Nonparametric Tests**;** Mann Whitney U Test; Sign Test; Wilcoxon Signed Rank Test and Kruskal-Wallis Test.

Unit 5: Ethics: Need for Ethics in Professional Life; Kohlberg's Theory of Moral Development and Its Applicability to Engineers. Professional Ethics: Values in Work Life; Professional Ethics and Ethos; Codes of Conduct. Research Ethics, Plagiarism, Case Studies on Ethics.

Unit 6: Introduction to IPR: Nature of Intellectual Property Rights: Patents; Designs; Trademarks; Copyright; Trade Secrets; Industrial Design; Geographical Indicators; Integrated Circuits. International Character of IPRs, Role of IPRs in Economic Development.

Patents: Introduction to Patents, Inventions not Patentable, Procedure for grant of Patents, Rights and Obligations of a Patentee; IPR Infringement.

Case studies on IPRs.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Analyse research related Information.
2. Indulge in ethical research practices
3. Equipped to write technical reports and research paper.
4. Equipped with the process of patent filing
5. Possess awareness about consequences of IPR Infringement

Text Books:

1. Geoffrey R. Marczyk. Essentials of Research Design and Methodology, Wiley, (2008).
2. Wayne Goddard, Stuart Melville. Research methodology: An Introduction, Juta, (2004).

3. Thomas, C. George. Research Methodology & Scientific Writing, Ane Books Pvt. Ltd, (2016).
4. Menell, Peter S, Lemley, Mark A, Merges, Robert P. Intellectual Property in the New Technological Age, Vol. I Aspen Law & Business, (2019).
5. Menell, Peter S, Lemley, Mark A, Merges, Robert P. Intellectual Property in the New Technological Age, Vol. II Aspen Law & Business, (2019).
6. Narayanan, P., Intellectual Property Law, Eastern Law House, (2008).

PCS220: MULTIMEDIA PROCESSING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To impart knowledge and practical skills in multimedia processing, focusing on fundamentals, enhancement, restoration, feature extraction, and energy-efficient algorithms for real-world applications.

Digital Image Fundamentals: Image perception - light, luminance, brightness, and contrast; Examples of fields that use digital image processing; Image formation model, image sampling and quantization, basic relationships between pixels, Types of Images: Binary, Grayscale, Color; Color representation: Color models; Fundamentals of Pseudo-color and Full-color image processing.

Image Enhancement and Restoration in Spatial and Frequency (Transform) domain: Introduction to spatial and frequency (transform) domain, Transformation Techniques from spatial to frequency domain (and reverse): DFT (Discrete Fourier Transform), DCT (Discrete Cosine Transform), DWT (Discrete Wavelets Transform), etc.; Image enhancement: Efficient algorithms for point processing in real-time, low-power systems, Neighbourhood processing, Histogram processing; Image Smoothing and Sharpening Lightweight smoothing/sharpening filters for resource-constrained devices; Lowpass and High pass filtering; Image restoration: degradation model; inverse filtering.

Feature Extraction: Introduction to type of features: Region, Edge, Boundary, Texture, Color; Feature Extraction Techniques: Canny, Optimizing SIFT for resource-constrained environments, etc. Introduction to Neural Network and Deep Learning (Convolutional Neural Networks (CNNs)), Overview on use cases of CNNs in Image Processing, Study of prominent CNN architectures: AlexNet, ResNet, EfficientNet, etc.

Energy-efficient algorithms for application areas of Image Processing: Study of image processing techniques implemented in various application areas (not limited to): Image compression, Image Segmentation, Image security: Watermarking, Steganography, Visual Cryptography, Object Detection and Classification, etc.

Video and Audio Processing: Analog video, Digital video, Time varying image formation model, Geometric image formation, sampling of video signal, 2D Motion Estimation: Optical flow, Pixel based motion estimation, Region based Motion estimation, Multi resolution motion estimation, Application of motion estimation in video coding. Speech/Audio Processing: Time domain analysis, Frequency domain analysis, Cepstral analysis, LPC analysis, Speech coding, Speech recognition, Speech enhancement, Text to speech conversion. Signal Processing Models of Audio Perception, Psycho-acoustic analysis, Spatial Audio Perception and rendering, Audio compression methods.

Laboratory work: Demonstrate the use of Multimedia Processing to create interactive multimedia processing applications.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Visualize the need and applications of image processing.
2. Apply various well known transformations and filters to enhance the visual quality of given grayscale/color image.
3. Comprehend the role of feature extraction in Image Processing.
4. Demonstrate the use of image processing techniques to ideate innovative solutions to real-world problems.
5. Understand the fundamentals of Video and Audio processing.

Text Books:

1. Gonzalez C. R., Woods E. R., Digital Image Processing, Pearson Education, 4th edition, (2017).
2. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1st edition, (2015).
3. Yao Wang, Jorn Ostermann and Ya Qin Zhang, Video Processing and Communications, Prentice Hall, 1st edition, (2002).
4. L.R. Rabiner and S. W. Schafer, Digital Processing of Speech Signals, Pearson Education, (2008).

Reference Books:

1. McAndrew A., Introduction to Digital Image Processing with Matlab, Thomson Course Technology, 1st edition, (2004).
2. Ben Gold & Nelson Morgan, Speech & Audio Signal Processing, 1st Edition, Wiley, (2011).

PCS222: DEEP LEARNING

L	T	P	Cr
3	0	2	4.0

Course objective: The main objective of this course is to enabling the student with basic deep learning architectures to build an intellectual machine for making decisions behalf of humans.

