CHAITANYA SCIENCE AND ARTS COLLEGE

(AUTONOMOUS)

PAMGARH, JANJGIR-CHAMPA (C.G.)

ACCREDITED "A" GRADE BY NAAC

DEPARTMENT OF COMPUTER SCIENCE

COURSE CURRICULUM & MARKING SCHEME

POSTGRADUATE PROGRAMME

MSC COMPUTER SCIENCE

PROGRAM CODE: CCMS06

THIRD & FOURTH SEMESTER

Approved By	Board of Studies	Academic Council
Date	16/08/2026	

ACADEMIC YEAR 2026-27

Website: https://chaitanyacg.ac.in/ **Email:**principalchaitanya417@gmail.com

SESSION 2026-27

Proposed Scheme For 2-Year Postgraduate Program

PROGRAMME STRUCTURE: PART	YEAR	SEMESTER	SEMESTER
Part-I	First Year	Semester-I	Semester-II
Part-II	Second Year	Semester-III	Semester-IV

COURSE STRUCTURE:

The syllabus with the paper combination is as under:

SEMESTER-III

PAPER	COURSE CODE	TITLE OF COURSE
I	MCST301	**THEORY OF COMPUTATION AND COMPILER DESIGN**
II	MCST302	**ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEM**
III	MCST303	**SOFT COMPUTING**
IV	MCST304	**.NET TECHNOLOGY**
V	MCSP305	**(PRACTICAL-1) MATLAB**
VI	MCSP306	**(PRACTICAL-2) PROGRAMMING THROUGH .NET**

SEMESTER -IV

PAPER	COURSE CODE	TITLE OF COURSE
I	MCST401	**ADVANCED TRENDS AND TECHNOLOGY IN COMPUTER SCIENCE**
II	MCST402	**ELECTIVE - I: DATA MINING AND DATA WAREHOUSING**
III	MCST403	**ELECTIVE - II: COMPUTER GRAPHICS AND MULTIMEDIA**
IV	MCSP404	**MAJOR PROJECT WORK** **Or** **Dissertation** **Seminar, viva**

SEMESTER III
Paper-1

PROGRAMME	**M.Sc. COMPUTER SCIENCE- SEMESTER III**
TITLE	**THEORY OF COMPUTATION AND COMPILER DESIGN**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
Course Objective	**CO1:** Explain the core concepts of Finite Automata, Regular Languages, Context-Free Grammars, Pushdown Automata, Turing Machines, and key properties including closure, pumping lemmas, and undecidability results such as the Halting Problem and Rice's Theorem. **CO2:** Apply automata models, regular expressions, and context-free grammars to design lexical analyzers, parsers, and solve language recognition problems. **CO3:** Analyze and compare various parsing techniques (LL, LR, SLR, LALR), intermediate code forms, runtime environments, and code optimization strategies in compiler design. **CO4:** Evaluate the efficiency of automata, grammars, and compiler phases, and design a basic compiler pipeline using tools like Flex and Bison/ANTLR for a small domain-specific language.
Course Outcomes	• Design and analyze finite automata, regular expressions, regular grammars, and regular languages, and apply their properties to solve language recognition problems. • Construct and evaluate context-free grammars (CFGs) and pushdown automata (PDAs), including normal forms, ambiguity, equivalence, and closure properties of context-free languages. • Explain the concepts of Turing machines, computability, decidability, recursive and recursively enumerable languages, and analyze classical undecidable problems. • Apply compiler design techniques including lexical analysis, syntax analysis, syntax-directed translation, intermediate code generation, runtime environments, code optimization, and code generation using modern compiler tools.

Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**FINITE AUTOMATA AND REGULAR LANGUAGES** Principles of Mathematical Induction, Finite Automata (DFA & NFA), Regular Expressions, FA and Regular Expressions equivalence, Regular Grammar, Closure properties of Regular Languages, Pumping Lemma for Regular Languages, Myhill-Nerode Theorem, Minimization of DFA, Applications of Regular Expressions, Moore & Mealy machines.	**15**
UNIT - II	**CONTEXT-FREE GRAMMARS AND PUSHDOWN AUTOMATA** Context-Free Grammars (CFG), Parse Trees, Ambiguity in Grammars and Languages, Normal Forms (CNF & GNF), Pushdown Automata (PDA), Languages of PDA, Equivalence of PDA and CFG, Deterministic PDA, Pumping Lemma for Context-Free Languages, Closure Properties of CFLs.	**15**
UNIT - III	**TURING MACHINES AND UNDECIDABILITY** Turing Machines (TM) and its variants, TM as acceptor and transducer, Recursive and Recursively Enumerable Languages, Undecidable Problems, Halting Problem, Post's Correspondence Problem, Rice's Theorem (statement & applications), Introduction to Computability vs Complexity.	**15**

