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

COURSE SCHEME

FOR

MCA

2025 Onwards

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MASTER OF COMPUTER APPLICATIONS (MCA)

SEMESTER-I

S. No.	Course No.	Title	L	T	P	Cr
1	MCA102	Data Base Management System	3	0	2	4.0
2	MCA103	Operating Systems	2	0	2	3.0
3	MCA105	Data Structures	3	0	2	4.0
4	MCA 302	Object Oriented Programming	3	0	2	4.0
5	MCA109	Data Science Foundation	3	0	2	4.0
6	MCA 207	Machine Learning	3	0	2	4.0
7	MCA107*	Computer Programming (Bridge Course)	3	0	2	0.0
		Total	16		14	23

SEMESTER-II

S. No.	Course No.	Title	L	T	P	Cr
1	MCA201	Theory of Computation	3	1	0	3.5
2	MCA202	Software Engineering	3	0	2	4.0
3	MCA204	Artificial Intelligence	3	0	2	4.0
4	MCA205	Design and Analysis of Algorithms	3	0	2	4.0
5	MCA108	Computer Networks	2	1	2	3.5
6		Elective-I	3	0	2	4.0
		Total	17		10	23

SEMESTER-III

S. No.	Course No.	Title	L	T	P	Cr
1.	MCA 203	Big Data Analytics and Business Intelligence	3	0	2	4.0
2	MCA303	IT Project Management	3	0	2	4.0
3	MCA312	Image and Video Processing	3	0	2	4.0
4	MCA391	Capstone Project	0	0	0	6.0
5		Elective-II	3	0	2	4.0
		Total	12	0	8	22.0

SEMESTER-IV

S. No.	Course No.	Title	L	T	P	Cr
1	MCA 492	Project Semester	-	-	-	15
		Total	-	-	-	15

Alternate Semester (For students having CGPA below 6.0)						
S. No.	Course No.	Title	L	T	P	Cr
1	MCA493	Project	0	0	0	8.0
2	MCA405	Ethical Hacking	3	0	2	4.0
3	MCA406	Social Network Analysis	2	0	2	3.0
		Total	8	0	6	15.0
		Grand Total of All Semesters				**83**

***To be offered to Non-Computer background students as a bridge course**

ELECTIVE-I

S. No.	Course No.	Title	L	T	P	Cr
1	MCA212	Deep Learning	3	0	2	4.0
2	MCA208	Cryptography and Network Security	3	0	2	4.0
3	MCA209	Source Code Management	3	0	2	4.0
4	MCA 213	Cloud Computing	3	0	2	4.0

ELECTIVE-II

S. No.	Course No.	Title	L	T	P	Cr
1	MCA 214	Natural Language Processing and Language Models	3	0	2	4.0
2	MCA305	Secure Coding	3	0	2	4.0
3	MCA306	Build and Release Management	3	0	2	4.0
4	MCA215	GPU Computing	3	0	2	4.0

MCA102 DATABASE MANAGEMENT SYSTEM				
	L	T	P	Cr
	3	0	2	4.0

Course objective: Emphasis is on the need of information systems. Main focus is on E-R diagrams, relational database, concepts of normalization and de-normalization and PL/SQL commands.

Introduction: Data, data processing requirement**,** desirable characteristics of an ideal data processing system, traditional file-based system its drawback, concept of data dependency, Definition of database, database management system, 3-schema architecture, database terminology, benefits of DBMS,

Database Analysis: Conceptual data modeling using E-R data model -entities, attributes, relationships, generalization, specialization, specifying constraints.

Relational Data Model, Relational Model Constraints Relational Algebra: Set-theoretic operations: Union, Intersection, difference, Cartesian product, Selection, Projection, Types of Join: Theta, Inner, Left Outer, Right Outer, Natural & Division, Relational calculus: Tuple and Domain relational calculus.

Relational DataBase Design: Functional Dependencies & Normalization for Relational Databases. Closure, Lossless join & Dependency preserving decomposition. Normal forms: INF, 2NF, 3NF, BCNF, 4NF and 5NF.

SQL: A Relational Database Language, SQL data types, Constraints: Primary key, foreign key, check, unique, not null. Table level and column level specifications, DDL, DML, and DCL Queries in SQL, Data retrieval statements and sub-queries, various Data type functions: Numeric, character, and date data type functions. Views, Sequences, Synonyms, Indexing.

Concurrency Control & Recovery Techniques: Transactions, Concurrency Control Techniques: Serializability, Locking Techniques, Time stamping Protocols, Recovery Techniques, Recovery concepts, Database backup and recovery from catastrophic failures

PL/SQL: Introduction to PL/SQL, data types, Cursors, Triggers, Functions, Procedures, Packages and Exception Handling.

Laboratory work: Lab work will be based on implementing database concepts using SQL and PL/SQL. Students will also implement one mini project.

Sample Mini Project:

1. Electronic Health Record (EHR) Management System.
2. Smart Agriculture Monitoring System.
3. Crime Reporting and Analysis System

Recommended Books:

1. H. F. Korth & Silverschatz, A., Database System Concepts, Tata McGraw Hill, 2010, 6th ed.
2. Elmasri & Navathe, Fundamentals of Database Systems, Addison-Wesley, 2011, 6thEd.
3. Hoffer, Prescott, Mcfadden, Modern Database Management, Paperback International, 2012, 11th Ed.
4. Martin Gruber, Understanding SQL, BPB Publication, 1994, Revised Ed.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Analyze the Information Systems as socio-technical systems, its need and advantages as compared to traditional file-based systems.
CLO2	Analyze and design database using E-R data model by identifying entities, attributes and relationships.
CLO3	Apply and create Relational Database Design process with Normalization.
CLO4	Comprehend the concepts of transaction management, concurrency control and recovery management.
CLO5	Demonstrate use of SQL and PL/SQL to implement database applications.

MCA103 OPERATING SYSTEMS				
	L	T	P	Cr
	2	0	2	3.0

Course Objective: Role and purpose of the operating system, Functionality of a typical operating system, managing atomic access to OS objects.

Operating System Principles: Structuring methods (monolithic, layered, modular, microkernel models), processes, and resources, Application bases, Concepts of APIs, Device organization, interrupts: methods and implementations, Concept of user/system state and protection, transition to kernel mode.

Concurrency: Implementing synchronization primitives and multiprocessor issues.

Scheduling and Dispatch: Dispatching and context switching, Pre-emptive and non-pre-emptive scheduling, Schedulers and policies, Processes and threads

Memory Management: Review of physical memory and memory management hardware, working sets and thrashing, Caching, Paging and virtual memory, Virtual file systems.

File Systems: Files: data, metadata, operations, organization, buffering, sequential, non-sequential, Directories: contents and structure, Naming, searching, Journaling and log-structured file systems.

Deadlock: Introduction, Analysis of conditions, Prevention & avoidance, Detection & recovery.

Security and Protection: Overview of system security, Security methods and devices, Protection, access control, and authentication.

Virtual Machines: Types of virtualizations (including Hardware/Software, OS, Server, Service, Network).

Self-Learning Contents:

Device Management: Characteristics of serial and parallel devices, Buffering strategies, Direct memory access, Disk structure, Disk scheduling algorithms.

Laboratory Work: To explore different operating systems like Linux, Windows etc. To implement main algorithms related to key concepts in the operating systems using a high-level language. Understanding of OS virtualization. Students has to submit a mini project.

Sample Mini Projects:

- Develop a basic task manager that monitors CPU and memory usage and suggests actions to save energy.
- Simulate efficient virtual memory management techniques for optimized performance in industrial systems.
- Modify a disk scheduling algorithm to reduce energy consumption.

Recommended Books

1. Silberschatz, A., Galvin, P.B. and Gagne, G., Operating System Concepts, John Wiley (2013), 9th Ed.
2. Stallings, Willam, Operating Systems Internals and Design Principles, Prentice Hall, (2014), 7th Ed.
3. Dhamdhere, D.M., Operating Systems: A Concept-Based Approach, McGraw Hill , (2008), 2nd Ed.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Comprehend the concepts related to the basics of operating System.
CLO2	Exemplify protection and security in the operating system.
CLO3	Simulate the various process management algorithms and specifications for simple as well as for concurrent processes.
CLO4	Grasp the concepts related to memory management.
CLO5	Implement the concepts related to file organization and handling in system.

MCA105 DATA STRUCTURES

L	T	P	Cr
3	0	2	4.0

Course Objectives: To become familiar with different types of Data structures and their applications. To become familiar with different types of algorithmic techniques and strategies.

Linear Data Structures: Arrays, Linked lists, Strategies for choosing the appropriate data structure, Abstract data types and their implementation: Stacks, Queues, Priority queues, Sets, Maps.