Artificial Neural Networks: Basic Concepts of Artificial Neurons, Single and Multi-Layer Perceptron, Learning Algorithm, Gradient Decent & Momentum Based Optimization, Activation Functions, Backpropagation.

Convolutional Neural Networks: Basic Concepts of Convolutional Neural Networks. Convolution and Pooling Operation, Convnet Architectures, Regularization, Dropout, Batch-Norm etc. Convnet Architectures - Alexnet, Zfnet, VGG, Googlenet, Resnet, Mobilenet etc. Real life applications in Healthcare Analytics & Agriculture Analytics

Recurrent Neural Networks: Recurrent Architecture, BPTT, Vanishing and Exploding Gradients, GRU, LSTM, Attention Mechanism and Transformers. Real Life Application such as Future Climate Prediction, Financial Risk Analysis

Autoencoders: Autoencoder and its Relation to PCA, Stack Autoencoders, Denoising Autoencoders Variational Autoencoders, Sparse Autoencoders and GANs. Synthetic data generation

Laboratory Work: To implement deep learning models using python and google open source library such as Tensorflow, Keras etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course the student will be able to:
1. Investigate and identify the advancements in learning techniques.
2. Implement various deep learning models. .
3. Demonstrate the applications deep learning in various fields.
4. Design and develop innovative solutions to real-life problems using deep learning open-source libraries

Text Books:

1. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, (2016).
2. Charu C, Aggarwal. Neural Networks and Deep Learning, Springer. 1st edition, (2018).

Reference Books:

1. S. Haykin, Neural Networks and Learning Machines, Pearson College Div, 3rd edition, (2008).
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learnin, Amazon Science, (2021).

PCS109: ADVANCED ALGORITHMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: The course aims to equip students with techniques for effective problem-solving in computing. It explores advanced algorithm design principles, fundamental algorithm analysis, and sophisticated data structures.

Fundamental Algorithm Paradigm: Divide-and-Conquer, Binary Search, Merge Sort.

Advanced Design and Analysis Techniques: Dynamic Programming, Greedy Approach, Sustainable resource allocation using greedy algorithms, Branch-and-Bound, Backtracking, Amortized Analysis, Illustrations of these techniques for Problem-Solving: Bin Packing, Knap Sack, Travelling Salesman Problem.

Numerical algorithms: Integer, Matrix and Polynomial Multiplication, FFT, Extended Euclid's algorithm, Modular Exponentiation, Primality Testing, Cryptographic Computations.

Geometric algorithms: Range searching, Convex Hulls, Segment Intersections, Closest Pairs

String Matching Algorithms: Suffix arrays, Suffix trees, tries (minimizing memory usage and optimizing computational efficiency), Rabin-Karp, Knuth Morris-Pratt, Boyer Moore algorithm (reducing energy consumption).

Approximation algorithms: Need of approximation algorithms: Introduction to P, NP, NP-Hard and NP-Complete; Deterministic, Non-Deterministic Polynomial time algorithms; Knapsack, TSP, Set Cover, Open Problems.

Randomized algorithms: Introduction, Type of Randomized Algorithms, Quick Sort, Min-Cut, 2-SAT; Game Theoretic Techniques, Random Walks.

Online Algorithms: Introduction, Online Paging Problem, Adversary Models, K-server Problem.

Laboratory Work: Design and implement algorithms for problems given above in a high level programming language.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After the successful completion of the course the students will be able to:

1. Explore the most sustainable algorithmic solution by enhancing efficiency, reducing energy consumption, and optimizing resource utilization.
2. Evaluate different algorithms, justify the selection of a specific one, and implement it within a given context.
3. Use advanced techniques to model a variety of real-world problems to solve and analyze them.
4. Implement advanced algorithmic techniques, such as String Matching Algorithms and Approximation Algorithms, while assessing their suitability for different contexts.

Text Books:

1. Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C., Introduction to Algorithms, MIT Press, 3rd edition, (2009).
2. Horwitz E., Sahni S., Rajasekaran S., Fundamentals of Computers Algorithms, Universities Press, 2nd edition, (2008).

Reference Books:

1. J. Kleinberg and E. Tardos, Algorithm Design, Pearson, 1st edition, (2013).
2. V. Aho, J. E. Hopcroft. and J. D. Ullman, The Design and Analysis of Algorithms, Addison-Wesley, (1974).
3. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press. 1st edition, (2004).
4. Vijay Vazirani, Approximation Algorithms, Springer. 1st edition, (2001).

PCS223: Advanced Computer Networks

L	T	P	Cr
3	0	2	4.0

Course Objectives: To give the students an understanding of the principles behind the latest advances in computer networks, including wireless media access, adhoc routing, congestion control and security.

Wireless Standards: IEEE 802.11a/b/n/g/p, 802.15, and 802.16 standards for Wireless PAN, LAN, and MAN

IPv6: Header, Addressing, Neighbour Discovery, Auto-Configuration, Header Extensions and options, support for QoS, security, etc., DHCPv6

Mobile IPv6: rationale and operation - intra and inter site

IP Multicasting: Multicast routing protocols, Virtual private network service, multiprotocol label switching. Peer-to-Peer network architectures, Overlay networks, flat routing protocols (DHTs), and peer-to-peer architectures.

Distributed Routing: RIP, OSPF and BGP Routing;

Mobile Adhoc Networks: Advanced Routing Protocols and Improvements in AODV, DSR, DSDV etc., Hot Potato(deflection) Routing Schemes and cut-through routing

Transport Layer Advances: TCP Improvements and Extensions, Performance issues, Client-server communication

QUIC: UDP-Based Multiplexed and Secure Transport, Using TLS to Secure QUIC, Handshake of QUIC compared to TCP with TLS 1.2, QUIC Loss Detection and Congestion Control, Relationship between HTTP, TLS and QUIC.

