SECOND YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –III

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.1	Applied Mathematics-III	3	1	--	3	100	25	--	--	--	**125**
COMP 3.2	Data Structures and Algorithms-I	3	1	2	3	100	25	--	25	--	**150**
COMP 3.3	Economics and Organizational Behaviour	3	--	--	3	100	25	--	--	--	**125**
COMP 3.4	Object-Oriented Programming using C++	3	1	2	3	100	25	--	25	--	**150**
COMP 3.5	Logic Design	3	1	2	3	100	25	--	--	25	**150**
COMP 3.6	Software Engineering	3	1	2	3	100	25	25	--	--	**150**
TOTAL		**18**	**05**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

SECOND YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –IV

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.1	Discrete Mathematics	3	1	--	3	100	25	--	--	--	**125**
COMP 4.2	Computer Organization	3	1	2	3	100	25	25	--	--	**150**
COMP 4.3	Microprocessors and Interfacing	3	1	2	3	100	25	--	25	--	**150**
COMP 4.4	Data Structures and Algorithms-II	3	1	2	3	100	25	--	--	25	**150**
COMP 4.5	Signals and Systems	3	1	--	3	100	25	--	--	--	**125**
COMP 4.6	Java Programming	3	1	2	3	100	25	--	25	--	**150**
TOTAL		**18**	**06**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

FE 1.5 FUNDAMENTALS OF COMPUTER ENGINEERING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
FE 1.5	Fundamentals of Computer Engineering	3	--	2	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of basic concepts of computer science and engineering.
2. An introduction to the fundamentals of hardware, software and programming.
3. An introduction to mathematical software.
4. An understanding of cyber laws and computer security.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the use of mathematical software and solve simple mathematical problems.
2. Explain the needs of hardware and software required for a computation task.
3. State typical provisions of cyber law that govern the proper usage of Internet and computing resources.
4. Explain the working of important application software and their use to perform any engineering activity.
5. Demonstrate the use of Operating system commands and shell script.

UNIT -1 (12 hours)

Overview, Introduction to computers: Generation of Computers. Software and hardware, Types of Computers, Computer Networks and Internet. Data and program representation. Working of CPU, Making computers faster and better now and in the future. Storage systems characteristics, types of storage systems, Magnetic disk systems, Optical disk systems and Flash Memory systems. Keyboards, Pointing devices, Scanners, Readers and Digital cameras, Audio input, Display devices, Printers, Audio output.

UNIT- 2 (12 hours)

Introduction to System software and Application software, the operating system (OS). OS for Desktop PCs, servers, handheld PCs, Smartphone and larger

computers. Linux and Windows Operating system commands and shell scripts. Concepts of Word processing, Spreadsheet, Database, Presentation graphics and multimedia. Introduction to Assemblers, Interpreters, Compilers and Debuggers.

UNIT-3 **(12 hours)**

Basic Concepts of Technology and Law, Understanding the Technology of Internet, Scope of Cyber Laws, Cyber Jurisprudence, Encryption, Science of Cryptography, Symmetric and Asymmetric Cryptography. Electronic Banking: Banking and Bookkeeping, Legal Recognition of Digital Signature. The Cyber Crime, Tampering with Computer Source Document, Hacking with Computer System.

UNIT-4 **(12 hours)**

MATLAB and Its family, Menus and toolbars, Types of windows and types of files, MATLAB Help system, Basic calculations in MATLAB, Vectors and arrays, Multi-dimensional arrays, Element by element operations, Polynomial operations using arrays, X-Y Plotting functions, Subplots, 3-D Plots and Contour plots.

Recommended Readings:

1. Deborah Morley and Charles S. Parker; Fundamentals of Computers; Cengage Learning, India edition; 2009.
2. Alexis Leon and Mathews Leon; Fundamentals of Information Technology; Vikas Publication, Chennai.
3. Francis Scheid; Theory and Problems of Introduction to Computer Science Schaum's Outline Series; Tata McGraw Hill publication.
4. Information Technology: Tools and Application, Ed. UPTEC Computer Consultancy Limited, Elsevier Publication, 2004.
5. Rudra Pratap ;Getting started with MATLAB: A quick introduction for scientists and engineers; Oxford University press; 2003.
6. W. L. Palm III ; Introduction to MATLAB 7 for Engineers; McGraw Hill; 2005.
7. Rajeshree R Khande and Manisha Maddel ; Internet Programming & Industrial Law; Vision Publications, Pune.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1) Five programs using MATLAB (Programs will be on Basic Calculation, Calling Data file and Sending results to Data file, Control structure, Plots and Subplots, creating and using built in functions)
2) Five programs using linux shell scripting. (Using any scripting language like PERL or PYTHON)
3) Five experiments involving packages for Word Processing, Spread Sheet, Presentation, Graphics and Database.

FE 2.3 PROGRAMMING LANGUAGES

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
FE 2.3	Programming Languages	3	--	2	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of basic concepts of computer programming and developer tools.
2. An introduction to the syntax and semantics of the "C" language as well as data types offered by the language.
3. An introduction to write programs using standard language infrastructure regardless of the hardware or software platform.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the use of algorithms and flowcharts to plan the solution of a computing problem.
2. Explain the use of formatted and unformatted input and output statements in "C".
3. State typical usage of sequence control statements of "C'.
4. Enlist the fundamental data types and data structures of "C".
5. Explain the usage of arrays and pointers in "C".
6. Differentiate between a structure and a union.
7. Explain the commands of File Management in "C".

UNIT - 1 (12 Hours)

Programming Basics: Notions of algorithms, flowcharts and programming, iteration and recursion. Imperative style of programming, Functional style of programming, correctness and efficiency issues. Features of block-structured languages, Functions and procedures, Parameter passing, Top-down style and stepwise-refinement with concrete examples Fundamental algorithms: Exchanging values of two variables, counting, summation of a set of numbers , generation of prime numbers , reversal ,series.

UNIT - 2 (12 Hours)

Overview of Programming language C, constants variables and data types, operators and expressions, data input output, decision making and looping: If, If-else, while, do-while, for, switch. Function declarations and prototypes, pass by value, and pass by reference. User defined function in C, iterative function and recursive functions.

UNIT - 3 (12 Hours)

Arrays: One dimension array, array initialization, Searching, Insertion, deletion of an element from an array; finding the largest/smallest element in an array, two dimension array, addition/multiplication of two matrices, transpose of a square matrix; passing array to function , character array and string. Pointers: Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, arrays and pointers, pointer arrays.

UNIT - 4 (12 Hours)

Structure & Unions**:** Defining a structure, declaring structure variables, Accessing structure members, structure initialization, copying & comparing structure variables, operation on individual members, Array of structures, structure & functions, Unions, Size of Structure.
Files management in C: Defining & opening a file, closing a file, I/O operations on files, Error handling during I/O files, Random Access to files. Introduction to Dynamic Memory Allocation

Recommended Readings:

1. Herbert Schildt ; C: The Complete Reference, 4th Edition; Tata McGraw Hill; 2000.
2. Stephen Prata ; C Primer Plus 5th Edition; SAMS Publishing; 2005.
3. Brian W. Kernighan and Dennis M. Ritchi; C Programming Language 2nd Edition; Pearson Education; 2006.
4. Samuel P. Harbison and Guy L. Steele; C: A Reference Manual , 5th Edition; Prentice Hall; 2003.
5. Yashwant Kanetkar; Let Us C; BPB Publications, 9th Edition; 2008.
6. King K.N; C Programming: A Modern Approach, 2nd Edition; W. W. Norton and Company; 2008.
7. Dromey R.J ; How to Solve it by Computer, Prentice Hall India Series; 2000.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Program to find area and circumference of circle.
2. Program to convert temperature from degree centigrade to Fahrenheit.
3. Program to calculate sum of 5 subjects & find percentage.
4. Program to show swap of two no's without using third variable.
5. Program to print a table of any number.
6. Program to find greatest in 3 numbers.
7. Program to show the use of conditional operator.
8. Program to find whether given no is even or odd.
9. Program to shift inputed data by two bits to the left.
10. Program to use switch statement. Display Monday to Sunday.
11. Program to display first 10 natural no & their sum.
12. Program to print Fibonacci series up to 100.
13. Program to find factorial of a number.
14. Program to find whether given no is a prime no or not.
15. Program to display series and find sum of 1+3+5+……..+n.
16. Program to use bitwise AND operator between the two integers.
17. Program to add two number using pointer.
18. Program to show sum of 10 elements of array & show the average.
19. Program to find sum of two matrices.
20. Program to find multiplication of two matrices.
21. Program to find transpose of a matrix.
22. Program to find the maximum number in array using pointer.
23. Program to reverse a number using pointer.
24. Program to show input and output of a string.
25. Program to find square of a number using functions.
26. Program to show call by value.
27. Program to show call by reference.
28. Program to find factorial of a number using recursion.
29. Program to find whether a string is palindrome or not.

COMP 3.1 Applied Mathematics-III

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.1	Applied Mathematics-III	3	1	--	3	100	25	--	-	--	**125**

Course Objectives: The aim of learning this course is to provide students with the mathematical knowledge and skills necessary to support their concurrent and subsequent engineering studies.

Course Outcomes:

After successful completion of this course the student would be able to

1. Compute the rank and inverse of a matrix and solve system of linear equations.
2. Compute Eigen values and Eigen vectors of a given matrix, apply Cayley Hamilton theorem.
3. Understand the basic concepts of probability, random variables, mean, variance, standard deviation and probability distributions, correlation and regression.
4. Use tools like Laplace transforms and Fourier transforms in formulating and solving Engineering problems.
5. Understand Z- transforms and its properties and apply it in solving difference equations.

UNIT - 1 (12 Hours)

Linear Algebra: Types of matrices, adjoint, inverse. Elementary transformations.Rank of a matrix, normal form, echelon form. Linear system of equations AX = B and AX = 0. Linearly independent and dependent vectors, Eigen values and Eigen vectors, Cayley Hamilton Theorem, minimal equation, Diagonalization.

UNIT - 2 (14 Hours)

Probability and Probability distributions: Definition, properties, Axioms of probability, Conditional probability, Baye's theorem, Random Variables. Discrete probability distribution, Continuous probability distribution, Distribution function. Expectation and Variance, Moment generating function. Special distributions: Binomial, Poisson, Geometric, Normal, Uniform and exponential. Correlation and regression.

UNIT - 3 (10 Hours)

Laplace Transforms: Definition, Existence conditions, properties, inverse Laplace transforms. Laplace transform of periodic functions, Convolution theorem, Laplace transform of Dirac-Delta function, Application of Laplace transforms in solving linear differential equations with initial conditions and system of linear simultaneous differential equations.

UNIT - 4 (12 Hours)

Fourier and Z-transforms: Definition, properties, inverse. Convolution theorem. Applications of Fourier and Z-transforms.

Recommended Readings:

1. Grewal B. S.; Higher Engineering Mathematics; Khanna Publications, New Delhi.
2. H. K. Dass; Advanced Engineering Mathematics; S. Chand &Co.
3. Erwin Kreyzing; Advanced Engineering Mathematic; Wiley.
4. Kandasamy, P.; Engineering Mathematics; Chand & Co., New Delhi.
5. Srimanta Pal, Subodh C. Bhunia; Engineering Mathematics; Oxford University Press.
6. Dr. D. S. C ; Engineering Mathematics- Part III ; Prism Books Pvt. Ltd.
7. Montgomery, D. C., Probability and Statistics for Engineers; Prentice Hall of India.

COMP 3.2 DATA STRUCTURES AND ALGORITHMS-I

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.2	Data Structures and Algorithms-I	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An ability to use data structures as the foundational base for computer solutions to engineering problems.
2. An understanding of the different logical relationships among various data items.
3. Ability to understand the generic principles of computer programming as applied to sophisticated data structures.
4. An ability to plan, design, execute and document sophisticated technical programs to handle various sorts of data structures.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design algorithms using principles of recursion.
2. Demonstrate the use of data structures like linked lists , stacks and queues.
3. Explain the applications of linked lists, stacks and queues in Computer Engineering.
4. Apply the knowledge of data structures to a given problem.
5. Illustrate searching, sorting and hashing techniques.

UNIT -1 (12 Hours)

Defining, Declaring and Initialization of Structure variables. Accessing members of a structure, Array of structures, Nested Structures, Pointers to structures. Passing structure, structure members, structure arrays and pointer to structure as function parameters. Self referential structures. Introduction to Data Structures: Linear and Non Linear Data Structures, Static and Dynamic Data Structures. Array Implementation of LIFO and FIFO data structures: Stack and Queue.

UNIT -2 (12 Hours)

Concept of Linked Lists. Singly linked lists, Doubly linked lists and circular linked lists. Insertion, deletion, update and copying operations with Singly linked

lists, Doubly linked lists and Circular linked lists. Reversing a singly linked list. Circular Doubly Linked List: Linked list with Header Node, Sorted Linked List, Merging, Concatenation Comparison of Arrays and linked Lists. Linked List Implementation of Stack, Linked List implementation of Queue, Circular Queue, Deque and Priority Queue.

UNIT -3 (12 Hours)

Recursion: Writing a recursive function, Flow of control in recursive functions, Winding and unwinding phase, Recursive data structures, Implementation of recursion. Tail recursion. Indirect and Direct Recursion. Applications of Linked Lists**:** Polynomial arithmetic with linked list, Creation of polynomial linked list, Addition of 2 Polynomials, Multiplication of 2 polynomials. Applications of Stacks**:** Reversal of a String, Checking validity of an expression containing nested parenthesis, Function calls, Polish Notation: Introduction to infix, prefix and postfix expressions and their evaluation and conversions. Application of Queues: Scheduling, Round Robin Scheduling.

UNIT -4 (12 Hours)

Searching: Sequential Search, Binary Search. Hashing: Hash Functions: Truncation, Midsquare Method, Folding Method, Division Method. Collision Resolution: Open Addressing: Linear Probing, Quadratic Probing, Double Hashing, Deletion in open addressed table, implementation of open addressed tables, Separate Chaining Bucket Hashing. Sorting: Sort order, Types of sorting, Sort stability, Sort by address, In place sort, Sort pass, Sort efficiency. Selection sort, Bubble sort, Insertion sort, Shell sort, Merge sort, Quick sort, Radix sort and Address Calculation Sort.

Recommended Readings:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Yedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; Prentice Hall of India; 1996.
3. Rajesh K. Shukla; Data Structures using C and C++; Wiley India; 2009.
4. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
5. Jean Paul Tremblay, Paul G. Sorenson; An introduction to data structures with applications; Tata McGrawHill; 1984.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Implementation of array of structures.
2. Implementation of pointers to structures.
3. Implementation of stack using array.
4. Implementation of queue using array.
5. Implementation of singly linked list.
6. Implementation of doubly linked list.
7. Implementation of circular linked list.
8. Implementation of stack using linked list.
9. Implementation of queue using linked list.
10. Implementation of conversion of infix to postfix and evaluation of postfix.
11. Implementation of Tower of Hanoi using recursion.
12. Implementation of sequential search in an array.
13. Implementation of binary search in an array.

COMP 3.3 ECONOMICS AND ORGANIZATIONAL BEHAVIOUR

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.3	Economics and Organizational Behaviour	3	--	--	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of demand and supply.
2. An understanding of Financial management
3. An understanding of the role of Communication Function in organizations.
4. An understanding of the complexity of managing in a global world.
5. An understanding of the concepts of Organizational Behaviour.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain economics using demand and supply.
2. Apply the concepts of Financial Engineering
3. Explain the role of Communication Function in organizations.
4. Apply managerial concepts to solve complex problems related to global issues.
5. Explain the essential components of an organization.
6. Explain the essential requirements to become a successful entrepreneur.

UNIT -1 **(12 Hours)**

Introduction to Economics and general concepts: Demand and Supply- Demand curve, Supply curve, Market Equilibrium. National Income terms: GDP, Real v/s Nominal GDP, Net Domestic Product, GNP, National Income, Per capita income, Disposable Income, Price Index, Inflation. Estimation/Forecasting of Demand- Meaning, importance, methods–trend, exponential smoothing, regression analysis. Economies and diseconomies of Scale.

UNIT -2 **(12 Hours)**

Financial Management: An Overview – Evolution of financial management, Financial decisions in a firm, The Fundamental principle of Finance, business ethics and social responsibility. Time value of money – times lines and notations, future value of a single amount, present value of a single amount, future value of

an annuity, present value of an annuity. Capital Budgeting - Different Methods of Evaluation of Projects- Payback Period, Discounted Cash Flow methods- Net Present Value, Internal Rate of Return. Leasing – types of leases, Rationale of leasing, mechanics of leasing.

UNIT -3 (12 Hours)

Working Capital Management: Determinants of working capital, financing of working capital, dangers of excessive and shortage of working capital. Equity capital, preference capital and term loans. Preparation of Income statement, Balance sheet, Fund Flow statement. Understanding and analyzing them using financial ratios – liquidity, leverage and profitability ratios. Mergers, Takeovers and Acquisitions.