Basic Analysis: Differences among best, expected, and worst-case behaviours of an algorithm, Asymptotic analysis of upper and expected complexity bounds, Big O notation: formal definition and use, little o, big omega and big theta notation, Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential, Time and space trade-offs in algorithms, Recurrence relations, Analysis of iterative and recursive algorithms.

Searching and Sorting: Linear Search, Binary Search, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Quick Sort, Heap Sort, Merge Sort, Counting Sort.

Algorithmic Strategies with examples and problem-solving: Brute-force algorithms with examples, Greedy algorithms with examples, Divide-and-conquer algorithms with examples.

Non-Linear Data Structures: Hash tables, including strategies for avoiding and resolving collisions, Binary search trees, Common operations on binary search trees such as select min, max, predecessor, successor, insert, delete, iterate over tree, Graphs and graph algorithms, Representations of graphs, Depth- and breadth-first traversals, Heaps, Graphs and graph algorithms, Shortest-path algorithms (Dijkstra and Floyd), Minimum spanning tree (Prim and Kruskal).

Laboratory Work: Implementation of Arrays, Recursion, Stacks, Queues, Linked-Lists, Binary trees, sorting techniques, Searching techniques. Implementation of all the algorithmic techniques.

Recommended Books:

1. Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, The MIT Press (2009) 3rd Edition.
2. Langsam, Augenstein, & Tenenbaum, Data Structures Using C and C++, Pearson Education India (2015), 2nd Ed.

3. Tremblay & Sorenson, An Introduction to Data Structures with Application, McGraw Hill Education (2017), 2nd Ed.
4. Sedgewick & Wayne, Algorithms, Addison-Wesley Professional (2011), 4th Ed.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Understand and implement various linear data structures: Array, Stack, Queue and Link List
CLO2	Learn various searching and sorting techniques.
CLO3	Understand and implement various Non Linear data structures: Trees and Graphs
CLO4	Analyse, evaluate and choose appropriate data structure and algorithmic technique to solve real-world problems.

MCA207 MACHINE LEARNING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course provides an advanced level of understanding to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications.

Introduction: Well-Posed learning problems, Basic concepts, Designing a learning system, Types of machine learning: Learning associations, Supervised learning, Unsupervised, Semi-supervised and Reinforcement learning, Data Pre-processing Technique: Data Cleaning, Data Integration, Data Transformation, Data Reduction; Feature Scaling

Regression: Linear regression, Gradient Decent Optimisation based Linear Regression, Multiple regression, Polynomial regression, Evaluation parameters.

Classification: Bayesian learning, Logistic Regression, Nearest Neighbours, Decision Tree Learning: Entropy and Gini Index based decision tree**,** Support vector machines, Naïve algorithm and Evaluation parameters. Ensemble Learning.

Clustering: K-Means, k-Medoids, Agglomerative versus Divisive Hierarchical Clustering Distance Measures in Algorithmic Methods, Mean-shift Clustering.

Artificial Neural Network: Neural network representation, Neural Networks as a paradigm for parallel processing, Linear discrimination, pairwise separation, Gradient Descent, Logistic discrimination, Perceptron, Training a perceptron, Multilayer perceptron, Back propagation Algorithm.

Association Rules Learning: Need and Application of Association Rules Learning, Basic concepts of Association Rule Mining, Apriori algorithm.

Laboratory Work: To implement various real time learning problems using machine learning models.

Recommended Books

1. Mitchell T.M., Machine Learning, McGraw Hill (1997).
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2010).
3. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag (2006).
4. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press (2009).

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Identify and analyse different types of machine learning algorithms and apply pre-processing techniques, such as data cleaning, feature selection, and feature scaling, to improve the performance of machine learning models.
CLO2	Understand and Implement linear regression, multiple regression and polynomial regression techniques along with evaluation metrics.
CLO3	Evaluate the performance metrics along with implementation of various classification techniques such as Bayesian learning, Logistic regression, decision tree, SVM.
CLO4	Understand and implement unsupervised learning approaches.
CLO5	Design, implement and evaluate an Artificial Neural Network using appropriate activation functions, optimization algorithms, and error metrics to solve a given problem."

MCA302: OBJECT-ORIENTED PROGRAMMING				
	L	T	P	Cr
	3	0	2	4.0

Course Objective: This course introduces the principles of Object-Oriented Programming. Students will learn how to design and implement reusable, modular, and scalable code using classes, objects, inheritance, polymorphism, encapsulation, and other OOPs concepts.

Objects and Classes: Structure in C and C++, Class specification, Objects, Namespaces, Overview of pillars of OOPS, Inline functions, Passing objects as arguments, Returning object from a function, Array of objects, Static keyword with data member, member function and object, Friend function, and Friend classes, Pointer to objects, this pointer, Dynamic memory allocation, const‘ keyword with data member, member function and object.

Constructor and Destructor: Constructors and its types, Constructor Overloading, Constructors in array of objects, Constructors with default arguments, Dynamic Constructor, Destructor.

Inheritance: Introduction to Inheritance, Forms of Inheritance (Single, Multiple, Multilevel, Hierarchical and Hybrid) with various modes (Public, Private and Protected), Inheritance with Constructor and Destructor.

Polymorphism: Classification of Polymorphism (Compile-time and Run-time), **Compile Time-**Function Overloading, Operator Overloading (Unary operator and Binary operator with member function and friend function), Data Conversion (Basic to user-defined, user- defined to basic, one user-defined to another user-defined). **Run-time-** Pointers to derived class object, Overriding member function, Virtual functions, pure virtual functions, Abstract class.

Exception Handling, Templates and Standard Template Library: Exception handling mechanism, Usage of template, Function templates, Overloading of Function templates, Class templates, Introduction to Standard Template Library and its components. Algorithms, Containers (Array, Vector, Stack) and Iterators.

Laboratory Work : To implement object Oriented constructs using the C++ programming language.

Recommended Books:

1. C++: The Complete Reference, Schildt H., Tata McGraw Hill, 4th Ed., 2003
2. C++Primer, Lippman B.S., Lajoie J., and MooE.B., Addison-Wesley Professional, 5th Ed., 2013.
3. Object-Oriented Programming in C++, Lafore R., Pearson Education, 4th Ed., 2002
4. Object Oriented Programming with C++, E Balagurusamy, 8th Ed.,2017
5. The C++programming language, Stroustrup B., Pearson Education India, 4th Ed., 2013

Course Learning Outcomes : (CLOs): On completion of the course, students will able to:

CLO1	To recall the knowledge of structure and its variables to comprehend the concept of classes, objects, constructors and destructors for implementing the object oriented paradigms.
CLO2	To apply and analyze the inheritance on real life case studies via coding competences.
CLO3	To design and develop code snippets for polymorphism to proclaim coding potential; and management of run-time exceptions.
CLO4	To assess and interpret the knowledge of templates to appraise the standard template libraries.

MCA109 : DATA SCIENCE FOUNDATION

L	T	P	Cr
3	0	2	4.0

Course Objective: To elaborate on the basics of data science and provide a foundation for understanding the challenges and applications.

Introduction to Python: Basic syntax, variables, Random Numbers, Functions.

Data Structures in Python: List, Tuple, Sets, Dictionary, Operations on Data Structures (Declarations, Iterations, Adding/deleting element, min/max/sorting, merge, select). Use of Libraries, File Handling (Read, Write, Merge, etc).

Advance Topics in Python: Working with Numpy, Working with Scipy.

Getting Started with Raw Data: Indexing and slicing, Shape manipulation, Empowering data analysis with pandas: The data structure of pandas, Series, DataFrame, Panel, Inserting and exporting data: CSV,XLS

Data cleansing: Checking the missing data, Filling the missing data, String operations, Merging data, Data operations; Aggregation operations: Joins, inner join, left outer join, full outer join, groupby function.

Application of Inferential Statistics: Various forms of distribution, A normal distribution, A normal distribution from a binomial distribution, A Poisson distribution, A Bernoulli distribution, A z-score, A p-value, Type 1 and Type 2 errors, A confidence interval, Correlation, Z-test vs T-test, The F distribution, The chi-square distribution, Chi-square for the goodness of fit, The chi-square test of independence, ANOVA.

Plotting and Visualization in Python: Plotting using Matplotlib and Seaborn library (Histogram, Box Plot, Scatter Plot, Bar Graphs, Line Graph, etc). Binning Visualization, organizing data and designing dashboards using Tableau.

Histogram: Display the distribution of school enrolments to highlight gaps in access to quality education, BoxPlot: Compare urban sustainability metrics across cities to identify disparities and Useful for global comparisons of education outcomes, Scatter Plot, Bar Graphs: Illustrate regional crime rates to inform peace and justice strategies, Line Graph, etc., Dashboard for Climate Change.