Multipath TCP (MPTCP): TCP Extensions for Multipath Operation with Multiple Addresses,

TCP Congestion Control: fairness, scheduling and Delay modeling, QoS issues, differentiated services, ECN, Transport layer in Wireless Networks.

Advanced Networking Architectures: Software Defined Networking, Opportunistic Networks, Cognitive Radio Networks, Wireless Sensor Networks.

Network Security: Network Security principles, Security related issues in wireless networks, Public and Private Key Cryptography, Key distribution protocols.

Laboratory:

Laboratory experiments will be performed using tools like Network Simulators NS3/NS2, Omnet++, mininet, NetSim, Packet Tracer, GNS3 etc.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):

After completion of this course, the students will be able to:

1. Analyse and understand the advancements in wireless networks.
2. Apply the different protocols associated with advanced computer networks.
3. Implement different adhoc routing protocols to solve real world problems.
4. Explore and analyse different network vulnerabilities and their security.

Text Books:

1. G. R. Wright and W. R. Stevens. TCP/IP Illustrated, Volume 2: The Implementation, Addison Wesley, 2nd edition (1995).
2. Theodore Rappaport, Wireless Communications: Principles and Practice, Pearson, 2nd Edition (2010).
3. W. Stallings. Cryptography and Network Security: Principles and Practice, Prentice Hall, 2nd Edition (1998).

References:

1. Hesham Soliman, Mobilie IPv6: Mobility in a Wireless Internet, Pearson Education, 1st Edition (2004).
2. Jie Wu and Yunsheng Wang, Opportunistic Mobile Social Networks, CRC Press, 1st Edition (2015).
3. Respective Internet Drafts and RFCs of IETF.

Elective Courses

PCS212: SECURE CODING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course aims to provide an understanding of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities. It gives an outline of the techniques for developing a secure application.

Introduction to Security: Core Concepts: Security, CIA Triad (Confidentiality, Integrity, Availability), Malicious Software: Viruses, Trojans, Worms, Rootkits, Trapdoors, Botnets, and Key Loggers, Attack Types: Active and Passive Attacks, Exploits, Threats, Vulnerabilities, and Risks, Tools: Honeypots and their Applications in Security.

Secure Software Development Lifecycle (SSDLC): Proactive Security: Secure Requirements, Design, Development, Testing, and Maintenance. Secure Coding Best Practices: Principles of Secure by Design, Default, and Deployment (SD3). Security Principles: Defense in Depth, Principle of Least Privilege, Secure Product Development Timeline, Case studies on secure coding practices for environmental monitoring systems.

Threat Modeling and Risk Mitigation: Threat Identification: Using Attack Trees and Rating Threats (DREAD Framework), Mitigation Techniques: Security Best Practices for Application and Infrastructure, Authentication and Authorization: Models, Techniques, and Best Practices, Threat modelling in smart healthcare and education systems.

Software Security: Software Vulnerabilities: PE Code Injection, Return-to-libc Attack, Format String Vulnerability, Race Condition Vulnerabilities and Dirty COW Exploit.

Web Security: Web Vulnerabilities, Cross-Site Request Forgery (CSRF): Attack Mechanisms and Prevention, Cross-Site Scripting (XSS): Self-Propagating XSS Worms and Mitigations, SQL Injection Attacks: Exploits and Countermeasures, Client-Side Attacks and Emerging Threats.

Laboratory work:

Hands-on exercises include:

1. Identifying and fixing common coding vulnerabilities.
2. Developing secure application code and implementing protection mechanisms.
3. Simulating software and web attacks (e.g., SQL Injection, XSS) and designing countermeasures.
4. Projects and assignments focused on secure coding practices.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):

After completion of this course, the students will be able to:

1. Demonstrate skills needed to deal with common programming errors and develop secure applications.
2. Implement PE Code injection and demonstrate control hijacking via EIP manipulation
3. Demonstrate client-side attacks and identify the nature of threats to software and incorporate secure coding practices throughout the planning and development of software products.
4. Demonstrate SQL injection, XSS attack and suggest countermeasures for the same.

Text Books:

1. Howard, M. and LeBlanc, D., Writing Secure Code, Howard, Microsoft Press, 2nd Edition, (2002).
2. Deckard, J., Buffer Overflow Attacks: Detect, Exploit, Syngress, 1st Edition, (2005).
3. Wenliang Du, Computer Security: A hands-on approach, CreateSpace (2017).

Reference Books:

1. Swiderski, F. and Snyder, W., Threat Modeling, Microsoft Professional, 1st edition (2004).
2. Salt, C., J., SQL Injection Attacks and Defence, Elsevier, 2nd edition, (2012).

PCS217: CONTINUOUS DELIVERY AND DEVOPS

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course makes student learn the fundamental principles and practices associated with DevOPs. To apply the principles and practices of DevOps and automation on a project of interest and relevance to the student.

Introduction to DevOps: History of DevOps, DevOps Ecosystem, DevOps Objectives, DevOps Market Trends, Infrastructure As A Code, IaaS Overview, Paas Overview, DevOps on the Cloud, DevOps Production Model, Tool pipelining

DevOps and Automation: Version Control, Continuous Integration, Continuous Testing, Configuration Management, Continuous Deployment, Containerization, Continuous Monitoring, Tool pipelining

Version Control: Introduction to version control, Introduction to Git, importance of Git for an organization, Common commands in Git, Working with Remote Repositories, Branching and Merging in Git, Git workflows, Git cheat sheet.

