PG NEP 2 year Course Structure M.Sc. (Computer Science) (2 year 4 Semesters)				
First semester		Credit	Marks	Total
CMS2PCOR01T	Advanced Computer Architecture	4	50	Marks = 300 Credit = 22
CMS2PCOR02T	Advanced Operating System	4	50	
CMS2PCOR03T	Advanced Database Management System	4	50	
CMS2PCOR04T	Advanced Compiler Design	4	50	
CMS2PCOR05P	Advanced Database Management System Laboratory	4	50	
CMS2PAEC01M	Application Development	2	50	
Second Semester				
CMS2PCOR06T	Image Processing and Machine Learning	4	50	Marks = 250 Credit = 20
CMS2PCOR07T	Advanced Data Structures and Algorithms	4	50	
CMS2PCOR08T	Cyber Security and IoT	4	50	
CMS2PCOR09T	Artificial Intelligence and Big Data Analytics	4	50	
CMS2PCOR10P	Image Processing and Machine Learning Laboratory	4	50	
Third semester				
CMS2PCOR11T	Advanced Software Engineering	4	50	Marks = 300 Credit = 22
CMS2PCOR12T	Research Methodology	4	50	
CMS2PCOR13T	Information and Coding Theory	4	50	
CMS2PDSE01T	Elective - 1	4	50	
CMS2PCOR14M	Seminar and Term Paper Leading to Project	4	50	
CMS2PSEC01M	LATEX for Scientific Writing	2	50	
Fourth Semester				
CMS2PDSE02T	Elective - 2	4	50	Marks = 300 Credit = 24
CMS2PDSE03T	Elective - 3	4	50	
CMS2PCOR15M	Grand Viva	4	50	
CMS2PCOR16M	Literature Review	4	50	
CMS2PCOR17M	Project Work	4	50	
CMS2PCOR18M	Project Viva-voce	4	50	

- Elective 1
  1. Mobile Computing
  2. Fuzzy Logic and Fuzzy Systems
  3. Computer Vision and Robotics
  4. Quantum Computing
  5. Pattern Recognition
  6. Soft Computing
  7. Operations Research
- Elective 2
  1. Computational Intelligence
  2. VLSI Design
  3. Deep Learning
  4. Parallel Computing
  5. Natural Language Processing
  6. Software Project Management
  7. Data Science
- Elective 3
  1. Cloud Computing
  2. Data Warehousing and Data Mining
  3. Advanced Graph Theory
  4. Full Stack Development
  5. IoT and Embedded Systems
  6. Optimization Techniques
  7. Network Security
  8. Bioinformatics

N.B. More Elective papers to be added as and when necessary

Regulations for
Two-Year M.Sc. Course in Computer Science

1. The Department of Computer Science, West Bengal State University, Barasat shall provide instructions leading to 2-Year, 4-Semester M.Sc. Degree in Computer Science.

2. A candidate who has passed 3-Year B.Sc. Examination in Computer Science from West Bengal State University, Barasat (or from any other UGC recognised University or Institution) can apply for admission to the M.Sc. course. Admission for candidates from other Universities/Colleges will be governed according to the university rules.

3. The examinations for the M.Sc. course shall be held in 4 Semesters. At the end of each semester, an examination of the papers covered in that semester would be held. This examination will be referred to as the M.Sc. examination of that semester. In any semester, a study break between the completion of regular classes and the commencement of the Semester Examination will generally be a maximum of 10 calendar days.

4.1 The total credits for the 2-Year (4-Semester) course in Computer Science will be 88. The distribution of credits for each category is as follows:

Subject	Credit	Total Credit
Theory and Tutorial	4 x 14	56
Laboratory	4 x 2	8
Project	8 + 4	12
Viva-voce	4 x 1	4
Term Paper	4 x 1	4
AEC	2 x 1	2
SEC	2 x 1	2
Total credit of the course		88

4.2 Examination of a Theoretical Paper is of 2-hour duration and will usually carry a total of 40 marks. 10 marks for each theoretical paper will be set aside as part of continuous assessment to be evaluated by the teacher(s) assigned for those classes.