Data Analysis: Getting to know your data, Data Analysis Pipeline: Data pre-processing-Attribute values, Attribute transformation, Sampling, Dimensionality Reduction-PCA, Multidimensional Scaling, Non-linear Methods, Graph-based Semi-Supervised Learning, Representation Learning Feature subset selection, Distance and Similarity calculation.

Advances in Data Science: Basics of Correlation, Regression, Working with Pandas, Working Scikit-Learn, Feature Engineering. Emphasis will be placed on applying these tools in domains such as **environmental monitoring, smart cities, and sustainable development**.

Application Development: GUI and Database-based applications to support **sustainable and efficient decision-making systems**.

Laboratory Work: To implement general problems in Python; and develop database and web-based applications that can contribute to **sustainability goals**, such as resource optimization and smart solutions.

Recommended Books:

1. Madhavan, S., 2015. Mastering python for data science. Packt Publishing Ltd.
2. Joel Grus, Data Science from Scratch: First Principles with Python, (2nd Ed.), O'Reilly Media, 2019.
3. John M. Shea, Foundations of Data Science with Python, *(1st* Ed.), CRC Press, 2021.
4. Martin C. Brown, Python: The Complete Reference, McGraw Hill, 2018.
5. Allen B. Downey, Think Python, O'Reilly, 2016.
6. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data*, (2nd Ed.), O'Reilly Media, 2022.*

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	To manage, manipulate, clean, and analyze different types of data.
CLO2	To develop dashboards for real-time data sets.
CLO3	To visualize datasets using various techniques for better understanding.
CLO4	To understand data correlation, reduction, and summarization, aiding in sustainable data-driven strategies**.**
CLO5	To apply inferential statistics on real-time dataset.

MCA107 COMPUTER PROGRAMMING				
	L	T	P	Cr
	3	0	2	4.0

Course objective: This course is designed to explore computing and to show students the art of computer programming. Students will learn some of the design principles for writing good programs.

Computers Fundamentals: Classification of Computers, Application of Computers, Basic organization of computer, Input and Output Devices, Binary Number Systems, Computer memory, Computer Software, Role of computing in sustainable development, Green computing practices .

Algorithms and Programming Languages: Algorithm, Flowcharts, Pseudocode, Generation of Programming Languages.

Introduction to 'C' programming: Fundamentals, Structure of a 'C' program, Compilation and linking processes.

Expressions and Console I/O: Basic Data types, Identifier Names, Variables, Scope, Type qualifiers, Storage class specifiers, Constants, Operators, Reading and writing characters, Reading and writing strings, Formatted and console I/O, printf(), scanf(), Suppressing input.

Statements: True and False in C, Selection statements, Iteration statements, Jump statements, Expression statements, Block statements.

Arrays and Strings: Single dimension array, Two-dimensional array, Strings, Array of strings, Multi-dimensional array, Array, Variable length arrays.

Pointers: Pointer variables, Pointer operators, Pointer expressions, Pointers and arrays, Multiple indirections, Pointer initialization, Pointers to arrays, Dynamically allocated arrays, Problems with pointers.

Functions: General form of a function, Understanding the scope of a function, Function arguments, Command line arguments, Return statement, Recursion, Function prototype, Pointers to functions, Optimized pointer usage.

Structures, Unions, Enumerations, and Typedef: Structures, Array of structures, Passing structures to functions, Structure pointers, Arrays and structures within structures, Unions, Bit-fields, Enumerations, typedef.

File I/O: Streams and files, File system basics, fread() and fwrite(), fseek() and random access I/O, fprintf() and fscanf(), Standard streams.

Pre-processor and Comments: Pre-processor, #define, #error, #include, Conditional compilation directives, #undef, Single line and multiple line comments.

Laboratory work: To implement Programs for various kinds of programming constructs in C Language.

Recommended Books:

1. Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, Prentice Hall(1988), 2nd Ed.
2. Schildt H., *C:* The Complete Reference, Tata Mcgraw Hill, (2000), 4th Ed.
3. Kanetkar Y., Let Us C, BPB Publications(2012),13th Ed.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Learn the implementation of simple 'C' program, data types and operators and Console I/O function.
CLO2	Learn the Implementation of decision control statements, loop control statements and case control structures.
CLO3	Understand the declaration and implementation of arrays, pointers and functions.
CLO4	Learn the structures declaration, initialization and implementation.
CLO5	Understand the file operations, Character I/O, String I/O, File pointers and importance of pre-processor directives

MCA201 THEORY OF COMPUTATION				
	L	T	P	Cr
	3	1	0	3.5

Course Objective: This course introduces the basic theory of computer science and formal methods of computation. The course exposes students to the computability theory, as well as to the complexity theory.

Regular Languages: Alphabets, Language, Regular Expression, Definitions of Finite State Machine, Transition Graphs, Deterministic & Non-deterministic Finite State Machines, Regular Grammar, Thompson's Construction to Convert Regular Expression to NDFA & Subset Algorithm to convert NDFA to DFA, Various recent development in the Conversion of Regular Expression to NFA, Minimization of DFA, Finite State Machine with output-Moore machine and Melay Machine, Conversion of Moore machine to Melay Machine & Vice-Versa.

Properties of Regular languages: Conversion of DFA to Regular Expression, Pumping Lemma, Properties and Limitations of Finite state machine, Decision properties of Regular Languages, Application of Finite Automata.

Context Free Grammar and Push Down Automata: Context Free Grammar, Derivation tree and Ambiguity, Application of Context free Grammars, Chomsky and Greibach Normal form, Properties of context free grammar, CKY Algorithm, Decidable properties of Context free Grammar, Pumping Lemma for Context free grammar, Push down Stack Machine, Design of Deterministic and Non-deterministic Push-down stack.

Turing Machine: Turing machine definition and design of Turing Machine, Church-Turing Thesis, Variations of Turing Machines, combining Turing machine, Universal Turing Machine, Post Machine, Chomsky Hierarchy, Post correspondence problem.

Uncomputability: Halting Problem, Turing enumerability, Turing Acceptability and Turing decidabilities, unsolvable problems about Turing machines, Rice's theorem.

Recommended Books:

1. Hopcroft J. E., Ullman J. D. and Motwani R., Introduction to Automata Theory, Languages and Computation, Pearson Education, 3rd Edition, (2006).
2. John C. Martin, Introduction to Languages and the Theory of Computation, McGraw-Hill Higher Education (2011).

3. Daniel A. Cohen, Introduction to Computer Theory, John Wiley and Sons, 2nd Ed., 1996
4. Michael Sipser, Introduction to the Theory of Computation, Thomson, 2nd Ed, 2007

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Comprehend regular languages and finite automata and develop ability to provide the equivalence between regular expressions, NFAs, and DFAs.
CLO2	Disambiguate context-free grammars by understanding the concepts of context-free languages and push-down automata.
CLO3	Apply the concepts of recursive and recursively enumerable languages and design efficient Turing Machines.
CLO4	Solve analytical problems in related areas of theory in computer science

MCA202 SOFTWARE ENGINEERING				
	L	T	P	Cr
	3	0	2	4.0

Course Objectives: To apply principles of software engineering for developing commercially viable software and to learn emerging trends in software engineering.

Introduction: Introduction to Software Engineering, importance of Software, The Software Evolution, Software Characteristics, Software Applications, Software Crisis: Problem and Causes.

Software Process: Software process models (Waterfall, Incremental, and Evolutionary process models, Overview of Agile Process.

Requirements Engineering: Problem Analysis, Requirement elicitation and Validation, Requirements modeling: Scenarios, Information and analysis classes, flow and behavioural modeling, documenting Software Requirement Specification (SRS), DFD, Data Dictionary

Software Design: System design principles: levels of abstraction (architectural and detailed design), separation of concerns, information hiding, Structured design (top-down functional decomposition), object-oriented design, data-structured centered, function oriented, service oriented, Architectural Design– Data Flow Style, MVC architecture, Call and Return Style architecture, Data Centered Style, Client – Server architecture, 2 – tier and 3- tier architecture, Pipe and Filter Architecture, Introduction to API architecture: REST API, SOAP API. Understanding REST API and SOAP architecture works, HTTP methods GET, POST, PUT, DELETE, PATCH, OPTIONS, HEAD, API testing, Microservices, Difference between API and Microservices, Domain-driven design (DDD), Service decomposition strategies, API design for microservices (REST, gRPC).

Software Construction: Coding Practices: Techniques, Integration Strategies, Internal Documentation, Verification.

Software Verification and Validation: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Design, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit Testing, Integration Testing, Top Down and Bottom- U p Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Debugging.

Agile Project Management and Sustainability: 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.

Laboratory Work: Implementation of Software Engineering concepts using tools like Rational Suite etc. Software Suit through projects.