Scoapni 16/08/2026 Atul

16/08/26 Indra

SYLLABUS AND MARKING SCHEME FOR THE FIRST & SECOND SEMESTER
SESSION-2026-27
SEMESTER III

Sr. No.	Subject Code	Title	Marks		Total
			Internal	External	
I	MCST301	THEORY OF COMPUTATION AND COMPILER DESIGN	30	70	100
II	MCST302	ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEM	30	70	100
III	MCST303	SOFT COMPUTING	30	70	100
IV	MCST304	.NET TECHNOLOGY	30	70	100
V	MCSP305	(PRACTICAL-1) MATLAB			100
VI	MCSP306	(PRACTICAL-2) PROGRAMMING THROUGH .NET			100
		Total			**600**

SEMESTER-IV

SR. No.	Subject Code	Title	Marks		Total
			Internal	External	
I	MCST401	ADVANCED TRENDS AND TECHNOLOGY IN COMPUTER SCIENCE	30	70	100
II	MCST402	ELECTIVE - I: DATA MINING AND DATA WAREHOUSING	30	70	100
III	MCST403	ELECTIVE - II: COMPUTER GRAPHICS AND MULTIMEDIA			
IV	MCSP404	REASEARCH BASED (MAJOR PROJECT) OR SOFTWARE BASED (MAJOR PROJECT)			400
		Total			**600**

16/08/2026

SYLLABUS AND MARKING SCHEME FOR THE FIRST & SECOND SEMESTER
SESSION-2026-27
SEMESTER III

Sr. No.	Subject Code	Title	Marks		Total
			Internal	External	
I	MCST301	THEORY OF COMPUTATION AND COMPILER DESIGN	30	70	100
II	MCST302	ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEM	30	70	100
III	MCST303	SOFT COMPUTING	30	70	100
IV	MCST304	.NET TECHNOLOGY	30	70	100
V	MCSP305	(PRACTICAL-1) MATLAB			100
VI	MCSP306	(PRACTICAL-2) PROGRAMMING THROUGH .NET			100
		Total			**600**

SEMESTER-IV

SR. No.	Subject Code	Title	Marks		Total
			Internal	External	
I	MCST401	ADVANCED TRENDS AND TECHNOLOGY IN COMPUTER SCIENCE	30	70	100
II	MCST402	ELECTIVE - I: DATA MINING AND DATA WAREHOUSING	30	70	100
III	MCST403	ELECTIVE - II: COMPUTER GRAPHICS AND MULTIMEDIA	30	70	100
IV	MCSP404	MAJOR PROJECT WORK Or DISSERTATION SEMINAR, VIVA			100
		Total			**400**

SEMESTER III

Paper-1

PROGRAMME	**M.Sc. COMPUTER SCIENCE- SEMESTER III**
TITLE	**THEORY OF COMPUTATION AND COMPILER DESIGN**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
Course Objective	**CO1:** Explain the core concepts of Finite Automata, Regular Languages, Context-Free Grammars, Pushdown Automata, Turing Machines, and key properties including closure, pumping lemmas, and undecidability results such as the Halting Problem and Rice's Theorem. **CO2:** Apply automata models, regular expressions, and context-free grammars to design lexical analyzers, parsers, and solve language recognition problems. **CO3:** Analyze and compare various parsing techniques (LL, LR, SLR, LALR), intermediate code forms, runtime environments, and code optimization strategies in compiler design. **CO4:** Evaluate the efficiency of automata, grammars, and compiler phases, and design a basic compiler pipeline using tools like Flex and Bison/ANTLR for a small domain-specific language.
Course Outcomes	• Design and analyze finite automata, regular expressions, regular grammars, and regular languages, and apply their properties to solve language recognition problems. • Construct and evaluate context-free grammars (CFGs) and pushdown automata (PDAs), including normal forms, ambiguity, equivalence, and closure properties of context-free languages. • Explain the concepts of Turing machines, computability, decidability, recursive and recursively enumerable languages, and analyze classical undecidable problems. • Apply compiler design techniques including lexical analysis, syntax analysis, syntax-directed translation, intermediate code generation, runtime environments, code optimization, and code generation using modern compiler tools.

Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**FINITE AUTOMATA AND REGULAR LANGUAGES** Principles of Mathematical Induction, Finite Automata (DFA & NFA), Regular Expressions, FA and Regular Expressions equivalence, Regular Grammar, Closure properties of Regular Languages, Pumping Lemma for Regular Languages, Myhill-Nerode Theorem, Minimization of DFA, Applications of Regular Expressions, Moore & Mealy machines.	**15**
UNIT - II	**CONTEXT-FREE GRAMMARS AND PUSHDOWN AUTOMATA** Context-Free Grammars (CFG), Parse Trees, Ambiguity in Grammars and Languages, Normal Forms (CNF & GNF), Pushdown Automata (PDA), Languages of PDA, Equivalence of PDA and CFG, Deterministic PDA, Pumping Lemma for Context-Free Languages, Closure Properties of CFLs.	**15**
UNIT - III	**TURING MACHINES AND UNDECIDABILITY** Turing Machines (TM) and its variants, TM as acceptor and transducer, Recursive and Recursively Enumerable Languages, Undecidable Problems, Halting Problem, Post's Correspondence Problem, Rice's Theorem (statement & applications), Introduction to Computability vs Complexity.	**15**