4.3 For Theoretical papers, paper setters and examiners will be appointed from the associated Board of Examiners duly constituted.

4.4 Evaluation of performance in a Practical paper will be based on Sessional works and end-semester viva-voce examination in that paper. The distribution of marks for each Practical paper would be as follows:

i) 50% for experiments performed in the Laboratory – the Sessional works to be evaluated by the teachers assigned for that course.

ii) 40% for viva-voce on the experiments to be conducted by a Board constituted by the faculty members and/or external examiners.

iii) 10% for Laboratory report to be evaluated by the viva-voce Board.

Only the total marks are to be shown in the mark-sheet.

4.5 In order to pass a semester examination, a candidate will have to score a minimum of 40% marks in each theoretical paper and 50% marks in each practical paper, and a candidate must appear in each theoretical and practical paper of a semester examination. Pass marks for project, grand viva-voce, and term paper examinations will be 50% as it is in practical papers.

4.6 Each candidate will have to complete a Term paper assignment in the 3rd semester. He/She will have to make a report and deliver a presentation before a Board of Examiners, on the topic of the term paper leading to the final year project.

4.7 Evaluation of the performance in a Term paper will be done by the Board of examiners only.

4.8 Each student will have to undertake project work at the beginning of the 3rd semester. The project work would have to be completed under the supervision of faculty member(s) of the department at the end of the 4th semester; a student will have to submit, through the respective supervisor(s), a dissertation on the project work. The project work will be assessed by a Board of Examiners being constituted by the faculty members of the department and External Examiner(s).

4.9 At the end of the 4th semester, a student will have to appear at a Grand Viva-voce examination. The grand viva-voce will be conducted by a Board of Examiners being constituted by the faculty members of the department and External Examiner(s).

5.1 A candidate shall be eligible to appear at the Semester Examinations provided he/she is present in regular course of studies with proper attendance as per the University rules.
5.2 The 2nd to 4th Semester classes will begin immediately after the completion of the previous semester examination.
5.3 All candidates who have completed a semester examination shall join the next semester classes. Candidates failing to qualify in a semester examination shall automatically revert back to the respective semester in the next academic session immediately after publication of the result. However, the candidate failing in a paper in the previous semester has to clear the paper(s) as per the rule stated in 5.4 below.
5.4 A candidate will get a maximum of three consecutive chances including the first one in his/her regular year in order to pass each of the Semester Examinations.
6. The final result (combining all the Semester results) will be determined by adding marks scored by the candidate for all theoretical and practical papers separately. A candidate obtaining a minimum of 40% marks in each theoretical paper and a minimum of 50% marks in each practical paper will be declared as "Pass" with Second Class. However, if the total marks scored by a candidate is 60% comprising all 'pass marks' in all Semester examinations will be declared as "Pass" with First Class.

Program Objective(s)

After completing master and few years of masters, the Computer Science students would	
PO I	Technical Expertise: Implement fundamental domain knowledge of core courses for developing effective computing solutions by incorporating creativity and logical reasoning.
PO II	Successful Career: Deliver professional services with updated technologies in computer science based careers.
PO III	Soft Skills:Develop leadership skills and incorporate ethics, team work with effective communication and time management in the profession.
PO IV	Life Long Learning: Conduct research among computing professionals as per market needs.

Program Specific Outcome(s)

Students will be able to

PSO1: Apply knowledge of mathematics, science and algorithms in solving complex Computer engineering problems.

PSO2: Generate solutions by conducting experiments and applying techniques to analyze and interpret data

PSO3: Design component, or processes to meet the needs within realistic constraints.

PSO4: Identify, formulate, and solve Software Engineering, Networking and Data Mining problems.

PSO5: Comprehend professional and ethical responsibility in the computing profession.

PSO6: Express effective communication skills.

PSO7: Participate in global, economic, environmental, and societal context.

PSO8:Recognize the need for, and an ability to engage in life-long learning.

PSO9: Knowledge of contemporary issues and emerging developments in the computing profession.