Continuous Integration: Introduction to Jenkins and its Architecture, Jenkins Management, Build Setup, Git and Jenkins Integration

Continuous Testing: Agile Testing Techniques, Test-Driven Development (TDD), Behaviour Driven Development (BDD), Acceptance Test Driven Development (ATDD) Life Cycle, User Acceptance Test, Definition of Done (DoD), fit test, early testing and traditional testing techniques, Introduction to Selenium,Selenium – Webdriver, X-Path, Creating Test Cases in Selenium WebDriver (Waits), Handling different controls on Webpage, Best practices to support saves time and resources

Containerization: Benefits and use cases for containerized environments, Shipping Transportation Challenges, Introduction to Docker, understanding images and containers, Introduction to Container, Container Life Cycle, Sharing and Copying to promote efficient use of resources, Base Image, Docker File, working with containers, Publishing Image on Docker Hub, Install Docker on a local machine, define a container environment using a Dockerfile, Store and share a Docker deployment, Container Deployment, Container orchestration..

Laboratory work:

Exploring Git Commands through Collaborative Coding, Implement GitHub Operations, Applying CI/CD Principles to Web Development Using Jenkins, Git, and Local HTTP Server, Create the GitHub Account to demonstrate CI/CD pipeline using Cloud Platform like AWS or Microsoft Azure, Demonstrating Infrastructure as Code (IaC) with Terraform

Course Outcomes (COs)/Course Learning Outcomes (CLOs):

After completion of this course, the students will be able to:

1. Identify and explain the need, and basic principles of DevOps and Continuous Delivery pipeline and its automation benefits..
2. Describe and apply the continuous delivery engineering practices and release process. in real-world applications.
3. Identify & use the test-driven deployment and various tools/frameworks used for continuous integration and delivery in DevOps.
4. Demonstrate the ability to implement containerized applications using Docker and utilizing Docker Compose for multi-container orchestration.

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis, The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, IT Revolution Press, (2016), 2nd ed.
2. Sharma S., The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise, Wiley, (2017), 1st ed.

Reference Books:

1. Christopher Cowell, Automating DevOps with GitLab CI/CD Pipelines: Build efficient CI/CD pipelines to verify, secure, and deploy your code using real-life, Packt Publishing (2023), 1st ed.
2. Bernd Öggl, Michael Kofler, Docker: Practical Guide for Developers and DevOps Teams - Unlock the Power of Containerization: Skills for Building, Securing, and Orchestrating with Docker, Rheinwerk Computing, (2023), 1st ed..

PCS221: CLOUD INFRASTRUCTURE AND SERVICES

L	T	P	Cr
3	0	2	4.0

Course Objective: To learn the advanced concepts of cloud infrastructure and services and its implementation for assessment of understanding the course by the students.

Introduction: Cloud Computing, History and evolution, Overview of Types of Computing: Cluster, Grid, Utility and Autonomic Computing, Applications of cloud computing for various industries, economics and benefits of cloud computing. Green IT procurement and lifecycle management.

Cloud Computing Architecture and challenges: Cloud Architecture, Types of Clouds: Public, Private & Hybrid Clouds, Cloud based services: Iaas, PaaS and SaaS. Security, Elasticity, Resource management and scheduling, QoS (Quality of Service) and Resource Allocation, Cost Management, Big Data.

Data Centre and Warehousing: Classic Data Center, Virtualized Data Center (Compute, Storage, Networking and Application), Overview of data center infrastructure and energy consumption. Green data centers: design, cooling, and energy management. Renewable energy in powering data centers. Carbon footprint measurement and reduction in cloud computing.

Cloud Implementations and Environments: Amazon Web Services, The Elastic Compute Cloud (EC2), The Simple Storage Service (S3), The Simple Queuing Services (SQS), Google AppEngine - PaaS, Windows Azure, Aneka, A Comparison of Cloud Computing Platforms.

Virtualization: Virtualization, Server consolidation Advantages and Disadvantages, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, para-virtualization, full-virtualization, Cloning, Snapshot and Template, VMware cloud – IaaS.

Cloud based Data Storage: Introduction to Hadoop and Map Reduce for Simplified data processing on Large clusters, Distributed File system, Data Replication, Shared access to data stores, introduction to Python, Design of data applications based on Map Reduce, Task Partitioning, Data partitioning, Data Synchronization.

Laboratory Work: To implement Cloud, Apache and Hadoop framework and related services. To understand various concepts practically about virtualization, data storage. To implement a few algorithms with the help of MapReduce and some high level language. To create a cloud instance, add storage and set up a web server on the instance, review and deletion of the instance.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Illustrate the existing hosting platforms and computing paradigms currently being used in industry and academia.

2. Describe data centre needs, its virtualization techniques and types of cloud.
3. Apply virtualization in Amazon Web Services, Azure, Aneka etc.
4. Interpret Sustainable Data Centres alongwith their applications for energy efficiency.
5. Explain and Identify Hadoop file system with MapReduce Programming.

Text Books:

1. Raj Kumar Buyya, James Broberg, Andrezei M.Goscinski, Cloud Computing: Principles and Paradigms, Wiley,1st edition, (2013).
2. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, 1st edition, (2008).

Reference Books:

1. Anthony Velte, Toby Velte and Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata McGrawHill, 3rd edition, (2017).
2. Judith Hurwitz, Robin Bllor, Marcia Kaufman, Fern Halper, Cloud Computing for Dummies, 1st edition, (2009).
3. Tom E. White, Hadoop: The Definitive Guide, O'Reilly Media, 4th edition, (2015).