Recommended Books:

1. Pressman R., Software Engineering, A Practitioner ‘s Approach, McGraw Hill International, 7th Ed. (2010).
2. Sommerville I., Software Engineering, Addison-Wesley Publishing, (2010) 9th Ed.
3. Jalote P., An integrated Approach to Software Engineering, Narosa (2005).
4. Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide, 2nd Ed. (2005).

Course Learning Outcomes (CLOs): On completion of the course, students will be able to

CLO1	Analyze software development process models for software development life cycle.
CLO2	Comprehend the use of agile development methodologies including UI sketching, user stories.
CLO3	Elicit, describe, and evaluate a system's requirements and analyze them using various UML models.
CLO4	Demonstrate the use of design principles in designing data, and architectural design.
CLO5	Test the system by planning appropriate test cases and applying relevant test strategies.

MCA204 ARTIFICIAL INTELLIGENCE

L	T	P	Cr
3	0	2	4.0

Fundamental Issues: Overview of AI problems, AI applications, Intelligent behaviour, Turing test, Approaches of AI.

Intelligent agents: reactive, deliberative, goal-driven, utility-driven, and learning agents.

Problem-solving through Search: Problem spaces (states, goals and operators), Sustainable Problem Spaces & Modeling, blind, heuristic, Sustainable Heuristics and Evaluation Functions, problem-reduction, A*, AO*, minimax, constraint propagation, stochastic, and evolutionary search algorithms, Case study on sustainable algorithm.

Knowledge Representation: Propositional and predicate logic, Resolution in predicate logic, Question answering, Theorem proving, Semantic networks, Frames and scripts, conceptual graphs, conceptual dependencies.

Planning: Planning as search, partial order planning, construction and use of planning graphs, Planning as Search with Sustainability Objectives, benefits of sustainable AI planning.

Reasoning with Uncertain Knowledge: Probability, Bayes rule, Bayesian networks, probabilistic inference, Markov Networks, Hidden Markov Models. Decision-Making: basics of utility theory, decision theory, sample applications.

Expert Systems: Architecture of an expert system, existing expert systems: MYCIN, RI. Expert system shells.

Laboratory work: Programming in C/C++/Python/R: programs for Search algorithms-Depth first, breadth first, best first, hill climbing, Implementation of games: 8-puzzle, Tic-tac-toe using heuristic search, designing an expert system using logic in Prolog, Implementing an intelligent agent.

Recommended Books:

1. Rich E., Artificial Intelligence, Tata McGraw Hills (2009) 3rd Ed.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia (2009) 6th Ed.
3. Patterson D.W, Introduction to AI and Expert Systems, Mc GrawHill 1998, 1st ed.
4. Russel S., Norvig P., Artificial Intelligence: A Modern approach, prentice Hall 2014, 3rd Edition.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Learn and understand applications of artificial intelligence and categorize various problem domains and search methods
CLO2	Analyze basic and advanced search techniques including game playing, evolutionary search algorithms, and constraint satisfaction.
CLO3	Learn knowledge representation using various forms in artificial intelligence and understand the importance of probability in knowledge representation for reasoning under uncertainty.
CLO4	Learn to represent knowledge using Bayesian networks and drawing Hidden Markov Models.
CLO5	Design and implement an expert system.

MCA205 DESIGN AND ANALYSIS OF ALGORITHMS				
	L	T	P	Cr
	3	0	2	4.0

Course Objectives: The objective of course is to provide an understanding of various techniques/methods such as Greedy, Dynamic Programming, Branch and Bound and Backtracking. It covers good principles of algorithm design, analysis of algorithms, and advanced data structures.

Introduction: Algorithm Definition and Characteristics of Algorithm. Analysing algorithms, Asymptotic analysis of Complexity Bounds – Best, Average and Worst-Case behaviour, Performance Measurements of Algorithm, Time and Space Trade-Offs, Analysis of Recursive Algorithms through Recurrence Relations: Substitution Method, Recursion Tree Method and Masters' Theorem.

Fundamental Algorithmic Strategies: Various design techniques are covered such as:

Divide and Conquer: General method, binary search, merge sort, quick sort.

Greedy: General method, activity selection, job sequencing, fractional knapsack problem, Coin Changing problem.

Dynamic Programming: General method, matrix multiplication, 0/1 knapsack, optimal binary search tree, longest common subsequence.

Backtracking: General method, N queen problem, sum of subsets, graph coloring.

Branch and Bound: General Method. 0/1 Knapsack, Travelling Salesman Problem

Graph and Tree Algorithms: Analysis of Graph Traversal algorithms - Depth First Search (DFS) and Breadth First Search (BFS); Analysis of - shortest path algorithms, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

String Matching Algorithms: Introduction to string matching, Rabin-Karp, Knuth-Morris-Pratt, Boyer Moore algorithm.

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Solutions for some NP complete problems using Approximation, Randomized, Online, and Genetic Algorithms.

Laboratory Work: Implementation of Different Algorithms based on various algorithmic strategies using C/C++.

Recommended Books:

1. Anany Levitin, ―Introduction to the Design and Analysis of Algorithms‖, Third Edition, Pearson Education, 2012.

2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Computer Algorithms/ C++, Second Edition, Universities Press, 2007.
3. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, ―Data Structures and Algorithms‖, Pearson Education, Reprint 2006.
4. Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, The MIT Press (2009) 3rd Edition.
5. Sedgewick & Wayne, Algorithms, Addison-Wesley Professional (2011), 4th Edition.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Analyze the complexity of algorithms, to provide justification for the selection, and to implement the algorithm in a particular context.
CLO2	Apply various algorithmic design paradigms such as greedy, dynamic, backtracking etc. to solve common computing problems.
CLO3	Identify basic properties of graphs and apply their algorithms to solve real life problems.

MCA108 COMPUTER NETWORKS				
	L	T	P	Cr
	2	1	2	3.5

Course Objective: The subject will introduce the fundamentals of computer networks, related concepts and theories.

Introduction: Introduction to Computer Networks, Components of Computer Network, Types of Communication Modes, Network Topologies, Classification of networks: LANs, WANs and MAN, Networking Devices.

Reference Models: The OSI Reference Model, The TCP/IP Reference Model, Design Issues for the Layers, Connection-Oriented and Connectionless Services.

Local Area Networks: Ethernet, IEEE standards 802.3, 802.5. Wireless LANs: IEEE 802.11 and Bluetooth.

Physical Layer: Analog & Digital Signals, Transmission Impairment, Data Rate limits, Performance, Digital conversion (Only Line coding: Polar, Bipolar and Manchester coding), Analog-to-digital conversion (only PCM), Transmission Modes, Digital-to-Analog conversion.

Data Link Layer: Error Detection & Correction: Introduction, Block coding, Cyclic codes, Checksum, Framing, Flow and Error Control Protocols, Noiseless Channels, Noisy channels.

Network Layer: Network Layer Design issues, Routing algorithms, Congestion Control Algorithms, Quality of Service, Internetworking and The Network Layer in the Internet, Unicast and Multicast Routing. Distance-Vector Routing, Link-State Routing, Shortest path computation, Dijkstra's algorithm, Network Layer Protocols (IP, ICMP), IP addressing, IPv4, IPV6, Address binding with ARP.

The Transport Layer: Process-to-Process Delivery, The Transport Service, Elements of Transport Protocols, Congestion Control, The Internet Transport Protocol: UDP, The Internet Transport Protocol – TCP.

The application Layer: Domain Name Space, Domain Resource Records, Domain Name Servers. Electronic mail: SMTP, The World Wide Web: Static and dynamic web pages, web applications, HTTP.

Self-Learning Contents:

Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.), Distributed applications (client/server, peer-to-peer, cloud, etc.), HTTP, Electronic mail, File transfer, Telnet.

Laboratory Work: Designing conceptual networks using E-Draw/Dia etc. and implementing topologies such as Bus, Ring, Star, Mesh and configuring Various Routing algorithms using Packet tracer or GNS3 platform.

Recommended Books

1. Forouzan, B.A., Data communication and Networking, McGraw Hill, (2006), 4th Ed.
2. Tanenbaum, A.S., Computer Networks, Prentice Hall, (2010), 5th Ed.
3. Kurose and Ross, Computer Networking: A Top Down Approach, Addison-Wesley, (2012), 6th Ed.
4. Stallings, W., Computer Networking with Internet Protocols and Tech, Prentice Hall of India (2010), 9th Ed.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Understand the computer networks and types of communication modes.
CLO2	Understand the concept of data communication, error detection and correction, access and flow control.
CLO3	Demonstrate the operation of various routing protocols and their performance analysis.
CLO4	Illustrate design and implementation of datalink, transport and network layer protocols within a simulated/real networking environment.