PSO10: Utilize the techniques, skills and modern computer Engineering tools, Software and techniques necessary for Engineering practice.

PSO11: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.

PSO12: Design research problems and conduct research in a computing environment.

Mapping of PEO and PSO

Programme Objective(s)	Program Specific Outcome(s)	
PO I	Technical Expertise: Implement fundamental domain knowledge of core courses for developing effective computing solutions by incorporating creativity and logical reasoning.	1, 2, 4, 7, 8, 10
PO II	Successful Career: Deliver professional services with updated technologies in computer science based career.	3, 5, 6, 11
PEO III	Soft Skills: Develop leadership skills and incorporate ethics, team work with effective communication and time management in the profession.	3, 9
PO IV	Life Long Learning: Conduct research among computing professionals per market needs.	12

M.Sc. (Computer Science)

(4/2 Semesters)

First Semester

CMS2PCOR01T: ADVANCED COMPUTER ARCHITECTURE **Full Marks: 50**

Objectives

The course should enable the student

1. To learn the basics of stored program concepts.
2. To learn the principles of pipelining.
3. To learn mechanism of data storage
4. To distinguish between the concepts of serial, parallel, pipeline architecture.

Outcomes

The Student should be able to

1. Learn pipelining concepts with a prior knowledge of stored program methods
2. Learn about memory hierarchy and mapping techniques.
3. Study of parallel architecture and interconnection networks .

Detailed syllabus:

Computer Architecture and Organization. Control unit design, Basic Parallel Processing Architecture, Taxonomy-SISD. MISD, SIMD, MIMD structures, Serial, Parallel and Concurrent Computation, CISC Vs RISC, Structure of Instruction of instruction sets and Desirable Attributes. Basic Concepts of pipelining, Instruction Pipelining. Hazards, Reservation Tables, Collision, Latency, Dynamic pipeline, Vector processing and Vector processors. Cache Memory and Virtual Memory: Structure, Analysis and Design. I/O Systems: Design Issues, Performance Measures. Loosely Coupled and Tightly Coupled Systems, Concurrency and Synchronization, ScaLaboratoryility, Models of Consistency, Application of SIMD Structure. Definition. Types of Interconnected Networks; Baselines, Shuffle- Exchange, Omega, Cuba, Comparison and Application. Mapping Algorithms to array structures, Systolic processors. Mapping design and Optimization, Wave Front Array processor. Data Flow Graphs, Petri nets, Static and Dynamic DFA. Different Models, Languages, Compilers, dependency Analysis. Message Passing, Program mapping to Multiprocessors, Synchronization. Case Study: Basic Features of Current Architectural Trends. DSP Processor, Dual core Technology

Books:

1. K. Hwang, Advanced Computer Architecture: Parallelism, ScaLaboratoryility, Programmability, McGraw-Hill,
2. K. Hwang, F.A. Briggs, Computer architecture and parallel processing, McGraw-Hill,

References Books:

1. D.A. Patterson, J.L. Hennessy, D. Goldberg, Computer Architecture : A Quantitative Approach 2nd Edn, Addison- Wesley.
2. Harold Stone, High-performance Computer Architecture (3rd edition), Addison Wesley.
3. P. V. S. Rao, Perspectives in Computer Architecture, PHI.
4. Hayes, Computer Architecture and Organization, 2nd and 3rd Edn

CMS2PCOR02T: - ADVANCED OPERATING SYSTEM **Full Marks:50**

Objectives

The course should enable the student

1. Learn operating system design and concept
2. Understand process, process life cycle
3. Difference between different types of operating system

Outcomes

The student should be able to

1) The students will understand the design approaches of advanced operating systems
2) Analyze the design issues of distributed operating systems.
3) Evaluate design issues of multi processor operating systems.
4) Identify the requirements of operating systems.
5) Formulate the solutions to schedule the real time applications.

Detailed syllabus:

Review of operating system. Introduction to Parallel and Distributed Systems. State recovery and clock models for distributed systems. Classification of control algorithms for dist and parallel systems process and mode synchronization, classical OS, Process Migration, termination detection, Remote Procedure Call. Case study on various operating systems.

