Dr YSR ANU College of Engineering & Technology

ACHARYA NAGARJUNA UNIVERSITY

SCHEME OF INSTRUCTION AND EXAMINATION, w.e.f. 2019-2022 (R19)

COMPUTER SCIENCE & ENGINEERING (CSE)

COMPUTER SCIENCE & INFORMATION TECHNOLOGY(CI)

III/IV B.TECH - SEMESTER I (R19 Regulation - Structure & Syllabus)

S.No.	Course Details		Category	Scheme of Instruction			Scheme of Examination		
	Code	Subject Name		Hours in a Week			Marks		Credits
				L	T	P	Internal	External	
1	CSE / CI 311	Automata Theory & Compiler Design	PC	3	0	0	40	60	3
2	CSE / CI 312	Java Programming	PC	3	0	0	40	60	3
3	CSE / CI 313	Design & Analysis of Algorithms	PC	3	0	0	40	60	3
4	CSE / CI 314	Job Elective – 1	JE-1	3	0	0	40	60	3
5	CSE / CI 315	Professional Elective-1	PE-1	3	0	0	40	60	3
6	CSE / CI 351	Java Lab	PC	0	0	3	40	60	1.5
7	CSE / CI 352	Design & Analysis of Algorithms	PC	0	0	3	40	60	1.5
8	CSE / CI 353	Job Elective -1 Lab	JE-1 Lab	0	0	3	40	60	1.5
9	CSE / CI 354	Mobile Application development Lab	Skill	0	0	3	40	60	2
Total Credits									21.5

JE-1 (Lab Oriented)

A. VLSI Design
B. Digital Signal Processing
C. Internet of Things (IoT)
D. Microprocessors and Microcontrollers

PE-1:

A. Computer Graphics
B. Advanced Computer Architecture
C. Software Engineering
D. Distributed Systems

Dr YSR ANU College of Engineering & Technology

ACHARYA NAGARJUNA UNIVERSITY

SCHEME OF INSTRUCTION AND EXAMINATION, w.e.f. 2019-2022 (R19)

COMPUTER SCIENCE & ENGINEERING (CSE)

COMPUTER SCIENCE & INFORMATION TECHNOLOGY(CI)

III/IV B.TECH - SEMESTER II (R19)(R19 Regulation - Structure & Syllabus)

S.No.	Course Details		Category	Scheme of Instruction			Scheme of Examination		
	Code	Subject Name		Hours in a Week			Marks		Credits
				L	T	P	Internal	External	
1	CSE / CI 321	Cryptography & Network Security	PC	3	0	0	40	60	3
2	CSE / CI 322	Artificial Intelligence & Machine Learning	PC	3	0	0	40	60	3
3	CSE / CI 323	Computer Networks	PC	3	0	0	40	60	3
4	CSE / CI 324	Job Elective – 2	JE-2	3	0	0	40	60	3
5	CSE / CI 325	Professional Elective-2	PE-2	3	0	0	40	60	3
6	CSE / CI 361	AI&ML Lab	PC	0	0	3	40	60	1.5
7	CSE / CI 362	Computer Networks Lab	PC	0	0	3	40	60	1.5
8	CSE / CI 363	JE2 Lab	JE 2 Lab	0	0	3	40	60	1.5
9	CSE / CI 364	.NET programming	SKILL	0	0	3	40	60	2
Total Credits									21.5`

JE-2 (Lab Oriented)

A. Soft Computing
B. Data Engineering
C. Digital Image Processing
D. Software Testing Methodologies

PE-2

A. Wireless Networks
B. High Performance Computing
C. Cloud Computing Architecture and Its Applications
D. Advanced Databases

CSE / CI 311	Automata Theory & Compiler Design

Objectives:

The learning objectives of this course are to:

Introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability. Compiler design is to explore the principles, algorithms, and data structures involved in the design and construction of compilers. Topics include context-free grammars, lexical analysis, parsing techniques, symbol tables, error recovery, code generation, and code optimization.

UNIT-I

Finite Automata & Regular Languages - Languages vs. Problems. Finite State Automata, Regular Languages. Closure properties, Limitations, Pumping Lemma, Myhill-Nerode relations, Quotient Construction. Minimization Algorithm. Non-determinism & Regular Expressions - Notion of non-determinism. Acceptance condition. Subset construction. Pattern matching and regular expressions.

UNIT-II

Grammars & Context-free Languages (CFLs) - Grammars and Chomsky Hierarchy, CFLs, Regular Grammars, Chomsky Normal Form, Pumping Lemma for CFLs, Inherent Ambiguity of Context-Free Languages, Cock-Younger-Kasami Algorithm, Applications to Parsing. Pushdown Automata (PDA).

Unit-III

Turing Machines & Computability - Introduction to Turing Machines, Configurations, Halting Vs. Looping. Multi-tape Turing machines. Recursive and Recursively enumerable languages. Undecidability of Halting Problem. Reductions.

Unit-IV

Introduction to Compiler - Phases and passes, Bootstrapping, Finite state machines and regular Expressions and their applications to lexical analysis, Implementation of lexical analyzers, Lexical-analyzer generator.

Basic Parsing Techniques- Parsers, top-down parsing, bottom-up parsing, LR parsing, Canonical LR parsing, LALR parsing.

Unit-V

Syntax-directed Translation - Syntax-directed Translation schemes, Implementation of Syntaxdirected Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples. Symbol Tables: Data structure for symbols tables, representing scope information. Run-Time

Introduction to code optimization- Loop optimization, the DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis.

Reference Books:

1. Automata and Computability, Dexter C. Kozen, Springer Publishers, 2007.
2. Introduction to Automata Theory, Languages and Computation, Hopcroft, Motwani, and Ullman, Pearson Publishers, Third Edition, 2006.
3. Elements of the Theory of Computation, H. R. Lewis and C.H. Papadimitriou, Prentice Hall Publishers, 1981
4. Introduction to Languages and the Theory of Computation, John. C. Martin, Tata McGraw-Hill, 2003.
5. Formal Languages and Automata Theory, E.Srinivasa Reddy, B.S.Publications
6. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Pearson.
7. Compiler Construction-Principles and Practice, Kenneth C Louden, Cengage Learning.
8. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
9. Compilers Principles, techniques and tools, Alfred V. Aho Ravi Sethi D. Ullman Pearson Education, 2007
10. Introduction to compiler design, Torben Egdius Mogensen, Pearson Education 2011

Course Outcomes:

- Understand the basic properties of formal languages and grammars.
- Differentiate regular, context-free and recursively enumerable languages.
- Make grammars to produce strings from a specific language.
- Including decidability and intractability.
- At the end of the subject, students will understand different considerations and phases of compilation, the impact of language attributes upon the compilation process, the effect of hardware feature on the generated code and the practical fundamentals of compiler implementation.

CSE / CI 312	Java Programming

COURSE OBJECTIVES:

1. To teach principles of Object-Oriented Programming paradigm including abstraction, encapsulation, inheritance and polymorphism.

2. To impart fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.

3. To inculcate concepts of inheritance to create new classes from existing one & Design the classes needed given a problem specification;

4. To familiarize the concepts of packages and interfaces.

5. To facilitate students in handling exceptions.

6. To demonstrate the concept of event handling used in GUI.

UNIT - I

JAVA BASICS: Review of Object Oriented concepts, History of Java, Java buzzwords, JVM architecture, Data types, Variables, Scope and life time of variables, arrays, operators, control statements, type conversion and casting, simple java program, constructors, methods, Static block, Static Data, Static Method String and String Buffer Classes, Using Java API Document.

UNIT - II

INHERITANCE AND POLYMORPHISM: Basic concepts, Types of inheritance, Member access rules, Usage of this and Super key word, Method Overloading, Method overriding, Abstract classes, Dynamic method dispatch, Usage of final keyword.

PACKAGES AND INTERFACES: Defining package, Access protection, importing packages, Defining and Implementing interfaces, and Extending interfaces.

I / O STREAMS: Concepts of streams, Stream classes- Byte and Character stream, Reading console Input and Writing Console output, File Handling.

UNIT - III

EXCEPTION HANDLING: Exception types, Usage of Try, Catch, Throw, Throws and Finally keywords, Built-in Exceptions, Creating own Exception classes.

MULTI THREADING: Concepts of Thread, Thread life cycle, creating threads using Thread class and Runnable interface, Synchronization, Thread priorities, Inter Thread communication.

UNIT – IV

AWT CONTROLS: The AWT class hierarchy, user interface components- Labels, Button, Text Components, Check Box, Check Box Group, Choice, List Box, Panels – Scroll Pane, Menu, Scroll Bar. Working with Frame class, Color, Fonts and layout managers.

EVENT HANDLING: Events, Event sources, Event Listeners, Event Delegation Model (EDM), Handling Mouse and Keyboard Events, Adapter classes, Inner classes.

UNIT – V

SWINGS: Introduction to Swings, Hierarchy of swing components. Containers, Top level containers - JFrame, JWindow, JDialog, JPanel, JButton, JToggleButton, JCheckBox, JRadioButton, JLabel,JTextField, JTextArea, JList, JComboBox, JScrollPane.APPLETS: Life cycle of an Applet, Differences between Applets and Applications, Developing applets, simple applet.

REFERENCE BOOKS:

1. Herbert Schildt (2010), The complete reference, 7th edition, Tata Mc graw Hill, New Delhi

2. Cay S. Horstmann and Gary Cornell, "Core Java: Volume I – Fundamentals", Eighth Edition, Sun Microsystems Press, 2008.