PCS224: NATURAL LANGUAGE PROCESSING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course provides a broad introduction to natural language processing with a focus on deep learning applications. It explores cost-effective approaches to automated knowledge acquisition in the emerging field of natural language understanding using advanced neural models and GPU computing.

Introduction to NLP: NLP Applications, Challenges of NLP, Principles and Traditional Methods: Rule-based Systems, Statistical NLP, Introduction to Linguistics for NLP: Morphology, Syntax, Semantics, and Pragmatics, Role of Machine Learning and Deep Learning in NLP.

Text Preprocessing and Machine Learning for NLP: Text Preprocessing Techniques: Tokenization, Stemming, Lemmatization, Stop-word Removal, Lowercasing. Text Representation Methods: One-hot Encoding, Bag-of-Words (BoW), TF-IDF. Supervised Learning: Naive Bayes, Logistic Regression, and SVM for NLP.

Word Embeddings and Recurrent Models: Feedforward NN, Word2Vec, GloVe, Contextualization (ELMo, etc.), Deep Recurrent Models: RNNs, GRUs, LSTMs.

Transformer-Based Models for NLP: Introduction to Transformers: Self-Attention Mechanism, BERT and its Variants: Architecture, Pre-training, and Fine-tuning, Other Transformer Models: GPT, Megatron, etc. Working with Open Source NLP Datasets and Benchmarks: GLUE, SQuAD, etc.

Applications of NLP: Exploring NLP Problem Statements: Information Retrieval, Intent Slot Filling, Machine Translation, Punctuation & Capitalization Restoration, Question Answering, Relation Extraction, Sentiment Analysis, Token Classification., NLP for Multilingual Processing

Large Language Models (LLMs): Introduction to large-scale pre-trained language models such as GPT, PaLM, LLaMA, and Falcon. Overview of their architecture, capabilities, and applications in generative NLP tasks. Exposure to concepts like prompt engineering, few-shot and zero-shot learning, in-context learning, and ethical considerations in deploying LLMs for real-world use cases.

Laboratory Work: To apply machine learning and deep learning techniques to a wide range of Natural Language Processing tasks using popular frameworks and tools, and to explore algorithms and models for text classification, sentiment analysis, information retrieval, machine translation, and other core NLP applications

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course the student will be able to:

1. Understand core NLP concepts and linguistic foundations, and apply basic text processing and machine learning methods to NLP tasks.

2. Apply word embedding techniques and construct deep learning models such as RNNs, GRUs, and LSTMs for sequence-based NLP tasks.
3. Analyze transformer-based architectures and apply pre-trained models to solve NLP tasks using standard benchmarks and datasets
4. Apply NLP techniques and large language models to solve real-world language understanding and generation tasks across various applications.

Text Books:

1. Smith, D. A., Deep Learning for Natural Language Processing, Packt Publishing, (2021).
2. Kalyan K. S., Sangeetha S., Natural Language Processing: A Textbook with Python Implementation, Springer, (2023).
3. Lewis T., Liu L., van der Plas L., Natural Language Processing with Transformers, Revised Edition, O'Reilly Media, (2023).

Reference Books:

1. Bird S., Klein E., Loper E., Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, O'Reilly Media, (2009).
2. Atkinson-Abutridy J., Large Language Models, CRC Press, (2023).
3. Goldberg, Y, Neural Network Methods for Natural Language Processing, Morgan & Claypool, (2017).

PCS225: QUANTUM COMPUTING

L	T	P	Cr
3	0	2	4.0

Course Objectives: The objective of this course is to provide the students an introduction to quantum computation after covering the concepts of linear algebra, vector space and quantum mechanics.

Mathematics and Quantum Mechanics foundation: Linear and complex vector space, Hilbert spaces (finite dimensional), Tensor Products, Dirac's notation, Probabilities and measurements, Postulates of quantum mechanics, Measurements in bases other than computational basis, Introduction of qubit, Bloch sphere representation of qubit, Quantum Superposition and entanglement, Super-dense coding, teleportation, Density operators, Euler identity.

Quantum Computing: classical gates, Single qubit gates, multiple qubit gates, quantum gates, universal quantum gates, Quantum circuits, design of quantum circuits, Energy efficiency concepts in quantum computing, Recent developments in industry and hardware development, Government Initiatives in quantum computing.

Quantum Algorithms: Deutsch's algorithm, Deutsch Jozsa's algorithm, Grover algorithm, Shor's factoring algorithm, Simon's Algorithm.

Quantum Cryptography: Basic concept of classical and quantum cryptography, BB84, B92 and E91 protocol.

Quantum Error Correction: Basic concept of quantum error correction, bit and phase flip error, Shor's 9 qubit error correction algorithm.

Lab: Implementation of Quantum concepts in any quantum simulator.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Comprehend the basic concepts of quantum computing.
2. Illustrate the concepts of quantum gates and quantum circuits.
3. Apply the concept of quantum computing in designing of quantum algorithms.
4. Acquire basic knowledge of quantum protocols.

Text Books:

1. NielsenM. A., Chuang I. L., Quantum Computation and Quantum Information, Cambridge University Press (2010) 10th ed.
2. Benenti G, Casati G., Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific (2007).

Reference Books:

1. PeresA., Quantum Theory: Concepts and Method, Kluwer Academic Publishers (2002) 1st ed.
2. Yanofsky N. S., Mannucci, M. A., Quantum Computing for Computer Scientists, Cambridge University Press, 2008.

PCS210: BLOCKCHAIN TECHNOLOGIES

L	T	P	Cr
3	0	2	4.0

Course Objective: This course provides a broad introduction to blockchain technology and its applications. It explores the fundamental principles of cryptography, decentralized networks, and consensus mechanisms, along with the hands-on development of smart contracts and enterprise blockchain solutions.