Text Book:
1. Tanenbum, A.S., Distributed O/S , Pearson Education.
2. Singhal, Shivaratri, Advanced Concepts In O/S, Tmh.

References Books:
1. P. K.Sinha, Distributed O/S, Phi
2. Balakrishna Prasad, Operating Systems and Systems Programming - 2nd Edn., Scitech
3. Avi Silberschatz , Peter Baer Galvin , Greg Gagne , Operating System Concepts Eight Edition
4. Allen B. Downey, Think Os A Brief Introduction To Operating Systems, Green Tea Press
5. G. Coulouris, J. Dollimore, T. Kindberg, G. Blair, Distributed Systems Concepts And Design 5th Edition

CMS2PCOR03T: ADVANCED DATABASE MANAGEMENT SYSTEMS **Full Marks: 50**

Objectives

The course should enable the student

1. To make a study of SQL and relational database design.
2. To understand the internal storage structures of different file and indexing techniques will help in physical DB design.
3. To know the fundamental concepts of transaction processing- concurrency control techniques and recovery procedure.
4. To have an introductory knowledge about the emerging trends in the area of distributed DB- OODB- Data mining and Data Warehousing.
5. To learn the basics of query evaluation and optimization techniques.

Outcomes

The Student should be able to

1. Explore the basic concepts of database systems.
2. Write SQL queries for a given scenario.
3. Describe relational database theory, and be able to write relational algebra expressions for queries.
4. Design logical data models
5. Evaluate and optimize queries
6. Implement transaction processing and concurrency control
7. Develop Object oriented dB, Distributed dB using XML, data warehousing

Detailed syllabus:

Query Processing, Query Optimization Algorithms. Transaction concepts, Recovery and Concurrency Control, Locking and Timestamp based protocols, Multiversion and Optimistic Concurrency Control schemes, Threats and countermeasures. Object-oriented and Object Relational Databases, Distributed Databases, Data Warehouse and Data Mining, Database Security, Emerging Technologies.

Text Book:
1. Elmasri, Navathe, Fundamentals of Database System, 3/e, Pearson Education.
2. Ozsu, Principals of Distributed Database System, Pearson Education.

References Books:
1. R. Chakrabarti, S. Dasgupta, ADVANCED DATABASE MANAGEMENT SYSTEM, Wiley
2. Carolyn Begg, Thomas Connolly, Database Systems, 4th Edition, Addison-Wesley
3. Dr. S. Sumathi, S. Esakkirajan , Fundamentals of Relational Database Management Systems, ISBN
4. Raghu Ramakrishnan , Database Management System 2nd Edition
5. Korth, Database Management System

CMS2PCOR04T: Advanced Compiler Design **Full Marks: 50**

Objectives

The course should enable the students to

1. Understand the basic principles of the compiler, Compiler construction tools and lexical analysis.
2. Learn the context of Context Free Grammar, Parsing and various parsing techniques.
3. Learn the process of intermediate code
Generation.
4. Learn the process of Code Generation and various Code optimization techniques.
5. Understand the need of Program verification and Lambda calculus in verification.

Outcomes

The student should be able to

1. Differentiate the various phases of a compiler.

2. Apply parsing techniques and be able to write Context Free Grammars for various languages.
3. Design the structure of intermediate code for various types of statements and expressions.
4. Design code generators and apply code optimization techniques.
5. Can design own compiler of any work specific application.

Detailed syllabus:
Compiler design, various phases; lexical analyzer, token, lexeme, and patterns. Regular definitions, Transition Diagrams, Syntax Analysis, ambiguity, associatively, precedence, Top down Parsing, recursive-descent parsing, predictive parsing, Bottom up Parsing, Operator precedence grammar, LR parsers Syntax directed definitions: inherited and synthesized attributes. Type checking, Symbol Tables. Runtime systems, Activation tree, Activation record, Basic Blocks, Dataflow analysis, Code optimization and code generation.