3. Head First Java, O'rielly publications

4. T. Budd (2009), An Introduction to Object Oriented Programming, 3rd edition, Pearson Education, India.

5. Nino, F. A. Hosch (2002), An Introduction to programming and OO design using Java, John Wiley & sons, New Jersey.

6. Y. Daniel Liang (2010), Introduction to Java programming, 7th edition, Pearson education, India.

COURSE OUTCOMES:

At the end of the course students will be able to:

1. Analyse the necessity for Object Oriented Programming paradigm over structured programming and become familiar with the fundamental concepts in OOP like encapsulation, Inheritance and Polymorphism

2. Design and develop java programs, analyse, and interpret object-oriented data and report results.

3. Design an object-oriented system, AWT components and multithreaded processes as per needs and specifications.

4. Participate and succeed in competitive examinations like GATE, Engineering services, recruitment interviews etc.

CSE / CI 313	Design & Analysis of Algorithms

Objectives:

- Analyse the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Unit I

Background: Introduction, algorithms specification, time and space complexity, performance analysis.
Divide and Conquer: Binary search, merge sort, quick sort, Strassen's matrix multiplication, maximum and minimum problem.

Unit II

Greedy Methods: General method, optimal merge patterns, optimal storage on tapes, Knapsack problem, job scheduling problem, single source shortest path problem.

Unit III

Dynamic Programming: General method, multistage graphs, 0/1 Knapsack problem, longest common subsequence, string editing, matrix chain multiplication, travelling salesman problem, optimal binary search trees.

Unit IV

Back Tracking: General method, 4-queen problem, sum of subset problem, graph colouring, Hamiltonian cycles.

Unit V

Branch and Bound: General method, 0/1 knapsack problem, travelling salesman problem.
NP Hard and NP Complete: deterministic and nondeterministic algorithms, NP Hard and NP complete.

Reference Books:

1. Cormen T. H, Leiserson C. E, Rivest R. L, and Stein C., Introduction to Algorithms, Prentice-Hall of India, 2nd Ed., 2001.
2. Brassard G., Fundamentals of Algorithmics, Prentice-Hall of India, 2003.
3. Aho A. V., Design and Analysis of Algorithms, Addison Wesley, 2001.
4. Horowitz E., Computer Algorithms, Galgotia Publications, 1998

Course Outcomes:

Students who complete the course will have demonstrated the ability to do the following:

- Ability to understand mathematical formulation, complexity analysis and methodologies to solve recurrence relations for algorithms.
- Ability to design algorithms using standard paradigms like: Greedy, Divide and Conquer, Dynamic Programming, Backtracking and Branch and Bound.
- Ability to understand NP class problems and formulate solutions using standard approaches.
- Ability to apply algorithm design principles to derive solutions for real life problems and comment on complexity of solution.

CSE / CI 314 A	VLSI Design

Course Objectives:

The objectives of the course are to:

1. Give exposure to different steps involved in the fabrication of ICs using MOS transistor, CMOS/BICMOS transistors.

2. Study the fundamentals of CMOS circuits and its characteristics.

3. Explain electrical properties of MOS and BiCMOS devices to analyze the behavior of inverters designed with various loads.

4. Give exposure to the design rules to be followed to draw the layout of any logic circuit.

5. Provide concept to design different types of logic gates using CMOS inverter.

UNIT – I

An introduction to MOS technology: Introduction to IC technology, Basic MOS transistors, NMOS fabrication, CMOS fabrication and BiCMOS technology. Basic Electrical Properties of MOS and BiCMOS Circuits: Ids versus Vds relationships, threshold voltage Vt, Transconductance (gm), Figure of merit (ω), Pass transistor, NMOS inverter, Pull-up to pull- down ratio, CMOS inverter, BICMOS inverters, Latch-up in CMOS circuits.

UNIT – II

MOS and BICMOS circuit Design processes: MOS layers, Stick diagrams, Layout diagrams, Design rules and layout, Sheet resistance Rs, Standard unit of capacitance, The Delay unit, Inverter delays, Propagation delays, Wiring capacitances, Scaling models, Scaling factors for device parameters.

UNIT – III

Subsystem design and layout: Architectural issues, Switch logic, Gate Logic, examples of Structured Design (combinational logic). Design of an ALU subsystem: Design of 4-bit adder, adder element requirements, a standard adder element, Implementing ALU functions with an adder. A further consideration of adders: Manchester carry chain, carry select adder, carry skip adder.

UNIT – IV

VLSI design flow, Introduction to ASICs, Full Custom ASICs, standard cell based ASICs, Gate array based ASICs, Programmable logic devices, PLAs, PALs, CPLDs and FPGAs.

UNIT – V

VHDL Hardware Description Language: Program Structure, Types and Constants, functions and Procedures, Libraries and Packages, Structural Design Elements, Dataflow design Elements, Behavioral design Elements, VHDL programs, The Time Dimension and Simulation, Synthesis.

REFERENCE BOOKS:

1. Douglas A.Pucknell and Kamran Eshranghian, Basic VLSI Design, 3 rd edition, PHI, 2002.

2. Debaprasad Das, VLSI Design, Oxford University Press, 2nd edition, 2015.

3. Michael John Sebastian Smith, Application Specific Integrated Circuits, Addison Wesley, 2003.

4. K Lal Kishore and VSV Prabhakar, VLSI Design, I K International Publishing House, 2009

5. J.Bhasker, A VHDL Primer, Pearson Education India, 3rd edition, 2015.

6. John F Wakerly, Digital Design Principles & Practices, 3rd Edition, Pearson Education, 2002.

7. Neil H E Weste and Kamran Eshranghian, Principles of CMOS VLSI Design, A system perspective, 2nd edition, Pearson Education, 2002.

8. Stephen Brown and Z Vonko Vranesic, Fundamentals of Digital Logic with VHDL Design, TMH, 2002.

9. Douglas L. Perry, VHDL Programming by Example, McGraw Hill Education, 4th edition 2017

Course Outcomes:

On successful completion of this course, the student should be able to:

1. Acquire qualitative knowledge about the fabrication process of integrated circuit using MOS transistors.

2. Choose an appropriate inverter depending on specifications required for a circuit.

3. Draw the layout of any logic circuit which helps to understand and estimate parasitics of any logic circuit.

4. Design different types of logic gates using CMOS inverter.

5. Provide design concepts to design building blocks of data path of any system using gates.

6. Understand basic programmable logic devices

7. Understand the modeling Styles in VHDL.

CSE / CI 314 B	Digital Signal Processing

Course Objectives:

The primary objective of this course is to provide a thorough understanding and working knowledge of design, implementation and analysis DSP systems.

UNIT – I

Discrete Signals and Systems: Introduction to digital signal processing, Advantages and applications, Discrete time signals, LTI system: Stability and causality, Frequency domain representation of discrete time signals and systems. Review of Z-transforms and Inverse Z-transforms.

UNIT – II

DFT and FFT: Discrete Fourier Series, Properties of DFS, Discrete Fourier Transform, Properties of DFT, Linear convolution using DFT, Computations for evaluating DFT, Decimation in time FFT algorithms, Decimation in frequency FFT algorithm, Computation of inverse DFT.

UNIT – III

IIR Filter Design Techniques: Introduction, Properties of IIR filters, Design of Digital Butterworth and Chebyshev filters using bilinear transformation, Impulse invariance transformation methods. Design of digital filters using frequency transformation method.

UNIT – IV

FIR Filter Design Techniques: Introduction to characteristics of linear phase FIR filters, Frequency response, Designing FIR filters using windowing methods: rectangular window, Hanning window, Hamming window, Generalised Hamming window, Bartlett triangular window,Kaiser window, Processing Comparison of IIR and FIR filters.

UNIT – V

Realization of Digital Filters: Direct, Canonic, Cascade, Parallel and Ladder realizations Effect of finite register length in FIR filter design, Introduction to Multi rate Signal Processing-Decimation, Interpolation, sampling rate convertion

REFERENCE BOOKS:

1. Lonnie C Ludeman, Fundamentals of Digital Signal Processing, John Wiley & Sons, 2003.

2. S K Mitra, Digital Signal Processing: A Computer Based Approach, 2nd Edition, TMH, 2003

3. Alan V Oppenheim and Ronald W Schafer, Digital Signal Processing, Pearson Education/PHI, 2004.

4. P. Ramesh Babu, Digital Signal Processing, 2nd Edition, Scitech Publications, 20041.

5. Johnny R. Johnson, Introduction to Digital Signal Processing, PHI, 2001.

6. Andreas Antoniou, Digital Signal Processing,TMH, 2006.

7. John G.Proakis, Dimitris G Manolakis, digital Signal Processing: Principles, Algorithms and Applications, Pearson Education / PHI, 2003.

Course Outcomes:

Upon successful completion of this course the students will have developed following skills/abilities:

1. Interpret, represent and process discrete/digital signals and systems.

2. Thorough understanding of frequency domain analysis of discrete time signals.

3. Ability to design & analyze DSP systems like FIR and IIR Filter etc.

4. Practical implementation issues such as computational complexity, hardware resource limitations as well as cost of DSP systems or DSP Processors.

CSE / CI 314 C	Internet of Things

Course Objectives:

1. Students will be explored to the interconnection and integration of the physical world and the cyber space.
2. They are also able to design & develop IOT Devices.