16/08/26 16/8/26 Scanned with Tadpoli 16/8/26

UNIT - IV	**COMPILER DESIGN** Phases of Compiler, Lexical Analysis (Role, Finite Automata based implementation, Flex), Syntax Analysis (Top-down & Bottom-up parsing, LL(1), LR(0), SLR(1), LR(1), LALR parsers, Yacc/Bison/ANTLR), Syntax Directed Translation, Intermediate Code Generation (Three Address Code), Runtime Environments (Activation Records, Stack & Heap Management, Garbage Collection basics), Code Optimization (Local & Global, Peephole, DAG), Issues in Code Generation, Introduction to LLVM architecture.	**15**

Text Books (Recommended):

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computation", 3rd Edition, Pearson Education.
2. 2007. 2. J. Martin, "Introduction to Languages and the Theory of computation" Third Edition, Tata Mc Graw Hill, 2007
3. Alfred Aho, Ravi Sethi and Ullman, J.D., "Compilers Principles, Techniques and Tools", Pearson Education, 2003.

Reference Books:

- Theory of Computing by KLP Mishra, S.Chandrashekran Pub.
- Compiler Design, A.A. Pentambekar, Technical Publications.
- "Introduction to the Theory of Computation" by Michael Sipser.

SEMESTER III
Paper-II

PROGRAMME	**M.Sc. COMPUTER SCIENCE- SEMESTER III**	
TITLE	**ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEM**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
Course Objective	**CO1:** Understand the fundamental concepts of Artificial Intelligence, intelligent agents, knowledge representation, and search techniques for problem solving. **CO2:** Apply symbolic knowledge representation and logical reasoning techniques to represent knowledge and perform intelligent inference. **CO3:** Develop basic AI programs using LISP and PROLOG, and understand the principles and techniques of Natural Language Processing. **CO4:** Analyze AI planning methods, expert systems, and modern AI concepts including machine learning and neural network fundamentals.	
Course Outcomes	**CO1:** Explain the fundamental concepts of Artificial Intelligence, intelligent agents, search techniques, knowledge representation schemes, and logical reasoning methods including propositional and first-order predicate logic. **CO2:** Apply various search algorithms, knowledge representation techniques, and AI programming constructs in LISP/PROLOG to solve simple reasoning and problem-solving tasks. **CO3:** Analyze different planning strategies, natural language processing steps, and the architecture of expert systems along with their reasoning mechanisms **CO4:** Evaluate the strengths and limitations of traditional AI approaches versus modern techniques like machine learning, and design basic components of an expert system or planning agent for a given domain.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE AND SEARCH TECHNIQUES** Introduction to AI, Intelligent Agents (Agents and Environments), AI Techniques, History and Foundations of AI, State-of-the-Art Areas. Knowledge Representation: Knowledge-Based Systems, Procedural vs Declarative Knowledge. Problem Solving: State Space Search, Uninformed Search (DFS, BFS), Informed Search (Best-First, A* Algorithm), Problem Reduction, AO* Algorithm, Constraint Satisfaction Problems (CSP).	**15**
UNIT - II	**KNOWLEDGE REPRESENTATION AND REASONING** Symbolic and Structured Representations: Frames, Conceptual Dependency, Semantic Nets, Scripts. Forward vs Backward Reasoning. Symbolic Logic: Propositional Logic, First-Order Predicate Logic (FOPL) – Syntax, Semantics, Representing Instance and is-a Relationships, Computable Functions and Predicates. Introduction to Inference and Resolution in Logic.	**15**
UNIT - III	**AI PROGRAMMING AND NATURAL LANGUAGE PROCESSING** AI Programming: Introduction to LISP (Basic list manipulation, Predicates, Logical functions, I/O, Simple programs), Introduction to PROLOG. Natural Language Processing: Role of Knowledge in NLP, Approaches to NLU, Steps in NLP, Syntactic Processing, Semantic Analysis, Augmented Transition Networks, Overview of Modern NLP (Tokenization, Embeddings).	**15**

UNIT - IV	**PLANNING, EXPERT SYSTEMS AND MODERN AI** Planning: Components of Planning System, Goal Stack Planning, Hierarchical Planning, Partial-Order Planning, Reactive Systems. Expert Systems: Structure, Types (Rule-Based, Model-Based, Case-Based, Hybrid), Knowledge Acquisition & Validation, Blackboard Architecture. Modern AI Bridge: Introduction to Machine Learning basics, Neural Networks fundamentals, and limitations of traditional Expert Systems in current landscape.	**15**

Text Books (Recommended):

1. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson.
2. Elaine Rich, Kevin Knight & Shivashankar B. Nair, "Artificial Intelligence", 3rd Edition, Tata McGraw Hill.