UNIT -4 (12 Hours)

Understanding of Organizational Behavior Nature of Organizations. Nature and Importance of Communication. The Two-Way Communication Process, Communication Barriers, Communication Symbols, Downward and Upward Communication, Formal and Informal Communication. Forms of Communication. Model of Motivation. Motivational Drives. Human Needs, Types of Needs. Maslow's Hierarchy of Needs. Hezberg's Two-Factor Theory. Behavior Modification. Goal Setting, Motivational Applications. The Expectancy Model. Nature of Work Change. Three Stages in Change. Reaching a New Equilibrium. The Organizational Learning Curve for Change. Appraising and Rewarding Performance. Organizational Behavior and Performance Appraisal. Economic Incentives Systems. The Reward Pyramid.

Recommended Readings:

1. R. L. Varshney and K L Maheswari; Managerial Economics; Nineteenth, Revised and Enlarged Edition; Sultan Chand and Sons Publications.
2. Peterson, Lewis; Managerial Economics; P.H.I.
3. Prasanna Chandra; Fundamentals of Financial Management, Third Edition;Tata McGraw-Hill, New Delhi.
4. Richard M. Lynch and Robert W. Williamson; Accounting for Management, Planning and Control, Third Edition; Tata McGraw-Hill, New Delhi.
5. John W. Newstrom and Keith Davis; Organizational Behavior (Human Behavior at Work), Tenth Edition; Tata McGraw Hill.

COMP 3.4 OBJECT ORIENTED PROGRAMMING USING C++

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.4	Object Oriented Programming using C++	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of the concept of object oriented programming.
2. An understanding of the concepts of data hiding, data abstraction, polymorphism inheritance and exception handling.
3. Ability to understand the generic principles of object oriented programming using "C++".
4. An understanding the use of templates in "C++".
5. An ability to plan, design, execute and document sophisticated object oriented programs to handle different computing problems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design algorithms using principles of object oriented programming.
2. Demonstrate the concepts of data abstraction and data hiding using 'C++".
3. Explain the applications of polymorphism and inheritance in object oriented programming.
4. Apply the knowledge of standard template library achieve reusability.
5. Illustrate stream I/O and exception handling.

UNIT -1 (12 Hours)

Introduction to Classes and Objects, Functions and an Introduction to Recursion, Arrays and Vectors, Pointers.

UNIT -2 (12 Hours)

Class scope and accessing class members, Constructors and destructors, Const objects and const member functions, Friend functions and friend classes, This pointer, Static class member, Data Abstraction and information hiding, Operator overloading, Inheritance, Polymorphism.

UNIT -3 (12 Hours)

Templates, Stream input/ output, Exception handling, File processing.

UNIT -4 (12 Hours)

String processing. Standard Template Library: Introduction to standard template library. Sequence Containers: vector, list, deque. Associative Containers: Set, Multiset, Map. Container Adapter: stack, queue, priority queue adapter
Algorithms: fill, fill_in, generate, generate_n, Lexicographical compare, Replace, replace_if

Recommended Readings:

1. Paul Deitel and Harrey Dietel; C++, How to Program; seventh edition.
2. Stanley Lippman; C++ Primer; Fifth edition.
3. Herbert Schildt; Complete Reference; Fourth edition.
4. Bjarne Stroustrup; C++ Programming Language; Fourth edition.
5. D Ravichandran; Programming with C++; Third Edition.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Classes and objects.
2. Friend function and friend classes.
3. Function overloading.
4. Operator Overloading.
5. Constructors and Destructors.
6. Inheritance.
7. Polymorphism and virtual functions.
8. Stream Input Output.
9. Exception Handling.
10. Templates.
11. File Handling.
12. Standard Template Library.

COMP 3.5 LOGIC DESIGN

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.5	Logic Design	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of various Number Systems & Codes along with Boolean algebra.
2. An ability to solve Boolean algebra problems.
3. An ability to design combinational and sequential circuits.
4. An understanding of programmable logic devices.

Course Outcomes:

The student after undergoing this course will be able to:

1. Convert the numbers from one radix to another and perform arithmetic operations using the 1's and 2's compliments.
2. Solve Boolean Expressions using Boolean algebra, K-maps and VEM and implement them using logic gates.
3. Design any given combinational circuits.
4. Explain different flip flops, registers and their applications.
5. Design sequential circuits and state machines.
6. Design synchronous and asynchronous counter circuits.
7. Explain arithmetic circuits like adders and multipliers and their applications.
8. Compare the characteristics of programmable logic devices.

UNIT -1 (12 Hours)

Introduction: Digital and analog systems, Logic levels and Pulse Waveforms.
Number systems – Decimal, Binary, Representation of Signed numbers and binary arithmetic, Octal number system, Hexadecimal number system.
Binary codes – Classification, 8421 BCD code, XS-3 code, Gray code, Error correction and detection codes. Logic gates-AND, OR, NOT, Universal , X-OR, X-NOR gates.
Boolean algebra: Logic operations, Laws of Boolean Algebra, Duality, Reducing Boolean expressions, Boolean functions and their representations, Boolean

expressions in SOP and POS forms, Computation of total gate inputs, Boolean expressions and logic diagrams, Conversion of AOI to NAND / NOR logic.

UNIT -2 (12 Hours)

Minimization of Switching Functions: Two, Three, Four variable K-Map, Don't Care Combinations, Quine-McCluskey method.
Combinational logic Design: Adders, Subtractors, Binary Parallel Adder/ Subtractor, Look Ahead Carry Adder, Code Converters, Parity generators/checkers , Comparators, Encoders, Decoders, Multiplexers and De-multiplexers, Modular design using IC chips.

Programmable logic devices: ROM, ROM Organization, Combinational Circuit implementation, Types of ROM, Combinational Programmable logic devices, PAL, PLA, PROM, Programmable logic devices Comparison.

UNIT -3 (12 Hours)

Flip-flops: Classification of Sequential Circuits, Latches & flip-flops - D flip-flop, JK flip-flop, T flip-flop. Flip-flop operating characteristics, Race around condition, Master slave flip-flop, conversion of one flip-flop to another , Applications of flip-flop.
Shift Registers: Buffer register, Data Transmission in Shift Registers, Serial-In Serial-Out Shift register, Serial-In Parallel-Out Shift register, Parallel-In Serial-Out Shift register, Parallel-In Parallel-Out Shift register, Bidirectional shift register, Universal Shift register, Applications of Shift register.

UNIT - 4 (12 Hours)

Counters: Asynchronous counters, Design of asynchronous counters, Synchronous counters, Shift register counters.
Sequential Circuits : Finite state model, Memory elements, Synthesis of synchronous sequential circuits, Serial Binary Adders, Sequence Detector.

Recommended Readings:

1. A. Anand Kumar; Fundamentals of Digital circuits; PHI, Second Edition.
2. Thomas L. Floyd; Digital Fundamentals; Prentice Hall.
3. Morris Mano; Digital Logic and Computer Design; PHI Publication.
4. Malvino & Leach; Digital Principles and Applications; TMH Publication.
5. R. P. Jain; Modern Digital Electronics; TMH Publication.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Performance of Basic Logic Gates.
2. a. De Morgan's Theorem (first and second law).
   b. Associative, Cumulative and Distributive laws.
3. Universal Gates-NAND and NOR .
4. Binary to Gray code conversion.
5. Half Adder and Full Adder.
6. Half Substractor and Full Substractor.
7. Sum of Product.
8. BCD to XS-3.
9. BCD to Seven-Segment Display.
10. Flip-Flop.
    a. SR-Flip Flop and D-Flip Flop.
    b. JK-Flip Flop and T-Flip Flop.

COMP 3.6 SOFTWARE ENGINEERING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
					Th Duration (Hrs)	Marks					
		L	T	P		Th	S	TW	P	O	Total
COMP 3.6	Software Engineering	3	1	2	3	100	25	25	--	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of the current issues and practices in software engineering with an emphasis on the software development process.
2. An ability to understand the software planning and management.
3. Ability to plan software requirements specifications, system modeling, quality specifications, and program specifications.
4. An understanding of software design approaches.
5. An understanding of the requirements of software project management.
6. An ability to recognize social, ethical, cultural, and safety issues in software deployment.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design a specification a software system for any existing system.
2. Plan a design of software system as per the specification.
3. Implement a software system it with readable, reusable, modular and object-oriented techniques.
4. Design a test procedure for validity, correctness and completeness.
5. Implement a software maintenance schedule.
6. Demonstrate the skills of a Software Designer, Software Architect or Project Manager for the development of software to solve business and technical problems.
7. Explain the methodologies, architectural approaches, project management techniques, and team dynamics.

UNIT -1 **(12 Hours)**

Scope of software engineering: Historical Aspects, Economic Aspects, Maintenance Aspects, Requirements, Analysis and Design Aspects, Team Development Aspects. Software Life-Cycle Models: Code-and-Fix Life-Cycle Model, Waterfall Life-Cycle Model, Rapid-Prototyping Life-Cycle Model, Open Source Life-Cycle Model, Agile Processes, Synchronize-and-Stabilize Life-Cycle Model, Spiral Life-Cycle Model. Software Process: The Requirements Workflow, The Analysis Workflow, The Design Workflow, The Implementation Workflow,

The Test Workflow, Post-delivery Maintenance, Retirement Capability Maturity Models Teams: Team Organization, Democratic Team Approach, Classical Chief Programmer Team Approach, Synchronize-and-Stabilize Teams, Teams for Agile Processes, Open-source Programming Teams.

UNIT -2 (12 Hours)

The Tools of the Trade: CASE, Taxonomy of CASE, Scope of CASE, Software Versions, Configuration Control. From Modules to Objects: Cohesion, Coupling, Data Encapsulation – Data Encapsulation and Development, The Object-Oriented Paradigm.

Reusability and Portability: Reuse Concepts, Impediments to Reuse, Objects and Reuse, Reuse during Design and Implementation – Design Reuse, Portability. Techniques for Achieving Portability. More on UML: Class Diagrams, Use-Case Diagrams, Interaction Diagrams, State Charts, Activity Diagrams.

UNIT -3 (12 Hours)

Testing: Quality Issues, Non-Execution-Based Testing, Execution-Based Testing, Testing versus Correctness Proof and stopping criteria. Planning and Estimating: Planning and the Software Process, Estimating Duration and Cost. Requirements: Determining what the Client Needs, Overview of the Requirements Workflow, Understanding the domain, The Business Model, Initial Requirements, Metrics and Challenges for Requirement Workflow.Classical Analysis: The Specification Document, Informal Specifications, Structured Systems Analysis, Metrics and Challenges for Classical Analysis.

UNIT -4 (12 Hours)

Design and Abstraction, Operation Oriented Design, Data Flow Analysis, Data-Oriented Design, Object-Oriented Design, Challenges and Metrics for Design Workflow Implementation: Choice of Programming Language, Fourth-Generation Language, Good Programming Practice, Coding Standards, Code Reuse, Integration, Test Case Selection, Black-Box Unit-Testing Techniques, Glass-Box Unit-Testing Techniques, Code Walkthroughs and Inspections, Integration Testing, Product Testing, Acceptance Testing. Post delivery Maintenance: Development and Maintenance, Management of Post delivery Maintenance, Maintenance of Object Oriented Software, Post delivery Maintenance Skills versus Development Skills, Reverse Engineering, Testing during Post delivery Maintenance, Metrics and Challenges for Post delivery Maintenance.

Recommended Readings:

1. Stephen R. Schach; Software Engineering; TMH, Seventh Edition.
2. Roger S. Pressman; Software Engineering – A practitioner's approach; McGraw Hill, Seventh Edition.
3. Edward Kit; Software Testing in the Real World: Improving the Process; Addison – Wesley Publishing company; 1995.
4. Pankoj Jalote; Software Project Management in Practice; PEA
5. Ian Sommervilee; Software Engineering; 10th Edition Pearson.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. The Term Work Marks to be awarded based on the assessment of experiments conducted.)

1. Introduction to Software Crisis and Software Processes.
2. Requirements (Requirements Document).
3. Specifications (Software Requirement Specification).
4. Design.
5. Cost Estimation.
6. Implementation.
7. Black Box Testing.
8. White Box Testing.
9. Software Reliability.
10. Software Maintenance.

COMP 4.1 Discrete Mathematics

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.1	Discrete Mathematics	3	1	--	3	100	25	--	-	--	**125**

Course Objective: This course is designed to introduce students to the techniques, algorithms, and reasoning processes involved in the study of discrete mathematical structures that are essential to the field of Computer Science.

Course Outcomes: On completing this course students will be able to

1. Perform operations on discrete structures such as sets, functions, relations, and sequences.
2. Know the properties of equivalence relations and partial orderings.
3. Apply algorithms and use definitions to solve problems to prove statements in elementary number theory.
4. Construct mathematical arguments using logical connectives and quantifiers and verify the correctness of an argument using propositional and predicate logic and truth tables.
5. solve problems using the basic principles of counting theory, including permutation, combinations, and the pigeonhole principle
6. Solve problems involving recurrence relations and generating functions.
7. Understand lattices and Boolean algebras.
8. Explain basic definitions and properties associated with simple planar graphs, including isomorphism, connectivity, and Euler's formula, and describe the difference between Eulerian and Hamiltonian graphs.
9. Use graphs and trees as tools to solve combinatorial optimization problems

UNIT - 1 (12 Hours)

Set Theory : Sets, Set Operations, Relations and their properties, Equivalence Relations, partial orderings.
Functions: One-to-One and Onto Functions, Inverse Function, Composition of functions, Graphs of functions and some important functions.
Integers: Integers and division (excluding applications of congruences and cryptology), primes and greatest common divisors, Integers and algorithms.

UNIT - 2 (12 Hours)

Propositional Calculus: Propositional logic, propositional equivalences, predicates and quantifiers, rules of inference.
Boolean Algebra: Boolean functions, representing Boolean functions.
Mathematical Induction: Principle of Mathematical Induction and applications.

UNIT - 3 (12 Hours)

Counting: The basics of counting, pigeonhole principle, permutations and combinations, binomial coefficients.
Advanced Counting Techniques: Recurrence relations, solving linear recurrence relations, inclusion –exclusion principle, applications of inclusion – exclusion principle.

UNIT - 4 (12 Hours)

Graph theory: Graphs and graph models, graph terminology and special types of graphs, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths, shortest path problems, planar graphs, graph coloring.
Trees: Introduction to Trees, applications of trees, tree traversal, Spanning Trees, Minimal Spanning Trees.

Recommended Readings:

1. Kenneth H. Rosen; Discrete Mathematics and Its Applications; Tata McGraw Hill (6th edition).
2. B Kolman, R.C. Busby and Sharon C. Ross; Discrete Mathematical Structures; Prentice Hall.
3. J. P. Tremblay and R. Manohar, McGraw Hill; Discrete Mathematical Structures with Applications to Computer Science; New York McGraw Hill.
4. Swapan Kumar Sarkar; Discrete Mathematics; S.Chand Publication.
5. Dr. D. S. C ;Discrete Mathematical Structures; Prism Books Pvt. Ltd.
6. G.V.Kumbhojkar; Discrete Structures And Graph Theory; Pradeep Prakashan.

COMP 4.2 COMPUTER ORGANIZATION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.2	Computer Organization	3	1	2	3	100	25	25	--	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of relationship between hardware and software.
2. An ability to recognize how machine organization impacts the efficiency of applications written in a high-level language.
3. An ability to understand the system performance and concepts behind advanced pipelining techniques.
4. An understanding of different ways of communicating with I/O devices and standard I/O interfaces.
5. An ability to develop solutions for basic programs using assembly language.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit.
2. Identify high performance architecture design.
3. Create an assembly language program to program a microprocessor system.
4. Design a pipeline for consistent execution of instructions with minimum hazards.
5. Explain the ways to take advantage of instruction level parallelism for high performance processor design.
6. Demonstrate memory hierarchy and its impact on computer cost/performance.

UNIT - 1 (12 Hours)

Introduction to Computer Organization: Computer components, Functions, interconnection Structure, Bus Interconnection. Computer Arithmetic: Integer Representation-unsigned numbers, signed numbers, signed magnitude,2's compliment, Biased Representation. Integer Arithmetic: Addition, Subtraction, Multiplication unsigned, signed (Booths Algorithm),Division-unsigned, signed.

Floating-Point Representation: IEEE 32 bits, 64 bits. Floating-Point Arithmetic: Addition, Subtraction, Multiplication, Division.

UNIT - 2 (12 Hours)

Internal Memory: Semiconductor Memory - Memory Hierarchy, Characteristics of Memory System, Semiconductor RAM Memories, Internal Organization of Memory Chip, Static RAM, Asynchronous DRAM, Synchronous DRAM, Connection of Memory to the processor, RAM Bus memory. Cache Memory: Basics of Cache, Structure, Read operation, Elements of Cache Design. Associative Memory: External Memory: Magnetic Disk, RAID, optical Memory. Virtual Memory: Logical VS Physical Address space, working Principle, Mapping Functions, Replacement Policy.