MCA203 BIG DATA ANALYTICS AND BUSINESS INTELLIGENCE				
	L	**T**	**P**	**Cr**
	3	**0**	**2**	**4.0**

Course Objective: To have a basic understanding of most recent advancements in Big Data and using insights, statistical models, visualization techniques for its effective application in Business Intelligence.

Introduction to Data Analytics: Data and its types, Data Visualization, Correlation, Regression: Univariate, Multivariat**e** for Environmental and Social Impact Data, Classification and Clustering to identify Patterns.

Big Data Technology Landscape: Fundamentals of Big Data Types, Big data Technology Components for Environmental and Social Impact Analysis, Big Data Architecture, Big Data Warehouses, Functional vs. Procedural Programming Models for Big Data for Developing Sustainable Data Solutions.

Introduction to Business Intelligence: Business View of IT Applications, Digital Data OLTP vs. OLAP, Why, What and How BI?, BI Framework and Components, BI Project Life Cycle – using a business Case Study, Business Intelligence vs. Business Analytics in Driving Efficient Business Practices.

Big Data Analytics: Big Data Analytics, Role of Data analyst, Framework for Big Data Analysis to Address Cost Effective Challenges, Predictive, Descriptive, and Prescriptive Approaches for Big Data Analysis, Resource Management, and Energy Efficiency, Security and Governance for Big Data, Big Data on Cloud for Scalable Solutions and Real-time Monitoring.

Business implementation of Big Data: Big Data Implementation for Sustainable Solutions, ETL in Big Data, Introduction to Hadoop Ecosystem, HDFS, Map-Reduce Programming, Big Data workflow, Operational Databases, Graph Databases in a Big Data Environment, Real-Time Data Streams and Complex Event Processing. Latest trends in Big Data, Challenges Related to Computational cost, Resource Optimization in Storing and Processing Large Datasets.

Laboratory Work: Introduction to most recent advancements in Big Data technology along with their usage and implementation with relevant tools and technologies.

Recommended books:

1. Michael Minelli, Michele Chambers, Ambiga Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley CIO Series (2013), 1st Ed.
2. T.white, Hadoop: The Definitive Guide, O' Reilly Media (2012), 3rd Ed.
3. Kimball, Ralph, Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data, Wiley India (2004), Paperback.
4. Ralph Kimball, Margy Ross, The Data Warehouse Toolkit, The Complete Guide to Dimensional Modeling, John Wiley & Sons, Inc. (2002), Second Edition.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to:

CLO1	Translate a business challenge into an analytics challenge and deploy a structured lifecycle approach to data science and big data analytics projects.
CLO2	Basic understanding of the usage of Big Data in present World.
CLO3	Analyse big data, create statistical models, and identify insights that can lead to actionable results.
CLO4	Select visualization techniques, communicate analytic insights to business sponsors, and others and explain how advanced analytics can be leveraged to create competitive advantage.

MCA303: IT PROJECT MANAGEMENT

L	T	P	Cr
3	0	2	4.0

Course objective: Learn and Explore SPM activities through knowledge of IT project management and planning.

Introduction to IT Project Management: The characteristics of IT projects, Objectives of project management: time, cost and quality, Basics of Project Management, Stakeholders, Stages of Project, The Feasibility Study, Cost-benefit Analysis, Planning, Project Execution, Project and Product Life Cycles, Project Management Knowledge areas, Project Management Tools & Techniques, Project success factors, role of project manager.

Project Management and Planning: System view of project management, Understanding organizations, stakeholder's management, project phases and project's life cycles. Introduction to Agile software, Why planning is necessary, Iterative steps for planning, Project Plan documentation methods, Software Requirement Specification. Agile Project Management. Agile impacts all knowledge areas after Agile project management.

Measurement and Control: Measurements for project monitoring, what and when to measure, Plan versus Control, managing the plan, The Deadline Effect. Reviews, feedback and reporting mechanisms, revisiting the plan.

Project Scope Management: Scope Planning & Scope management plans, Function point calculation, Scope definitions & project scope statement, Work Breakdown Structure (WBS), WBS dictionary, scope verification, scope control.

Time Management: Project time management, activities sequencing, network diagrams, activity resource estimation, activity duration estimation, schedule development, Gantt Charts, Critical path method, Programme evaluation & review technique (PERT) and CPM, concept of slack time, schedule control.

Project Cost management: Basis principles of cost management, Cost estimating, type of cost estimate, cost estimate tools & techniques, COCOMO, Putnam/ SLIM model Estimating by Analogy, cost budgeting, cost control, earned value management, project portfolio management

Project Quality Management: Quality Planning, quality Assurance, Quality control, Tool & techniques for quality control, Pareto Analysis, Six Sigma, CMM, ISO Standards, Juran Methodology

Project Human Resource Management: Human resource planning, project organizational charts, responsibility assignment metrics, acquiring project team, resource assignment, resource loading, resource levelling, Different team structures developing project teams.

Project Communication Management: Communication Planning, Performance reporting, managing stakeholders, improving project communication

Project risk management: Risk Management planning, common sources of risk, risk identification, risk register, qualitative risk analysis, using probability impact matrixes, expert judgement, qualitative risk analysis, decision trees & expected monetary value, simulation, sensitivity analysis, risk response planning, risk monitoring & control.

Project procurement management: Procurement management plans, contract statement of work, planning contracts, requesting seller responses, selecting sellers, administrating the contract, closing the contract

Software Configuration Management: Why versions exist, why retain versions, SCI, Releases vs. version. Change Control and Management.

Laboratory work: Using Function Point calculation tools for estimation, comparing with COCOMO estimates, Implementation of various exercises using PERT, CPM methods, Preparing schedule, resource allocation etc. using MS Project or Fissure. sim or VENSIM can also be used, preparing an RMMM Plan for a case study, Preparing Project Plan for a for Lab Project or case study. Exploring about PMBOK (Project Management Body of Knowledge) and SWEBOK (Software Engineering Body of Knowledge) from related website, Implementation of software project management concepts using related tools and technologies.

Recommended Books:

1. Hughes B. and Cotterell M. and Mall R., Software Project Management, Tata McGraw Hill (2011) 5th Ed.
2. Pressman R., A practitioner's Guide to Software Engineering, Tata McGraw Hill (2014) 7th Ed.
3. Stellman A., Greene J., Applied Software Project Management, O'Reilly Media, Inc. (2008).
4. Futrell T. R., Shafer F. D. and Shafer I. L., Quality Software Project Management, Prentice Hall (2002).

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO 1	Describe and apply basic concepts related to IT project planning, scope and feasibility.
CLO 2	Analyze IT project estimation techniques.
CLO 3	Comprehend the concept of team structure and project communication management.
CLO 4	Acquire knowledge about quality assurance, quality control, and risk management.
CLO 5	Describe various project management activities such as tracking, project procurement, configuration management, monitoring.

MCA312: IMAGE AND VIDEO PROCESSING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course introduces students to the fundamentals of digital imaging. It provides exposure to image enhancement techniques, image degradation and restoration methods and various image segmentation techniques. Students will also learn about fundamentals of video processing and motion estimation.

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

Image enhancement in the spatial domain: Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

Image Enhancement Frequency (Transform) domain: Introduction to frequency (transform) domain, DFT (Discrete Fourier Transform), Properties of 2-D DFT; Image enhancement: Smoothing and Sharpening: Low pass and High pass filtering.

Image restoration: A model of the image degradation/restoration process, noise models, and restoration in the presence of noise–only, spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms.

Morphological Image Processing: Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms.

Image Segmentation: Morphological operation, Point-Line-Edge Detection; Thresholding; Segmentation by Region Growing; Segmentation by Region Splitting and Merging; Region Segmentation by Clustering; Region Segmentation by Graph-Cut; Texture Segmentation.

Application Areas of Image Processing: Image compression: JPEG compression; Huffman coding. Image security: Watermarking, Steganography, Visual Cryptography.

Object Detection and Classification: Introduction to Neural Network, Convolutional Neural Networks (CNNs), Overview on use cases of CNNs in Image Processing.

Video 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.

Laboratory work: Demonstrate the use of Image Processing Toolbox on MATLAB/PYTHON to create interactive image processing applications like image enhancement, image compression, image segmentation, feature extraction etc.

Recommended Books:

1. Gonzalez C. R., Woods E. R., Digital Image Processing, Pearson Education, 4th Ed.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989.
3. McAndrew A., Introduction to Digital Image Processing with Matlab, Thomson Course Technology (2004)
4. Yao Wang, Jorn Ostermann and Ya Qin Zhang, Video Processing and Communications, Prentice Hall, 2002
5. Low A., Introductory Computer Vision and Image Processing, McGraw-Hill (1991), 1st Ed.
6. M. Tekalp ,"Digital video Processing", Prentice Hall International.