Reference Books:

- Ivan Bratko, "Prolog Programming for Artificial Intelligence".
- Joseph C. Giarratano & Gary D. Riley, "Expert Systems: Principles and Programming

SEMESTER III

Paper-III

PROGRAMME	**M.Sc. COMPUTER SCIENCE- SEMESTER III**
TITLE	**SOFT COMPUTING**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
Course Objective	**OBJECTIVES:** • Understand the fundamental concepts of soft computing, its techniques, hybrid approaches, and applications in solving complex real-world problems. • Analyze the architecture, learning paradigms, and training algorithms of Artificial Neural Networks for pattern recognition and intelligent decision-making. • Apply fuzzy logic concepts, fuzzy inference systems, and fuzzy set operations to model and solve problems involving uncertainty and imprecision. • Understand genetic algorithms, evolutionary optimization techniques, and hybrid soft computing models for designing intelligent computational systems.
Course Outcomes	**CO1:** Explain the concepts of Soft Computing, its tools (ANN, Fuzzy Logic, GA), hybrid approaches, and their applications compared to traditional computing. **CO2:** Apply neural network architectures, learning algorithms, fuzzy operations, and genetic algorithm operators to solve basic pattern recognition and optimization problems. **CO3:** Analyze the working of different soft computing techniques, their strengths, limitations, and suitability for various real-world scenarios. **CO4:** Evaluate and design hybrid soft computing models (such as Neuro-Fuzzy ANFIS or GA-based systems) and implement simple experiments using MATLAB for practical problem-solving.

Syllabus	Description	No. of Lectures / Hours
UNIT - I	**INTRODUCTION TO SOFT COMPUTING** What is Soft Computing? Comparison with Hard Computing, Different Tools of Soft Computing (ANN, Fuzzy Logic, Genetic Algorithms, Swarm Intelligence), Areas of Application. Hybrid Soft Computing Approaches: Overview of Neuro-Fuzzy, Neuro-Genetic, and Fuzzy-Genetic Systems. Introduction to MATLAB/Python tools for Soft Computing experiments.	**15**
UNIT - II	**ARTIFICIAL NEURAL NETWORKS** Architecture of ANN, Biological Neuron vs Artificial Neuron, Types of ANN (Single Layer Perceptron, Feed-forward, Feedback Networks), XOR Problem and Limitations. Learning Paradigms: Supervised, Unsupervised, Reinforcement. Learning Rules (Hebbian, Delta, Widrow-Hoff, Competitive, Winner-Take-All). Back Propagation Algorithm (EBPA), Kohonen Networks, Hopfield Networks, Associative Memories. Training Algorithms and Practical Examples.	**15**
UNIT - III	**FUZZY LOGIC AND FUZZY SYSTEMS** Classical Sets vs Fuzzy Sets, Membership Functions, α-cuts, Properties and Theorems, Extension Principle. Operations on Fuzzy Sets (Complement, Union, Intersection, Aggregation). Fuzzy Arithmetic, Linguistic Variables, Fuzzy Relations and Measures. Fuzzy Inference Systems and Applications of Fuzzy Logic in Control and Decision Making.	**15**

16/08/26 16/08/2026 16/8/26 16/08/26

UNIT - IV	**GENETIC ALGORITHMS AND HYBRID SYSTEMS** Introduction to Optimization and Evolutionary Computing. Genetic Algorithm (GA) Operators: Selection (various techniques), Crossover, Mutation. Chromosome Representations (Binary, Real, etc.), Basic GA and Variants. Applications of GA. Hybrid Soft Computing: Design of Neuro-Fuzzy Models (ANFIS), Neuro-Genetic, Fuzzy-Genetic, and Neuro-Fuzzy-Genetic Systems. Implementation Experiments using MATLAB.	**15**

RECOMMENDED TEXT BOOKS (Selected & Curated)

1. **S. Rajasekaran and G.A. Vijayalakshmi Pai**, "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications", PHI Learning.
2. **Timothy J. Ross**, "Fuzzy Logic with Engineering Applications", 4th Edition, Wiley.

REFERENCE BOOKS

1. **J. S. R. Jang, C. T. Sun, E. Mizutani**, "Neuro-Fuzzy and Soft Computing", Pearson Education.
2. **David E. Goldberg**, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley.
3. **Stuart Russell & Peter Norvig** (relevant chapters from AI: A Modern Approach)