UNIT - 3 (12 Hours)

Input/Output: External Devices, I/O Modules, Programmed I/O, Interrupt Driven I/O, Direct Memory Access, I/O Channel and Processor. CPU Structure and Functions: Processor Organization, Register Organization, Instruction Pipeline, Basic Concepts of Pipelining. RISC CPU Architecture: Instruction Execution Characteristics, Use of Large Register File, Compiler based register optimization, Reduced Instruction Set Architecture, RISC v/s CISC.

UNIT - 4 (12 Hours)

Buses: Bus interconnections, VGA, Asynchronous v/s Synchronous Buses, PCI Bus, SCSI
Control Unit Operation: Micro Operations, Control of the CPU, Hardwired Implementation Micro programmed Control: Basic Concepts, Microinstruction Sequencing, and Microinstruction Execution. Parallel Processing: Multi Processing, Cache Coherence /MESI Protocol.

Recommended Readings:

1. William Stalling; A textbook of Computer Organization and Architecture; Edition VI.
2. M. Morris Mano ; A textbook of Computer Organization and Architecture.
3. Douglas V. Hall ; Microprocessors and Interfacing.
4. David A. Patterson, John L. Hennesy ; Computer Organization And Design, Edition III.
5. Carl Hamacher, Zvonko Vranesic, Safal Zaky ; Computer Organization; Edition V.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. The Term Work Marks to be awarded based on the assessment of experiments conducted.)

1. Assembly language program to perform addition on
    a. 8 bit data.
    b. 16 bit data.
    c. 32 bit data.
2. Assembly language program to perform subtraction on
    a. 8 bit data.
    b. 16 bit data.
3. Assembly language program to perform multiplication on
    a. 8 bit data.
    b. 16 bit data.
4. Assembly language program to find average of two numbers.
5. Assembly language program to find two's compliment of number.
6. Assembly language program to check status of sign flag.
7. Assembly language program to compute Xn.
8. Assembly language program to perform bubble sort in descending order.
9. Assembly language program to find largest from array of 8 bit numbers.

COMP 4.3 MICROPROCESSORS AND INTERFACING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.3	Microprocessors and Interfacing	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An in-depth understanding of the Intel 8086 architecture and programming model.
2. An ability to write Assembly language programs for a given task.
3. An understanding of different types of memories, peripheral IC's like 8255, 8259 and 8254and their interfacing with the processor.
4. An ability to interface various I/O devices with the processor.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe the architecture and explain the working of each block in 8086 processor.
2. Analyze the instruction set of 8086 processor.
3. Analyze the timing sequence of various instructions.
4. Create Assembly language programs for a given task.
5. Explain the basic programmable ICs like 8255, 8259 and 8254.
6. Design interfacing of memories and various I/O devices with the processor.

UNIT -1 **(12 Hours)**

The 8086 Microprocessor family overview, 8086 Internal Architecture. Introduction to programming the 8086. 8086 Family Assembly Language Programming: Program Development steps, Constructing the machine codes for 8086 Instructions, Writing programs for use with an Assembler, Assembly Language Program Development Tools

Implementing Standard Program Structures in 8086 Assembly Language: Simple Sequence programs, Jumps, Flags, Conditional Jumps, If –Then, If-Then-Else and Multiple If-Then-Else programs, While-do programs, Repeat-Until Programs. 8086 Instruction Descriptions and Assembler Directives: Instruction Description, Assembler Directives.

UNIT -2 (12 Hours)

Strings, Procedures and Macros: The 8086 String Instructions, Writing and using Procedures, Writing and using Assembler Macros. 8086 System Connections Timing and Troubleshooting: A Basic 8086 Microcomputer System, An example minimum mode System, The SDK-86. 8086 Interrupts and Interrupt Applications: 8086 Interrupts and Interrupt responses, Hardware Interrupt Applications, 8254 Software Programmable Timer/Counter, 8259A Priority Interrupt controller, Software Interrupt Applications.

UNIT -3 (12 Hours)

Digital Interfacing: Programmable Parallel Ports and Handshake Input Output, 8255A Programmable Parallel Interface- Internal Block diagram and System Connections, 8255A Operational Modes and Initializations, 8255A Handshake Applications Examples. Cache Memories Coprocessor: 8086 Maximum mode, Direct Memory Access (DMA) Data transfer, Interfacing and Refreshing Dynamic RAMS, A Coprocessor- The 8087 Math Coprocessor.

UNIT -4 (12 Hours)

The 80286,80386 and 80486 Microprocessors: Multiuser/Multitasking Operating System Concepts, The Intel 80286 Microprocessor, The Intel 80386 32 bit Microprocessor. The 80286, 80386 and 40386 Microprocessors: The Intel 80486 Microprocessor. An Introduction to the Pentium Processors.

Recommended Readings:

1. Douglas V. Hall; Microprocessors and Interfacing; TMH,Revised Second Edition.
2. John F. Uffenbeck; The 8086/8088 family design, programming and interfacing; (PHI).
3. Liu and Gibson; Microprocessor Systems: The 8086/8088 family architecture programming and design; PHI.
4. Richard C. Detmer Jones and Bartlett; Introduction to 8086 Assembly Language And Computer Architecture; Publishers ISBN 0-7637-1773-8.
5. Barry B. Brey; The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium and Pentium preprocessor architecture, Programming and Interfacing; PHI.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. 8086 ALP to evaluate the expression.
2. 8086 ALP to find sum and average of n numbers.
3. 8086 ALP to find even and odd number.
4. 8086 ALP to implement linear search.
5. 8086 ALP to find the square of a number using macro and procedure.
6. 8086 ALP to implement bubble sort .
7. 8086 ALP to compare two strings.
    a. Using string instructions.
    b. Without using string instructions.
8. 8087 ALP to evaluate the expression.
9. 8087 ALP to evaluate the identity $\sin^2\emptyset+\cos^2\emptyset=1$.
10. 8087 ALP to evaluate to compute standard deviation.
11. 8087 ALP.
    a.To compute X^y .
    b.To evaluate the expression.

COMP 4.4 DATA STRUCTURES AND ALGORITHMS-II

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.4	Data Structures and Algorithms-II	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

The subject aims to provide the student with:

1. An ability to use data structures as the foundational base for computer solutions to engineering problems.
2. An understanding of the different logical relationships among various data items.
3. Ability to understand the generic principles of computer programming as applied to sophisticated data structures.
4. An ability to plan, design, execute and document sophisticated technical programs to handle various sorts of data structures.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the use of complex data structures like trees and graphs.
2. Explain the applications of tree and graph structures in Engineering.
3. Apply the knowledge of data structures to a given problem.
4. Explain the storage management schemes.

UNIT -1 (12 Hours)

Introduction to Trees: Terminology, Types of Binary trees. Array and Linked List representation of binary trees in memory: Traversal in Binary Tree: Non recursive traversal in binary tree: Preorder, in-order, post-order and Level order traversal. Creation of binary trees from the traversal of binary trees. Binary search tree: Traversal in binary search tree, searching in a binary search tree, finding nodes with minimum and maximum key, insertion and deletion in a binary search tree. Threaded Binary Tree: Finding in-order successor and predecessor of a node in threaded tree. Insertion and deletion in threaded binary tree.

UNIT - 2 (12 Hours)

AVL Tree**:** Searching and traversing in AVL trees. Tree Rotations: Right Rotation, Left Rotation. Insertion and Deletion in an AVL Tree. Red Black Trees: Searching,

Insertion and Deletion. Heap: Insertion, Deletion, Building a Heap, Selection algorithm, implementation of Priority Queue. Weighted Path Length. Multi-way Search Tree. B-tree: Searching, Insertion, Deletion from leaf node and non-leaf node. B+ Tree: Searching, Insertion, Deletion. Digital Search Tree.

UNIT - 3 (12 Hours)

Introduction to Graphs: Undirected Graph, Directed Graph, graph terminology, Connectivity in Undirected and Directed Graphs. Spanning tree, spanning forest. Representation of graph: adjacency matrix, adjacency list, Transitive closure of a directed graph and path matrix. Traversals: Breadth First Search, Depth First Search. Connected components in a graph. Network flow problems- a simple maximum flow algorithm.

UNIT - 4 (12 Hours)

Applications of Trees: Huffman Tree, Binary Tree Sort, Heap Sort and Huffman Tree.
Applications of Graphs: Warshall's algorithm. Shortest Path Algorithms: Dijkstra's Algorithm, Bellman Ford Algorithm, Floyd's Algorithm. Minimum Spanning Tree: Prim's Algorithm, Kruskal's Algorithm. Topological Sorting. Storage Management: Sequential Fit Methods: First Fit, Best Fit and Worst Fit methods. Fragmentation, Freeing Memory, Boundary Tag Method. Buddy Systems: Binary Buddy System, Fibonacci Buddy System.
Compaction, Garbage Collection.

Recommended Readings:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Yedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; Prentice Hall of India; 1996.
3. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
4. Alfred V. Aho, John E. Hopcroft, J. D. Ullman; Data Structures and Algorithms; Addison Wesley; 1983.
5. Jean Paul Tremblay, Paul G. Sorenson; An introduction to data structures with applications; Tata McGrawHill; 1984.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Implementation of operations in a binary search tree.
2. Implementation of insertion, deletion and traversal for fully in-threaded binary search tree.
3. Implementation of AVL tree.
4. Implementation of red black tree.
5. Implementation of insertion and deletion in heap.

6. Implementation of operations in a B tree.
7. Implementation of adjacency matrix creation.
8. Implementation of addition and deletion of edges in a directed graph using adjacency matrix.
9. Implementation of insertion and deletion of vertices and edges in a directed graph using adjacency list.
10. Implementation of traversal of a directed graph through BFS.
11. Implementation of traversal of a directed graph through DFS.
12. Implementation of finding shortest distances using Dijkstra's algorithm.

COMP 4.5 SIGNALS AND SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.5	Signals and Systems	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of time-domain representation and analysis of signals and systems.
2. An ability to perform frequency-domain representation and analysis using Fourier tools.
3. An ability to perform frequency-domain representation and analysis using Laplace transform.
4. An understanding of sampling theory.

Course Outcomes:

The student after undergoing this course will be able to:

1. Determine the mathematical representation and classification of signals and systems.
2. Determine the response of an LTI system using convolution and classical methods. Analyze system properties based on impulse response.
3. Determine and analyze the responses of LTI system to periodic signals using Fourier series.
4. Determine and analyze the responses of LTI system to arbitrary time signals using Fourier transform.
5. State sampling theory and its application and convolution and correlation of signal.
6. Determine the properties of continuous time signals and system using Laplace transforms.

UNIT -1 (12 Hours)

Signal, Continuous-time signals, Discrete-time Signals, Graphical, Functional, Tabular and Sequence Representation of Discrete-time Signals. Basic Operations on Signals. Classification of Signals. Signals and Vectors: Vector addition, Scalar multiplication, Dot product (inner product), Norm, Distance, Angle, Projection. Vector Space: Orthogonality in vectors. A System. Classification of Systems. Invertibility and Inverse Systems.

UNIT - 2 (12 Hours)

Time domain analysis of discrete-time systems: Solution of difference equations. natural, forced, total response, and impulse response. Representation of discrete-time signals in terms of impulses. Impulse response and convolution sum. Properties of convolution. Convolution of two sequences. Causality, FIR and IIR systems, stability, BIBO stability criterion, step response, correlation of two sequences. Inverse system and deconvolution. Time domain analysis of continuous-time systems: Solution of differential equations. Natural, forced and total response. Representation of a continuous-time signal. Convolution integral, Properties of convolution, Impulse response of interconnected systems. Causality, Graphical Procedure to Perform Convolution, Stability Step Response, Correlation.

UNIT – 3 (12 Hours)

Fourier Series Analysis of Continuous-Time Periodic signals: Fourier Series Representation of Periodic Signals, Evaluation of Fourier Coefficients, Symmetry Conditions, Cosine Representation, Exponential Fourier series, Existence of Fourier Series, Properties of Continuous-time Fourier Series, Power Representation using the Fourier series, Fourier Spectrum, Gibb's Phenomenon. The Continuous-Time Fourier Transform. Development of Fourier Transform. Fourier Transform of some Standard signals. Properties of Fourier Transform. Fourier Transform of a Periodic signal. Modulation. System Analysis with Fourier Transform.

UNIT - 4 (12 Hours)

Signal and System Analysis using the Laplace Transform: Convergence of the Laplace Transform, s-Plane, The Unilateral Laplace Transform, Properties of unilateral Laplace transform. Inversion of Unilateral Laplace Transform, Inversion of the Bilateral Laplace Transform. Solution of Differential Equations using Laplace Transform. Analysis of Electrical Networks using Laplace Transform., Stability, Block Diagram Representation of Summer, Gain, Feedback, Integrator, cascade connection of blocks and parallel connection of blocks. Signal flow graph, System Realization. State Space Analysis. Sampling: Analog to Digital Conversion, Sampling and Aliasing, Impulse sampling, Sampling Theorem, Anti Aliasing Filter, Pulse Sampling, Flat-Top sampling, Signal Reconstruction, Bandpass Signals. Sampling Bandpass Signals.

Recommended Readings:

1. P. Ramesh Babu, R. Anandanatarajan; Signals and Systems; Scitech Publications, 4th Edition; 2006.
2. Rodger E.Ziemer, William H. Tranter, D. Ronald Fannin; Signals and Systems - Continuous and Discrete; Pearson Education, 4th Edition; 1983.
3. Simon Haykin and Barry Van Veen ; Signals and Systems; John Wiley &Sons (Asia)Pvt. Ltd; 2/e.
4. Oppenheim and Willskay with Hamid Nawab ; Signals and Systems; Prentice Hall of India.
5. Linder, Introduction to Signals and Systems; McGraw Hill.
6. Nagrath, Sharan, Rajan and Kumar; Signals and Systems; McGraw Hill.

COMP 4.6 JAVA PROGRAMMING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.6	Java Programming	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of how things work in the web world.
2. An understanding of the client-side implementation of web applications.
3. An ability to understand the generic principles of object oriented programming using "Java".
4. An understanding the use of Graphics programming in "Java".
5. An ability to plan, design, execute and document sophisticated object oriented programs to handle different computing problems using "Java".

Course Outcomes:

The student after undergoing this course will be able to:

1. Design algorithms using principles of object oriented programming
2. Demonstrate the use-cases, pseudocode, and an incremental coding plan for a given Problem specification.
3. Explain the operations of common data structures and algorithms.
4. Design a "Java" program to solve a given problem specification.
5. Illustrate stream I/O, Graphics programming and exception handling.

UNIT – 1 (12 Hours)

Fundamentals of Object Oriented Programming, Java Evolution , Overview of Java Language, Constants, Variables and Data Types, Operators and Expressions, Decision Making and Branching, Decision Making and Looping, Classes, Objects and Methods.

UNIT – 2 (12 Hours)

Arrays, Strings and Vectors, Interfaces: Multiple Inheritance, Packages: Putting Classes together.

UNIT – 3 (12 Hours)

Multithreaded Programming, Managing Errors and Exceptions, Applet Programming.

UNIT – 4 (12 Hours)

Graphics Programming, Managing Input/ Output Files in Java, Java Collections.

Recommended Readings:

1. E. Balagurusamy; Programming with Java A Primer; Tata McGrawHill Companies 5th edition.
2. John P. Flynt ;Java Programming; Thomson 2nd.
3. Ken Arnold ;Java Programming Language; Pearson.
4. Hervert schildt; The complete reference JAVA2; TMH.
5. Cay Horstmann; Big Java; 2nd edition; Wiley India Edition.
6. Sachin Malhotra, Saurabh Chaudhary; Programing in Java; Oxford University Press, 2010.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Programs using constructor and destructor.
2. Creation of classes and use of different types of functions.
3. Count the number of objects created for a class using static member function.
4. Write programs on interfaces.
5. Write programs on packages.
6. Write programs using function overloading.
7. Programs using inheritance.
8. Programs using IO streams.
9. Programs using files.
10. Write a program using exception handling mechanism.
11. Programs using AWT.
12. Programs on swing.
13. Programs using JDBC.