UNIT-I

Introduction to IoT Defining IoT, Characteristics of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Communication models & APIs

UNIT–II

Elements of IoT: Hardware Components-Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces. Software Components-Programming API's (using Python / Node.js / Arduino), Communication Protocols- ZigBee, Bluetooth, 6LoPAN, LoRa, MQTT, CoAP,XMPP.

UNIT–III

M2M and IoT Design Methodology: M2M- Differences and Similarities between M2M and IoT, SDN and NFV for IoT; IoT Design Methodology.

UNIT-IV

Domain specific applications of IoT: Home automation, Industry applications, Surveillance Applications, Other IoT applications, challenges of IoT.

UNIT–V

Developing IoTs: Introduction to Python, Implementing IoT concepts with python.

IoT Case Studies: Smart Lighting, Home Intrusion Detection, Smart Parking, Weather Monitoring System, Weather Reporting Bot, Air Pollution Monitoring, Forest Fire Detection, Smart Irrigation.

Reference books:

1. From Internet of Things to Smart Cities: Enabling Technologies,Hongjian Sun, Chao Wang, BasharI Ahmad

2. Internet of Things: A Hands-On Approach,ArshdeepBahga, Vijay Madisetti ,universities press

3. Learning Internet of Things By Peter Waher Packt Publishing Ltd

4. Internet of Things with Python ,GastnCHillar,Packt Publishing Ltd

5. Adrian McEwen, Hakim Cassimally, Designing the Internet of Things, John Wiley and Sons,1st Edition, 2014.

6. Raj Kamal, ”Internet of Things: Architecture and Design”, McGraw Hill Education, 1stedition, 2017.

Course Outcomes:

1. Able to understand the application areas of IOT ·
2. Able to realize the revolution of Internet in Mobile Devices,Sensor Networks ·
3. Able to understand building blocks of Internet of Things and characteristics.

CSE / CI 314 D	Microprocessor and Microcontrollers

Course Objectives:

1. To understand the architecture of 8085 microprocessor.
2. To learn 8086 architecture Instruction set
3. To learn and understand 8051 Architecture assembly Language programming

UNIT – I

MICROPROCESSOR: Introduction to microcomputers and microprocessors, introduction and architecture of 8086 family, addressing modes, instruction description and assembler directives of 8086 microprocessors.

UNIT - II

8086 PROGRAMMING AND SYSTEM CONNECTIONS: Program development steps, writing programs for use with an assembler, assembly language program development tools, writing and using procedures and assembler macros. 8086 interrupts and interrupt responses.

UNIT - III

Digital Interfacing: Programmable parallel ports, handshake IO, interface Microprocessor to keyboards.
Analog interfacing: DAC principle of operation, specifications and different types of DAC's and interfacing.
Programmable devices: Introduction to Programmable peripheral devices 8255, 8254, 8259, 8251, DMA data transfer, RS232 communication standard.

UNIT-IV

Micro controllers:Introductionto Micro controllers,comparing microprocessors and microcontrollers Architecture of 8051, Registers, Pin configuration of 8051, I/O Ports, Memory Organization, Addressing Modes.

UNIT - V

Programming & Interfacing 8051- Instruction set, Assembly language Programming,Counters& Timers, Serial data Communication– Interrupts,
Interfacing of 8051 – keyboard, Displays, ADC converters.

REFERENCE BOOKS:

1. Microprocessor architecture programming & applications with the 8085,S.Ramesh Gaonkar, PRI Publishers. 6th Edition
2. Advanced Microprocessors & Peripheral interfacing, Ray Bhurchandi, 3rd edition, MC graw hill Publications
3. The INTEL Microprocessors, Brey, 6th edition, PHI Publishers 4. The 8051 Microcontroller and architecture, Kenneth J. Ayala, PRI Publishers 2nd edition
4. Microprocessor and Microcontrollers, N.Senthil Kumar, M.Saravanan, S.Jeevanathan, Oxford Publishers. 1st Edition, 2010
5. The X86 Microprocessors, Architecture, Programming and Inerfacing, Lyla B. Das, Pearson Publications, 2010

Course Outcomes:

After completion of this subject the students will be able to:
1. Do programming with 8086 microprocessors Understand concepts of Intel x86 series of advanced Processors.
2. Able to understand the basic concepts of 8051 architecture Design and implement some specific real time applications Using 8051 Microcontroller.

CSE / CI 315 A	Computer Graphics

Course Objectives:

The student should be made to:

- Gain knowledge about graphics hardware devices and software used.
- Understand the two-dimensional graphics and their transformations.
- Understand the three-dimensional graphics and their transformations.
- Be familiar with understand clipping techniques.

UNIT-I:

2D PRIMITIVES: Elements of pictures created in computer graphics, Graphics input primitives and devices.

Drawing primitives in open GL and Basic open GL programming, open GL basic Graphics primitives, Output primitives – Line, Circle and Ellipse drawing algorithms – Attributes of output primitives.

UNIT-II:

2D GEOMETRIC TRANSFORMATIONS: 2D Viewing – Window-Viewport Transformation - Two dimensional Geometric transformations– Line, Polygon, Curve and Text clipping algorithms.

UNIT-III:

3D CONCEPTS: Three-dimensional object representations – Polygon surfaces- Polygon tables- Plane equations – Polygon meshes; Curved Lines and surfaces, Quadratic surfaces; Blobby objects; Spline representations – Bezier curves and surfaces -B-Spline curves and surfaces.

TRANSFORMATION AND VIEWING: Three dimensional geometric and modelling transformations; Three-dimensional viewing

UNIT-IV:

MULTIMEDIA BASICS: Introduction and definitions ,applications , elements , Animations , Compression ,Types of Compressions: Lossless , Lossy , Video compression , Image Compression , Audio compression , Data and file format standards –,Multimedia data structures: KD Trees ,R trees.

UNIT-V:

MULTIMEDIA AUTHORING AND APPLICATIONS: Creating interactive multimedia, Multimedia Authoring Systems, Multimedia Authoring, Software Applications, Video On demand, Virtual Reality, Augmented Reality, Content based retrieval in digital libraries.

Reference Books:

1. John F. Hughes, Andries Van Dam, Morgan Mc Guire ,David F. Sklar , James D. Foley, Steven K. Feiner and Kurt Akeley ,"Computer Graphics: Principles and Practice", , 3rd Edition, Addison-Wesley Professional,2013.

2. Donald D. Hearn, M. Pauline Baker and Warren Carithers, ―Computer Graphics with OpenGL‖, Fourth Edition, Pearson Education, 2010.
3. Ze-Nian Li and Mark S.Drew, ―Fundamentals of Multimedia‖, First Edition, Pearson Education, 2007.
4. F.S.Hill, ―Computer Graphics using OPENGL‖, Second edition, Pearson Education, 2003.
5. Prabhat K Andleigh, Kiran Thakrar, ―Multimedia systems design‖, First Edition, PHI, 2007.

Course Outcomes:

At the end of the course, the student should be able to:

- Design two-dimensional graphics.
- Apply two dimensional transformations.
- Design three-dimensional graphics.
- Apply three dimensional transformations.
- Apply clipping techniques to graphics.

CSE / CI 315 B	Advanced Computer Architecture

Course Objectives:

- To make students know about the Parallelism concepts in Programming
- To give the students an elaborate idea about the different memory systems and buses.
- To introduce the advanced processor architectures to the students.
- To make the students know about the importance of multiprocessor and multicomputers.
- To study about data flow computer architectures

Unit-I

Theory of Parallelism: Parallel computer models, The State of Computing, Multiprocessors and Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks.

Program and network properties: Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

Unit-II

Principals of Scalable performance: Principals of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speedup performance laws, Scalability Analysis and Approaches, Hardware Technologies.

Processes and Memory Hierarchy: Advanced Processor Technology, Superscalar and Vector Processors, Memory Hierarchy Technology, Virtual Memory Technology.

Unit-III

Bus Cache and Shared memory: Bus Cache and Shared memory, Bus systems, Cache Memory organizations, Shared- Memory Organizations, Sequential and weak consistency models.

Pipelining and superscalar techniques: Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

Unit-IV

Parallel and Scalable Architectures: Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms.

Multivector and SIMD computers: Vector Processing Principals, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations.

Unit-V

Scalable, Multithreaded and Dataflow Architectures: Latency-hiding techniques, Principals of Multithreading, Fine-Grain Multicomputers, Scalable and multithreaded Architectures, Dataflow and hybrid Architectures.

REFERENCE BOOKS:

1. Kai Hwang and Naresh Jotwani, Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability, McGraw Hill Education 3/e. 2015.

2. Computer Architecture, Fourth edition, J. L. Hennessy and D.A. Patterson. ELSEVIER.

3. Advanced Computer Architectures, S.G. Shiva, Special Indian edition, CRC, Taylor &Francis.

4. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.

5. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.

Course Outcomes:

- Discuss memory organization and mapping techniques.
- Describe architectural features of advanced processors.
- Interpret performance of different pipelined processors.
- Explain data flow in arithmetic algorithms.

CSE / CI 315 C	Software Engineering

Course Objectives

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.

- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of proc.ess: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models. Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

UNIT - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. System models: Context models, behavioral models, data models, object models, structured methods.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT - IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT - V

Metrics for Process and Products: Software measurement, metrics for software quality. Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

REFERENCE BOOKS:

1. Software Engineering, A practitioner’s Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.

2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education. 4. Desikan and G. Ramesh, “Software Testing: Principles and Practices”, Pearson Education.
5. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
6. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.
7. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education
8. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India

Course Outcomes:

- Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).

- Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.

- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report.

CSE / CI 315 D	Distributed Systems

Course Objectives:

- To learn the principles, architectures, algorithms and programming models used in distributed systems.
- To examine state-of-the-art distributed systems, such as Google File System.
- To design and implement sample distributed systems.

UNIT I

Characterization of Distributed Systems: Introduction, Examples of Distributed systems, Resource sharing and web, challenges.

System Models: Introduction, Architectural and Fundamental models.

UNIT II

Time and Global States: Introduction, Clocks, Events and Process states, Synchronizing physical clocks, Logical time and Logical clocks, Global states, Distributed Debugging.

Coordination and Agreement: Introduction, Distributed mutual exclusion, Elections, Multicast Communication, Consensus and Related problems.

UNIT III

Inter Process Communication: Introduction, The API for the internet protocols, External Data Representation and Marshalling, Client-Server Communication, Group Communication.

Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications.

UNIT IV

Distributed File Systems: File service architecture - network file system- Andrew file system recent advances

Transactions and concurrency control: nested transactions, locks, optimistic concurrency control, comparison of methods for concurrency control

UNIT V

Distributed Transactions: Flat and Nested Distributed Transactions, distributed deadlocks, transactions recovery.

Replication System model and group communication: fault tolerant services, transactions with replicated data.

REFERENCE BOOKS:

a. Distributed Systems, Concepts and Design, George Coulouris, J Dollimore and Tim Kindberg, Pearson Education, 5th Edition,2012.
b. Andrew S.Tanenbaum, Maarten Van Steen, ―Distributed Systems‖, Third Edition (2017), Pearson Education/PHI..
c. Distributed Systems, An Algorithm Approach, Sikumar Ghosh, Chapman & Hall/CRC, Taylor & Fransis Group, 2007.

Course Outcomes:

- Students will identify the core concepts of distributed systems: the way in which several machines orchestrate to correctly solve problems in an efficient, reliable and scalable way.

- Students will examine how existing systems have applied the concepts of distributed systems in designing large systems, and will additionally apply these concepts to develop sample systems.

CSE / CI 351	Java Lab

COURSE OBJECTIVES:

1. To teach fundamentals of object oriented programming in Java. Understand various concepts of JAVA.
2. To familiarize Java environment to create, debug and run simple Java programs.
3. To demonstrate java compiler and eclipse platform and learn how to use Net Beans IDE to create Java Application.

Programs:

1. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer. (use Scanner class to read input)
2. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.
3. Write a Java program to multiply two given matrices.
4. Write a Java program that checks whether a given string is a palindrome or not.
5. Write a Java program to create a Student class and find the grade of the student.
6. Write a java program to create an abstract class named Shape contains number Of Sides ( ) method and Trapezoid, Triangle and Hexagon classes extends the class Shape.
7. Write a Java program to read copy content of one file to other by handling all file related exceptions.
8. Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
9. Write a Java program that reads a file and displays the file on the screen.
10. Write a Java program that displays the number of characters, lines and words in a text file.
11. Write a Java program that creates three threads. First thread displays “Good Morning” everyone second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.
12. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.
13. Write a Java program for handling mouse events.
14. Write a Java program for handling key events using Adapter classes
15. Develop simple calculator using Swings.

COURSE OUTCOMES:

At the end of the course students will be able to:

1. Implement Object oriented features using Java
2. Apply the concept of polymorphism and inheritance.
3. Implement exception handling
4. Develop network and window application using awt and swings.

CSE / CI 352	Design & Analysis of Algorithms Lab

1. Write a java program for merge sort using divide and conquer technique.
2. Write a java program for binary search using divide and conquer technique.
3. Write a java program for maximum and minimum problem using divide and conquer technique.
4. Write a java program for solving knapsack problem using greedy method
5. Write a java program for solving single source shortest path problem using greedy method
6. Write a java program for solving multistage graph problem using dynamic method
7. Write a java program for solving string editing problem using dynamic method
8. Write a java program for solving matrix chain multiplication problem using dynamic method
9. Write a java program for solving 4-queens problem using backtracking
10. Write a java program for solving sum of subsets problem using backtracking
11. Write a java program for solving graph colouring problem using backtracking
12. Write a java program for solving Hamiltonian cycle problem using backtracking
13. Write a java program for solving travelling sales man problem using branch and bound.

CSE / CI 353 A	VHDL Lab

VHDL Modelling and Synthesis of the following Experiments

1. Logic gates
2. Adders (Hall adder & Full Adder)
3. Code Converter (Binary to Gray & Gray to Binary)
4. 4x16 Decoder
5. 16x4 Encoder
6. Comparator
7. Arithmetic Logic Unit (ALU)
8. BCD to 7-Segment Display
9. Multiplexer/De-multiplexer
10. Flip Flops: JK/T/D
11. Counter
12. Moore state Machine
13. Mealy State Machine
14. Traffic light controller
15. Universal Asynchronous Receiver Transmitter (UART)

CSE / CI 353 B	Digital Signal Processing Lab

Experiments Based on Tool Boxes

1. Simulation of AM.
2. Simulation of FM.
3. Simulation of LPF and HPF.
4. Fourier Transforms.
5. Simulation of M-ary PSK.
6. Simulation of DPCM.
7. Evaluation of DFT and IDFT of 16 Sample Sequence using DIT Algorithm.
8. Evaluation of DFT and IDFT of 16 Sample Sequence using DIF Algorithm.
9. Design of IIR Butterworth Filter using Impulse Invariant Method.
10. Design of FIR Filter using Windowing Technique.
11. Convolution of Two Signals.
12. Correlation of Two Signals.
13. DFT Analysis of a Noise Corrupted Signal.

CSE / CI 353 C	Internet of Things Lab

Digital Sensors:

1. Write an Arduino/python program for LED RED, GREEN and BLUE sensors
2. Write an Arduino/python program for touch sensor
3. Write an Arduino/python program for push button sensor
4. Write an Arduino/python program for motion sensor
5. Write an Arduino/python program for buzzer ringing based on the input

Analog Sensors:

1. Write an arduino/python program for temperature sensor
2. Write an arduino/python program for gas sensor
3. Write an arduino/python program for rotation sensor
4. Write an arduino/python program for light sensor
5. Write an arduino/python program for ultrasonic sensor
6. Write an arduino/python program for moisture sensor
7. Write an arduino/python program for sound sensor
8. Write an arduino/python program for magnetic sensor
9. Write an arduino/python program for sending message to the mobile

Note: Can use any simulation tools for implementing above list of programs.

CSE / CI 353 D	Microprocessor and Microcontrollers Lab

Course Objectives:

The course should enable the students to:

1: Introduce the programming and interfacing techniques of 8086 microprocessor.

2: Analyse the basic concepts and programming of 8051 microcontroller.

Course Outcomes:

Students will be able to:

1: Develop 8086 programming skills in assembly language.

2: Understand the instruction set of 8051 microcontrollers, and have the ability to program 8051 using proper simulation tools.

List of Experiments:

Experiments Based on ALP (8086):

1. Programs on Data Transfer Instructions.

2. Programs on Arithmetic and Logical Instructions.

3. Programs on Branch Instructions.

4. Programs on Subroutines.

5. Sorting of an Array.

6. Programs on Interrupts (Software and Hardware).

7. 8086 Programs using DOS and BIOS Interrupts.

Experiments Based on Interfacing & Microcontroller (8051):

8. DAC Interface-Waveform generations.

9. Stepper Motor Control.

10. Keyboard Interface / LCD Interface.

11. Data Transfer between two PCs using RS.232 C Serial Port.

12. Programs on Data Transfer Instructions using 8051 Microcontroller.

13. Programs on Arithmetic and Logical Instructions using 8051 Microcontroller.

14. Applications with Microcontroller 8051.

CSE / CI 354	Mobile Application Development Lab

OBJECTIVES:

- To understand the components and structure of mobile application development frameworks for Android and windows OS-based mobiles.
- To understand how to work with various mobile application development frameworks.
- To learn the basic and important design concepts and issues of development of mobile applications.
- To understand the capabilities and limitations of mobile devices.

LIST OF EXPERIMENTS:

1. Develop an application that uses GUI components, Font and Colours.
2. Develop an application that uses Layout Managers and event listeners.
3. Develop an application that makes use of databases.
4. Develop an application that makes use of Notification Manager.
5. Develop a native application that uses GPS location information.
6. Implement an application that for basic calculator.
7. Implement an application that creates an alert upon receiving a message.
8. Write a mobile application that makes use of RSS feed.
9. Develop a mobile application to send an email.
10. Develop a Mobile application for simple needs (Mini Project).

OUTCOMES:

Upon Completion of the course, the students will be able to:

- Develop mobile applications using GUI and Layouts.
- Develop mobile applications using Event Listener.
- Develop mobile applications using Databases.
- Develop mobile applications using RSS Feed, Internal/External Storage, SMS and GPS.
- Analise and discover own mobile app for simple needs.