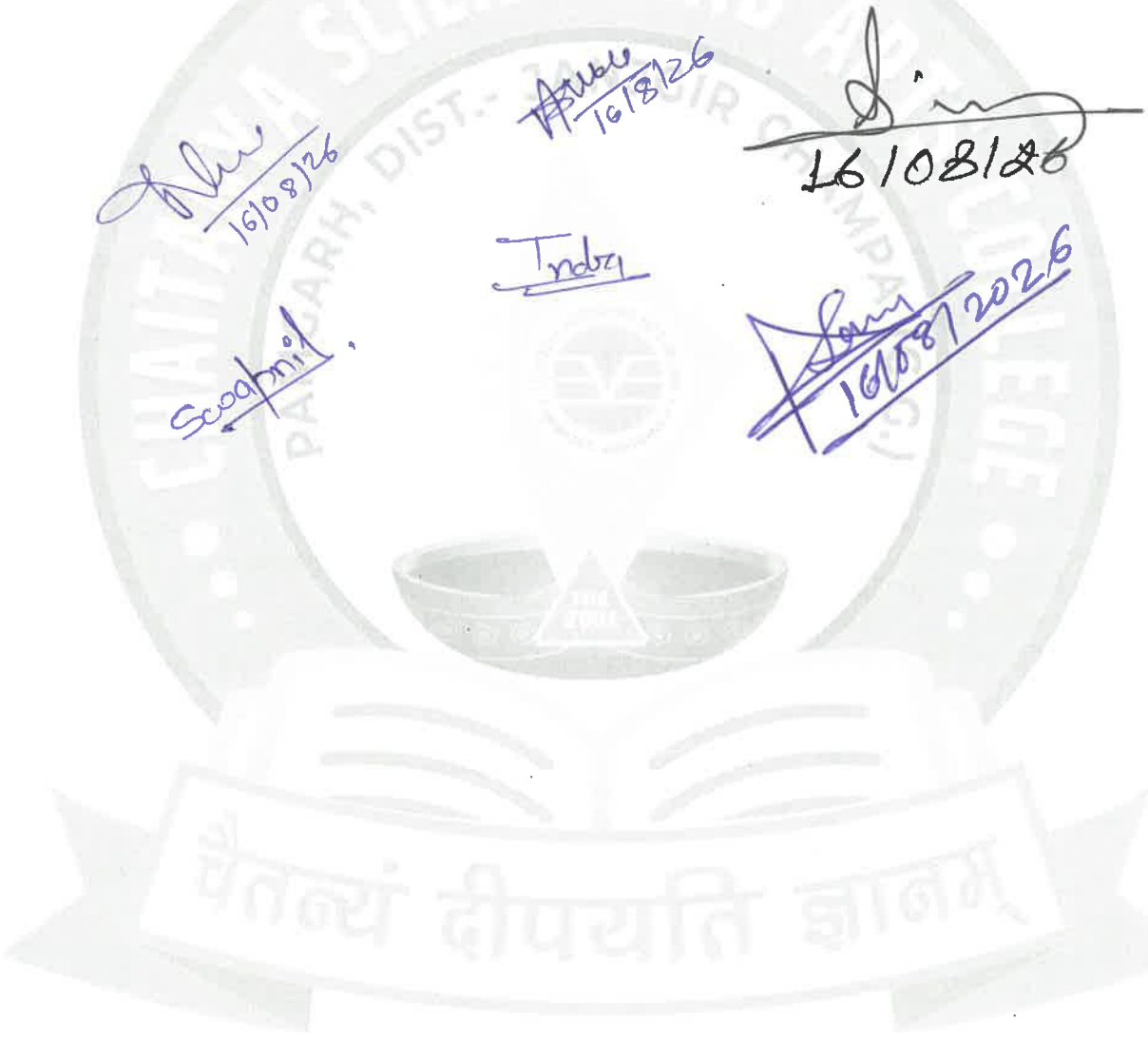
SEMESTER III
Paper-IV

PROGRAMME	**M.Sc. COMPUTER SCIENCE- SEMESTER III**
TITLE	**.NET TECHNOLOGY**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
Course Objective	1. Understand the architecture of the .NET platform and develop object-oriented applications using C# with modern .NET features. 2. Apply VB.NET programming concepts to design Windows-based applications with graphical user interfaces, file handling, exception handling, and database connectivity using ADO.NET. 3. Develop dynamic web applications using ASP.NET Core, Razor, Entity Framework Core, data binding, validation, and state management techniques. 4. Design secure and scalable .NET applications using web services, LINQ, authentication and authorization, caching, cloud deployment, and modern .NET technologies such as Blazor and microservices.
Course Outcomes	**CO1:** Explain the architecture of .NET framework, evolution to modern .NET, C# and VB.NET programming constructs, and core components like CLR, assemblies, and ADO.NET. **CO2:** Apply object-oriented concepts, controls, and data handling techniques to develop Windows and web applications using C#, VB.NET, and ASP.NET. **CO3:** Analyze various web development features including validation, data binding, navigation, authentication, and state management in ASP.NET Core applications. **CO4:** Evaluate and design advanced .NET solutions involving web services, LINQ, caching, deployment strategies, and hybrid applications using modern tools like Entity Framework and Blazor.

Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**.NET FRAMEWORK AND C# PROGRAMMING** Introduction to .NET Technology and Evolution (.NET Framework to Modern .NET 8+), CLR, MSIL, CTS, Class Library. Visual Studio IDE. C# Fundamentals: Data Types, Statements, Arrays, Strings, OOP (Classes, Objects, Inheritance, Properties, Indexers, Delegates, Events, Generics), Collections, Exception Handling, Threading, Reflection, Assemblies, and Versioning.	**15**
UNIT - II	**VB.NET AND WINDOWS APPLICATIONS** VB.NET Fundamentals: Variables, Operators, Control Structures, Procedures, Functions, Forms, GUI Controls, Menus, Loops, Validation, File Handling, Exception Handling. Object-Oriented Features: Classes, Inheritance, Generics, Collections. Database Access with ADO.NET.	**15**
UNIT - III	**ASP.NET CORE WEB DEVELOPMENT** Introduction to ASP.NET Core, Web Forms vs MVC vs Minimal APIs. Working with HTML/Razor Controls, Server Controls, Validation, Master Pages/Themes. Data Access with ADO.NET & Entity Framework Core. GridView/Data Binding, Navigation Controls (Menu, TreeView, SiteMap). Configuration and State Management.	**15**
UNIT - IV	**ADVANCED .NET APPLICATIONS AND SERVICES** Web Services (REST APIs, Minimal APIs), LINQ to SQL/Entities, Authentication & Authorization (Identity). Caching, AJAX Integration. XML/JSON Handling. Deployment: SQL Server Management (Backup/Restore), FTP, Email Services. Introduction to Blazor, Microservices, and Cloud Deployment (.NET 8+ features).	**15**