Text Book:
1. Aho, Compilers: Principals, Techniques and Tools, Pearson Education.
2. Muneeswaran, Compiler Design, Oxford

References Books:
1. Sudha Sadasivam, Compiler Design - 2nd Edn., SCITECH
2. Niklaus Wirth, Theory and Techniques of Compiler Construction, Addison-Wesley
3. Compiler Design, Santanu Chattopadhay, PHI.

CMS2PCOR05P: ADVANCED DATABASE MANAGEMENT SYSTEMS Lab Full Marks: 50

Objectives
The course should enable the students to:
1. Learn to write a query
2. Learn to design a simple dB using data modeling techniques
3. To provide the knowledge of various dB tools

Outcomes
At the end of the course the student should be able to:
1. Populate and query a database using SQL DML/DDL commands
2. Write programs using PL/SQL including stored procedures , cursors, packages etc.
3. Construct real time database application using current techniques

Detailed syllabus:
The SQL programming course is very important to prepare various software. This has immense importance nowadays. This course will help students to develop various software which in turn will increase their potential for employability and entrepreneurship. This is one of the state-of-the-art fields of study in today's society.
Introduction to SQL, Database Schema Design, Database Creation, SQL Programming and Report Generation using a RDBMS. Students are to be exposed to front-end development tools, ODBC; Internet based access to databases and database administration. Assignments on developing programs and functions related to the theoretical paper coverage on DATABASE LABORATORY.

CMS2PAEC01M: Application Development Lab Full Marks: 50

Objectives
The course should enable the students to:
1. To provide a comprehensive understanding of mobile application development for both Android (using Kotlin) and cross-platform (using Flutter/Dart)
2. To develop skills in designing intuitive user interfaces, managing data, and integrating external services and APIs
3. To understand and apply best practices in mobile app architecture, storage optimization, and user experience design for cross-platforms
4. To gain practical experience in integrating cloud services and preparing apps for deployment

Outcomes
At the end of the course the student should be able to:
1. Design and develop interactive mobile user interfaces using Kotlin for Android and Flutter for cross-platform development
2. Implement efficient data management solutions using local storage, cloud databases, and state management techniques.
3. Integrate device features, background processing, and external services into mobile applications on both platform platforms
4. Apply modern mobile app architecture principles and best practices to create production-ready applications for

Android and Flutter

Detailed syllabus:
Design and development of desktop, web, and mobile applications using modern programming languages, frameworks, and integrated development environments. The Laboratory focuses on practical implementation of software development concepts and user interface design.
Implementation of frontend development techniques using HTML, CSS, JavaScript, Bootstrap, and modern frameworks for creating responsive and interactive user interfaces.
Development of backend applications using server-side programming languages, database connectivity, API integration, session management, authentication, and CRUD operations in application development.
Database design and management using SQL and NoSQL databases, normalization, query processing, transaction handling, and integration of databases with software applications.
Development and deployment of full stack applications, testing and debugging techniques, version control using Git and GitHub, software maintenance, and collaborative development practices.
Mini projects and case studies involving real-world applications such as management systems, e-commerce applications, cloud-based applications, mobile applications, and intelligent software solutions.

Text Books
1. Web Development with Node and Express — Ethan Brown; O'Reilly Media.
2. Learning Web Design — Jennifer Niederst Robbins; O'Reilly Media.

Reference Books
1. Eloquent JavaScript — Marijn Haverbeke; No Starch Press.
2. Pro MERN Stack — Vasan Subramanian; Apress Publications.