REFERENCES:

1. Build Your Own Security Lab, Michael Gregg, Wiley India

Dr YSR ANU College of Engineering & Technology

ACHARYA NAGARJUNA UNIVERSITY

SCHEME OF INSTRUCTION AND EXAMINATION, w.e.f. 2019-2022 (R19)

COMPUTER SCIENCE & ENGINEERING (CSE)

COMPUTER SCIENCE & INFORMATION TECHNOLOGY(CI)

III/IV B.TECH - SEMESTER II (R19)(R19 Regulation - Structure & Syllabus)

S.No.	Course Details		Category	Scheme of Instruction			Scheme of Examination		
	Code	Subject Name		Hours in a Week			Marks		Credits
				L	T	P	Internal	External	
1	CSE / CI 321	Cryptography & Network Security	PC	3	0	0	40	60	3
2	CSE / CI 322	Artificial Intelligence & Machine Learning	PC	3	0	0	40	60	3
3	CSE / CI /ECE 323	Computer Networks	PC	3	0	0	40	60	3
4	CSE / CI 324	Job Elective – 2	JE-2	3	0	0	40	60	3
5	CSE / CI 325	Professional Elective-2	PE-2	3	0	0	40	60	3
6	CSE / CI 361	AI&ML Lab	PC	0	0	3	40	60	1.5
7	CSE / CI/ 362	Computer Networks Lab	PC	0	0	3	40	60	1.5
8	CSE / CI 363	JE2 Lab	JE 2 Lab	0	0	3	40	60	1.5
9	CSE / CI 364	.NET programming Lab	SKILL	0	0	3	40	60	2
Total Credits									21.5`

JE-2 (Lab Oriented)

E. Soft Computing
F. Data Engineering
G. Digital Image Processing
H. Software Testing Methodologies

PE-2

E. Wireless Networks
F. High Performance Computing
G. Cloud Computing Architecture and Its Applications
H. Advanced Databases

CSE / CI 321	Cryptography & Network Security

Course Objectives:

- To understand basics of Cryptography and Network Security.
- To be able to secure a message over insecure channel by various means.
- To learn about how to maintain the Confidentiality, Integrity and Availability of a data.
- To understand various protocols for network security to protect against the threats in the networks.

UNIT I

Need for Security – Attacks, Services and Mechanisms, Classical encryption Techniques, Block ciphers and data encryption, standard. Advanced encryption standard, evaluation criteria of AES, Symmetric ciphers- multiple encryption and triple DES, Block cipher modes of operation, Stream ciphers and RC4, Stream ciphers – Blowfish, Modern Symmetric encryption - IDEA,

UNIT-II

Introduction to number theory- Prime numbers, Fermat's and Euler's theorems, Chinese Remainder Theorem, Discrete logarithms, Public key cryptography - Principles of public key cryptosystems and RSA, Key management, Diffie-Hellman key exchange, Elliptic curve arithmetic, Elliptic curve cryptography, Key Distribution, Message authentication and Hash functions-Authentication functions, Security and Hash functions and MACs, HMAC, CMAC, Digital signatures and authentication protocols, Authentication protocols, Digital signature standard.

UNIT III

Attacks- Denial-of-service/Distributed denial-of-service attacks, Back door, Spoofing, Man-inthe-middle, Replay, TCP/Hijacking, Fragmentation attacks, Weak keys, Mathematical attacks, Social engineering, Port scanning, Dumpster diving, Birthday attacks, Password guessing, Software exploitation, Inappropriate system use, Eavesdropping, War driving, TCP sequence number attacks, War dialing/demon dialing attacks.

UNIT IV

Other public Key Cryptosystems –Public key algorithms using GMP, Introduction to packet sniffing tool, Architecture of SSL, Attacks on SSL, Introduction to Intruder detection System, Snort and stenographic tools.

UNIT V

Wireless and IP Security-IEEE 802.11 Wireless Security, WEP, WEP security upgrades, IEEE 802.11i, Wireless application protocol, IP Security architecture, Authentication header, Encapsulating security pay load, combining security associations.

Reference Books:

1. William Stallings, Cryptography and Network Security, Pearson Education, 2006

2. Eric Cole, Dr. Ronald Kurtz and James W. Conley, Network Security Bible, Wiley Publishers, 2009

3. Jason Albanese and Wes Sonnenreich, Network Security Illustrated, MGH Publishers, 2003

COURSE OUTCOMES:

After successful completion of the course, the learners would be able to

- Describe network security services and mechanisms.
- Symmetrical and Asymmetrical cryptography.
- Data integrity, Authentication, Digital Signatures.
- Various network security applications, IPSec, Firewall, IDS, Web security, Email security, and Malicious software etc.

CSE / CI 322	Artificial Intelligence & Machine Learning

Course Objectives:

1. the ability to adapt, contribute and innovate new technologies and systems in the key domains of Artificial Intelligence and Machine Learning.
2. Will be able to successfully pursue higher education in reputed institutions with AI Specialization.
3. Graduates will have the ability to explore research areas and produce outstanding contribution in various areas of Artificial Intelligence and Machine Learning.
4. Graduates will be ethically and socially responsible solution providers and entrepreneurs in the field of Computer Science and Engineering with AI/ML Specialization.

UNIT-I

Introduction: Cousins of Artificial Intelligence, Applications, Stages and types of AI, intelligent agents,

Problem Solving: Solving Problems by Searching, heuristic search techniques, constraint satisfaction problems.

UNIT-II

Game Playing: Minimax search, Evaluation functions, Markov Decision Processes, Reinforcement learning for games.

Knowledge and Reasoning: Building a Knowledge Base: Propositional logic, first order logic, Theorem Proving in First Order Logic. Uncertain Knowledge and Reasoning, Probabilities, Introduction to Natural Language Processing.

UNIT-III

Introduction to machine learning: Concept Learning and the General to Specific Ordering: Concept learning task, concept learning as search, Find-S: finding a Maximally Specific hypothesis, Version Spaces and the Candidate-Elimination algorithm, remarks on Version Spaces and Candidate-Elimination and inductive bias.

Decision Tree Learning: Decision Tree representation, appropriate problems for Decision Tree learning, hypothesis space search in Decision Tree learning, inductive bias in Decision Tree learning and issues in Decision Tree learning.

UNIT-IV

Artificial Neural Networks: Neural Network representations, appropriate problems for Neural Network learning, Perceptrons, Multilayer Networks and the Backpropagation algorithm and remarks on the Backpropagation algorithm.

Bayesian Learning: Bayes theorem and concept learning, maximum likelihood hypotheses for predicting probabilities, minimum description length principle, Bayes optimal classier, Gibbs algorithm, Naive Bayes classier, Bayesian belief networks and EM algorithm.

UNIT-V

Instance Based Learning: Introduction, k-Nearest Neighbour learning, locally weighted regression, radial basis functions, Case Based Reasoning and remarks on Lazy and Eager learning.

Genetic Algorithms: Introduction, hypothesis space search, Genetic programming and models of evolution and learning.

Reference Books:

1. Artificial Intelligence, A modern Approach, Second Edition by Stuart Russell, Peter Norvig.
2. Tom M. Mitchell, "Machine Learning", Mc. Graw Hill Publishing.

Course Outcomes:

After completing this course, the student will be able to:

1. Demonstrate fundamental understanding of artificial intelligence (AI) and expert systems.

2. Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

3. Demonstrate proficiency in applying scientific method to models of machine learning.

4. Discuss the awareness of ANN and different optimizations techniques.

CSE / CI / ECE 323	Computer Networks

Course Objectives:
At the end of the course, the students will be able to:
1. Build an understanding of the fundamental concepts of data communication and computer networking.
2. Understand how errors detected and corrected that occur in transmission
3. How collisions to be handled when many stations share a single channel
4. Know about routing mechanisms and different routing protocols
5. Understand transport layer functions
6. Know about different application layer protocols

UNIT I:
Introduction: Uses of Computer Networks, Network Hardware, LANs, MANs, WANs, Network Software.
Reference Models: The OSI Reference Model, TCP/IP Reference Model, the comparison of OSI, and TCP/IP reference models.
The Physical Layer: Guided transmission media: Magnetic Media, Twisted Pair, Coaxial Cable, and Fiber Optics.

UNIT II:
The Data Link Layer: Data link layer design issues, Error detection and correction, Elementary data link protocols, and Sliding window protocols.
The Medium Access Control Sub layer: The channel allocation problem, multiple access protocols, ETHERNET, and Wireless LANs.

UNIT III:
The Network Layer: Network Layer Design Issues, Routing Algorithms: Shortest Path, Flooding, DVR, and Link State routing algorithm, Congestion Control Algorithms, and Quality of Service. IP protocol and IP address.

UNIT – IV

The Transport Layer: The Transport Service, Elements of Transport Protocols, and the Internet Transport Protocols: UDP- Remote Procedure Call, The Real-Time Transport Protocol, TCP- Introduction to TCP, The TCP Service model, The TCP Protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release, TCP Connection Management Modeling, TCP Transmission Policy, Congestion Control, TCP Timer Management.

UNIT - V

Application Layer: The Domain Name System (DNS) – Resource Records, Name Servers, E-Mail – Architecture and Services, POP3, IMAP, World Wide Web – Architectural Overview, Server side, Uniform Resource Locators, Statelessness and Cookies.

Reference Books:

1.Andrew S Tanenbaum, Computer Networks.4 ed, Pearson Education / PHI.
2.Behrouz A.Forouzan, Data Communications and Networking. 4 ed, TATA McGraw Hill
3.Kurose and Ross, Computer Networks – A Top-down Approach Featuring the Internet.'Pearson Education.

Course Outcomes:

After completing this course the student must demonstrate the knowledge and ability to:

1. Describe the basis and structure of an abstract layered protocol model
2. Independently understand basic computer network technology.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5. Identify the different types of network devices and their functions within a network
6. Understand and building the skills of subnetting and routing mechanisms.
7. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation
8. Understand how the Internet works today.