Cryptography: Traditional and Modern Cryptography techniques. Symmetric key cryptography, Asymmetric key cryptography (ECC and RSA), Signatures, hash.

Blockchain definition, shortcomings of current transaction systems, distributed network, difference between blockchain and traditional database, evaluation of blockchain. Core Components of Blockchain Architecture, Bitcoin's block structure, node, Merkle Trees, Shared ledger, Mining, validators.

Consensus and cryptography behind the blockchain: Bitcoin Blockchain transaction flow. Blockchain need, use cases of blockchain, Types of Blockchain Architecture (public, private, consortium). Working of Consensus, Bitcoin – I (The Basics, PoW and Beyond, The Miners), Permissioned Blockchain, proof of stake, delegated proof of stake, round robin, PBFT, POET.

Ethereum: Public consortium blockchain: Introduction of Ethereum, Ethereum account, Ethereum network, Ethereum client, Ethereum gas, Ethereum virtual machine, Ethereum block, header, Ether.

Solidity language: Writing smart contracts: Ethereum development: Preparing smart contract, development tools: remix, geth and mist etc., token standard.

Hyperledger: Private consortium: Hyperledger Burrow, Hyperledger Sawtooth, Hyperledger fabric, Hyperledger indy, Hyperledger iroha. Hyperledger suitability according to project. Tools in Hyperledger: Caliper, composer, explorer.

Evil Sides of Blockchain and Legal Regulations for Blockchain: Ransomware, Money Laundering, Cyber Currencies, Cyber Security Exchanges-The 'Dark' Side of Blockchain, Criminal Use of Payment Blockchains, The Role of Financial Regulations for Blockchain, Does Blockchain Need Legal Regulations, Global Digital Assets Regulatory Trends, **Green Blockchain Initiatives**

Laboratory Work: To design and implement algorithms using smart contracts. Laboratory experiments will be performed using blockchain tools: remix, geth and mist etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Analyze modern and traditional cryptographic techniques and their role in blockchain security.
2. Analyze different Blockchain architectures, including public, private, and consortium models.

3. Illustrate the structure of Bitcoin blocks and the Ethereum network.
4. Develop smart contracts using Ethereum and related development tools.
5. Demonstrate the functionality of Hyperledger Fabric and its suitability for various projects.

Text Books:

1. Bashir I., Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications, Packt Publishing, 3rd edition, (2023).
2. Bashir I., Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications, Packt Publishing, 3rd edition, (2023).
3. Narayanan A., Bonneau J., Felten E., Miller A., and Goldfeder S., Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 1st edition, (2016).

Reference Books:

1. Antonopoulos A. M., Mastering Bitcoin: Programming the Open Blockchain, O'Reilly Media, 2nd edition, (2017).
2. Antonopoulos A. M. and Wood G., Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, 1st edition, (2018).
3. Stallings W., Cryptography and Network Security: Principles and Practice, Pearson Education, 8th edition, (2022).
4. Paar C. and Pelzl J., Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 1st edition, (2010).

PCS226: SOURCE CODE MANAGEMENT

L	T	P	Cr
3	0	2	4.0

Course Objectives: The objective of the course is to teach techniques to combine software development and IT operations using DevOps. It helps to understand faster software development practices with higher quality.

Traditional Software Development: The Advent of Software Engineering, Waterfall method, Developers vs IT Operations conflict.

Rise of Agile methodologies: Agile Vs Waterfall Method, Iterative Agile Software Development, Individual and team interactions over processes and tools, working software over comprehensive documentation, Customer collaboration over contract negotiation, responding to change over following a plan.

Definition and Purpose of DevOps: Introduction to DevOps, DevOps and Agile, Minimum Viable Product, Application Deployment, Continuous Integration, Continuous Delivery.

CAMS (Culture, Automation, Measurement and Sharing): CAMS – Culture, Automation, Measurement, Sharing, Test-Driven Development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness, Organizational Learning.

Typical Toolkit for DevOps: Introduction to continuous integration and deployment, Version control system.

Source Code Management History and Overview: History of Linux and the development of Git by Linus Torvalds. Version Control Systems: Introduction to centralized (SVN) and distributed (Mercurial, Git) version control systems. Examples: Overview and comparison of SVN (Subversion), Mercurial, and Git.

Source Code Management Best Practices: Commit, Importance of updated version, Log entry, Review, Branches.

Version Control System: Version control system vs Distributed version control system: Local repository, Advantages of distributed version control system, The Multiple Repositories Models, completely resetting local environment, Revert - cancelling out changes.

Green DevOps & Sustainable Software Engineering: Writing energy-efficient code, Measuring software energy consumption, Cloud computing sustainability strategies, Green hosting and carbon-aware infrastructure choices..

Laboratory work:
Basic structure and Implementation of various distributed version control systems for source code management.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Investigate and identify the need for migrating from traditional software development to the Agile model and then to DevOps.
2. Design, develop, and implement solutions based on the basic principles of DevOps and Continuous Delivery.
3. Analyze and explore the history and overview of Source Code Management, along with real-world examples.
4. Demonstrate and apply the concepts of centralized and distributed version control systems by performing basic operations using various version control tools.
5. Differentiate between centralized and distributed version control systems and demonstrate the use of various version control tools through basic operations.

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, IT Revolution Press, 2nd edition, (2021).
2. Len Bass, Ingo Weber, and Liming Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley, 1st edition, (2015).