THIRDYEAR :COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER - V

Subject Code	Nomenclature of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination					
						Marks				
		L	T	P#	ThDuration(Hrs)	Th	S	TW	P	O	Total
COMP 5.1	Data Communication	3	0	0	3	100	25	--	--	--	**125**
COMP 5.2	Automata Languages and Computation	3	1	2	3	100	25	25		--	**150**
COMP 5.3	Cryptography and Coding Theory	3	1	0	3	100	25	--	--	--	**125**
COMP 5.4	Computer Hardware Design	3	1	2	3	100	25	--		25	**150**
COMP 5.5	Database Management System	3	1	2	3	100	25	--	25		**150**
COMP 5.6	Operating Systems	3	1	2	3	100	25		25	--	**150**
TOTAL		**18**	**05**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

#A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

THIRDYEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER - VI

Subject Code	Nomenclature of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
						Marks					
		L	T	P#	ThDuration(Hrs)	Th	S	TW	P	O	Total
COMP 6.1	Software Testing and Quality Assurance	3	0	0	3	100	25	--	--	--	**125**
COMP 6.2	Design and Analysis of Algorithms	3	1	0	3	100	25	--	--	--	**125**
COMP 6.3	Artificial Intelligence	3	1	2	3	100	25	--	25	--	**150**
COMP 6.4	Computer Graphics	3	1	2	3	100	25	--	25		**150**
COMP 6.5	Embedded System Design	3	1	2	3	100	25	25	--	--	**150**
COMP 6.6	Computer Networks	3	1	2	3	100	25			25	**150**
TOTAL		**18**	**05**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

#A candidate is considered to havesuccessfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

APPENDIX

COMP 5.1 DATA COMMUNICATION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.1	Data Communication	3	1	-	3 hrs	100	25	-	-	-	**125**

Course Objectives:

1. To focus on imparting knowledge about various components of data communications emphasizing on the physical layer and data link layer of the OSI stack.
2. To understand the conceptual and analytical differences between analog and digitalcommunication.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the basic concepts of data communication components used at various transmission speeds.
2. Identify the characteristics and analyze specific role of Data Communication technologies such as ATM, wireless, satellite and fiber optic communication.

UNIT - 1 (12 Hours)

Introduction to Data Communication, Networks, Protoc. ols and Standards. Network Models: Layered Task, The OSI Reference Model, TCP/IP protocol Suite, Addressing
Data and Signals: Analog and Digital, Periodic analog signals, digital signal, transmission impairment, data rate limits, performance.Digital Transmission: digital-to digital conversion, analog-to-digital conversion transmission modes

UNIT - 2 (12 Hours)

Analog Transmission: Digital-to- analog conversion, analog-to-analog conversion. Multiplexing and Spread Spectrum. Transmission Media: Guided Media, Unguided Media
Switching: Circuit Switched Networks, Datagram Networks. Telephone Network: Dial-up modems, DSL, Cable TV networks

UNIT - 3 (12 Hours)

Error Detection and Correction: , Block Coding, Linear Block Codes, Cyclic Block Codes,
Data Link Control: Framing, Flow and Error Control, Protocols, Noiseless channels, noisy channels, HDLC .Random Access. Channelization.

UNIT - 4 (12 Hours)

Connecting LAN's, Backbone Networks and Virtual LAN's. Wireless WAN's: Cellular Telephony, satellite Networks. Virtual- Circuit Networks: Frame Relay, ATM, ATM LAN

Recommended Readings:

1. B.A. Forouzan; Data Communication and Networking; Tata McGraw Hill, 4th Edition
2. Andrew S. Tanenbaum ;Computer Networks; Pearson Education
3. William Stallings; Data and Computer Communication; Seventh edition

COMP5.2 AUTOMATA LANGUAGE AND COMPUTATION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP5.2	Automata language and Computation	3	0	2	3	100	25	--	--	--	**125**

Course Objectives:

1. To introduce concepts in automata theory and theory of computation.
2. To learn different formal language classes and their relationships.
3. To design grammars and recognizers for different formal languages.
4. To study the decidability and intractability of computation problems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Identify formal language classes and Explain the properties of languages, grammars and automata.
2. Apply the techniques to transform between equivalent deterministic and non-deterministic finite automata and regular expressions.
3. Design grammars and automata (recognizers) for different language classes.
4. Perform the Simplification of automata and Context free grammars.
5. Explain the concepts of context-free languages, pushdown automata and Turing recognizable languages.

UNIT – 1 (12 Hours)

Mathematical Preliminaries and Notation: Sets, Functions and Relations, Graphs and Trees, Proof Techniques. Languages, Grammars and Automata. A Hierarchy of Formal Languages and Automata, Recursive and Recursively Enumerable Languages, Unrestricted Grammars, Context-Sensitive Grammars and Languages, Closure properties of Regular and Context Free languages. The Chomsky Hierarchy. Finite Automata: Deterministic Finite Accepters, Deterministic Accepters and Transition Graphs.

UNIT - 2 (12 Hours)

Nondeterministic Finite Accepters, Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata, Regular Expressions: Connection Between Regular Expressions and Regular Languages.

Right- and Left-Linear Grammars. Equivalence of Regular Languages and Regular Grammars. A Pumping Lemma for regular languages.

UNIT - 3 **(12 Hours)**

Context-Free Grammars, Leftmost and Rightmost Derivations, Derivation Trees, Parsing and Ambiguity. Methods for Transforming Grammars: Substitution Rule, Removing Useless Productions, lambda-Productions, and Unit-Productions. Chomsky Normal Form, and Greibach Normal Form. Deterministic and Nondeterministic Pushdown Automata, Pushdown Automata for Context-Free Languages, Context-Free Grammars for Pushdown Automata. A Pumping Lemma for Context-Free Languages.

UNIT - 4 **(12 Hours)**

Definition of a Turing Machine, Turing Machines as Language Accepters, Turing Machines as Transducers, Combining Turing Machines for Complicated Tasks, Turing's Thesis.. Turing Machines with More Complex Storage, Multitape Turing Machines, Multidimensional Turing Machines. Nondeterministic Turing Machines, A Universal Turing Machine, Linear Bounded Automata. Computability and Decidability: Turing Machine Halting Problem.

.

Recommended Readings:

1. Peter Linz; An introduction to Formal Languages and Automata; Jones & Bartlett Learning, 2006
2. K.L.P Mishra, N. Chandrasekaran; Theory of Computer Science – Automata, languages and Computation; PHI Publications; Third Edition ; 2008.
3. John C Martin; Introduction to languages and the theory of computation; Tata McGraw Hill, Fourth Edition, 2010.
4. John E. Hopcraft and Jeffery D. Ullman; Introduction to Automata Theory, Languages and Computation; Narosa Publishing House.
5. Michael Sipser; Introduction to Theory of Computation; PWS Publishing Company.
6. A.A Puntambekar; Formal Languages and Automata Theory; Technical Publications Pune;

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Design and implementation of Non-Deterministic Finite Automata (NFA).
2. Design and implementation of Deterministic Finite Automata (DFA).
3. Transform the equivalent DFA from the Non-Deterministic Finite Automata.
4. Obtain the minimized DFA from the given DFA.
5. Convert the given RE into Finite Automata.
6. Design and Implementation of Mealy machine.
7. Design and Implementation of Moore machine.
8. Prove Pumping lemma for regular languages.
9. Prove Pumping lemma for context-free languages.
10. Design and Implementation of Pushdown Automata.
11. Design and Implementation of Turing Machine.
12. Convert the grammar to its equivalent automata and vice versa.

COMP5.3 CRYPTOGRAPHY AND CODING THEORY

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP5.3	Cryptography and Coding Theory	3	1	0	3	100	25	--	--	--	**125**

Course Objectives:

1. To familiarize students with the theory and practice of the coding and encryption schemes.
2. To learn fundamental properties, most relevant examples and most important applications of Block Codes.
3. To understand various techniques for symmetric and asymmetric cryptography.
4. To familiarize the techniques for hashing and digital signature.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the foundations of number theory and its applications in building crypto systems.
2. Analyze which error-correction coding scheme is most appropriate for a givendemand.
3. Explain the relations existing among different areas of mathematics, especially algebra, coding theory and the theory of self -correcting codes.
4. Describe which encryption techniques are currently used in the different digital systems and to how they work.

UNIT – 1 (12 Hours)

Basic Number Theory: Basic Notions, Congruences, The Chinese Remainder Theorem, Modular Exponentiation, Fermat and Euler, Primitive Roots,

Classical Cryptosystems: Shift Ciphers, Affine Ciphers ,TheVigenere Cipher, Substitution Ciphers , Sherlock Holmes , The Playfair and ADFGX Ciphers One-Time Pads, Pseudo-random Bit Generation, LFSR Sequences, Enigma.

UNIT - 2 **(12 Hours)**

The Data Encryption Standard: A Simplified DES-Type Algorithm, Differential Cryptanalysis,DES, Modes of Operation, Breaking DES, Meet-in-the-Middle Attacks Password Security. The Advanced Encryption Standard: RijndaelAlgorithm, The Layers, Decryption, Design Considerations.

UNIT - 3 **(12 Hours)**

The RSA Algorithm, Attacks on RSA, Primality Testing, Factoring, The RSA Challenge, The Public Key Concept,Discrete Logarithms, Computing Discrete Logs, Bit Commitment Diffie-Hellman Key Exchange, The ElGamal Public Key Cryptosystem.

UNIT - 4 **(12 Hours)**

Hash Functions: A Simple Hash Example, The Secure Hash Algorithm, Birthday Attacks, Digital Signatures, RSA Signatures, The ElGamal Signature Scheme, Hashing and Signing..

Error Correcting Codes, Bounds on General Codes, Linear Codes, Hamming Codes, Golay Codes, Cyclic Codes, BCH Codes and Reed-Solomon Codes

Recommended Readings:

1) Introduction to Cryptography with Coding Theory, 2nd edition, Wade Trappe and Lawrence C. Washington, Pearson Education, 2011
2) Lin, S.; Costello, D.J.. Error control coding: fundamentals and applications. Prentice-Hall, 2004.
3) R. Hill ;A first course in Coding Theory,; Oxford, 0-19-8538030
4) J. A. Buchmann;Introduction to Cryptography; Springer, 0387950346
5) W. C. Huffman & V. Pless; Fundamentals of Error-Correcting Codes; Cambridge, 521782805
6) S. Roman; Introduction to Coding and Information Theory,;Springer
7) A. Menezes, P. van Oorschot, and S. Vanstone; Handbook of Applied Cryptography; CRC
8) N. Ferguson & B. Schneier;Practical Cryptography; John Wiley
9) W. Mao; Modern Cryptography Theory and Practice; Prentice Hall,
10) H. Niederreiter editor;Coding Theory and Cryptology; World Scientific/Singapur U. Press

COMP 5.4 COMPUTER HARDWARE DESIGN

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.4	Computer Hardware Design	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To understand the basic principles of Digital System design.
2. To use hardware description language, VHDL, in the design process.
3. To automatically synthesize digital hardware from VHDL description using CAD tools.
4. To equip students with skills to design systems at high level and express algorithms in VHDL.

Course Outcomes:

The student after undergoing this course will be able to:

1. Relate the constructs of VHDL to corresponding hardware
2. Design digital logic circuits using Computer Aided Tools.
3. Model, Simulate and Synthesize digital systems using VHDL.
4. Design and prototype with programmable logic.
5. Test a design described in VHDL using a test bench written in VHDL.

UNIT - 1 (12 Hours)

Introduction to VHDL: Computer-Aided Design, Hardware Description Languages, VHDL Description of Combinational Circuits, VHDL Modules, Sequential Statements and VHDL Processes, Modeling Flip-Flops Using VHDL Processes, Processes Using Wait Statements, Two Types of VHDL Delays: Transport and Inertial Delays, Compilation, Simulation, and Synthesis of VHDL Code, VHDL Data Types and Operators, Simple Synthesis Examples, VHDL Models for Multiplexers, VHDL Libraries, Modeling Registers and Counters Using VHDL Processes, Behavioral and

Structural VHDL, Variables, Signals, and Constants, Arrays, Loops in VHDL, Assert and Report Statements. Brief Overview of Programmable Logic Devices, Simple Programmable Logic Devices (SPLDs), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Arrays (FPGAs)

UNIT - 2 (12 Hours)

BCD to 7-Segment Display Decoder, A BCD Adder, 32-Bit Adders, Traffic Light Controller, State Graphs for Control Circuits, Scoreboard and Controller, Synchronization and Debouncing, A Shift-and-Add Multiplier, Array Multiplier, A Signed Integer,Fraction Multiplier, Keypad Scanner, Binary Dividers. State Machine Charts, Derivation of SM Charts, realization of SM Charts, Implementation of the Dice Game, Microprogramming, Linked State Machines.

UNIT - 3 (12 Hours)

Implementing Functions in FPGAs, Implementing Functions Using Shannon's Decomposition, Carry Chains in FPGAs, Cascade Chains in FPGAs, Examples of Logic Blocks in Commercial FPGAs, Dedicated Memory in FPGAs, Dedicated Multipliers in FPGAs, Cost of Programmability, FPGAs and One-Hot State Assignment, FPGA Capacity: Maximum Gates Versus Usable Gates, Design Translation (Synthesis), Mapping, Placement, and Routing. Representation of Floating-Point Numbers, Floating-Point Multiplication, Floating-Point Addition, Other Floating-Point Operations

UNIT - 4 (12 Hours)

VHDL Functions, VHDL Procedures, Attributes, Creating Overloaded Operators, Multi-Valued Logic and Signal Resolution,The IEEE 9-Valued Logic System, SRAM Model Using IEEE 1164, Model for SRAM Read,Write System, Generics, Named Association, Generate Statements, Files and TEXTIO.Testing Combinational Logic , Testing Sequential Logic, Scan Testing, Boundary Scan, Built-In-Self-Test.

Recommended Readings:

1. Charles Roth; Digital Systems Design Using VHDL; Cengage Learning ,2006
2. Charles Roth ,Lizy John; Digital Systems Design Using VHDL; 2nd Ed., Thomson, 2008.
3. Volnei A. Pedroni; Circuit Design with VHDL; MIT Press, 2004
4. Stephen Brown, Z. Vranesic; Fundamentals of Digital Logic Design with VHDL Second Edition; McGrawHill.2007
5. JayaramBhasker; VHDL Primer, A; 3rd Edition; Prentice Hall; 1999

List of Experiments in Computer Hardware Design:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Introduction to lab hardware & software (Tutorial)
2. Combinational Logic Design Using Schematic Capture
3. Implementation of decoders.
4. Implementation of adders and subtractors
5. Sequential Logic Design Using Schematic Capture
6. Combinational Logic Design Using VHDL
7. Sequential Logic Design Using VHDL
8. Parameterized VHDL Universal Register/Counter
9. Implementation of interfacing with stepper motor, seven segment displays etc.

COMP 5.5 DATABASE MANAGEMENT SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
						Marks					
		L	T	P	ThDuration(Hrs)	Th	S	TW	P	O	Total
COMP 5.5	Database Management System	3	1	2	3	100	25	--	25		**150**

Course Objectives:

1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. To understand the relational database design principles.
4. To become familiar with the basic issues of transaction processing and concurrency control.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the basic elements of a relational database managament system.
2. Ability to identify the data models for relevant problems.
3. Ability to design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respect data.
4. Apply normalization for the development of application softwares.

UNIT -1 (12 Hours)

Introduction to Databases: Characteristic of Database Approach,Advantages of using DBMS approach.Overview of Database Languages and Architecture: Data Models, Schemas and instances, Three Scheme Architecture and Data Independence. Relational model concepts, Constraints and relational Database schema ,Update operation, transactions and dealing with constraint violation.Formal Relational languages: Unary relational operations, relational algebra operations from set theory, Binary relational operations JOIN and DIVISION. Additional relational operations.

UNIT - 2 (12 Hours)

SQL data Definition and Data types, specifying constraints in SQL, Basic retrieval in SQL. INSERT,DELETE and UPDATE statement in SQL. More Complex SQL retrieval queries. Specifying constraints as Assertions and Actions as triggers. VIEWS in SQL. Schema Change statement.

UNIT – 3 (12 Hours)

Using High level conceptual Data model for Database Design, Entity type, entity set, attributes and keys, relationship type,relationship set roles and structural constraints, weak entity type,ER diagrams, Naming conventions and Design issues, subclasses, super classes and Inheritance,Specialization and Generalization in EER. Mapping the conceptual design into logical design: ER to relational mapping.Introduction to Normalization. Informal design guidelines for relational schemas, Functional dependencies, Normal forms based on primary keys(1NF,2NF ,3NF),BCNF.Database Design Theory: Inference rules, Equivalence and minimal cover.

UNIT - 4 (12 Hours)

Foundation of Database Transaction Processing: Transaction and system concepts, desirable properties of transaction, Characterizing schedules based on recoverability Characterizing schedules based on serializability Introduction to protocols for concurrency control in Databases: Two phase locking technique for concurrency control , concurrency control based on timestamp ordering, Multiversion concurrency control technique, validation concurrency control technique.Introduction to Database security: Security Issues

Recommended Readings:

1. RamezElmasri, ShamkatB.Navathe; Database Systems; 6th edition; Pearson Education; 2013.
2. A.Silberschatz, H.F. Korth, S.Sudarshan;Data base System Concepts; McGraw Hill; VI edition, 2006.
3. Raghurama Krishnan, Johannes Gehrke;Data base Management Systems; TATA McGrawHill; 3rd Edition; 2003.
4. Thomas Connolly, Carolyn Begg;Database Systems; A Practical approach to Design implementation and Management Fourth edition; Pearson education.
5. P.K. Das Gupta; Database Management System Oracle SQL and PL/SQL; PHI.
6. C.J. Date; Introduction to Database Systems; Pearson Education.