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Comprehend the need and usage of concepts of image processing
CLO2	Enhance the visual quality of given grey/color image using well known transformations and filters.
CLO3	Apply and comprehend the role of feature extraction in Image Processing.
CLO4	Demonstrate the use of image processing techniques to ideate innovative solutions to real-world problems.
CLO5	Understand the fundamentals of video processing.

MCA405 ETHICAL HACKING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course is designed to impart a critical and theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools and the services related to Ethical Hacking.

Introduction: Understanding the importance of security, the Concept of ethical hacking, and essential Terminologies-Threat, Attack, Vulnerabilities, Target of Evaluation, Exploit. Phases involved in hacking.

Footprinting: Introduction to footprinting, Understanding the information gathering methodology of the hackers, and Tools used for the reconnaissance phase.

Scanning: Detecting live systems-on the target network, - Discovering services running listening on target systems, understanding port scanning techniques, Identifying TCP and LIDP services running on the target network, and Understanding active and passive fingerprinting.

System-Hacking: Understanding Sniffers, Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing.

Session Hijacking: Understanding Session Hijacking, Phases involved in Session Hijacking, Types of Session Hijacking, and Session Hijacking Tools.

Hacking Wireless Networks: Introduction to 802.11, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Wireless DOS attacks, WLAN Scanners, WLAN Sniffers, Hacking Tools, Securing Wireless Networks.

Cryptography: Symmetric and Asymmetric Cryptography, Classical Encryption techniques, Substitution techniques, Block Ciphers Principles, Fiestel Structure, DES, Double and Triple DES, AES, Public Key Cryptography, RSA, Diffie-Hellman Key Exchange, Cryptographic Hash Functions, and Digital Signatures.

Laboratory Work:

Lab Exercises including using scanning tools like Nmap, Wireshark IPEYE, IPsecScan, SuperScan etc. and Hacking Tools like Trinoo, TFN2K, Zombic, Zapper OpenSSL for Digital Certificate generation etc.

Recommended Books

1. Simpson T. M., Backman K., Corley J., Hands-On Ethical Hacking and Network Defense, Delmar Cengage Learning (2011) 2nd Edition.
2. Fadia A. and Zacharia M., Network intrusion alert: an ethical hacking guide to intrusion detection, Boston, MA: Thomas Course Technology 3rd Edition (2008).
3. Cryptography And Network Security, William Stallings, 3rd Edition, Pearson Publications.
4. Mathew T., Ethical Hacking, OSB Publication (2003). 2nd Edition
5. McClure S., Scambray J. and Kurtz G., Hacking Exposed 7: Network Security Secrets and Solutions, McGrawHill (2012) 7th Edition.

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Understand the different phases involved in hacking.
CLO2	Utilize the scanning tools used for the information gathering.
CLO3	Recognize the phases in session hijacking and use the tools for counter-measuring the various sniffing attacks.
CLO4	Analyse different types of attacks on the wireless networks.
CLO5	Describe and apply different types of cryptographic algorithms for securing the data.

MCA406: SOCIAL NETWORK ANALYSIS				
	L	T	P	Cr
	2	0	2	3.0

Course Objectives: To enable students to put Social Network Analysis projects into action in a planned, informed and efficient manner.

Introduction to Social Networks: Types of Networks: General Random Networks, Small World Networks, Scale-Free Networks; Examples of Information Networks; Static Unweighted and weighted Graphs, Dynamic Unweighted and weighted Graphs, Network Centrality Measures; Strong and Weak ties.

Walks: Random walk-based proximity measures, Other graph-based proximity measures. Clustering with random-walk based measures, Algorithms for Hitting and Commute, Algorithms for Computing Personalized Pagerank and Sim- rank.

Community Detection: Basic concepts, Algorithms for Community Detection: Quality Functions, The Kernighan-Lin algorithm, Agglomerative/Divisive algorithms, Spectral Algorithms, Multi-level Graph partitioning, Markov Clustering; Community Discovery in Directed Networks, Community Discovery in Dynamic Networks, Community Discovery in Heterogeneous Networks, Evolution of Community, Education Networks, Knowledge flow, resources and Innovations.

Link Prediction: Feature based Link Prediction, Bayesian Probabilistic Models, Probabilistic Relational Models, Linear

Event Detection: Classification of Text Streams, Event Detection and Tracking: Bag of Words, Temporal, location, ontology based algorithms. Evolution Analysis in Text Streams, Sentiment analysis.

Social Influence Analysis: Influence measures, Social Similarity - Measuring Influence, Influencing actions and interactions. Homophily, Influence maximization.

Laboratory work:

Implementation of various concepts taught in the course using Python/R Programming

Recommended Books:

1. Social Network Analysis, Tanmoy Chakraborty, Wiley, 2021
2. Charu C. Aggarwal, Social Network Data Analytics, Springer; 2011.
3. S.Wasserman, K.Faust: Social Network Analysis: Methods and Applications, Cambridge Univ Press, 1994
4. Scott, J. (2007). Social network analysis: A handbook (2nd Ed.). Newbury Park, CA: Sage.
5. Knoke (2008). Social Network Analysis, (2nd Ed). Sage.

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Formalize different types of entities and relationships as nodes and edges and represent this information as relational data.
CLO2	Plan and execute network analytical computations
CLO3	Use advanced network analysis software to generate visualizations and perform empirical investigations of network data.
CLO4	Interpret and synthesize the meaning of the results with respect to a question, goal, or task.
CLO5	Collect network data in different ways and from different sources while adhering to legal standards and ethics standards

Elective Courses

MCA 212 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: Artificial Neural Networks- Biological vs Artificial Neuron, Mathematical Model of a Neuron, Perceptron, Perceptron Learning Algorithm & Limitations of Perceptron.

Activation Function: - Threshold Logic Unit, Step Function, Linear, Nonlinear activation functions- Sigmoid, Tanh, and ReLU Softmax.

Multilayer Perceptron: - Multi-Layer Perceptron, Optimization Approaches -Gradient Decent & Momentum based & Backpropagation Algorithm.

Convolutional Neural Networks: - Basic Concepts of Convolutional Neural Networks. Convolution and Pooling Operation, Regularization, Dropout, Batch-Norm & Introduction to Transfer Learning Models. Application of CNN in medical image analysis.

Recurrent Neural Networks: Recurrent Architecture, BPTT, Vanishing and Exploding Gradients, GRU & LSTM. Time series forecasting in real life applications.

Autoencoders: Autoencoder and its Relation to PCA, Stack Autoencoders, Denoising Autoencoders Variational Autoencoders & GANs.

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

Recommended Books:

1. Ian Goodfellow and Yoshua Bengio and Aaron Courville,"Deep Learning", MIT Press, 2016
2. Michael Nielsen, "Neural Network and Deep Learning", Online Book 2016

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Describe the Feed-forward and Deep networks
CLO2	Design single and multi-layer feed-forward deep networks.
CLO3	Implement Deep Neural Networks to solve real life problems.
CLO4	Analyse the performance of Deep Networks.

MCA208: CRYPTOGRAPHY AND NETWORK SECURITY				
	L	T	P	Cr
	3	0	2	4.0

Course Objectives: This course is designed to impart a critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools.

Introduction: Security Trends, Possible Types of Attacks, Model for Network Security, active and passive fingerprinting, Introduction to Protocols, Plain Text and Cipher Text, Encryption and Decryption, Communication using Symmetric Cryptography, Substitution Ciphers and Transposition Cipher, Block cipher, Stream cipher.

Cryptography: Modes of operation, Symmetric and Asymmetric cryptography, Classical Encryption techniques, Substitution techniques, Block Ciphers Principles, Fiestel Structure, DES, Double and Triple DES, AES, Public Key Cryptography, RSA, Diffie-Hellman Key Exchange, Cryptographic Hash Functions, and Digital Signatures, Digital Signatures with Encryption, Random and Pseudo Random Sequence Generation, Energy efficient cryptographic algorithms.

Network Security: Essential Terminology, Information Gathering Methodology, Active and Passive reconnaissance, Packet sniffing and spoofing, comprehend active and passive sniffing, Session Hijacking and its types, Spoofing vs Hijacking, Attacks on TCP protocol, SYN flood, TCP reset attack, DNS poisoning, Authoritative replies, ARP spoofing and poisoning. Scanning, Elaboration phase, active scanning, DNS Zone transfer, port scanning techniques, Identifying TCP and UDP services running on the target network, DoS (Denial of Service) attacks, Wireless DoS attacks, Role of WEP, Cracking WEP Keys, WLAN scanners and sniffers, securing wireless networks, Case studies of how secure networks prevent cyber-crimes and protect institutions.