CSE / CI 324(A)	Soft Computing

Course Objectives:

This course makes the students to Understand

- Fundamentals of Neural Networks & Feed Forward Networks.
- Associative Memories & ART Neural Networks.
- Fuzzy Logic & Systems.
- Genetic Algorithms and Hybrid Systems.

UNIT I

FUZZY SYSTEMS:

Introduction to Neuro – Fuzzy and Soft Computing – Fuzzy Sets – Basic Definition and Terminology – Set-theoretic Operations – Member Function Formulation and Parameterization – Fuzzy Rules and Fuzzy Reasoning – Extension Principle and Fuzzy Relations – Fuzzy If-Then Rules – Fuzzy Reasoning – Fuzzy Inference Systems – Mamdani Fuzzy Models – Sugeno Fuzzy Models – Tsukamoto Fuzzy Models – Input Space Partitioning and Fuzzy Modeling.

UNIT II

OPTIMAIZATION:

Derivative-based Optimization – Descent Methods – The Method of Steepest Descent – Classical Newton's

Method – Step Size Determination – Derivative-free Optimization – Genetic Algorithms – Simulated Annealing – Random Search – Downhill Simplex Search.

UNIT III

NEURALNETWORKS

Supervised Learning Neural Networks – Perceptrons - Adaline – Backpropagation. Mutilayer Perceptrons – Radial Basis Function Networks –Unsupervised Learning Neural Networks – Competitive Learning Networks – Kohonen Self-Organizing Networks – Learning Vector Quantization – Hebbian Learning.

UNIT IV

NEUROFUZZY MODELING:

Adaptive Neuro-Fuzzy Inference Systems – Architecture – Hybrid Learning Algorithm – Learning Methods that Cross-fertilize ANFIS and RBFN – Coactive Neuro Fuzzy Modeling – Framework Neuron Functions for Adaptive Networks – Neuro Fuzzy Spectrum.

UNIT V :

APPLICATIONS OF COMPUTATIONAL INTELLIGENCE:

Printed Character Recognition – Inverse Kinematics Problems – Automobile Fuel. Efficiency Prediction – Soft Computing for Color Recipe Prediction.

REFERENCES:

1.J.S.R.Jang, C.T.Sun and E.Mizutani, "Neuro-Fuzzy and Soft Computing", PHI, 2004, Pearson Education 2004.
2.Timothy J.Ross, "Fuzzy Logic with Engineering Applications", McGraw-Hill, 1997.
3. Davis E.Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y., 1989.
4. S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms.

Course Outcomes:

On completion of this course the students will be able to

- Identify and employ suitable soft computing techniques in classification and optimization problems.
- Design hybrid systems to suit a given real – life problem.

CSE / CI 324(B)	Data Engineering

Objective:

This course gives an introduction to methods and theory for development of data warehouses and data analysis using data mining. Data quality and methods and techniques for pre-processing of data.

Algorithms for classification, clustering and association rule analysis. Practical use of software for data analysis.

UNIT – I

Data Warehouse – Introduction, A Multi-dimensional data model, Data Warehouse Architecture, Data Warehouse Implementation.

Data Mining – Introduction, Data Mining, Kinds of Data, Data Mining Functionalities, Classification of Data Mining Systems, Major issues in Data Mining.

UNIT – II

Data Pre-processing – Data cleaning, Data Integration & Transformation, Data Reduction, Discretization & Concept Hierarchy Generation, Data Mining Primitives.

Mining Association roles in large databases – Association rule mining, mining single-dimensional Boolean Association rules from Transactional Databases, Mining Multi-dimensional Association rules from relational databases & Data Warehouses.

UNIT – III

Cluster Analysis – Introduction, Types of data in Cluster analysis, A categorization of major clustering methods, partitioning methods, Hierarchical methods, Density-Based Methods: DBSCAN, Grid-based Method: STING; Model-based Clustering Method: Statistical approach, Outlier analysis.

UNIT – IV

Classification & Prediction – Introduction, Classification by Decision tree induction, Bayesian Classification,Classification by Back propagation, Other Classification Methods, Prediction, Classifier accuracy.

UNIT-V

Mining Complex Type of Data – Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Mining Spatial Databases, Mining Multimedia Databases, Mining Text Databases, Mining the World Wide Web.

Textbooks:

1.Data Mining Concepts & Techniques – Jiawei Han MichelineKamber – Morgan Kaufmann Publishers.

Reference Books:

1. Data Warehouse Toolkit – Ralph Kinball – John Wiley Publishers.

2. Data Mining (Introductory and Advanced Topics) – Margaret H.Dunham – Pearson Education.
3. Data Warehousing in the real world – A Practical guide for Building decision support systems – Sam Anahory, Dennis Murray – Pearson Education.
4. Introduction to Data Mining with case studies – G.K.Gupta, PHI Publications, 2006

Course Outcomes

- This course gives an introduction to methods and theory for development of data warehouses and data analysis using data mining.
- Data quality and methods and techniques for pre-processing of data.
- Modelling and design of data warehouses.

- Algorithms for classification, clustering and association rule analysis.

CSE / CI 324(C)	Digital Image Processing

Course Objectives:

1. To study the image fundamentals and mathematical transforms necessary for image processing.
2. To study the image enhancement techniques
3. To study image restoration procedures.
4. To study the image compression procedures.

UNIT – I

Digital image fundamentals:

Introduction: Digital Image- Steps of Digital Image Processing Systems-Elements of Visual Perception - Connectivity and Relations between Pixels. Simple Operations- Arithmetic, Logical, Geometric Operations.

Mathematical Preliminaries - 2D Linear Space Invariant Systems - 2D Convolution - Correlation 2D Random Sequence - 2D Spectrum.

UNIT – II

Image transforms and enhancement:

Image Transforms: 2D Orthogonal and Unitary Transforms-Properties and Examples. 2D DFT- FFT – DCT -Hadamard Transform - Haar Transform - Slant Transform - KL Transform -Properties And Examples.Image

Enhancement:- Histogram Equalization Technique- Point Processing-Spatial Filtering-In Space And Frequency -Nonlinear Filtering-Use Of Different Masks.

Unit – III

Image restoration and construction:

Image Restoration: Image Observation and Degradation Model, Circulant And Block Circulant Matrices and Its

Application In Degradation Model - Algebraic Approach to Restoration- Inverse By Wiener Filtering - Generalized

Inverse-SVD And Interactive Methods - Blind Deconvolution-Image Reconstruction From Projections.

Unit – IV

Image compression & segmentation

Image Compression: Redundancy And Compression Models -Loss Less And Lossy.

Loss Less- Variable-Length, Huffman, Arithmetic Coding - Bit-Plane Coding, Loss Less Predictive Coding, Lossy Transform (DCT) Based Coding, JPEG Standard - Sub Band Coding.

Image Segmentation: Edge Detection - Line Detection - Curve Detection - Edge Linking And Boundary Extraction, Boundary Representation, Region Representation And Segmentation, Morphology-Dilation, Erosion, Opening And Closing. Hit And Miss Algorithms Feature Analysis

Unit – V

Color and multispectral image processing

Color Image-Processing Fundamentals, RGB Models, HSI Models, Relationship Between Different Models.

Multispectral Image Analysis - Color Image Processing Three Dimensional Image Processing-Computerized Axial

Tomography-Stereometry-Stereoscopic Image Display-Shaded Surface Display.

Reference Books:

1 Digital Image Processing, Gonzalez.R.C & Woods. R.E., 3/e, Pearson Education, 2008.

2. Digital Image Processing, Kenneth R Castleman, Pearson Education,1995.

3. Digital Image Procesing, S. Jayaraman, S. Esakkirajan, T. Veerakumar, McGraw Hill Education ,2009.

Pvt Ltd, NewDelhi

4. Fundamentals of Digital image Processing, Anil Jain.K, Prentice Hall of India, 1989.

5. Image Processing, Sid Ahmed, McGraw Hill, New York, 1995.

Course Outcomes:

1. Review the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques.
5. Interpret Image compression standards.
6. Interpret image segmentation and representation technique.

CSE / CI 324(D)	Software Testing Methodologies

COURSE OBJECTIVES

The objective of this course is:

1. Introducing various design approaches, models and metrics.
2. Presenting various techniques and strategies of software testing and inspection and pointing out the importance of testing in achieving high-quality software.
3. Understand concept of reliability, the role it plays in software engineering, and how it is modeled and measured.
4. Showing how software product and process are managed and controlled for maintaining software quality assurance.
5. Highlighting importance of software maintenance, restructuring, and reengineering.
6. Presenting the various techniques of software cost estimation and risk assessment.

Unit I: Introduction

Faults, Errors, and Failures, Basics of software testing, Testing objectives, Principles of testing, Requirements, behavior and cor-rectness, Testing and debugging, Test metrics and measurements, Verification, Validation and Testing, Types of testing, Software Quality and Reliability, Software defect tracking.

Unit II: White Box and Black Box Testing

White box testing, static testing, static analysis tools, Structural testing: Module/Code functional testing, Code coverage testing, Code complexity testing, Black Box testing, Requirements based testing, Boundary value analysis, Equivalence partitioning, state/ graph based testing, Model based testing and model checking, Differences between white box and Black box testing.