List of Experiments in Database Management Systems:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To study Data Definition language Statements.
2. To study Data Manipulation Statatements.
3. Study of SELECT command with different clauses
4. Study of GROUP functions (avg, count, max, min, Sum).
5. Study of various type of SET OPERATORS (Union, Intersect, Minus).
6. Study of various type of Integrity Constraints.
7. Study of Various type of JOINS.
8. To study Views
9. To Study Triggers

COMP 5.6 OPERATING SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.6	Operating Systems	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. To provide students a comprehensive introduction to understand the underlying principles, techniques and approaches which constitute a coherent body of knowledge in operating systems.
2. To provide students an understanding of concurrent processes, multi-threads, CPU scheduling, memory management, file system, storage subsystem, and input/output management.
3. To provide students knowledge of the components and management aspects of different types of operating systems.
4. To equip students with necessary skills required for Shell Programming.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the fundamental concepts of operating systems, its evolution and various architectures.
2. Understand relationship between subsystems of a modern operating system.
3. Gain knowledge on processes, threads, concurrency control, synchronization, CPU scheduling and semaphores.
4. Analyze algorithms used in process management, deadlock handling, memory management, file systems and I/O systems.
5. Be familiar with various types of operating systems including UNIX.

UNIT -1 (12 Hours)

Process management:Processes concept, Processscheduling,Operations on processes, Interprocess communication. Threads:Overview,Multithreading models,Threading issues. CPU Scheduling:Basic Concepts,Scheduling Criteria,Scheduling Algorithms: FCFS,SJF, SRTF / SRTN,Priority Scheduling ,Round Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Fair Share Scheduling, Multiprocessor Scheduling,Real – Time Scheduling

UNIT - 2 (12 Hours)

Process Synchronization:Critical Section Problem, Petersons solution, Synchronisation hardware support,Mutex locks,Semaphores,Classical problems of syncronisation using semaphores (Producer – Consumer problem, Readers – Writers problem,Dining philosophers Problem), Monitors.

Deadlocks:System model,Deadlock characterization,Methods for handling deadlocks,Deadlock prevention,Deadlock avoidance,Deadlock detection,Recovery from deadlock.

UNIT – 3 (12 Hours)

Memory Management: Background, Swapping, Contiguous allocation, Segmentation, Paging, Structure of the page table

Virtual Memory: Demand Paging, Copy on write, Pagereplacement algorithms (FIFO, Optimal page replacement, Least Recently used), Allocation of frames, Thrashing.

File System Interface: File Concept, Access methods, Directory and Disk Structure.

File system implementation:File system structure,Implementation,Directory implementation,Allocation methods

UNIT –4 (12 Hours)

I/O Systems: I/O Hardware, Application I/O Interface, Kernel I/O subsystem, Transforming I/O requests to hardware operations.

Secondary Storage structure: Disk structure and attachment, Disk scheduling, Disk management

Linux Commands:Basic Linux commands,Essential Shell Programming.

Recommended Readings:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; Operating System Concepts; 9th Edition
2. William Stallings; Operating systems internals and design principles;7th edition
3. Sumitabha Das; UNIX – Concepts and applications;4th edition
4. A.S Tanenbaum; Operating systems, Design and implementation;3rd edition
5. Milenkovic; Operating Systems,2nd edition

6. William E. Shotts, Jr;The Linux Command Line: A Complete Introduction;3rd edition

List of Experiments in Operating systems:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Study of various types of operating systems
2. Implementation of system calls
3. Non preemptive CPU scheduling algorithms
4. Preemptive CPU scheduling algorithms
5. Implementation of threads
6. Process synchronization using semaphores
7. Implementation of deadlock avoidance scheme
8. Memory allocation techniques
9. Page replacement methods
10. Disk scheduling algorithms
11. Shell scripting

COMP6.1 SOFTWARE TESTING AND QUALITY ASSURANCE

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.1	Software Testing and Quality Assurance	3	0	0	3	100	25	--	--	--	**125**

Course Objectives:

1. To develop and implement an effective testing strategy,.
2. To plan and prepare appropriate tests for all phases of software development
3. To measure and control the quality of the testing.
4. To understand the significance of finding and resolving errors early.
5. To develop strategies and techniques for building-in quality from the start.

Course Outcomes:

The student after undergoing this course will be able to:

1. Manage, plan and prepare rigorous, formal, visible and repeatable tests that will fully exercise software, in the development of quality systems.
2. Apply.different testing approaches to all stages of software development
3. Prepare test plans, strategy, specifications, procedures and controls to provide a structured approach to testing.
4. Apply the techniques and methods covered to testing packages.
5. Describe the different types of testing tools available and identify the appropriate types of tools for their needs.

UNIT -1 (12 Hours)

Quality perspective and expectations, Quality framework and ISO 9126, Correctness and defects. Quality Assurance: Classification, Defect prevention, Defect reduction, Defect containment. Quality Assurance in context: Handling discovered defects during QA activities, QA activities, Verification and validation perspective.

Quality Engineering: Activities & Process, Quality planning, Quality assessment & improving, ISO 9000 series standards. Capability Maturity Model integration for software engineering.

UNIT -2 (12 Hours)

Purpose, activities, process and context, issues and questions about testing, Functional v/s structural testing, Coverage based v/s usage based testing.Test planning and preparation, Test execution, result checking and measurement, Analysis and follow up, Activities, people and management. Coverage and usage testing based on checklists and partitions: Checklist based testing and limitations. Testing for partition coverage, Usage based statistical testing with Musa's operational profiles. Input domain partitioning and testing, simple domain analysis and extreme point combination strategies, Testing strategies based on boundary analysis.

UNIT - 3 (12 Hours)

Defect prevention and process improvement: Basic concepts and generic approaches, Root cause analysis for defect prevention, Training for defect prevention, Defect prevention techniques.Basic Control flow testing, Data Dependency and data flow testing.Finite State Machines (FSM)and testing, FSM testing, FSM based testing of web based applications, Markov Chains and Unified Markov Models (UMM) for testing, Using UMM's for usage based statistical testing, Testing based on Web usages.

UNIT - 4 (12 Hours)

Software testing tools and overview: Need for automated testing tools, Taxonomy of testing tools, Functional/Regression testing tools, Performance testing tools, Testing management tools, Source code testing tools, Selection of testing tools. Case study: Overview of WinRunner, Loadrunner, Quick Test Professional and SQA Robot.

Recommended Readings:

1. Software Quality Engineering – Testing, Quality Assurance and Quantifiable Improvement by Jeff Tian, Edition 2006, ISBN: 81-265-0805-1
2. Software Testing Tools by Dr. K.V.K.K. Prasad.
3. Effective methods for Software testing by William E. Perry, 3rd edition.
4. Introducing Software testing by Louise Tamares, ISBN: 81-7808-678-6

COMP 6.2 DESIGN AND ANALYSIS OF ALGORITHMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.2	Design and Analysis of Algorithms	3	1		3	100	25		-	-	**125**

Course Objectives:

1. To learnalgorithm designing techniques and understand the asymptotic behavior of algorithms in general and sorting algorithms in particular.
2. To acquire the knowledge of analyzing algorithms using the Recurrence relation techniques.
3. To gain the complete understanding of Dynamic Programming, Greedy Algorithmic strategies, Backtracking methods and Branch and Bound Techniques.
4. To comprehend the analysis of graph algorithms and to learn about the single source shortest paths and all pairs shortest path algorithms and their analysis in graph theory.
5. To get an in-depth understanding of randomized algorithms, NP-Completeness and reduction and FFT and sorting networks.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe and analyze the complexity and asymptotic behavior of algorithms in general and sorting algorithms in particular.
2. Trace time complexity of algorithms using the Recurrence relation techniques.
3. Explain and apply dynamic programming, greedy algorithms, backtracking and branch and bound techniques to various problems.
4. Analyze graph algorithms and find the single source shortest paths and all pairs shortest paths in any graph.
5. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.

UNIT - 1 (12 Hours)

Introduction to Algorithms. Algorithm specification, Performance analysis, Growth of Function, Recurrences, Randomized Algorithms.
Divide and conquer: General Method, Binary Search, Finding the maximum and minimum, MergeSort, QuickSort, Selection Sort, Strassen's Matrix Multiplication. Greedy method strategy: General method, Knapsack problem, Job sequencing with deadlines.

UNIT - 2 (12 Hours)

Dynamic Programming: General method, Multistage graphs, Optimal Binary Search Trees, 0/1 Knapsack, Travelling Salesperson problem, Flow Shop Scheduling.
Backtracking: General method, 8-Queens Problem, Sum of Subsets Problem, Graph Coloring, Hamiltonian Cycles, Knapsack Problem.
Branch and Bound: The Method,0/1 Knapsack Problem, Traveling Salesperson.

UNIT - 3 (12 Hours)

Internet Algorithms: Strings and patterns matching algorithm. Tries.Text compression.Text similarity testing.
Introduction to parallelism models: Simple algorithms for parallel computers, CRCW and EREW algorithms. Probabilistic Algorithms: Expected versus average time, Pseudorandom generation,Buffon's needle, numerical integration, Probabilistic counting,Monte Carlo algorithms .

UNIT - 4 (12 Hours)

NP-Hard and NP-Complete Problems: Basic concepts, NP-Hard Graph Problems: CDP, NCDP, CNDP, DHC NP-Hard Scheduling Problems: Scheduling Identical Processors, Flow-Shop Scheduling, Job Shop Scheduling. Approximation algorithms: Vertex cover problem, traveling-salesperson problem, set-covering problem.

Recommended Readings:

1. Computer Algorithms by Horowitz, Sartaj Sahni. Rajasekharan – Galgotia, ISBN: 9788175152571
2. Computer Algorithms – Saar Baase. PHI , ISBN: 0201612445
3. Fundamentals of Algorithms by Gilles Brassard and Paul Bratly. PHI, ISBN: 9780133350685.
4. Algorithm Design Foundation, Analysis and Internet Examples by Michael Goodrich & Roberto Tamassia,, Second Edition, Wiley student Edition.
5. Introduction to algorithms by Thomas H cormen, Charles E Leiserson, Ronald L Rivest.PHI, ISBN: 81-203-1353-4

COMP6.3 ARTIFICIAL INTELLIGENCE

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.3	Artificial Intelligence	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. To study the concepts of Artificial Intelligence
2. To learn the methods of solving problems using Artificial Intelligence
3. To know various AI search algorithms
4. To study the techniques nod machine learning, natural language processing, experts systems and planning.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the concepts of A.I. problem solving.
2. Identify the knowledge representation schemes.
3. Apply the various reasoning mechanism in decision making.
4. Explain the concepts of planning, perception and learning and their applications.
5. Design an expert system for a given application domain.

UNIT -1 (12 Hours)

Introduction to Artificial Intelligence and A.I. Techniques. Problems, Problem Spaces & Search. Defining the Problem as state space search. Production Systems, Problem characteristics, Production System Characteristics, Design issues in Searching. Uninformed Search Techniques: BFS, DFS, Depth-limited search and Iterative deepening DFS. Heuristic Search Techniques: Hill Climbing, Best First Search, Problem Reduction, Constraint satisfaction. Means Ends Analysis.

UNIT -2 (12 Hours)

Knowledge Representation: Representation & Mapping, Approaches to knowledge Representation, Predicate Logic, Representing simple facts & logic. Representing instance & ISA relationship, Computable functions & predicates. Unification. Resolution. Symbolic Reasoning under uncertainly. Introduction to non-monotonic Reasoning. Logic for non-monotonic Reasoning. Weak slot and filter structures: Semantic nets, Frames. Strong Slot and Filter Structures: Conceptual dependency, Scripts

UNIT - 3 (12 Hours)

Game Playing: Mini Max Search Procedure, Adding alpha-beta cut offs. Planning: Components of a planning system, Blocks World Problem. Goal Stack Planning Non–linear Planning, Hierarchical Planning. Introduction to natural language processing:

UNIT - 4 (12 Hours)

Learning: Rote learning, learning by taking advice, learning in problem solving. Inductive learning: version space, decision trees: statistical learning: Naïve Bayes Technique. Neural networks: topology and activation functions, Perceptron learning algorithm and back propagation algorithm.
Expert systems: Characteristics and design, Types of expert system, Expert system shells

Recommended Readings:

1. Elaine Rich and Kevin Knight; Artificial Intelligence; Third Edition, TMH
2. Struart Russell and Peter Norvig;Artificial Intelligence, a Modern Approach, Third Edition, Pearson Education.
3. Ela Kumar; Artificial Intelligence; I. K. International Publishing House, 2011.
4. Nils J. Nillson; Artificial Intelligence: A new Synthesis; Harcourt Asia
5. Patrick Winston; Artificial Intelligence; Pearson Education
6. George F. Luger; Artificial Intelligence : Structures and strategies for complex problem solving, Pearson education.
7. Saroj Kaushik; Artificial Intelligence; Cengage Learning, 2011

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Implementation of BFS
2. Implementation of Depth limited DFS
3. Implementation of Hill Climbing Search
4. Implementation of A* search
5. Implementation of Crypt arithmetic problem
6. Implementation of Unification Algorithm.
7. Implementation of TIC TAC TOE problem using Min-Max-Search
8. Implementation of an Expert System
9. Implementation of Logical functions using neural networks.

COMP 6.4 COMPUTER GRAPHICS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.4	Computer Graphics	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. This course is designed to provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.

2. A thorough introduction to computer graphics techniques, including 3D modeling, rendering and animation. Topics cover: geometric transformations, geometric algorithms, 3D object models (surface and volume), visible surface detection algorithms, image synthesis, shading and mapping, global illumination and animation techniques

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the concepts of computer graphics system
2. Implement the algorithms for two dimensional transformations.
3. Demonstrate the techniques of clipping
4. Explain the basics of 3D Graphics and three dimensional transformations.

UNIT - 1 (12 Hours)

Overview of graphic systems: Raster scans systems, Random scan systems.

Output Primitives. Points and lines, Line drawing algorithms, DDA, Bresenhams line algorithm Circle generating algorithms, Properties of circles, Midpoint circle algorithm, Ellipse generating algorithm, Properties of Ellipses, Midpoint ellipse algorithm, Filled area primitives, Scan line polygon Fill algorithm, Inside – outside tests, Scan line fill of curved boundary, Boundary fill algorithm, Flood fill algorithm, Fill area functions.

UNIT - 2 (12 Hours)

Two Dimensional Geometric Transformations: Basic Transformations, Translation, Rotation, Scaling, Composite transformation, Translations, Rotations, Scaling, Other transformations- Reflection, Shear.

Two-Dimensional Viewing: The viewing pipeline, Viewing coordinate reference frame, Window to viewport coordinate transformation, 2-D viewing functions, Clipping operations, Point Clipping, Line clipping , Cohen- Sutherland Line Clipping, Polygon Clipping, Sutherland Hodgeman Polygon clipping, Weiler- Atherton Polygon Clipping, Curve clipping, Text clipping.

UNIT - 3 (12 Hours)

Three Dimensional Concepts: 3- Dimensional display methods, Parallel projections Perspective projection, Depth cueing, Surface rendering, Exploded and cutaway views. Three Dimensional Object representations- Polygon surfaces, Polygon tables, Three Dimensional Geometric and Modeling transformations- Translation Rotation, Coordinate Axes, rotations , Scaling , Reflections , Shears
Three Dimensional Viewing, Transformation from world to viewing coordinates Projections.
Picture Structure: Defining Symbols By Procedures, Display Procedures, Boxing, Structured Display Files. Techniques for Achieving Realism. Curves And Surfaces: Shape Description Requirements, Parametric Functions, Bezier Methods. B-Spline Methods.

UNIT - 4 (12 Hours)

Classification of visible – surface detection algorithms, Back – Face detection , Depth buffer method , A – Buffer method , Scan – Line method , Depth Sorting method , BSP- Tree method, Area Sub-division method.

Color Models and Color Applications- Properties of light ,Standard primaries and the, Chromaticity Diagram, XYZ Color model, CIE Chromaticity Diagram, RGB color model, YIQ Color Model , CMY Color Model, HSV Color Model, HLS Color Model

Computer Animation: Design of animation sequences, General computer animation functions, Raster Animations, Computer animation languages, Motion specification, Direct motion specification, Goal directed systems Kinematics and dynamics.