RECOMMENDED TEXT BOOKS

1. **Christian Nagel**, "Professional C# and .NET", Latest Edition, Wiley (Wrox).
2. **Andrew Lock**, "ASP.NET Core in Action", 3rd Edition, Manning. **REFERENCE BOOKS**
1. **Mark J. Price**, "C# 13 and .NET 9 – Modern Cross-Platform Development", Packt Publishing
2. **Adam Freeman**, "Pro ASP.NET Core 8", Apress.
3. **Dino Esposito**, "Programming ASP.NET Core", Microsoft Press.



SEMESTER-III

PRACTICALS

LAB-1: MATLAB
LAB-2: PROGRAMMING THROUGH .NET
Break-up of Marks for Practical Examination:

Sr. No.	Component	Maximum Marks	Minimum Passing Marks
1.	Lab Record	20	-
2.	Viva-voce	40	-
3.	Program Development and Execution	40	-
Total	**100**	**100**	**36**

Passing Criteria: Minimum 36 marks out of 100 are required to pass the practical examination.

LAB-1: MATLAB (Soft Computing Implementation)
Objectives: To implement soft computing techniques using MATLAB for neural networks, fuzzy systems, genetic algorithms, and hybrid models.
Suggested List of Experiments:

1. Introduction to MATLAB environment, matrix operations, and plotting.
2. Implementation of Single Layer Perceptron and solving XOR problem.
3. Back Propagation Algorithm for multi-layer feed-forward network.
4. Kohonen Self-Organizing Map and Hopfield Network.
5. Fuzzy Set operations, Membership Functions, and Fuzzy Inference System.
6. Genetic Algorithm implementation for optimization problems (crossover, mutation, selection).
7. Design and simulation of ANFIS (Adaptive Neuro-Fuzzy Inference System).
8. Neuro-Genetic or Fuzzy-Genetic hybrid model experiment.
9. Case study: Pattern recognition / Function approximation using ANN.
10. Mini Project: Soft computing solution for a real-world problem (e.g., control system or clustering).

LAB-2: PROGRAMMING THROUGH .NET
Objectives: To develop Windows and Web applications using C#, VB.NET, ASP.NET Core, and advanced .NET features.
Suggested List of Experiments:

1. C# console applications using OOP concepts (Inheritance, Delegates, Events, Generics).
2. Windows Forms application with controls, menus, and event handling.
3. VB.NET application with multiple forms, validation, and file handling.
4. Database connectivity using ADO.NET (CRUD operations).
5. ASP.NET Core MVC web application with Razor pages and validation.
6. Data binding with Grid View / Data Table and Entity Framework Core.
7. Implementation of Authentication & Authorization using ASP.NET Identity.
8. Web Services / REST API development and consumption.
9. AJAX integration and Caching in web applications.
10. Mini Project: Full-stack .NET application (e.g., E-commerce / Student Management System) with deployment features.

SEMESTER IV
Paper-1

PROGRAMME	**M.SC. COMPUTER SCIENCE SEMESTER IV**
TITLE	**ADVANCED TRENDS AND TECHNOLOGY IN COMPUTER SCIENCE**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
Course Objective	1. Understand the fundamental concepts, architectures, and applications of Big Data Analytics, including Hadoop, Spark, and real-time data processing techniques. 2. Analyze cloud computing architectures, service and deployment models, virtualization, cloud platforms, security, and resource management for scalable computing solutions. 3. Understand the principles, architectures, and applications of grid and cluster computing for high-performance and distributed computing environments. 4. Apply the concepts of pervasive and edge computing to design context-aware, IoT-enabled, and scalable distributed systems while addressing security and emerging technology challenges.
Course Outcomes	**CO1:** Explain the core concepts, architectures, and applications of Big Data Analytics, Cloud Computing, Grid/Cluster Computing, and Pervasive/Edge Computing paradigms. **CO2:** Apply Big Data tools, cloud service models, and cluster/grid resource management techniques to solve large-scale computing problems. **CO3:** Analyze the strengths, limitations, and suitability of different advanced computing technologies for real-world scenarios including scalability, security, and performance. **CO4:** Evaluate emerging trends in pervasive and edge computing and design integrated solutions combining cloud, big data, and IoT technologies.