Second Semester

CMS2PCOR06T: Image Processing and Machine Learning **Full Marks: 50**

Objectives
The course should enable the student:
1. Learn digital image fundamentals and image enhancement
2. Understand the concept of Image Restoration and segmentation
3. Be familiar with image compression and image representation
4. Learn to machine learning algorithms Understand the Neural networks
5. Understand the Neural networks

Outcomes:
The Student should be able to:
1. Explain the fundamentals of image processing and image enhancement
2. Apply image processing restoration techniques and segmentation methods
3. Apply image processing Compression and representations,
4. Develop algorithms for machine leavening
5. Explain the analysis of neural network techniques

Detailed syllabus:
Introduction – Origin – Steps in Digital Image Processing – Components – Elements of Visual Perception Image Sampling and Quantization – Relationships between pixels - color image processing – RGB color model – HSV and LAB Color model, Image Transform –DCT Image enhancement Spatial Domain: Gray level transformations Histogram processing – Basics of Spatial Filtering– Smoothing and Sharpening Spatial Filtering – Frequency Domain: Introduction to Fourier Transform – Smoothing and Sharpening frequency domain filters – Ideal, Butterworth and Gaussian filters Noise models – Mean Filters – Order Statistics – Adaptive filters – Band reject Filters – Band pass Filters –Notch Filters – Optimum Notch Filtering – Inverse Filtering – Wiener filtering Segmentation: Detection of Discontinuities–Edge Linking and Boundary detection – Region based segmentation--Morphological based boundary detection.
Compression: Fundamentals – Image Compression models –Lossy Compression – Lossy Predictive Coding –Compression Standards - JPEG- MPEG, Boundary representation – Chain Code – Polygonal approximation signature, boundary segments – Boundary description – Shape number – Fourier Descriptor, moments- Regional Descriptors – Topological feature, Texture
Machine learning Introduction – Supervised Unsupervised learning and Reinforcement Learning Classification and Regression: K-Nearest Neighbor, Linear Regression, Logistic Regression, Support Vector Machine (SVM) K- means clustering.
Biological and Artificial Neural Network – Activation Function -McCulloch Pitts Neuron -thresholding logic Perceptron- Multilayer Perceptron Feedforward Neural Network – Backpropagation Neural Network-Adam Optimizer

Text Books

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson Education, 2010.
2. A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall India, 1988.

Reference Books

1. Madhuri A. Joshi, Digital Image Processing: An Algorithmic Approach, PHI Publisher, 2006.
2. S. Sridhar, Digital Image Processing, Oxford University Press, 2011.
3. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, Tata McGraw Hill Pvt. Ltd., 2011.
4. William K. Pratt, Digital Image Processing, John Wiley and Sons, 2002.
5. Malay K. Pakhira, Digital Image Processing and Pattern Recognition, PHI Learning Pvt. Ltd., 2011.

CMS2PCOR07T: Advanced Data Structures and Algorithms **Full Marks: 50**

Objectives

The course should enable the student

1. To introduce algorithms analysis and design techniques
2. To understand algorithms of various data structures used for searching, sorting,
3. indexing operation and dynamic programming

Outcomes

The Student should be able to

1. Ability in using the appropriate algorithm for searching, sorting, indexing operations
2. Designing of new algorithms
3. Analyzing complexity issues of algorithms

Detailed syllabus:

Basic concepts about Algorithms, Data Structures, Recursion, Iteration, Big-O Notation, Brief Foundations and Applications of Trees – Definitions, Representations, Binary Tree and Its Usefulness, Binary Search Tree, Tree Traversal, Threaded Binary Trees, Binary Tree Representation of any Tree other than Binary Tree, Decision Trees, Balanced Tree Schemes – AVL Trees, 2-3 Trees. Basic concepts about Searching, B-Trees, Hashing. complexity issues of different Sorting Algorithms. Binomial Heaps, Fibonacci Heaps, Amortized Analysis of Algorithms, Divide and Conquer Algorithms: Multiplications of Large Integers , Strassen's Matrix Multiplication algorithm. Dynamic Programming :shortest path, chained matrix multiplication, optimal binary search trees, Travelling salesman problem. Greedy Algorithm Knapsack problem. Computational Complexity and Intractability. Introduction to NP.

Text Book:

1. T. H. Cormen et al -Introduction to Algorithms, PHI
2. E.Horowitz, S. Sahani- Fundamentals of Computer Algorithms –Galgotia.