Recommended Readings:

1. Donald Hearn and M. P. Baker ; Computer Graphics; Prentice Hall of India Pvt. Ltd.
2. William Newman and Robert Sproull; Principles of Interactive Graphics; Tata McGraw hill Publishing company Ltd.
3. N. Krishnamurthy; Introduction to Computer Graphics; TMH
4. Steven Harrington; Computer Graphics; Tata McGraw Hill.
5. Foley, Van Dam, Feiner and Hughe; Computer Graphics: Principles and Practice

List of Experiments in Computer Graphics:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To study basic Graphics Primitive functions
2. To draw a Line using DDA line drawing algorithm
3. To draw a line using Breshenham's algorithm
4. To draw circle using midpoint circle algorithm
5. To draw an ellipse using mid-point ellipse algorithm.
6. To translate, rotate and scale the 2D object.
7. To translate, rotate and scale the 3D object.
8. To fill polygon using boundary fill algorithm.
9. To fill polygon using flood fill algorithm.
10. To implement Cohen–Sutherland 2D clipping and window–viewport mapping
11. To perform 2-D animation

COMP 6.5 EMBEDDED SYSTEM DESIGN

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
					Th Duration (Hrs)	Marks					
		L	T	P		Th	S	TW	P	O	Total
COMP 6.5	Embedded System Design	3	1	2	3	100	25	-	-	25	150

Course Objectives:

The subject aims to provide the student with:

1. To conceptualize the basics of embedded systems.
2. To conceptualize the basics of organizational and architectural issues of a microcontroller
3. To learn programming techniques used in microcontroller
4. To understand fundamentals of real time operating system

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain about microcontrollers embedded processors and their applications
2. Develop the programs for microcontroller.
3. Describe the role of embedded systems in industry.

UNIT – 1 (12 Hours)

Overview of Embedded System Architecture, Application areas, Categories of embedded systems, specialties of embedded systems. Recent trends in embedded systems. Brief introduction to embedded microcontroller cores CISC, RISC, ARM, DSP and SoC (System on Chip).Introduction to 8051 Microcontroller, Architecture, Pin configuration, Memory organization, Input /Output Ports, Counter and Timers, Serial communication, Interrupts.

UNIT –2 (12 Hours)

Assembly Language Programming of 8051:Instruction set, Addressing modes, Development tools, Assembler Directives, Programming based on Arithmetic & Logical operations, I/O parallel and serial ports, Timers & Counters, and Interrupt Service Routine.

UNIT –3 (12 Hours)

Embedded / Real Time Operating System:Architecture of kernel, Task and Task scheduler, Interrupt service routines, Semaphores, Mutex, Mailboxes, Message queues, Event registers, Pipes, Signals, Timers, Memory management, Priority inversion problem. Off-the-Shelf Operating Systems, Embedded Operating Systems, Real Time Operating System (RTOS) and Handheld Operating Systems.

UNIT –4 (12 Hours)

Embedded System - Design case studies:Digital clock, Battery operated smart card reader, Automated meter reading system, Washing Machine, Microwave Owen, Automotive Embedded Systems. Embedded software development tools: Code generation tools, Simulator, Testing and debugger, Integrated Development Environments (IDE) for 8051 systems, Memory and Processor sensitive program and device drivers.

Recommended Readings:

1. M. A. Mazidi, J. G. Mazidi, R. D. McKinlay; The 8051 microcontroller & Embedded systems; , Pearson
2. Kenneth J. Ayala, Dhananjay V. Gadre; . The 8051 microcontroller & Embedded systems; Cengage Learning.
3. Dr. K. V. K. K. Prasad; Embedded / real – time systems: concepts, design & programming, Black Book;Dreamtech press, Reprint edition 2013.
4. Raj Kamal; Embedded System: architecture, programming and design; TMH.
5. Frank Vahid; Tony Givargis,;John Wiley; Embedded System Design;
6. Laya B. Das, Pearson; Embedded systems an integrated approach;
7. Frank Vahid,;TonyGivargis;Embedded system design A Unified hardware/software Introduction.
8. Shibu K.V; Introduction to Embedded Systems; Mc Graw Hill

List of Experiments:

(At least 8 experiments should be conducted based on the broad areas listed below)

Using Keil

1. Write a program to send ASCII values 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e to port 1
2. Write a program to toggle the bits of P1
3. Write a program to send and receive data serially
4. Programming based on arithmetic operations in 8051
5. Programming based on logical operations in 8051
6. Programming based on timers in 8051
7. Programming based on interrupts in 8051

Based on RTOS

1. To implement Shortest Job First Scheduling algorithm
2. To implement Priority Inheritance Protocol
3. Case Study: Reliability & Fault tolerance in RTOS

Case Study on Embedded System

1. Digital clock,
2. Battery operated smart card reader,
3. Automated meter reading system,
4. Washing Machine,
5. Microwave Owen,
6. Automotive Embedded Systems

COMP 6.6 COMPUTER NETWORKS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.6	Computer Networks	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To provide an introduction to basic concepts of communication and networks.
2. To understand the underlying principles and approaches in computer networks.
3. To provide students knowledge of data link, network, transport and application protocols.
4. To equip students with necessary skills required for network programming.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the fundamental concepts of computer networks, its evolution and architectures.
2. Have a good understanding of the OSI and TCP/IP Reference models
3. Have a detailed understanding of data link, network, transport and application layer protocols.
4. Analyze the requirements for a given organization and select the most appropriate networking architecture and technologies.
5. Have knowledge of network programming and tools.

UNIT -1 (12 Hours)

Reference Models: The OSI Reference Model, The TCP/IP Reference Model, A Comparison of the OSI and TCP/IP Reference Models. The Physical Layer: The Theoretical Basis for Data Communication, Fourier Analysis, Bandwidth-Limited Signals, The Maximum Data Rate of a Channel. Data Link Layer Design Issues, Error-Correcting Codes, Error –Detecting Codes, Elementary Data Link Protocols- An Unrestricted Simplex Protocol, A Simplex Stop-and-Wait Protocol, A Simplex Protocol for a Noisy Channel, Sliding Window Protocols - A One-Bit Sliding Window Protocol, A Protocol Using Go Back N.

UNIT - 2 (12 Hours)

Multiple Access Protocols – ALOHA, Carrier Sense Multiple Access Protocols, Collision-Free Protocols, Limited-Contention Protocols, Wavelength Division Multiple Access Protocols, Wireless LAN Protocols, Ethernet - Ethernet Cabling, Manchester Encoding, The Ethernet MAC Sublayer Protocol. Network Layer Design Issues,Routing Algorithms- Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing,Congestion Control Algorithms -General Principles of Congestion Control, Congestion Prevention Policies, Congestion Control in Virtual-Circuit Subnets and Datagram Subnets, Load Shedding. The IP Protocol, IP Addresses, Internet Control Protocols.

UNIT - 3 (12 Hours)

The Transport Layer: The Transport Service, Services Provided to the Upper Layers,Transport Service Primitive, Berkeley Sockets. Elements Of Transport Protocols- Addressing, Establishing a Connection, Releasing a Connection, The Internet Transport Protocols: UDP, TCP- The Service Model, The protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release. The Application Layer:Domain Name System- Name Space, Resource Records, Name Servers, Electronic Mail- Architecture and Services, The User Agent, Message Formats, Message Transfer, Final Delivery

UNIT - 4 (12 Hours)

Introduction to NS2: Simulator structure, Simulator input and Output, NS2 Installation steps, NS2 Directories & Files, Network Animator, NS2 Program Structure. Basics of Protocol Simulation Using NS2: Tcl, Program execution, Basic programming constructs,arrays, Lists, Dictionaries, Procedures, File handling, Object oriented Tcl, General structure of AWK Scripting, Gnuplot. Wired Network simulation and wireless network simulation.

Recommended Readings:

1. Andrew S. Tannenbaum; Computer Networks; PHI, 4th Edition
2. Ajit Kumar Nayak, SatyanandaChampatiRai, Rajib Mall; Computer Network Simulation Using NS2,CRC.
3. Behrouz A. Forouzan; Data communication and Networking; Tata McGraw Hill.
4. Williams Stallings; Data and Computer Communications; PHI.
5. Fred Halsall; Data Communications, Computer Networks and Open Systems; Pearson Education

List of Experiments in Computer Networks:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Study of Networking and Internetworking devices
2. Implementation error detection mechanisms
3. Implementation of error correction codes
4. Study of Network Layer- IP Addresses
5. Implementation of shortest Path Routing algorithms
6. Socket Programming in TCP/UDP
7. Installation of NS2
8. Implementation of Destination sequenced distance vector
9. Implementation of Dynamic source routing
10. Implementation of Temporally ordered routing algorithm
11. Implementation of Adhoc on demand distance vector

APPENDIX

QUESTION PAPER PATTERN

Syllabus in each subject will have 4units.

Question paper shall be drawn as follows:

Question No	From Units	No. of Questions to be Set	No. of Questions to be Answered	Remarks
1-3	1-2	3 x 20marks	2 x 20 marks	Each unit shall have minimum 20 marks
4-6	3-4	3 x 20 marks	2 x 20 marks	Each unit shall have minimum 20 marks
7-8	1-4	2 x 20 marks	1 x 20 marks	---
		8 - 160 marks	5 - 100 marks	

SAMPLE QUESTION PAPER

SUBJECT: MARKS: 100

MAXIMUM DURATION: 3 hours

Instructions to the candidates:

1.

2

Part –A (Questions to be drawn from units 1 & 2)

Answer any **TWO** questions from the following: 2 x 20= 40 Marks

Question-1 .....20 Marks

a)

b)

..

Question-2 .....20 Marks

a)

b)

..

Question-3 .....20 Marks

a)

b)

..

Part –B (Questions to be drawn from units 3 & 4)

Answer any **TWO** questions from the following: 2 x 20= 40 Marks

Question-4 .....20 Marks

a)

b)

..

Question-5 .....20 Marks

a)

b)

..

Question-6 .....20 Marks

a)

b)

..

Part –C (Questions to be drawn from all units i.e. units 1 - 4)

Answer any **ONE** question from the following: 1 x 20= 20 Marks

Question-7 .....20 Marks

a)

b)

..

Question-8 .....20 Marks

a)

b)

..

..

BOARD OF COMPUTER ENGINEERING
SCHEME AND SYLLABUS FOR SEM VII AND VIII OF RC-2016-17

COURSE

FINALYEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER - VII

Subject Code	Nomenclature of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP7.1	Compiler Construction	3	1	2	3	100	25	--	--	25	**150**
COMP7.2	Data Mining	3	1	2	3	100	25	--	--	--	**125**
COMP7.3	Image Processing	3	1	2	3	100	25	--	--	25	**150**
COMP7.4	Elective I	3	1	2	3	100	25	--	--	--	**125**
COMP7.5	Elective II	3	1	-	3	100	25	--	--	25	**150**
COMP7.6	Project	--	--	4	--	--	--	--	--	25	**25**
TOTAL		**15**	**05**	**12**	**--**	**500**	**125**	**--**	**-**	**100**	**725**

#A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

Electives: A student must take One Elective from each Group.

Elective I		Elective II	
COMP7.4.1	VLSI Design	COMP7.5.1	Entrepreneurship Development
COMP7.4.2	Data Compression	COMP7.5.2	Geographical Information System
COMP7.4.3	Fuzzy Logic and Neural Networks	COMP7.5.3	Design Patterns and Frameworks
COMP7.4.4	Web Technologies	COMP7.5.4	Project Management and Quality Assurance
COMP7.4.5	Cloud Computing	COMP7.5.5	Big Data Analytics

FINAL YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER - VIII

Subject Code	Nomenclature of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP8.1	Distributed Operating Systems	3	1	2	3	100	25	--	--	25	**150**
COMP8.2	Network Security	3	1	2	3	100	25	--	--	--	**125**
COMP8.3	Elective III	3	1	2	3	100	25	--		25	**150**
COMP8.4	Elective IV	3	1	2	3	100	25	--	--	25	**150**
COMP8.5	Project	--	--	8	--	--	--	75	--	75	**150**
TOTAL		**12**	**04**	**16**	**--**	**400**	**100**	**75**	**-**	**150**	**725**

#A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

Electives: A student must take One Elective from each Group.

Elective III		Elective IV	
COMP8.3.1	Operation Research	COMP8.4.1	Genetic Algorithms
COMP8.3.2	Multimedia Systems	COMP8.4.2	Real Time Systems
COMP8.3.3	Bio Informatics	COMP8.4.3	Mobile Computing
COMP8.3.4	Storage Area Networks	COMP8.4.4	Machine Learning
COMP8.3.5	Web Services	COMP8.4.5	Digital Signal Processing

COMP 7.1 COMPILER CONSTRUCTION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.1	Compiler Construction	3	1	2	3	100	25	-	-	25	**150**

Course Objectives:

1. This subject introduces various language translators involved in the process of translating a modern high-level language to executable code. The subject discusses phase/pass structure of Assembler, Macro preprocessor, Linker, Loader, and Compiler in greater detail.

Course Outcomes:

The student after undergoing this course will be able to:

1. Know the major steps involved in translating a high-level programming language down to a low-level target machine language.
2. understand the relationship between machine and assembly language, compilers, interpreters, linkers, loaders, assemblers and macro preprocessors.

UNIT - 1 (12 Hours)

Language processor concepts. Data Structures for language processors.
Introduction to Compiler, Phases of compilation, Bootstrapping and Porting, Compiler writing tools.
The role of a lexical analyser. Design of lexical analyzer. Implementation of lexical analyzer.
A Language for specifying lexical analyzer. Study of the features and applications of LEX/FLEX tool.

UNIT - 2 (14 Hours)

Overview of Context free grammar. Derivations and Parse trees, Ambiguity, Left recursion, Left factoring.
Top down parsing: Recursive descent parsing and Predictive parsers.
Bottom up parsing: Shift-reduce parsers. Operator precedence parsers, LR parsers. YACC parser generator

UNIT - 3 (14 Hours)

Intermediate Code Generation: Intermediate Language, Declarations, Assignment statements, Boolean expressions, Case statement, Procedure call.
Run Time environments: Source language issues, Storage organization, Storage allocation strategies.
Symbol tables: The content of a symbol table, Data structures for Symbol Table, Representing scope information.
Error detection and recovery: Lexical phase errors, Syntactic phase errors, Semantic errors.

UNIT - 4 **(12 Hours)**

Code generation: Issues in the design of a code Generator, Basic blocks and flow graphs, Next-use information, A simple Code generator, The DAG representation of Basic blocks, Peephole Optimization, Generating code from DAGS.

Code optimization: The principle sources of optimization, Optimization of basic blocks, Machine dependent optimization, Register allocation optimization.

Recommended Readings:

1. Aho and Ulman ; Principles of Compiler Design; Narosa publishing House, ISBN: 81-85015-61-9.
2. Aho, Ulman and Sethi; Compilers, Principles, techniques and tools; Pearson Education Asia, ISBN: 81-7808-046-X.
3. Vinu V. Das ; Compiler design with FLEX and YACC; PHI publication, ISBN:978-81-203-3251-5
4. Louden; Compiler Construction, Principles and Practice; Galgotia Publication, ISBN:0-534-93972-4

List of Experiments in Compiler Construction:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. A program to detect tokens from user defined expression.
2. A LEX program to find if the input is integer, real number or word.
3. A LEX program to add line numbers for given text.
4. A LEX program to convert decimal numbers to hexadecimal numbers.
5. A LEX program to compute average of given set of numbers.
6. A YACC program to parse an expression for a given grammar.

7. A program that combines YACC and LEX.

8. A program to obtain First and Follow for a user specified grammar.

9. A program to obtain Leading and Trailing for a user specified grammar.

10. To implement code generation algorithm.

COMP7.2 DATA MINING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP7.2	Data Mining	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

1. This course will focus on imparting a complete introduction to data mining for students.
2. It will provide a sound understanding of the foundations including fundamental concepts and algorithms of data mining which intend to search through data for hidden relationships and patterns.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the basic concept of preprocessing the data before using it in the mining algorithms and different types of data mining tasks.
2. Understand and apply the principles of various classification techniques
3. Understand and apply the principles of various association mining techniques.
4. Study and apply various clustering algorithms and anomaly detection techniques.

UNIT -1 (12 Hours)

Introduction –Challenges, Origin of Data Mining, Data Mining Tasks

Data -- *Types of Data*:Attributes and Measurement , Types of Data Sets
Data Quality:Measurement and Data Collection Issues, Issues Related to Applications
Data Pre-processing : Aggregation, Sampling, Dimensionality Reduction, Feature Subset Selection, Feature Creation, Discretization and Binarization, Variable Transformation.

UNIT - 2 (12 Hours)

Measures of Similarity and Dissimilarity
Similarity and Dissimilarity between Simple Attributes Dissimilarities between Data Objects Similarities between Data Objects Examples of Proximity Measures Issues in Proximity Calculation Selecting the Right Proximity Measures.

Summary Statistics
Frequencies and the Mode, Percentiles,
Measures of Location: Mean and Median

Measures of Spread: Range and Variance
Multivariate Summary Statistics, Other Ways to Summarize the Data

Data Cube and OLAP
Data Cube: A multidimensional data model.*Schemas for Multidimensional data model* : Star, Snowflakes and Fact Constellation schemas. *Dimensions*: The role of Concept Hierarchies. *Measures*: Categorization and Computation. *OLAP Operations*.