Laboratory work: To learn and experiment different types of scenarios related to Security of an organization. To implement core algorithms in high level language and to use related tools and technologies for various security concepts above.

Recommended books:

1. Hackers Beware, Eric Core, EC-Council Press, 2003
2. Network Security Essentials, William Stallings, Prentice Hall, 5th Edition, 2013

3. Firewalls and Internet Security, William R. Cheswick and Steven M. Bellovin, Addison-Wesley Professional, 2nd Edition, 2003.
4. Cryptography and Network Security, W. Stallings , Prentice Hall, 5th Edition, 2010

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Identify possible types of attacks on the networks and study different types of the algorithms for securing the data.
CLO2	Apply basic rules of public key and symmetric and asymmetric cryptography for practical cryptographic problems.
CLO3	Recognize the phases in session hijacking and use of tools for counter-measuring various Sniffing attacks.
CLO4	Analyzing attacks on Wireless Networks and use of tools for securing them.

MCA209 SOURCE CODE MANAGEMENT				
	L	T	P	Cr
	3	0	2	4.0

Course Objective: The objective of the course is to understand the ways to bridge software development and IT operations using DevOps. It also helps to various approaches manage source code in the DevOps pipeline.

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

Definition and Purpose of DevOps: Introduction to DevOps, DevOps and Agile, Minimum Viable Product, Application Deployment, Introduction to continuous integration and deployment, Version control system. CAMS (Culture, Automation, Measurement and Sharing, Test-Driven Development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness, Organizational Learning.

Cloud Sustainability: Choosing eco-friendly cloud services (e.g., AWS, Azure, or GCP with carbon-neutral data centers) aligns with DevOps goals and sustainability.

Source Code Management History and Overview: History - Linux and Git by Linus Torvalds, Examples - SVN, Mercury and Git.

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.

Laboratory work: **Laboratory work**:

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

Recommended Books:

1. The DevOps Handbook - Book by Gene Kim, Jez Humble, Patrick Debois, and Willis Willis
2. Source Code Management – Volume 1, Pearson and Xebia Press
3. Pro Git – Book by Scott Chacon and Ben Straub
4. What is DevOps? - by Mike Loukides
5. "Git for Teams: A User-Centered Approach to Creating Efficient Workflows in Git"- Book by Emma Jane Hogbin Westby, published by O'Reilly Media, 2015

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Identify the need for migrating from traditional software development to Agile model and then to DevOps
CLO2	Define and understand the basic principles and need of DevOps and Continuous deployment
CLO3	Understand the history and overview of Source Code Management, along with real-time examples
CLO4	Demonstrate & identify the use of various centralized and distributed version control systems and its basic operations.

MCA213: CLOUD COMPUTING				
	L	T	P	Cr
	3	0	2	4.0

Course Objectives: To learn the concepts of cloud infrastructure and services in addition to its implementation for assessment of understanding the course by the students

Introduction and Evolution of Computing Paradigms: General Benefits and Architecture, Business Drivers, Main players in the Field, Overview of Existing Hosting Platforms and their architecture, Cluster Computing, Grid Computing, Services: Infrastructure as a Service, Platform as a Service, Software as a Service, Storage as a Service, Database as a Service, Security as a Service, Deployment Models: Public, Private, Hybrid and Community Clouds. The impact of cloud computing on conventional and distance education systems.

Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualizations, Migration of processes, Classic Data Center, Virtualized Data Center (Compute, Storage, Networking, and Application), Business Continuity in VDC. VMware vCloud – IaaS, Network virtualization through Software Defined Networks. Data centre energy consumption; Renewable energy in powering data centres; Green data centres design, cooling, and energy management; Virtualization and server consolidation.

Cloud Computing Security: Cloud Computing Security Issues and Challenges, Security and Privacy in Cloud Computing, Security Management in Cloud Computing, Security in Public Clouds and Private Clouds.

Cloud Computing Management and Monitoring: Cloud Computing Management and Monitoring, Service Level Agreements, Cloud Computing Maintenance and Performance Management.

Classification of Cloud Implementations: Key Amazon offerings-Amazon Web Services, The Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queuing Services (SQS), Bundling Amazon instances, AWS Identity Management and Security in the Cloud, Messaging in the Cloud, RESTFul Web Services, Overview of Google AppEngine - PaaS, Windows Azure.

Cloud-based Data Storage: Introduction to Hadoop, Hadoop Ecosystem (Pig, Hive, Cassandra, and Spark), Introduction No-SQL databases, Map- Reduce framework for

Simplified data processing on large clusters using Hadoop, Data Replication, Shared access to data stores.

Related Technologies: Introduction to Fog Computing and Edge Computing, Usage of Cloud for IoT and Big Data Analytics.

Role of Cloud, edge and fog computing in Sustainability: cloud sustainability and its importance, Global regulations and standards related to cloud sustainability, Carbon footprint measurement and reduction in cloud computing, green software principles to develop energy-efficient applications.

Laboratory work: To implement Cloud, Apache, and basics of Hadoop framework, an open-source implementation of MapReduce, and its Java API, Hadoop Distributed File System (HDFS). Implementation of RESTFul Web Services. To understand various concepts about virtualization and data storage. Setting up a cloud environment on a local server and deploying virtual machines, implementing storage virtualization, and testing storage virtualization techniques.

Recommended Books :

1. Buyya K, R., Broberg J. and Goscinski M. A., Cloud Computing: Principles and paradigms, MIT Press (2011) 4th Ed.
2. Kai Hwang, Geoffrey Fox and Jack Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann (2012) 2nd Ed.
3. Miller M., Cloud Computing, Que Publishing (2008) 1st Ed.
4. Puttini R. and Mahmood Z., Cloud Computing: Concepts, Technology & Architecture, Service Tech press (2013) 1st Ed.
5. Velte A., Velte T., and Elsenpeter R., Cloud Computing: A practical Approach, Tata Mc GrawHill (2009) 1st Ed.
6. Hurwitz J., Bllor R., Kaufman M. and Halper F., Cloud Computing for dummies (2009) 1st Ed.

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Comprehend the basic concepts and architecture of Cloud computing
CLO2	Implement Cloud Services through AWS offerings and Restful web services
CLO3	Apply the knowledge of virtualization through different virtualization technologies.
CLO4	Analyze the security challenges associated with cloud computing services and deployment models.
CLO5	Perform operations on data sets using Map Reduce framework, SQL and NO SQL databases.

MCA 214 NATURAL LANGUAGE PROCESSING AND LANGUAGE MODELS

L	T	P	Cr
3	0	2	4.0

Course objective This course provides a broad introduction to deep learning and natural language processing. It offers some of the most cost-effective approaches to automated knowledge acquisition in the emerging field of natural language understanding using deep learning and GPU Computing.

Introduction: Introduction to Natural Language Processing, NLP Challenges, applications, scope. History: Rule-based, statistical, neural, LLMs, Pretrained language models.

Text Processing and Language Modeling: Text Processing Basics, lemmatization, stemming, tokenization (word, subword, BPE, WordPiece). Text normalization, corpus preparation. Bag-of-Words (BoW), Term Frequency–Inverse Document Frequency (TF-IDF), n-grams, phrase modeling, dense embeddings, handling special characters and Unicode.

NLP using Deep Learning: Language Model Evolution GloVe, Word2Vec, ELMo. Encoder-Decoder architecture, self-attention, Transformers. static vs. dynamic embeddings, large-scale models (Megatron), GPT-style LMS vs BERT, RoBERTa, DistilBERT

Language Models: LLM Foundations GPT-2/3/4, PaLM, LLaMA, Claude, RLHF, fine-tuning, causal vs. disguised pretraining objectives, in-context learning, RAG. LLM alignment and safety, LLM metrics: BLEU, ROUGE, perplexity, fidelity. Fairness, bias, poison, hallucinations. Moral and legal issues.

Applications of NLP: Exploring NLP Problem Statements- Information Retrieval, Intent Slot Filling, Machine Translation, Punctuation and Capitalization, Question and Answering Machine Machine, Relation Extraction, Sentiment Analysis, Token Classification.

Practical Labs

- Neural Modules (NeMo) for Training Conv AI Models.
- Sentiment Analysis, Text Classification.
- Intent Slot Filling for ChatBot using Joint Bert Model.

● Machine Translation.

● Question Answering Machine.

● Information Retrieval, Punctuation; Capitalization, Relation Extraction, Sentiment Analysis, Token Classification.

● Using OpenAI API

Recommended Books:

1. Denis Rothman, Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing, 2021.
2. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming,1995.
3. Denis Rothman, RAG-Driven Generative AI: Build custom retrieval augmented generation pipelines with LlamaIndex, Deep Lake, and Pinecone, Packt Publishing, 2024.
4. Ben Auffarth, Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs, Packt Publishing Ltd, Dec 2023
5. Valentina Alto, Building LLM Powered Applications: Create intelligent apps and agents with large language models, Packt Publishing, 2024.