Unit III: Integration, System, and Acceptance Testing

Top down and Bottom up integration, Bi-directional integration, System integration, Scenario Testing, Defect Bash, Functional versus Non-functional testing, Design/Architecture verification, Deployment testing, Beta testing, Scalability testing, Reliability testing, Stress testing, Acceptance testing: Acceptance criteria, test cases selection and execution,

Unit IV: Test Selection & Minimization for Regression Testing

Regression testing, Regression test process, Initial Smoke or Sanity test, Selection of regression tests, Execution Trace, Dynamic Slicing, Test Minimization, Tools for regression testing, Ad hoc Testing: Pair testing, Exploratory testing, Iterative testing, Defect seeding.

Unit V: Test Management and Automation

Test Planning, Management, Execution and Reporting, Software Test Automation: Scope of automation, Design & Architecture for automation, Generic requirements for test tool framework, Test tool selection, Testing in Object Oriented Systems.

REFERENCES:

1. Desikan and G. Ramesh, “Software Testing: Principles and Practices”, Pearson Education.
2. Aditya P. Mathur, “Fundamentals of Software Testing”, Pearson Education.
3. Naik and Tripathy, “Software Testing and Quality Assurance”, Wiley ,K. K. Aggarwal and Yogesh Singh, “Software Engineering”, New Age International Publication.
4. Software Testing techniques – Baris Beizer, Dreamtech, second edition.
5. Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.
6. Paul C. Jorgensen, Software Testing: A Craftsman‟s Approach, 3rd Edition, CRC Press, 2007.
7. Boris Beizer, Software Testing Techniques, Dreamtech, 2009.

COURSE OUTCOMES

At the end of the course student will be able to:

1. Use the appropriate methods and tools for estimating software cost.
2. Understand the difference between different software design models and techniques and how to apply them.
3. Recognize the importance of software reliability and how we can design dependable software, and what measures are used.
4. Understand the principles and techniques underlying the process of inspecting and testing software and making it free of errors and tolerable.
5. Recognize the importance of software standards and quality assurance.
6. Apply the appropriate software evolution methods for maintaining, restructuring available software and managing soft-ware development.

CSE / CI 325(A)	Wireless Networks

Course Objectives:

- The students should get familiar with the wireless/mobile market and the future needs and challenges.

- To get familiar with key concepts of wireless networks, standards, technologies and their basic Operations.

- To learn how to design and analyze various medium access.

- To learn how to evaluate MAC and network protocols using network simulation software tools.

- The students should get familiar with the wireless/mobile market and the future needs and challenges.

UNIT-I: Introduction- Wireless Networking Trends, Key Wireless Physical Layer Concepts, Multiple Access Technologies - CDMA, FDMA, TDMA, Spread Spectrum technologies, Frequency reuse, Radio Propagation and Modelling, Challenges in Mobile Computing: Resource poorness, Bandwidth, energy etc. Wireless Local Area Networks- IEEE 802.11 Wireless LANs Physical & MAC layer, 802.11 MAC Modes (DCF& PCF) IEEE 802.11 standards, Architecture & protocols, Infrastructure vs. Adhoc Modes, Hidden Node & Exposed Terminal Problem, Problems, Fading Effects in Indoor and outdoor WLANs, WLAN Deployment issues.

UNIT–II: Wireless Cellular Networks-1G and 2G, 2.5G, 3G, and 4G, Mobile IPv4, Mobile IPv6, TCP over Wireless Networks, Cellular architecture, Frequency reuse, Channel assignment strategies, Handoff strategies, Interference and system capacity, Improving coverage and capacity in cellular systems, Spread spectrum Technologies.

UNIT-III: WiMAX (Physical layer, Media access control, Mobility and Networking), IEEE802.22 Wireless Regional Area Networks, IEEE 802.21 Media Independent Handover Overview, Wireless Sensor Networks: Introduction, Application, Physical, MAC layer and Network Layer, Power Management, Tiny OS Overview.

UNIT–IV: Wireless PANs-Bluetooth AND Zigbee, Introduction to Wireless Sensors.

UNIT–V: Security-Security in wireless Networks Vulnerabilities, Security techniques, Wi-Fi Security, DoS in wireless communication. Advanced Topics: IEEE 802.11x and IEEE 802.11i standards, Introduction to Vehicular Adhoc Networks

Reference Books:

1. Schiller J., Mobile Communications, Addison Wesley, 2000. 2. Stallings W., Wireless Communications and Networks, Pearson Education 2005.

2. Stojmenic Ivan, Handbook of Wireless Networks and Mobile Computing, John Wiley and Sons Inc, 2002.

3. Yi Bing Lin and Imrich Chlamtac, Wireless and Mobile Network Architectures, John Wiley and Sons Inc, 2000.

4. Pandya Raj, Mobile and Personal Communications Systems and Services, PHI, 2000.

Course Outcomes:

At the end of the course, student will be able to :

- Demonstrate advanced knowledge of networking and wireless networking and understand various types of wireless networks, standards, operations and use cases.
- Be able to design WLAN, WPAN, WWAN, Cellular based upon underlying propagation and performance analysis.
- Demonstrate knowledge of protocols used in wireless networks and learn simulating wireless networks.
- Design wireless networks exploring trade-offs between wire line and wireless links.
- Develop mobile applications to solve some of the real-world problems.

CSE / CI 325(B)	High Performance Computing

Course Objectives:

1. Provide systematic and comprehensive treatment of the hardware and the software high performance techniques involved in current day computing.
2. Introduce the fundamentals of high-performance computing with the graphics processing units and many integrated cores using their architectures and corresponding programming environments.
3. Introduce the learner to fundamental and advanced parallel algorithms through the GPU and MIC programming environments

4. Provide systematic and comprehensive treatment of the components in the pipeline that extract instruction level parallelism.
5. Provide a strong foundation on memory hierarchy design and trade-offs in both uniprocessor and multiprocessors.
6. Illustrate the cache coherence and consistency problems in multiprocessors, and their existing solutions.

UNIT- I

Graphics Processing Units: Introduction to Heterogeneous Parallel Computing. GPU architecture. Thread hierarchy. GPU Memory Hierarchy.

UNIT-II

GPU Programming: Vector Addition, Matrix Multiplication algorithms. 1D, 2D, and 3D Stencil Operations. Image Processing algorithms – Image Blur, Gray scaling. Histogramming, Convolution, Scan, Reduction techniques.

UNIT-III

Many Integrated Cores: Introduction to Many Integrated Cores. MIC, Xeon Phi architecture. Thread hierarchy. Memory Hierarchy. Memory Bandwidth and performance considerations.

UNIT-IV

Shared Memory Parallel Programming: Symmetric and Distributed architectures. OpenMP Introduction. Thread creation, Parallel regions. Work-sharing, Synchronization.

UNIT-V:

Message Passing Interface: MPI Introduction. Point to Point communication, Collective communication. Data grouping for communication.

Reference Books:

1. Wen-Mei W Hwu, David B Kirk, Programming Massively Parallel Processors A Hands-on Approach, Morgann Kaufmann, 3e.
2. Rezaur Rahman, Intel Xeon Phi Coprocessor Architecture and Tools, Apress Open, 2013.
3. Barbara Chapman, Gabriele Jost, Ruud van der Pas, Using OpenMP, MIT Press, 2008.
4. Gropp, Lusk, Skjellum, Using MPI, Using MPI, 2014.

Course Outcomes:

1. The learner will be able to design, formulate, solve and implement high performance versions of standard single threaded algorithms.
2. The learner will know and will be able to demonstrate the architectural features in the GPU and MIC hardware accelarators.

3. The learner will be able to design programs to extract maximum performance in a multicore, shared memory execution environment processor.
4. The learner will be able to design and deploy large scale parallel programs on tightly coupled parallel systems using the message passing paradigm.

CSE / CI 325(C)	Cloud Computing Architecture and Its Applications

Course outcomes:

The students should be able to:

• Explain cloud computing, virtualization and classify services of cloud computing

• Illustrate architecture and programming in cloud

• Describe the platforms for development of cloud applications and List the application of cloud.

Unit-1: Introduction to Cloud Computing

Introduction ,Cloud Computing at a Glance, The Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms and Technologies, Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka.

Unit-2: Virtualization and Cloud Computing Architecture

Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V.

Cloud Computing Architecture, Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Definition, Cloud Interoperability and Standards Scalability and Fault Tolerance Security, Trust, and Privacy Organizational Aspects.

Unit-3: Cloud Application Platform

Aneka: Cloud Application Platform, Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools

Unit-4: Concurrent and Data Intensive Computing

Thread Programming, Introducing Parallelism for Single Machine Computation, Programming Applications with Threads, What is a Thread?, Thread APIs, Techniques for Parallel Computation with Threads, Multithreading with Aneka, Introducing the Thread Programming Model, Aneka Thread vs. Common Threads, Programming Applications with Aneka Threads, Aneka Threads Application Model, Domain Decomposition: Matrix Multiplication, Functional Decomposition: Sine, Cosine, and Tangent.

High-Throughput Computing: Task Programming, Task Computing, Characterizing a Task, Computing Categories, Frameworks for Task Computing, Task-based Application Models, Embarrassingly Parallel Applications, Parameter Sweep Applications, MPI Applications, Workflow Applications with Task Dependencies, Aneka Task-Based Programming, Task Programming Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.

UNIT-5: Data Intensive Computing and Cloud Platforms in Industry

Map-Reduce Programming, What is Data-Intensive Computing?, Characterizing Data-Intensive Computations, Challenges Ahead, Historical Perspective, Technologies for Data-Intensive Computing, Storage Systems, Programming Platforms, Aneka Map Reduce Programming, Introducing the Map Reduce Programming Model, Example Application.