UNIT - 3 (12 Hours)

Classification
General Approach to Solving a Classification Problem
Decision Tree Induction: Working, Construction, Methods for Expressing Attribute Test Conditions, Measures for Selecting the Best Split, Algorithm and Characteristics for Decision Tree Induction.
Model Overfitting: Overfitting Due to Presence of Noise, Overfitting Due to Lack of Representative Samples, Overfitting and the Multiple Comparison Procedures, Estimation of Generalization Errors, Handling Overfitting in Decision Tree Induction.

Rule-Based Classifier: Concept, Rule-Ordering Schemes, Building a Rule-Based Classifier
Direct Methods for Rule Extraction, Indirect Methods for Rule Extraction, Characteristics of Rule-Based Classifiers.
Nearest-Neighbor classifiers: Algorithm, Characteristics of Nearest-Neighbor Classifiers.

UNIT - 4 (12 Hours)

Association Analysis
Frequent Itemset Generation, The Apriori Principle, Frequent Itemset Generation in the Apriori Algorithm, Candidate Generation and Pruning, Support Counting, Computational Complexity, *Rule Generation*: Confidence-Based Pruning, Rule Generation in Apriori Algorithm, Maximal Frequent Itemsets, Closed Frequent Itemsets.
Alternative Methods for Generating Frequent Itemsets.

Cluster Analysis
K-means :The Basic K-means Algorithm, K-means: Additional Issues, Bisecting K-means
K-means and Different Types of Clusters, Strengths and Weaknesses.
Agglomerating Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm, Key Issues in Hierarchical Clustering, Strengths and Weaknesses.

Anomaly Detection

Statistical Approaches :Detecting Outliers in a Univariate Normal Distribution, Outliers in a Multivariate Normal Distribution, A Mixture Model Approach for Anomaly Detection
Strengths and Weaknesses.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Implementation of data preprocessing techniques
2. Implementation of measures of similarity and dissimilarity.
3. Study of Data Cube and OLAP Operations.
4. Implementation of K-Nearest Neighbor Classifier.
5. Implementation of Classification by Decision Tree Induction.
6. Implementation of Apriori Algorithm.
7. Implementation of FP Tree.
8. Implementation of K-Means Clustering Algorithm.
9. Implementation of Bisecting K-Means Clustering Algorithm.
10. Implementation of Agglomerative Hierarchical Algorithm.

Recommended Readings:

TEXT BOOK

1. Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education, ISBN:81-317-1472-1

REFERENCE BOOK

1. Data Mining - Concepts and Techniques by Jiawei Han and Micheline Kamber, Elsevier, Second Edition, Original ISBN: 978-1-55860-901-3, Indian Reprint ISBN: 978-81-3120535-8

COMP7.3 IMAGE PROCESSING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.3	Image Processing	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To provide an introduction to basic concepts and methodologies in digital image processing.
2. To develop a foundation that can be used as the basis for further study and research in image processing.

Course Outcomes:

The student after undergoing this course will be able to:

1. Have a fundamental understanding of digital image processing techniques, including image smoothing, enhancement, restoration, segmentation and morphology.
2. Be able to implement working code for basic image processing algorithms.
3. Have the skill base necessary to further explore advanced topics of Digital Image Processing.

UNIT -1 (12 Hours)

Basic Concepts of Images: Analog Signals; Digital signals – Sampling and Quantization; Grey-Scale Images – Resolution and Grey Levels; Color Images Models – RGB, YIQ, YUV, HIS, CMYL; Image Storage Formats – BMP, RAW, JPEG, GIF; Video.

Basic Image Processing Tools: Correlation Operation and Convolution Operation; Fourier Transform – Continuous Fourier Transform (One and two dimensional), Discrete Fourier Transform, Properties of Discrete Fourier Transform, Fast Fourier Transform; Discrete Cosine Transform; Gabor Transform; Wavelet Transform – Continuous and Discrete Wavelet Transform.

UNIT - 2 (12 Hours)

Basic Image Processing Tools: Pixel Brightness (Grey level) Transformation – Image Enhancement Based on Histogram, Contrast Stretching; Concepts and Models of Image Processing; Image Smoothing using Spatial Domain Methods –

Neighbourhood Averaging, Threshold Averaging, Gaussian Filtering, Median Filtering, Weighted Median Filtering; Image Smoothing using Frequency Domain Methods – Ideal Low Pass Filtering, Trapezoidal Low Pass Filtering, Butterworth Low Pass Filtering; Image Enhancement – Gradient Image, Gradient Operators, High Pass Filtering; Image Restoration – Image Degradation Model, Image Restoration Based on the Degradation Model, Inverse Filtering, Wiener Filtering, Geometric Rectification; Processing Methods using Partial Differential Equations – Diffusion Based Models, TV Based Models, Discrete Formats of PDE Models

UNIT - 3 **(12 Hours)**

Image Segmentation: Thresholding –Semi Thresholding and Band Thresholding, Histogram Based Thresholding (Mode Method and adaptive (Local) Method), Optimal (Iterative) Thresholding; Edge Based Segmentation – Edge Image Thresholding, Edge Relaxation, Border Tracing, Hough Transform; Region Based Segmentation – Region Growing Method, Region Merging Method, Region Split and Merge Method.

UNIT - 4 **(12 Hours)**

Mathematical Morphology:Some Basic Concepts of Set Theory – Sets and elements, Relationships between two sets, Operations involving sets; Morphology for Binary Image – (Dilation Operation, Erosion Operation; Opening and Closing Operation, Hit or Miss Transformation), Applications of Binary Morphological Operations (Thinning and Thickening, skeleton Methods); Morphology for Grey Scale Morphological Operations – Basic Grey Scale Morphological Operations (Dilation Operation, Erosion Operation), Application of Grey-Scale Morphological Operations.

Recommended Readings:

1. Meiqing Wang, Choi-Honglai;AConcise Introduction to Image Processing Using C++;Chapman & Hall/CRC.
2. William K. Pratt ;Digital Image Processing.
3. Rafael C. Gonzales, Richar E Woods; Digital Image Processing.
4. Madhuri A. Joshi; Digital Image Processing, An algorithmic Approach.
5. SonkaHlavac Boyle; Digital Image Processing and Computer Vision.

List of Experiments in Image Processing:

1. Convert 24bit color Image to Grey Scale.
2. Program to calculate Fourier Transform of an Image
3. Program to calculate Discrete Cosine Transform of an Image.
4. Program to calculate the grey scale Histogram of an image.
5. Program to perform Median Filtering.
6. Program to obtain the Gradiant Image using Sobel-Operator

7. Image Restoration using the second and fourth order partial Differential Equation
8. Program for Optimal Thresholding Segmentation
9. Program for Border-Tracing
10. Program for Binary Erosion
11. Program to generate the Binary Skeleton of an Image.

COMP 7.4.1 VLSI DESIGN

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.4.1	VLSI Design	3	1	2	3	100	25	--	--	--	**125**

Course Objective:

1. To study various aspects of VLSI Design
2. To understand working of MOS Transistor under various bias.
3. To understand various semiconductor Technology processes.
4. To understand VHDL.
5. To understand verification Testing of MOS Circuits.

Course Outcomes:

1. To analyse the characteristics of MOS device under dc Bias.
2. To implement Digital Circuits using VHDL.
3. To verify ATPG Techniques on to digital Circuits.
4. To Design circuits for CMOS Transistor.

UNIT -1 (12 Hours)

Introduction, A Brief History, MOS Transistors, CMOS Logic – Inverter, NAND Gate, Combinational Logic, NOR Gate, Compound Gates, Pass Transistors and Transmission Gates, Tristates, Multiplexers, CMOS Fabrication and Layout.
MOS Transistor Theory – Ideal I-V Characteristics, C-V Characteristics – Simple MOS Capacitance Models. Nonideal I-V Effects – Velocity Saturation and Mobility Degradation, Channel Length Modulation, Body Effect, Junction Leakage, Tunneling.
DC Transfer Characteristics- Complementary CMOS Inverter DC Characteristics, Beta Ratio Effects, Noise Margin.

UNIT - 2 (12 Hours)

CMOS Processing Technology: CMOS Technologies – Background, Wafer Formation, Photolithography, Well and Channel Formation, Silicon Dioxide, Isolation, Gate Oxide, Gate and Source/Drain Formation, Contacts and Metallization, Passivation, Metrology.
Circuit Characterization and Performance Estimation: Delay Estimation, RC Delay Models – Elmore Delay Model. Power Dissipation – Static Dissipation, Dynamic Dissipation, Interconnect – Resistance, Capacitance, Design Margin – Supply Voltage, Temperature, Process Variation, Design Corners. Reliability – Reliability Terminology, Electromigration, Self-heating, Hot Carriers, Latchup. Scaling – Transistor Scaling Interconnect Scaling Properties

UNIT - 3 (12 Hours)

Combinational Circuit Design: Circuit Families – Static CMOS – Bubble Pushing, Compound Gates, Asymmetric Gates, Skewed Gates. Cascode Voltage Switch Logic, Pass- transistor Circuits – CMOS with Transmission Gates, Complementary Pass Transistor Logic(CPL), More Circuit Families – Differential Circuits (Differential Split-Level and Cascode Nonthreshold Logic), BiCMOS Circuits. Analog Circuits: MOS Small-signal Model, Current Mirrors, Differential Pairs, Simple CMOS Operational Amplifier. CMOS Physical Design Styles: Static CMOS Gate Layout, General CMOS Layout Guidelines. Layout Optimization for Performance.

UNIT - 4 **(12 Hours)**

Design Methodology and Tools: Design Methodology – Structured Design Techniques, Microprocessor/DSP, Programmable Logic – Programmable Logic Devices, Field Programmable Gate Arrays(FPGA). Testing and Verification: Logic Verification, Basic Digital Debugging Hints. Manufacturing Tests – Manufacturing Test Principles – Fault Models, Observability, Controllability. Fault Coverage, ATPG, Delay Fault Testing. Design For Testability – Built-in Self-Test(BIST). Basic Programming using VHDL.

Recommended Readings:

1. *Ayan Banerjee,David Harris,Neil H.E. West;* **CMOS VLSI Design: A Circuits and Systems Perspective, (Third Edition);** *Pearson Education , 2011*
2. Neil H.E. West and Kamran Eshraghian; Principles of CMOS VLSI Design; Prentice Hall of India, 1995
3. Douglas Pucknell and Kamran Eshraghian ; Basic VLSI Design; Prentice Hall of India, 1990

List of Experiments

1. Introduction to VHDL and VLSI Design
2. Use of NAND and NOR Gates for realizing other gates using VHDL.
3. Design of Half adder and Full adder using VHDL
4. 4: 1 MUX Design using VHDL
5. Solving of a SOP Expression using VHDL
6. Asynchronous D-Flip Flop using VHDL
7. Decade Counter using VHDL
8. Serial Shift Register using VHDL

COMP7.4.2 DATA COMPRESSION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.4.2	Data Compression	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

1. To provide basic introduction to concepts and methodologies and data compression.
2. To develop knowledge about the conceptual and practical aspect of data compression .
3. To develop a foundation that can be used for further research in data compression.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the basic concept and techniques of data compression.
2. Understand the fundamental principles of data compression techniques which includes Huffman's coding, Arithmetic coding, dictionary techniques etc.
3. Be able to implement basic data compression algorithms

UNIT -1 (12 Hours)

Information and Coding: Information and Entropy - Characteristics of entropy; Noiseless and memoryless coding – the kraft inequality, Fundamental theorem of discrete coding.

Shannon-Fano Coding: Shannon coding; Shannon-Fano Coding

Huffman Coding: Huffman coding with low memory requirements; Adaptive Huffman coding.

UNIT - 2 (12 Hours)

Arithmetic coding: Implementation of arithmetic coding – integer implementation.

Dictionary Techniques: The LZ77 technique – LZSS Technique; LZ78 Technique – LZW Technique.

Sampling and Quantization: Sampling; Quantization (scalar quantization) – Uniform quantization; Vector quantization – The K means algorithm.

UNIT - 3 (12 Hours)

Predictive Coding: Delta modulation – Adaptive delta modulation, delayed coding and delta modulation; Differential pulse code modulation – adaptive differential pulse code modulation, adaptive prediction.

Transform Coding: Define a transform; The Karhunen–Loeve Transform;The Hadamard Transform; Discrete Fourier Transform; Discrete Wavelet Transform.

UNIT - 4 (12 Hours)

Subband Coding:Filters; Down sampling and Up sampling

Compression of still images:**JPEG**: The Base line system – Source Image format, DCT based coding; Hierarchical mode of operation; Sequential lossless mode of operation.

Recommended Readings:

1. Adam Drozdek;Elements of Data Compression;Thomson Brooks/Cole;
2. Khalid Sayood; Introduction to Data Compression; Elseivier;Second Edition.
3. Ida MengyiPu, Butterworth-Heinemann; Fundamental Data Compression.

List of Experiments in Data Compression:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Case study on information and entropy.
2. Case study noiseless and memoryless coding
3. Implementation of Shannon coding.
4. Implementation of Shannon – Fano coding.
5. Implementation of Huffman coding.
6. Implementation of LZ77 Technique.
7. Implementation of LZSS Technique.
8. Implementation of LZW Technique.

COMP7.4.3 FUZZY LOGIC AND NEURAL NETWORKS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.4.3	Fuzzy Logic and Neural Networks	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

1. To provide basic introduction to concepts and methodologies of Fuzzy Logic and Neural Networks.
2. To develop knowledge about the conceptual and practical aspect of Neural Networks and Fuzzy Logic.
3. To develop a foundation that can be used for further research in Fuzzy Logic and Neural Networks.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the basic concept and techniques of Neural Networks.
2. Differentiate between crisp set and fuzzy set.
3. Describe the learning rules used in Neural Networks.
4. Apply the concepts of Fuzzy Logic and Neural networks in practical applications.

UNIT -1 (12 Hours)

History of Neural Networks. Structure and function of a single neuron. Neural Net Architecture. Neural Learning. Common usage of neural networks in classification, clustering, vector quantization. pattern association, function approximation and forecasting. Evaluation of networks. Implementation of neural networks.

Perceptrons. Linear Separability Perceptron Training Algorithm, Guarantee of Success, Pocket algorithm, Adaline. Multilayer networks, Multilevel discrimination, Architecture, objectives and working of Backpropagation algorithm. Setting the parameter values of Backpropagation algorithm. Accelerating learning process and applications of Backpropagation algorithm.

UNIT - 2 (12 Hours)

Prediction tasks using Recurrent Networks and feedforward networks, Radial basis functions. Polynomial networks. Unsupervised learning. Hamming networks, simple competitive learning. counter-propagation network, adaptive resonance theory, Self

organizing maps. Non-iterative procedures for association, Discrete Hopfield Network, Brain-State_in_a_box Network, Boltzmann Machine, Bi-directional Associate memory.

UNIT - 3 **(12 Hours)**

History and Motivation for Fuzzy Logic. Classical sets, Fuzzy sets, Operations of Fuzzy sets, Properties of Fuzzy sets, A Geometric interpretation of Fuzzy sets, possibility theory. (03 hrs) Fuzzy relations, composition of Fuzzy relations, Fuzzy graphs and numbers, Functions with Fuzzy arguments, arithmetic operations on Fuzzy numbers. Basics of Fuzzy rules, Fuzzy mapping rules, Fuzzy implication rules, Fuzzy rule based models for function approximation, Theoretical foundation of fuzzy mapping rules, Types of fuzzy rule based models: Mamdani model, TSK model, and standard additive model.

UNIT - 4 **(12 Hours)**

Propositional logic and first order predicate calculus. Fuzzy logic: Fuzzy implication, approximate reasoning, Criteria of Fuzzy implications, Three families of Fuzzy implications. Possibility versus Probability, Probability of a Fuzzy event. Probabilistic interpretation of Fuzzy sets. Fuzzy Logic in Expert Systems. intelligent agents and Mobile robot navigation,. Fuzzy logic in database systems, Fuzzy relational data models and operations, Fuzzy object oriented database. Fuzzy information Retrieval and Web search.

Recommended Readings:

1. Kishan Mehrotra, Chilukuri Mohan, and Sanjay Ranka; Elements of Artificial Neural Networks by Penram International Publishing (India)
2. John Yen and Reza Langari,Fuzzy Logic, Intelligence, Control and Information; Pearson Education
3. Neural Networks and Fuzzy Systems: A dynamical Systems Approach toi Machine Intelligence, by Bart Kosko, PHI
4. Neural Networks: A comprehensive Foundation, - By Simon Haykin, Pearson Education
5. Introduction to Artificial Neural Networks, - By Jacek M. Zurada, Jaico PublishingHouse
6. Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications by S. Rajasekaran, G.A. Vijayalakshmi Pai, PHI

List of Experiments in Data Compression:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Implementation of basic logic gates using Neural networks

2. Designing a Neural Network to simulate any Boolean function.
3. Implementation of Perceptron Learning Algorithm
4. Implementation of Back propagation Algorithm
5. Implementation of Hebbian rule
6. Implementation of fuzzy set operations.
7. Implementation of fuzzy inference rules.
8. Implementation of an application using Neuro Fuzzy techniques.