Reference Books:

1. Mike Loukides, What is DevOps?, O'Reilly Media, 1st edition, (2012).
2. Michael Hüttermann, DevOps for Developers, Apress, 1st edition, (2012).
3. Dror G. Feitelson, Software Deployment, Updating, and Patching, MIT Press, 1st edition, (2021).

PCS252: GPU COMPUTING

L	T	P	Cr
3	0	2	4.0

Course Objective: To study architecture and capabilities of modern GPUs and learn programming techniques for the GPU such as CUDA programming model.

Introduction to GPU Computing: Heterogeneous Parallel Computing, Introduction to GPU Architecture. Parallel Programming Models: OpenCL, CUDA, OpenACC, and their Ecosystem.

History and Evolution of GPU Computing : Graphics Pipelines: From Fixed-Function to Programmable Pipelines, General-Purpose GPU Computing (GPGPU): Unified Graphics and Computing Processors, Scalable GPUs, Recent Trends: Tensor Cores, AI and Machine Learning Acceleration, Ray Tracing Hardware.

CUDA Programming Fundamentals: Basics of Data Parallelism and CUDA C: CUDA Program Structure, Kernels, and Threads, Memory Management: Global, Shared, and Constant Memory; Memory Transfer Between Host and Device, Performance Optimization: Tiling, Coalesced Memory Access, and Reducing Divergence.

Parallel Execution Model: CUDA Thread Hierarchies: Blocks, Grids, and Warps, Multidimensional Thread Mapping: Matrix Multiplication and Complex Kernel Design, Synchronization and Resource Allocation: Managing Dependencies, Transparent Scalability, and Latency Tolerance, parallel programming for early disease detection, vaccine development, and drug discovery.

Memory Optimization in CUDA: Memory Types: Shared Memory, Texture Memory, and Registers, Tiled Matrix Multiplication: Leveraging Shared Memory to Maximize Performance, Tools: Profiling and Debugging CUDA Applications Using NVIDIA Tools (e.g., Nsight, Visual Profiler).

OpenCL Programming: Data Parallelism and Device Architecture in OpenCL, Managing Devices and Kernels: Platform Model, Command Queues, and Execution Models, Example Applications: Electrostatic Potential Map, Image Processing Kernels, Utilizing parallel processing for flood prediction, earthquake simulations, and emergency response systems.

OpenACC Programming: OpenACC vs. CUDA: High-Level Abstractions and Ease of Use, Execution and Memory Models: Constructs for Loops, Kernels, and Data Management, Asynchronous Computation and Overlapping Data Transfers.

Laboratory work: Practice programs using CUDA, OpenCL and OpenACC.
Hands-on exercises include:

1. Writing and optimizing programs using CUDA for basic and advanced algorithms (e.g., matrix multiplication, reduction).
2. Developing parallel applications in OpenCL for diverse hardware platforms.
3. Implementing high-level GPU programs using OpenACC.
4. Profiling, debugging, and optimizing GPU programs for real-world applications.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Define the key concepts of GPU computing, including efficiency, scalability, and speedup.
2. Study common GPU architectures and programming models.
3. Design and implement efficient parallel algorithms for data-intensive applications using CUDA, OpenCL, and OpenACC.
4. Examine GPU applications for memory and computational efficiency, and assess performance using profiling tools.
5. Design GPU-accelerated solutions for real-world problems and analyze results.

Text Books:

1. J. Sanders and E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Nvidia, Addison-Wesley, 1st edition, (2010).
2. David B. Kirk, Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, 2nd edition, (2012).

Reference Books:

1. Wen-mei W. Hwu, A GPU Computing Gems Emerald Edition (Applications of GPU Computing Series), Morgan Kaufmann, (2011).
2. Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann (2012).
3. Benedict Gaster et al., Heterogeneous Computing with OpenCL 2.0, Morgan Kaufmann (2015).

PCS227: GENERATIVE AI

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course introduces Generative AI, its capabilities, and applications across domains such as text, image, audio/video, and code generation. It focuses on prompt engineering, foundational models, AI agent frameworks, and ethical implications of generative technologies.

Introduction: Introduction to Generative AI, Capabilities of Generative AI, Exploring Generative AI Applications Across Domains, Leveraging Generative AI tools for Text Generation, Image Generation, Audio and Video Generation, Code Generation.

Foundational Models and Platforms: Introduction to Large Language Models (LLMs), Transformer Networks, Generative AI Models, Foundational Models, Pre-trained Models: Text-to-Text Generation, Text-to-Image, Text-to-Code.

Prompt Engineering: Meaning of Prompt, Best Practices for Prompt Creation, Common Prompt Engineering Tools, Text-to-Text Prompt Techniques, Interview Pattern Approach, Chain-of-Thought Approach, Tree-of-Thought Approach

AI Agents: RAG Framework, RAG, Encoders, and FAISS, Introduction to LangChain, LangChain: Core Concepts, LangChain Documents for Building RAG Applications, LangChain Chains and Agents for Building Applications.

Ethical Issues and Responsible Generative AI: Limitations of Generative AI, Issues and Concerns About Generative AI, Hallucinations of Text and Image Generating LLMs, Code Generating LLMs, Considerations for Responsible Generative AI, Implementing Responsible Generative AI Across Domains, Economic and Social Implications of Generative AI.

Laboratory work:

Labs include hands-on sessions with generative tools (e.g., ChatGPT, DALL·E, LangChain), building AI agents using RAG and LangChain, prompt engineering techniques, and analyzing ethical issues such as hallucinations and bias in AI outputs.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Understand the basic principles, capabilities, and domains of application for generative AI technologies.
2. Apply prompt engineering techniques to effectively interact with large language models.
3. Utilize foundational models and AI tools to build generative applications.
4. Evaluate ethical, social, and technical considerations in the responsible use of Generative AI.