Cloud Platforms in Industry, Amazon Web Services, Compute Services, Storage Services, Communication Services, Additional Services, Google AppEngine, Architecture and Core Concepts, Application Life-Cycle, Cost Model, Observations, Microsoft Azure, Azure Core Concepts, SQL Azure, Windows Azure Platform Appliance.

Text Books:

1. RajkumarBuyya, Christian Vecchiola, and ThamaraiSelvi Mastering Cloud. Computing McGraw Hill Education

Reference Books:

1. Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, Elsevier 2013.

CSE / CI 325(D)	Advanced Databases

OBJECTIVES:

- To understand the design of databases.
- To acquire knowledge on parallel and distributed databases and its applications.
- To study the usage and applications of Object Oriented and Intelligent databases.
- To understand the emerging databases like Mobile, XML, Cloud and Big Data

UNIT I

PARALLEL AND DISTRIBUTED DATABASES

Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems.

Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism – Design of Parallel Systems

Distributed Database Concepts: Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing

UNIT II

INTELLIGENT DATABASES

Active Databases: Syntax and Semantics (Starburst, Oracle, DB2)- Taxonomy- Applications- Design Principles for Active Rules.

Temporal Databases: Overview of Temporal Databases TSQL2- Deductive Databases- Recursive Queries in SQL.

Spatial Databases: Spatial Data Types - Spatial Relationships - Spatial Data Structures- Spatial Access Methods - Spatial Database Implementation.

UNIT III

XML DATABASES

XML Databases: XML Data Model – DTD – XML Schema – XML Querying – Web Databases – Open Database Connectivity.

UNIT IV

MOBILE DATABASES

Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management - Location Dependent Data Distribution - Mobile Transaction Models - Concurrency Control - Transaction Commit Protocols

UNIT V

MULTIMEDIA DATABASES

Multidimensional Data Structures – Image Databases – Text / Document Databases – Video Databases – Audio Databases – Multimedia Database Design.

REFERENCES:

1. C.J.Date, A.Kannan, S.Swamynathan, —An Introduction to Database Systems‖, Eighth Edition, Pearson Education, 2006.

2. Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T.Snodgrass, V.S.Subrahmanian, Roberto Zicari, —Advanced Database Systems‖, Morgan Kaufmann publishers,2006.

3. Henry F Korth, Abraham Silberschatz, S. Sudharshan, —Database System Concepts‖, Sixth Edition, McGraw Hill, 2011.

4. R. Elmasri, S.B. Navathe, —Fundamentals of Database Systems‖, Sixth Edition, Pearson Education/Addison Wesley, 2010.

5. Vijay Kumar, —Mobile Database Systems‖, John Wiley & Sons, 2006.

OUTCOMES:

Upon completion of this course, a students should be able:

- To develop skills on databases to optimize their performance in practice.
- To analyze each type of databases and its necessity
- To design faster algorithms in solving practical database problems

CSE / CI 361	AI&ML Lab

Learn Prolog/ Python programming/R- language and implement programs on below topics

1. Write a LISP program to solve the water-jug problem using heuristic function.
2. Create a compound object using Turbo Prolog.
3. Write a program to use of BEST-FIRST SEARCH applied to the eight puzzle problem.
4. a) Implement A* Search algorithm.
   b) Implement AO* Search algorithm.
5. a) For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
   b) Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
6. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
7. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
8. a) Classification: Identifying to which category an object belongs to.
   b) Regression: Predicting a continuous-valued attribute associated with an object.
   c) Clustering: Automatic grouping of similar objects into sets
   d) Dimensionality reduction: Reducing the number of random variables to consider.
9. Pre-processing: Feature extraction and normalization.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

CSE / CI / 362	Computer Networks Lab

1. Study of different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.
2. Study of Network Devices in Detail.
3. Study of network IP.
4. Connect the computers in Local Area Network.
5. Study of basic network command and Network configuration commands.
6. Performing an Initial Switch Configuration
7. Performing an Initial Router Configuration

8. Configuring and Troubleshooting a Switched Network
9. Connecting a Switch
10. Using the Cisco IOS Show Commands
11. Interpreting Ping and Traceroute Output
12. Placing ACLs
13. Implementing an IP Addressing Scheme
14. Examining Network Address Translation (NAT)
15. Configuring RIP

CSE / CI 363 (A)	Soft Computing Lab

1. Create a perceptron with appropriate number of inputs and outputs. Train it using fixed increment learning algorithm until no change in weights is required and output the final weights.
2. a) Write a program to implement artificial neural network without back propagation.
   b) Write a program to implement artificial neural network with back propagation.
3. Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relation by Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.
4. Implement travelling sales person problem (tsp) using genetic algorithms**.**
5. Plot the correlation plot on dataset and visualize giving an overview of relationships among data on soya bins data. Analysis of covariance: variance (ANOVA), if data have categorical variables on iris data.
6. Implement linear regression and multi-regression for a set of data points
7. Implement crisp partitions for real-life iris dataset.
8. Write a program to implement Hebb's rule Write a program to implement Delta rule.
9. Write a program to implement logic gates.
10. Implement SVM classification by fuzzy concepts.

CSE / CI 363(B)	Data Engineering Lab

1. Analyzing data with ROLLAP, CUBE.

2. Perform Cube slicing – come up with 2-D view of data.

3. Apply Drill-down or Roll-down- going from summary to more detailed data.
4. Apply Dicing – projecting 2-D view of data.
5. Creating Star Schema/snowflake Schema.

6. Create and populate FACT table.
7. Build dimensions using tool.
8. ETL : Extraction Options
    a) Perform Full extraction
    b) Perform Incremental extraction
    c) Perform Change Data Capture(CDC)
9. ETL: Transformation Options
    a) Apply Transformation: during extraction, in staging area, during load, etc.
    b) Apply Multi-state transformation
    c) Apply Pipelined transformation

CSE / CI 363(C)	Digital Image Processing Lab

1. Write a MATLAB program to extract different Attributes of an Image.
2. Write a MATLAB program for image enhancement
3. Write a MATLAB program for Image Negation.
4. Write a MATLAB program for image compression
5. Write a MATLAB program for colour image processing
6. Write a MATLAB program for image segmentation
7. Write a MATLAB program for image morphology
8. Write a MATLAB program for Image Restoration
9. Write a MATLAB program for Power Law Transformation.
10. Write a MATLAB program for Histogram Mapping and Equalization.
11. Write a MATLAB program for Image Smoothening and Sharpening.
12. Write a MATLAB program for Edge Detection using Sobel, Prewitt and Roberts Operators.
13. Write a MATLAB program for Morphological Operations on Binary Images.
14. Write a MATLAB program for Pseudo Colouring.
15. Write a MATLAB program for Chain Coding.
16. Write a MATLAB program for DCT/IDCT Computation.

CSE / CI 363(D)	Software Testing Methodologies Lab

1. **CONSTRUCTS**

   Write programs in C language to demonstrate the working of the following constructs:
   a) while b) switch c) for d) if-else e) do-while

2. SYSTEM SPECIFICATIONS

a. Study the system specifications of ATM system and report various bugs in it.

b. Study the system specifications of banking application and report various bugs in it.

3. TEST CASES

a. Write the test cases for ATM system.

b. Write the test cases for banking application.

4. **TEST PLAN**

a. Create a test plan document for any application (e.g. Library management system)

5. **TESTING TOOL**

a. Study of any testing tool (e.g. Win runner)

6. **SELENIUM**

a. Study of web testing tool (e.g. Selenium).

7. **BUG TRACKING TOOL**

a. Study of bug tracking tool (e.g. Bugzilla)

8. **BUGBIT**

a. Study of bug tracking tool (e.g. Bugbit).

9. **TEST MANAGEMENT TOOL**

a. Study of any test management tool (e.g. Testdirector)

10**. OPEN SOURCE TESTING TOOL**

a. Study of any Open Source Testing Tool (e.g. Test Link).

11. **AUTOMATED FUNCTIONAL TESTING TOOL**

a. Study of QTP (Quick Test Professional) automated functional testing tool

12. **INTROSPECTION OF MATRIX MULTIPLICATION**

a**.** A program written in C language for matrix multiplication fails, introspect the causes for its failure and write down the possible reasons for its failure.

OBJECTIVES:

This Lab course will help students to achieve the following objectives:

1.Introduce to .Net IDE Component Framework.

2.Programming concepts in .Net Framework.

3.Creating website using ASP.Net Controls.

LIST OF EXPERIMENTS

1. Create a windows form with the following controls Textbox, Radio button, Check box, Command Button

2. Write a program for Menu option.

3. Create a program to connect with database and manipulate the records in the database using ADO .NET

4. Create a program to implement the concepts of OOPS for creating class, inheritance

5. Create a program to perform input validation using procedure.

6. Write a program to open a file and using I/O operations write contents into a file and read the contents from the file.

7. Create a window form using HTML controls.

8. Create a program to perform validation using validation controls.

9. Create a program in ASP .NET to connect with the database using ADODB connectivity and manipulate the records.

10. Write a program to store the employee details using class and methods in C# .NET

11. Write a program to Handle Exceptions

12. Write a program to create a form with Basic controls. In c#. NET

OUTCOMES:

At the end of this Lab course students will be able to:

1.Create user interactive web pages using ASP.Net.

2.Create simple data binding applications using ADO.Net connectivity.

3. Performing Database operations for Windows Form and web applications.