Syllabus	Description	No. of Lectures / Hours
UNIT - I	**BIG DATA ANALYTICS** Big Data Characteristics, Applications (Product Selection, Location-Based Services, Online Advertising). Architecture Components: Massively Parallel Processing (MPP), Hadoop Ecosystem, Spark Framework. Unstructured Data Analytics, Real-Time Adaptive Analytics, Data Privacy & Security, Ontologies, Implementation Tools and Case Studies.	**15**
UNIT - II	**CLOUD COMPUTING** Cloud Computing Overview, Service Models (IaaS, PaaS, SaaS, Faas), Deployment Models. Cloud Platforms (AWS, Azure, GCP), Virtualization, Resource Management, Web Services & APIs, Cloud Storage, Security & Privacy Issues, Serverless Computing, Cost Models, and Migration Strategies.	**15**
UNIT - III	**GRID AND CLUSTER COMPUTING** Grid Computing: Architecture, Resource Management, Middleware, Monitoring (GMA), Global Middleware Examples. Cluster Computing: Taxonomy, Hardware/Software Architecture, Node Management, Workload Scheduling, High-Performance Computing, Virtualization in Clusters, Comparison with Cloud/Grid.	**15**
UNIT - IV	**PERVASIVE & EDGE COMPUTING** Pervasive Computing: Principles, Characteristics, Evolution, Smart Devices, Context-Aware Systems, Embedded Controls. Edge Computing: Architecture, IoT Integration, Scalability, Security Challenges, Pervasive Applications, Mobile & Ubiquitous Systems, Future Trends (Fog Computing, 5G/6G Enablement).	**15**

RECOMMENDED TEXT BOOKS (Curated & Updated)

1. **Arvind Sathi**, "Big Data Analytics: Disruptive Technologies for Changing the Game", IBM Corp.
2. **Anthony T. Velte, Toby J. Velte & Robert Elsenpeter**, "Cloud Computing: A Practical Approach", McGraw Hill.
3. **Rajkumar Buyya et al.**, "Mastering Cloud Computing", Morgan Kaufmann (for Cloud + Edge integration).

REFERENCE BOOKS

1. **Jure Leskovec, Anand Rajaraman & Jeffrey D. Ullman**, "Mining of Massive Datasets".
2. **Maozhen Li & Mark Baker**, "The Grid: Core Technologies", Wiley.
3. **Uwe Hansmann et al.**, "Pervasive Computing: The Mobile World".

SEMESTER IV
Paper-II

PROGRAMME	**M.SC. COMPUTER SCIENCE SEMESTER IV**	
TITLE	**ELECTIVE - I COMPUTER GRAPHICS AND MULTIMEDIA**	
Total Marks	• **Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
Course Objective	• Understand the fundamentals of computer graphics systems, graphics hardware, display devices, output primitives, and graphics generation algorithms. • Apply two-dimensional geometric transformations, viewing techniques, clipping algorithms, and window-to-viewport transformations for graphical object manipulation. • Analyze three-dimensional transformations, viewing and projection techniques, object representation methods, and basic rendering concepts for 3D graphics. • Develop multimedia and interactive graphics applications using image editing techniques, vector graphics, OpenGL/WebGL concepts, and modern 3D modeling and animation tools.	
Course Outcomes	**CO1:** Explain the fundamental concepts of computer graphics systems, output primitives, 2D/3D transformations, projections, and multimedia components. **CO2:** Apply line drawing, circle, polygon filling, and geometric transformation algorithms to solve graphics problems. **CO3:** Analyze 2D and 3D viewing, clipping, and rendering techniques along with their mathematical foundations. **CO4:** Evaluate and design basic multimedia applications and interactive graphics scenes using modern tools like OpenGL/WebGL and image/animation software.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**INTRODUCTION TO COMPUTER GRAPHICS SYSTEM** Computer Graphics Fundamentals and Applications, Video Display Devices, Raster & Random Scan Systems, Input/Output Devices, Graphics Software and Standards. Output Primitives: Line Drawing Algorithms (DDA, Bresenham), Circle & Ellipse Generating Algorithms, Polygon Fill Algorithms (Scan-Line, Boundary-Fill, Flood-Fill), Inside-Outside Tests	**15**
UNIT - II	**2D TRANSFORMATIONS AND CLIPPING** 2D Viewing and Clipping (Point, Line, Polygon Clipping Algorithms). 2D Geometric Transformations: Translation, Rotation, Scaling, Reflection, Shear. Matrix Representation, Homogeneous Coordinates, Composite Transformations. Window-to-Viewport Transformation.	**15**
UNIT - III	**3D TRANSFORMATIONS AND VIEWING** 3D Geometric Transformations: Translation, Rotation, Scaling. 3D Viewing and Projections (Parallel, Perspective, Orthographic, Oblique, Isometric). 3D Object Representation: Polygon Surfaces, Bezier Curves & Surfaces, Visible Surface Detection. Introduction to 3D Rendering and Illumination Models.	**15**
UNIT - IV	**MULTIMEDIA AND INTERACTIVE GRAPHICS** Fundamentals of Multimedia Systems, Text, Image, Audio, Video, Animation. Image Editing Concepts (Layers, Masks, Filters, Color Correction, Special Effects). Introduction to Modern Tools: Adobe Photoshop/Alternatives, Vector Graphics (CorelDraw/Illustrator concepts), OpenGL/WebGL Basics, Blender Fundamentals for 3D Modeling & Animation. Multimedia Authoring and Applications.	**15**

RECOMMENDED TEXT BOOKS (Curated)

1. **D.F. Rogers**, "Procedural Elements for Computer Graphics", Tata McGraw Hill.
2. **J.D. Foley & A.D. Van Dam**, "Fundamentals of Interactive Computer Graphics", Addison-Wesley.