References Books:

1. S. Sahani, Data Structures, Algorithms And Applications In C++ 2nd Edition, ORIENT BLACKSWAN PVT LTD
2. Robert Sedgewick and Philippe Flajolet, An Introduction to the Analysis of Algorithms, Addison-Wesley
3. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley
4. Udi Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley
5. Thomas H. Cormen, Algorithms Unlocked, MIT

CMS2PCOR08T: Cyber Security and IoT **Full Marks: 50**

Outcomes

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

1: Understand the concepts, architecture, and applications of Cyber Security and IoT systems.
2: Identify cyber threats, attacks, and vulnerabilities in networks and IoT environments.
3: Apply cryptographic and network security techniques for securing communication systems.
4: Design and implement secure IoT applications using modern hardware and software platforms.
5: Analyze ethical, legal, and forensic aspects of cyber security.
6: Explore advanced research trends and emerging technologies related to cyber security and IoT.

Outcomes

1: Apply advanced knowledge of computer science principles in solving real-world problems.
2: Analyze complex computing problems using logical and critical thinking.
3: Design and develop secure and efficient computing solutions.

4: Use modern computing tools, technologies, and platforms effectively.
5: Demonstrate effective communication and teamwork skills.
6: Understand professional ethics, cyber laws, and social responsibilities.
7: Engage in independent learning and research activities.
8: Develop innovation and employability skills in emerging technologies.

Detailed syllabus:
Basic concepts of Cyber Security and Internet of Things (IoT), including information security principles, cyber threats, IoT architecture, sensors, actuators, smart devices, machine-to-machine communication, and applications of IoT in intelligent systems.
IoT architecture, wireless sensor networks, RFID, NFC, and communication protocols such as MQTT, CoAP, ZigBee, BLE, IPv6, and 6LoWPAN. It also covers cloud, edge, and fog computing technologies used in IoT environments.
Cyber attacks and network security mechanisms including malware, phishing, ransomware, DoS/DDoS attacks, spoofing, session hijacking, firewalls, IDS/IPS, VPNs, SSL/TLS, and IoT security challenges.
Cryptography and IoT security concepts including symmetric and asymmetric encryption, AES, RSA, hash functions, digital signatures, authentication, access control, lightweight cryptography, secure key management, and blockchain-based IoT security.
Ethical hacking and cyber forensics including reconnaissance, vulnerability assessment, penetration testing, password attacks, digital evidence collection, forensic investigation, incident response, cyber laws, and IT regulations.
Applications and emerging trends in Cyber Security and IoT such as smart homes, smart cities, industrial IoT, healthcare systems, AI and machine learning in security, cloud security, edge AI, cyber physical systems, and future research directions.

Text Books
1. Internet of Things: A Hands-on Approach by Arshdeep Bahga and Vijay Madisetti.
2. Cryptography and Network Security by William Stallings.

Reference Books
1. Network Security Essentials by William Stallings.
2. Practical Internet of Things Security by Brian Russell and Drew Van Duren.
3. Hacking: The Art of Exploitation by Jon Erickson.
4. Cyber Security and Cyber Laws by Alfred Basta and Nadine Basta.
5. IoT Security by Madhusanka Liyanage.

CMS2PCOR09T: Artificial Intelligence and Big Data Analytics **Full Marks: 50**

Objectives
The course should enable the student
1. To provide the foundations for AI problem solving techniques and knowledge representation formalisms

Outcomes
The Student should be able to
1. Ability to identify and formulate appropriate AI methods for solving a problem
2. Ability to implement AI algorithms
3. Ability to compare different AI algorithms in terms of design issues, computational complexity, and assumptions