Course Learning Outcomes (CLOs): On completion of the course, students will able to:

CLO1	Understand the concept of Neural Networks, large scale language model methodologies and its implementation in the context of Natural Language Processing.
CLO2	Apply text pre-processing techniques for designing language processing applications
CLO3	Design and develop transformer-based text analytics applications.
CLO4	Implement NeMo for training Deep Neural Network.

MCA305 SECURE CODING				
	L	**T**	**P**	**C**
	3	**0**	**2**	**4.0**

Course Objective: 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: Security, CIA Triad, Viruses, Trojans, and Worms, Security Concepts- exploit, threat, vulnerability, risk, attack. Decipher journey starting from FQDN to html page getting served to browser, Authoritative reply, revisit layer 2 and layer 3 of TCP/IP, DNS poisoning, ARP poisoning, C language obfuscation. ARP poisoning and its countermeasures. Buffer Overrun- Stack overrun, Heap Overrun, Array Indexing Errors, Format String Bugs, PE Code injection.

Malware Terminology: Rootkits, Trapdoors, Botnets, Key loggers, Honeypots. Active and Passive Security Attacks. IP Spoofing, Tear drop, DoS, DDoS, XSS, SQL injection, Smurf, Man in middle, Format String attack.

Types of Security Vulnerabilities: buffer overflows, Invalidated input, race conditions, access-control problems, weaknesses in authentication, authorization, or cryptographic practices. Access Control Problems.

Need for secure systems: Proactive Security development process, Secure Software Development Cycle (SSDLC), Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code – Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline.

Threat modelling process and its benefits: Identifying the Threats by Using Attack Trees and rating threats using DREAD, Risk Mitigation Techniques and Security Best Practices. Security techniques, authentication, authorization. Defence in Depth and Principle of Least Privilege, Threat modelling in smart healthcare and education systems.

Secure Coding Techniques: Protection against DoS attacks, Application Failure Attacks, CPU Starvation Attack, Case studies on secure coding practices for environmental monitoring systems.

Database and Web-specific issues: SQL Injection Techniques and Remedies, Race conditions, Time of Check Versus Time of Use and its protection mechanisms. Validating Input and Interprocess Communication, Securing Signal Handlers and File Operations. XSS scripting attack and its types – Persistent and Non-persistent attack XSS Countermeasures and Bypassing the XSS Filters.

Laboratory Work: In this Lab, the student will be able to practically understand how all the security attacks does has happened, as well as learn to recognize and remove common coding errors that lead to vulnerabilities. This lab also gives an outline of the techniques for developing a secure application code that consists of using network monitoring tools, implementing different types of attacks and some protection schemes. Evaluation will be mainly based on projects and assignments.

Recommended Books:

1. Howard, M. and LeBlanc, D., Writing Secure Code, Howard, Microsoft Press (2002) 2nd Edition.
2. Deckard, J., Buffer Overflow Attacks: Detect, Exploit, Syngress (2005) 1st Edition.
3. Swiderski, F. and Snyder, W., Threat Modeling, Microsoft Professional, (2004) 1st Edition.
4. Salt, C., J., SQL Injection Attacks and Defence, Elsevier (2012), 2nd Edition.

Course Learning Outcomes (CLOs): On completion of this course, students will be able to:

CLO1	Implement DNS and ARP poisoning attack and demonstrate counter measure against these.
CLO2	Implement PE Code injection and demonstrate control hijacking via EIP manipulation
CLO3	Demonstrate skills needed to deal with common programming errors and develop secure applications.
CLO4	Demonstrate client-side attacks and identify nature of threats to software and incorporate secure coding practices throughout the planning and development of software products.
CLO5	Demonstrate SQL, XSS attack and suggest countermeasures for the same.

MCA306 BUILD AND RELEASE MANAGEMENT				
	L	**T**	**P**	**Cr**
	3	**0**	**2**	**4.0**

Course Objective: : The main objective of this course is to help participants understand the process of build and release management.

Introduction to Build and Release Management: Introduction to build, understanding different phases of build and release management, Measuring and minimizing carbon footprints of software builds, introduction to release management, best practices for build and release management, Intelligent scheduling for release deployments to reduce energy consumption, concept of build abstraction and dependency abstraction.

Dependency Management: Introduction to dependency management, Best practices for versioning to reduce duplication and resource optimization. how to use source code repositories, managing transitive dependencies, dependency scope and discussion of various tools like Ant, Maven, and Gradle, Tools for measuring energy usage in software delivery pipelines.

Document and Reporting: Introduction to build document and reporting, incorporating sustainability principles in project documentation, different types of documentation, understanding site life cycle, advance site configurations and reports, generation of unit test reports, generation of code coverage reports, code coverage tools, code coverage pros and cons.

Release Cycle: Stages of project release life cycle, source code repositories, how to install and configure source code repositories and deploying build to production goals- prepare, perform, clean and rollback.

Laboratory work: Setting up Maven environment and understanding POM hierarchy, creation of a project using Maven and its configurations.

Recommended Books:

1. "Release It!: Design and Deploy Production-Ready Software" –Book by Michael T. Nygard, published by Pragmatic Bookshelf, 2018.

2. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" –Book by Jez Humble, David Farley, published by Addison-Wesley Professional, 2010.

References:

1. Get started with GitLab CI/CD | GitLab at https://docs.gitlab.com/ee/ci/
2. Jenkins User Documentation at https://www.jenkins.io/doc/

Course Learning Outcomes (CLOs): On completion of this course, students will be able to

CLO1	Explain the basics of build and release management by learning build abstraction and declarative dependency management.
CLO2	Describe dependency management and the associated concepts like repositories, dependency identification and scope, transitive dependencies, and examples for build tools.
CLO3	Discuss the process of documentation and reporting, using site life cycle, site configuration and generation of unit testing and code coverage reports
CLO4	Define the release cycle and the phases of release, preparing, cleaning and performing goals.

MCA 215: GPU COMPUTING				
	L	T	P	Cr
	3	0	2	4.0

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

Introduction : Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding Up Real Applications, Parallel Programming Languages and Models. Power-efficient GPU architectures: low power states, dynamic voltage and frequency scaling, Energy consumption in data centers and the role of GPUs.

History of GPU Computing: Evolution of Graphics Pipelines, The Era of Fixed-Function Graphics Pipelines, Evolution of Programmable Real-Time Graphics, Unified Graphics and Computing Processors, GPGPU, Scalable GPUs, Recent Developments, and Future Trends.

Computation: Kernels, launch parameters, Thread hierarchy, Warps/wavefronts, Thread blocks/workgroups, Streaming multiprocessors, 1D / 2D / 3D thread mapping, and Device properties.

Introduction to Data Parallelism and CUDA C: Data Parallelism, CUDA Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading. Parallelization techniques for optimal GPU utilization.

Data-Parallel Execution Model: CUDA Thread Organization, Mapping Threads to Multidimensional Data, Matrix-Matrix Multiplication—A More Complex Kernel, Synchronization and Transparent Scalability, Assigning Resources to Blocks, Thread Scheduling and Latency Tolerance.

CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Tiled Matrix – À Matrix Multiplication Kernel, Memory as a Limiting Factor to Parallelism.

An Introduction to OpenCL: Data Parallelism Model, Device Architecture, Kernel Functions, Device Management and Kernel Launch, Electrostatic Potential Map in OpenCL.

Parallel Programming with OpenACC: OpenACC Versus CUDA C, Execution Model, Memory Model, Basic OpenACC Programs, Parallel Construct, Loop Construct, Kernels Construct, Data Management, Asynchronous Computation and Data Transfer.

Self-Learning Content: Basics of Parallel and distributed Computing, CUDA programming model Industry, innovation and infrastructure **Laboratory work:** Practice programs using CUDA, OpenCL and OpenACC.
Recommended Books: 1. Sanders, J. and Kandrot, E., CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional (2012) 4th Edition. 2. Kirk, D. and Hwu, M., W., Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann (2016) 3rd Edition. 3. Grama, A., Gupta, Karypis, G., Kumar, V., Introduction to Parallel Computing, Addison Wesley, (2003) 2nd Edition. 4. Hwu, M., W., A GPU Computing Gems Emerald Edition (Applications of GPU Computing Series), Morgan Kaufmann (2011) 1st Edition.

COURSE LEARNING OUTCOMES (CLOs): On completion of the course, students will able to:

CLO1	Comprehend commonly used terms in parallel computing.
CLO2	Understand common GPU architectures and Programming Models.
CLO3	Implement algorithms efficiently for common application kernels.
CLO4	Develop efficient parallel algorithms to solve given problems.