Text Books:

1. Tunstall L., von Werra L., Wolf T., *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*, O'Reilly Media (2022), 1st ed.
2. Foster D., *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*, O'Reilly Media (2019), 1st ed.

3. Kambhatla N., *Generative AI: A Guide for Leaders and Practitioners*, Wiley (2023), 1st ed.

Reference Books:

1. Dignum V., *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*, Springer (2019), 1st ed.

PCS228: QUANTUM MACHINE LEARNING

L	T	P	Cr
3	0	2	4.0

Course Objectives: The objective of this course is to learn the quantum version of the classical machine learning algorithm and apply these in solving real-life problems.

Basic of Quantum Computing: An introductory overview of single and multi-qubit gates, rotation gates, density matrix and mixed states, reversible computing, probabilistic computation, Complexity classes including BPP, RP, CoRP, BQP, ZQP, QMA and PSAPCE.

Quantum Machine Learning Foundations: Quantum encoding strategies (basis, amplitude, angle, ZZ-feature map), variational quantum circuits, Quantum classifier, Quantum support vector machine, Quantum Fourier Transform Algorithm, HLL algorithm, Quantum Principal Component analysis.

Quantum Neural Networks and Hybrid Models: Quantum Convolutional neural networks (QCNNs), Quantum LTSM, Quantum Generative Adversarial Networks(QGANs), Quantum gradient descent, Hybrid quantum-classical workflows.

Quantum Approximate Optimization Algorithm: Quadratic Unconstrained Binary Optimization(QUBO) formulation, Ising model, Knapsack problem using QUBO model, Introduction to QAOA algorithms with maxcut problems, Quantum Annealing.

Frameworks, Cloud Platforms and Hardware: Discussion on cloud platform services like IBM Q Experience, Amazon Bracket, Pennylane. Quantum processors based on various technologies and their challenges, Quantum Roadmaps (IBM, Google, Microsoft, Rigetti and Xanadu).

Lab: Implementation of Quantum machine learning on any available quantum simulator.

Course learning outcomes (CLOs):
After the completion of the course, the student will be able to:

1. Comprehend the basic concepts of quantum encoding.
2. Illustrate the concepts of quantum support vector machine and quantum principal component analysis.
3. Acquire the knowledge of quantum convolutional neural network for problem solving.
4. Apply the concept of QAOA for solving optimization problems.

Text Books:

1. Nielsen M. A., Chuang I. L., Quantum Computation and Quantum Information, Cambridge University Press (2010) 10^{th} ed.
2. Maria Schuld, Francesco Petruccione, Machine Learning with Quantum Computers, Springer, 2021.

Reference Books:

1. PeresA., Quantum Theory: Concepts and Method, Kluwer Academic Publishers (2002) 1^{st} ed.
2. Yanofsky N. S., Mannucci, M. A., Quantum Computing for Computer Scientists, Cambridge University Press, 2008.

PCS397: DISSERTATION/INTERNSHIP INTERIM REPORT

L	T	P	Cr
0	**0**	**0**	**4.0**

Course Objectives: This course is designed to encourage design projects in which students apply the knowledge and skills gained during their Master of Engineering program to explore and develop a specific idea.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Investigate and identity the real-world problems.
2. Design, develop and implement a domain specific design/research problem.
3. Develop acumen for higher education and research.
4. Enhance technical report writing skills.

PCS391: SEMINAR

L	T	P	Cr
0	0	0	4.0

Course Objectives: This course aims to equip students with the skills needed to effectively discuss and present topics within a group. The Seminar course is the outcome of six months of study, research, exploration, and analysis of a particular topic. It is designed to evaluate the student's ability to deliver a well-structured presentation, engage an audience, and demonstrate strong communication skills. The course also fosters the development of lifelong learning as a core competency.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Identify and select a scholarly topic relevant to a specific domain.
2. Conduct a thorough investigation and compile historical and contextual information related to the chosen topic.
3. Analyze and articulate the real-world or domain-specific applications of the topic.
4. Develop skills in technical report writing, presenting information in a clear and structured manner.
5. Demonstrate effective communication skills through well-organized presentations and active audience engagement.

Evaluation Scheme:

- Presenting a topic to an audience in a given time with a professionally prepared content.
- Literature Survey/Content: This includes the depth knowledge of the related work done by others related to Seminar Topic
- Viva (answer to the queries)
- Report Writing

PCS392: DISSERTATION

L	T	P	Cr
0	0	0	16

Course Objectives: This course is designed to help the student obtain research skills which includes a thorough survey of a particular domain, finding a research problem and presenting a methodology to resolve the problem; with adequate experimental results to strengthen the contribution. The students are also given an exposure where they learn to write research papers and present the work in the conferences. Students are also supposed to learn about communicating the impact of their work by different tools which include video, poster and presentation.

Course Outcomes (COs)/Course Learning Outcomes (CLOs):
After completion of this course, the students will be able to:

1. Design and implementation of identified research problems or industrial projects.
2. Develop acumen for higher education and research.
3. Write technical reports and publish the research work in referred journals, national and international conferences of repute.
4. Foresee how their current and future work will influence/impact the economy, society and the environment.

Evaluation Scheme:

- ✓ Subject matter of Presentation
- ✓ Literature Review
- ✓ Discussion of Results and Inferences drawn
- ✓ Presentation Structuring
- ✓ Response to Questions
- ✓ Usefulness/Contribution to the profession
- ✓ Overall Perception
- ✓ Reflective Diary
- ✓ Publication
- ✓ Poster
- ✓ Video Presentation