REFERENCE BOOKS

1. **Edward Angel**, "Interactive Computer Graphics: A Top-Down Approach with WebGL", Pearson. *(Modern OpenGL/WebGL focus — highly recommended upgrade.)*
2. **Donald Hearn & M. Pauline Baker**, "Computer Graphics with OpenGL", Pearson.
3. **Rogers and Adams**, "Mathematical Elements for Computer Graphics", Tata McGraw Hill.

SEMESTER IV

Paper-II

PROGRAMME	**M.SC. COMPUTER SCIENCE SEMESTER IV**
TITLE	**ELECTIVE - II DATA MINING AND DATA WAREHOUSING**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
Course Objective	• Understand the fundamentals of data warehousing, data mining, and data preprocessing techniques for efficient knowledge discovery from large datasets. • Analyze data warehouse architectures, OLAP technologies, multidimensional data models, and data cube techniques for decision support systems. • Apply association rule mining, classification, and prediction techniques using appropriate algorithms to extract meaningful patterns from data. • Evaluate clustering techniques, advanced data mining or solving complex analytical problems.
Course Outcomes	**CO1:** Explain the fundamental concepts of data warehousing, data mining, preprocessing techniques, OLAP operations, and multidimensional modeling. **CO2:** Apply data preprocessing, association rule mining algorithms, and classification techniques on real datasets. **CO3:** Analyze clustering methods, frequent pattern mining, and prediction models along with their performance measures. **CO4:** Evaluate advanced data mining applications in web, text, multimedia, and social networks, and design basic data mining solutions for business problems.

Syllabus	Description	No. of Lectures / Hours
UNIT - I	**INTRODUCTION TO DATA WAREHOUSING AND PREPROCESSING** Fundamentals of Data Mining and Data Warehousing, Data Mining Functionalities, Classification of Data Mining Systems, Major Issues in Data Mining. Data Warehouse Architecture, Data Preprocessing: Data Cleaning, Data Integration, Data Transformation, Data Reduction, Data Discretization.	**15**
UNIT - II	**OLAP AND DATA WAREHOUSE IMPLEMENTATION** OLAP Technology, Multidimensional Data Model, Data Warehouse Implementation, Data Cube Technology, OLAP Operations, From Data Warehousing to Data Mining, Attribute-Oriented Induction, Efficient Methods for Data Cube Computation.	**15**
UNIT - III	**ASSOCIATION RULE MINING AND CLASSIFICATION** Mining Frequent Patterns and Associations, Apriori Algorithm, FP-Growth, Correlation Analysis, Constraint-Based Association Mining. Classification and Prediction: Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Back Propagation, Accuracy and Error Measures, Ensemble Methods.	**15**
UNIT - IV	**CLUSTERING, ADVANCED MINING AND APPLICATIONS** Cluster Analysis: Types of Clustering (Partitioning, Hierarchical, Density-Based, Grid-Based, Model-Based), Outlier Analysis. Mining Streams, Time-Series and Sequence Data, Graph Mining, Social Network Analysis. Applications: Mining Multimedia, Text, Web, Spatial Data, Trends in Data Mining, Research Prototypes and Social Impacts.	**15**

RECOMMENDED TEXT BOOKS

1. **Jiawei Han & Micheline Kamber**, "Data Mining: Concepts and Techniques", 3rd Edition, Morgan Kaufmann.
2. **Pang-Ning Tan, Michael Steinbach & Vipin Kumar**, "Introduction to Data Mining", Pearson Education.

REFERENCE BOOKS

1. **Arun K. Pujari**, "Data Mining Techniques", 2nd Edition, Universities Press.
2. **Margaret H. Dunham**, "Data Mining: Introductory and Advanced Topics", Pearson Education.
3. **Sam Anahory & Dennis Murray**, "Data Warehousing in the Real World", Pearson Education.

SEMESTER-IV
MAJOR PROJECT

Note:

1. It is compulsory for students to undertake the Major Project in a group of **maximum two students**.
2. The project must be carried out under **Government Sector / Public Sector / Private Ltd. Software Company**. Students should not undertake projects under local or small private institutions.
3. The project must be original work developed by the students themselves. Copying or using someone else's project is strictly prohibited.

Steps for Live Project Development

1. Getting customer's requirements
2. Preparing designs, database and business logics
3. Developing software application project
4. Testing and implementing the project
5. Troubleshooting the project application after implementation

Break-up of Marks for Major Project

Sr. No.	Component	Maximum Marks	Minimum Passing Marks
1.	Project Record	100	-
2.	Viva-voce	100	-
3.	Program Development and Execution	200	-
Total	**400**	**400**	**144**

Passing Criteria: Student must secure minimum **144 marks** out of **400** to pass the Major Project.

Evaluation Guidelines (Recommended):

- **Project Record** (100 Marks): Complete documentation including SRS, Design documents, ER Diagrams, Test Cases, and User Manual.
- **Viva-voce** (100 Marks): Depth of understanding, presentation, and defense of the project.
- **Program Development & Execution** (200 Marks): Working demonstration, code quality, innovation, and successful implementation.