Detailed syllabus:
Importance of Al, Scope of Al, Goals of AI, AI and Related fields, State-Space Graphs, Implicit and Explicit Graphs, Production Systems, Formulating the State-Space; Uniformed search: Depth-first Search, Breadth-first Search; Uniform Cost algorithm; Use of Heuristics, A* Algorithm, Admissibility of A*; Analysis and comparison of Search algorithms;
Two-agent games, AND/OR Graphs, Minimax Procedure, α-β pruning procedure, Learning evaluation functions; Introduction to ES, Knowledge-Based systems, Knowledge Representation: Rule_Based approach: Forward and Backward Chaining, Semantic-Nets Based approach, Frame Based approach; Introduction to Constrained Satisfaction Problems(CSP), Applications, Algorithms to CSPs, Symbolic constraints and Propagation; Introduction to programming in logic. Declarative and Procedural Meaning, Data Objects, Lists, Operators, Controlled Backtracking.
Soft computing basics. Fuzzy Systems: Fuzzy sets, Fuzzy logic. Fuzzy relations, Approximate Reasoning, Fuzzy logic control systems. Artificial Neural Networks: Feedforward Networks and Supervised Learning Perception learning rules, Adaline, Back propagation. Unsupervised Learning Networks. Genetic Algorithm (GA): Evolutionary Computing. Basics of Genetic Algorithms Reproduction, Crossover Mutation, Schemata, Fitness Function.

INTRODUCTION TO BIG DATA AND HADOOP Types of Digital Data, Introduction to Big Data, Big Data Analytics, History of Hadoop, Apache Hadoop, Analysing Data with Unix tools, Analysing Data with Hadoop, Hadoop Streaming, Hadoop Echo System .
HDFS(Hadoop Distributed File System) The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression,

Serialization, Avro and File-Based Data structures.

Text Book:

1. E. Rich and K. Knight: Artificial Intelligence, TMH
2. Dan W. Patterson: Introduction to Artificial Intelligence and Expert Systems

Reference Books:

1. S. Russel and P. Norvig, "Artificial Intelligence, A modern Approach"
2. Cloksin and Mellish , Programming In Prolog , Narosa Publishing House
3. Nillson, Principles Of Artificial Intelligence, Harcourt Asia and Morgan
4. Janakiraman, Sarukesi and Gopal Krishnan , Foundation Of Artificial Intelligence and Expert System, Macmillan
5. Hadoop: The definitive Guide, Tom White, O'reilly
6. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.

CMS2PCOR10P: Image Processing and Machine Learning Laboratory **Full Marks: 50**

Objectives

The course should enable the student

1. To understand basic image processing operations such as sampling, quantization, DCT, and histogram processing using Python.
2. To gain knowledge of image enhancement and image segmentation techniques using Python.
3. To understand and implement linear regression using Python.
4. To impart knowledge of classifiers and their implementation using Python.

Outcomes

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

1. Demonstrate the implementation of various image processing operations such as sampling, quantization, histogram processing, and image transforms using Python.
2. Design and implement image enhancement and segmentation techniques using Python.
3. Analyze and implement linear regression models and classifiers using Python for data analysis and pattern recognition tasks.

Detailed syllabus:

Implementation of basic image processing operations including image acquisition, image representation, image enhancement, filtering, histogram processing, edge detection, segmentation, morphological operations, and image transformation techniques using Python or MATLAB.

Experiments on machine learning techniques including data preprocessing, feature extraction, classification, clustering, regression analysis, model training, testing, and performance evaluation using machine learning libraries and tools.

Implementation of supervised learning algorithms such as linear regression, logistic regression, decision trees, support vector machines, k-nearest neighbors, and naïve Bayes classifiers for image and data analysis applications.

Implementation of unsupervised learning techniques including k-means clustering, hierarchical clustering, dimensionality reduction, principal component analysis, and feature selection methods.

Introduction to deep learning frameworks and practical implementation of convolutional neural networks, image classification, object detection, pattern recognition, and image-based intelligent systems using TensorFlow or PyTorch.

Mini projects and case studies related to medical image analysis, face recognition, handwritten character recognition, object tracking, intelligent surveillance, and real-world machine learning applications.

Text Books

1. Digital Image Processing — Rafael C. Gonzalez and Richard E. Woods; Pearson Education.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow — Aurélien Géron; O'Reilly Media.

Reference Books

1. Pattern Recognition and Machine Learning — Christopher M. Bishop; Springer.
2. Digital Image Processing Using MATLAB — Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins; McGraw Hill Education.