COMP7.4.4 WEB TECHNOLOGY

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP7.4.4	Web Technology	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

The purpose of this course is to provide students with a basic understanding of web programming. It will focus on the client-side as well as server-side implementation of web applications.

Course Outcomes:

1. Understand the basics of the internet and related underlying protocols involved in web development.
2. Discuss the insights of internet programming and implement complete application over the web.
3. Demonstrate the important HTML tags for designing static pages and separate design from content using Cascading Style sheet.
4. To design data using XML,perform validations and display in HTML format.
5. Utilize the concepts of JavaScript and Angular JS in developing dynamic web applications
6. Ability to develop server-side applications using PHP and JSP.

UNIT -1 (12 Hours)

Introduction to Web Technologies

History of the Web, OSI Reference Model, Understanding Web System Architecture, understanding 3-Tier Web Architecture, Layers in the TCP/IP Model, Web, Overview of HTTP,Using Cookies to Remember User Information, Exploring Web Technologies, Introduction to Web Services, About IIS,Services Supported by IIS 7, Installation of IIS 7, Administer Web Server Remotely, Creating Web Sites.

HTML and JAVASCRIPT Programming

HTML, Introducing HTML Document structure, Creating Headings on a web page, Working with Links, creating a paragraph, working with images, working with tables, working with frames, Introduction to Forms and HTML Controls, Introducing JavaScript.

UNIT - 2 (12 Hours)

Cascading Style Sheets

Coding CSS, Properties of Tags, Property Values, Other Style Properties, In-Line Style Properties, Embedded Style Sheets, Grouping, Inheritance, Class as Selector, ID as Selector, Contexual Selectors, Pseudo Classes and Pseudo-elements, Positioning, Backgrounds, Element Dimensions

Extensible Mark-Up Language (XML)

Introduction, HTML vs XML, Syntax of XML Document, XML Attributes, XML Validation, XML DTD, The Building Blocks of XML Documents, DTD Elements, DTD Attributes, DTD Entities, DTD Validations, XSL, XSL Transformation, XSL Namespaces, XML Schema

UNIT - 3 (12 Hours)

Angular JS

Introducing AngularJS

Introducing AngularJS, What Is MVC (Model-View-Controller), AngularJS Benefits, The AngularJS Philosophy, Starting Out with AngularJS , A Basic AngularJS Application ,Angular JS Hello World

Basic AngularJS Directives and Controllers

AngularJS Modules,Creating Our First Controller,Working with and Displaying Arrays,More Directives , Working with ng-repeat, ng-repeat Over an Object ,Helper Variables in ng-repeat , Track by ID , ng-repeat Across Multiple HTML Elements

Forms, Inputs, and Services
Working with ng-model,Working with Forms, Leverage Data-Binding and Models ,Form Validation and States ,Error Handling with Forms ,Displaying Error Messages ,Styling Forms and States ,Nested Forms with ng-form Other Form Controls: Textareas ,Checkboxes ,Radio Buttons ,Combo Boxes/Drop-Downs

Java Server Pages (JSP)
Introduction,Advantages of JSP, Developing first JSP, Components of JSP, Reading Request Information, Retrieving the Data posted, JSP Sessions, Cookies, Disabling Sessions.

UNIT - 4 (12 Hours)

Introducing PHP
Versions of PHP, Features of PHP, Advantages of PHP over other scripting languages, creating a PHP Script, running a PHP Script, Handling Errors in a PHP Script

Working with variables and constants
Using variables, using constants, exploring datatypes in PHP, Exploring operators in PHP.

Controlling Program Flow
Conditional Statements, Looping Statement, Break,Continue and Exit Statement

Working with Functions,Arrays,Files and Directories
User-Defined Functions in PHP, Built-in functions in PHP, Recursive, Variable and call-back Functions, Introducing Arrays, Types of Arrays, Traversing Arrays using Loops and Array Iterators, Built-in Array Functions, Working with Files, Working with Directories

Working with Forms and Databases
Introduction to Web Forms, working with the <form> Tag and Form Elements, processing a Web Form, validating a Form, Using PHP and MySQL

Exploring cookies and sessions
Working with cookies, Working with sessions

Recommended Readings:

1. Web Technology: A Developer's Perspective by N. P. Gopalan and J. Akhilandeswari, PHI ,Second Edition,ISBN:978-81-203-5006-9
2. Web Technologies Black Book by Kogent Learning Solutions, dreamtechpress,ISBN: 9788177228496
3. AngularJS: Up and Running By ShyamSeshadri and Brad Green ,First Edition, Shroff Publishers and Distributors,ISBN: 978-1-491-90194-6

List of Experiments:
(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Selection of a Project which will incorporate HTML5, CSS3, XML, XSLT, DOM , DTD, Javascript, Angular js , PHP, Mysql

2. Implementation of HTML5 and CSS3.
3. Implementation of DTD and XML
4. Implementation of DOM and XSLT.
5. Implementation of Javascript.
6. Implementation of Angular JS/Ajax.
7. Implementation of MySql.
8. Implementation of PHP. (Creation and connection)
9. Implementation of PHP. (Update and Search)
10. Implementation of PHP. (View and Delete) (Along with Final DEMO)

COMP 7.4.5 Cloud Computing

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.4.5	Cloud Computing	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

1. Analyze the components of cloud computing showing how business agility in an organization can be created
2. Compare and contrast the economic benefits delivered by various cloud models based on application requirements, economic constraints and business requirements.
3. Critically analyze case studies to derive the best practice model to apply whendeveloping and deploying cloud based applications.

Course Outcomes:

The student after undergoing this course will be able to:

1. Compare the advantages and disadvantages of various cloud computing platforms.
2. Analyze the performance, scalability, and availability of the underlying cloud technologies and software.
3. Solve a real-world problem using cloud computing through group collaboration.

UNIT - 1 (12 Hours)

Cloud Computing Fundamental: Cloud Computing definition, private, public and hybrid cloud.

Cloud types;IaaS, PaaS, SaaS. Benefits and challenges of cloud computing, public vs private clouds, role of virtualization in enabling the cloud;

Business Agility: Benefits and challenges to Cloud architecture. Application availability, performance, security and disaster recovery; next generation Cloud Applications.

UNIT - 2 (10 Hours)

Cloud Applications: Technologies and the processes required when deploying web services; Deploying a web service from inside and outside a cloud architecture, advantages and disadvantages.

Software as a Service (Saas)-Understanding the Multitenant Nature of SaaS Solutions, Understanding SOA.

Platform as a Service (PaaS)- IT Evolution Leading to the Cloud, Benefits of Paas Solutions, Disadvantages ofPaas Solutions.

Infrastructure as a Service (Iaas)-Understanding IaaS, Improving Performancethrough Load Balancing, System and Storage Redundancy, Utilizing Cloud-Based NAS Devices,

UNIT - 3 (13 Hours)

Cloud Services Management: Reliability, availability and security of services deployed from the cloud. Performance and scalability of services, tools and technologies used to manage cloud services deployment.

Cloud Economics: Cloud Computing infrastructures available for implementing cloud based services.

Economics of choosing a Cloud platform for an organization, based on application requirements, economic constraints and business needs (e.g Amazon, Microsoft and Google, Salesforce.com, Ubuntu and Redhat)

UNIT - 4 (12 Hours)

Application Development: Service creation environments to develop cloud based applications. Development environments for service development; Amazon, Azure, Google App.

Best Practice Cloud IT Model : Analysis of Case Studies when deciding to adopt cloud computing architecture. How to decide if the cloud is right for your requirements. Cloud based service, applications and development platform deployment so as to improve the total cost of ownership (TCO)

Recommended Readings:

1. GautamShroff, Enterprise Cloud Computing Technology Architecture Applications
   [ISBN: 978-0521137355]
2. Toby Velte, Anthony Velte, Robert Elsenpeter, Cloud Computing, A Practical Approach [ISBN: 0071626948]
3. Cloud Computing: Implementation, Management and Security, John W. Rittinouse, James F Ransome. CRC Press, rp2012.
4. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. George Reese, O'RedI SPD, rp2Oll.
5. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, SubraKtriaraswamy, ShahedLatif, O'Redç SPD, rp2Oll.

List of Experiments in Cloud Computing:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1.Software study for cloud computing Software.
2. Service Development & Usage over Cloud.
3. Managing Cloud Computing Resources
4.Using existing cloud characteristics & Service models
5.Performance evaluation of service over cloud.
6. Installation and Configuration of Cloud.
7. Create an application using Cloud.
8. Case Study

COMP 7.5.1 Entrepreneurship Development

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.5.1	Entrepreneurship Development	3	1	--	3	100	25	--	--	25	**150**

Course Objectives:

1. Study of this subject provides an understanding of the scope of an entrepreneur.
2. Analyze the key areas of development, financial assistance by the institutions, methods of taxation and tax benefits, etc.
3. Critically analyze case studies to derive the best practice model to apply when developing and deploying real life applications.

Course Outcomes:

The student after undergoing this course will be able to:

1. Identify the concept of entrepreneurship and its functions.
2. Learn the competencies of an Entrepreneur.
3. Able to differentiate Entrepreneur, Business man and employee.

UNIT - 1 (12 Hours)

ENTREPRENEUR AND ENTREPRENEURSHIP

Entrepreneur, Concept , definition, features , significance of Entrepreneurship,function,process, quality of Entrepreneur , mindset of employees vs entrepreneur, Entrepreneurship – Characteristicsentrepreneur vs Entrepreneurship,Intrapreneur –Myths ofEntrepreneurship, Role of entrepreneurship in Economicdevelopment, challenges of Entrepreneurship, Social responsibility ofEntrepreneurship

UNIT - 2 (14 Hours)

ENTREPRENEURIAL COMPETENCIES

Introduction – competencies of entrepreneurs-(1) Decision Making (2)Problem Solving (3) Risk Taking (4) Leadership(5) Communication(5)Dealing with customers, Entrepreneurial Values and attitudemotivation-Need Hierarchy Theory of Motivation - David MC llandNeed Theory of Motivation- Life Skills - Managing Self and Others,Positive Attitude Creativity, Team Building and motivation.

UNIT - 3 (13 Hours)

ENTREPRENEURSHIP JOURNEY

Self Assessment of Qualities, Skills, Resources and DreamsIdentify your personality type before starting a business venture –Trailblazers, Go-getters,Managers, Motivators, Authoritarians, Collaborators, Diplomats -Business Ideas- Generating Ideas- Ways to Generate Ideas- EnvironmentScanning- Creativity and Creative Problem Solving –Brainstorming Focus Groups- Feasibility Study- Types of feasibility study- MarketFeasibility- Technical Feasibility- Financial Feasibility- OrganizationalFeasibility- Features of Feasibility Study- Role of society and family inthe growth of an entrepreneur- Rural Entrepreneurship- Agripreneurship– Social Entrepreneurship – Women Entrepreneurship StudentEntrepreneurship.

UNIT - 4 (12 Hours)

ENTREPRENEUR THE INNOVATOR

Innovations leading entrepreneurial ventures - the role of technologyfor Entrepreneurship development- social media in creating new formsof business organizations - networks and co-operative clusters -Concept of Risk Taking- Types of Risk Taking: EntrepreneurshipDevelopment Club - Entrepreneur Support Scheme(ESS). – Businessincubation.

RECOMMENDED READINGS

1. Forbat, John, “Entrepreneurship” New Age International.
2. Havinal, Veerbhadrappa, “Management and Entrepreneurship” New Age International
3. Joseph, L. Massod, “Essential of Management", Prentice Hall of India.

4.Tendon ,C: Environment and Entrepreneur; Cliugh Publications, Allahabad.

5. SinerA David: EntrepreneuralMegabuks; John Wiley and Sons, New York.

6.Srivastava S. B: A Practical Guide to Industrial Entrepreneurs; Sultan Chand and Sons,New Delhi.

COMP 7.5.2 GEOGRAPHICAL INFORMATION SYSTEM

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
						Marks					
		L	T	P	ThDuration (Hrs)	Th	S	TW	P	O	Total
COMP 7.5.2	Geographical Information System	3	1	--	3	100	25	--	--	25	**150**

Course Objectives:

1. To understand the basic concepts of Geographical Information Systems.
2. To Learn the procedures employed in Geographical Information Systems.
3. To study the applications of the Geographical Information Systems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the GIS data processing.
2. Describe the concepts of data Modeling.
3. Explain the GIS Design issues w.r.t. an application.
4. Design a GIS system for the given environment.

UNIT – 1 (12 Hours)

Introduction to GIS: Definition, Evolution, Component of GIS. Functions and Characteristics of GIS applications. Contributing and Allied Disciplines. Map scale, Classes of map, Mapping process, Coordinate systems – plane and geographic, Map projection, Spatial framework for mapping locations – georeferencing, Topographic mapping, Attribute data for Thematic mapping. Digital Representation of Geographic data: Object representation and data analysis Relationship between Data representation and Data analysis.

UNIT - 2 (12 Hours)

Data Quality and Standards: Concepts and definition of data quality, Component of geographic data, Data quality assessment, Spatial data error management, Geographic data standards, Geographic data standards and GIS development. Raster based GIS data processing: Acquiring and Handling raster geographic data, Raster based GIS data analysis, output functions of raster data processing, Cartographic modeling. Vector based GIS data processing: Characteristics of vector based GIS data processing, Vector data input functions, Non topological GIS analysis , functions, Feature based topological functions, Layer based topological functions, Vector based output functions, Application programming.

UNIT - 3 (12 Hours)

Visualization of Geographic Information and Generation of Information Products: Cartography in GIS context, Human computer interaction and GIS, Visualization of geographic information, Principles of Cartographic design in GIS, Generation of information product. Data Modeling: Digital Terrain Modeling, Approaches to digital terrain data modeling, Acquisition of digital terrain data, Data processing, Analysis and visualization, Applications of digital terrain models.

Spatial modeling: Descriptive statistics, Spatial autocorrelation, Quadrant counts and Nearest-Neighbor analysis, Trend surface analysis, Gravity models, Network analysis, GIS modeling..

UNIT - 4 (12 Hours)

GIS Modeling: Binary Models, Index Models, Regression Models, Process Models.

GIS Project Design And management: Software engineering as applied to GIS, GIS project planning, System analysis and study of user requirement, Geographic database design methodology, GIS application software design methodology, System implementation and technology rollout, system maintenance and support.
GIS issues And Future of GIS: Issues of implementation, Trend of GIS development, GIS applications and GIS users.

Recommended Readings:

1. C.P. La, Albert K.W. Yeung; Concepts and Techniques of Geographic Information Systems; PHI, ISBN:81-203-2230-4
2. Kang-Tsung Chang; Introduction to Geographic Information Systems; TMH, ISBN:0-07-049552-1
3. Lan Heywood, Sarah Cornelius, Steve Carver; An Introduction to Geographical Information System;, Pearson Education, ISBN:81-7808-541-0

COMP 7.5.3 DESIGN PATTERNS AND FRAMEWORKS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.5.3	Design Patterns and Frameworks	3	1	--	3	100	25	--	--	25	**150**

Course Objectives:

1. To provide an introduction to basic concepts and methodologies in design patterns and frameworks.
2. To develop a foundation that can be used as the basis for further study and research in patterns and frameworks.

Course Outcomes:

The student after undergoing this course will be able to:

1. Have a fundamental understanding of design patterns and frameworks techniques, including Interface Patterns, Responsibility Patterns, Construction Patterns, and Operations Patterns.
2. Implement working code for basic design patterns.
3. Have the skill base necessary to further explore advanced topics of Design patterns and frameworks

UNIT -1 (12 Hours)

Interface Patterns: Introducing Interfaces – Summary, Beyond Ordinary Interfaces; Adapter – Adapting to an Interface, Class and Object Adapters, Adapting Data in .NET, Summary; Façade – An Ordinary Façade, Refactoring to FAÇADE, Facades, Utilities, Demos, Summary; Composite – An Ordinary Composite, Recursive Behavior in Composites, Composites, Tress and Cycles, Composites with Cycles, Consequences of Cycles, Summary; Bridge – An Ordinary Abstraction, From Abstraction to Bridge, Drivers as Bridges, Database Drivers, Summary.

UNIT - 2 (12 Hours)

Responsibility Patterns: Beyond Ordinary Responsibility; Singleton – Singleton Mechanics, Singleton and Threads, Recognizing SINGLETON, Summary; Observer – C# Support for Observer, Delegate Mechanics, A Classic Example(OBSERVER in GULs), Model/View/Controller, Layering, Summary; Mediator – A Classic Example (GUI Mediators), Relational Integrity Mediators, Summary; Proxy – A Simple Proxy, A Data Proxy, Remote Proxies, Summary; Chain of Responsibilities – An Ordinary CHAIN OF RESPONSIBILITY, Refactoring to CHAIN OF RESPONSIBILITY, Anchoring a Chain.

Flow through Pipes: Loss of head in pipes, major, minor losses,Darcy's weisbach equation, Hydraulic gradient andtotal energy line, Flow through siphon , Equivalent pipe -series &parallel pipes, Flow through nozzle, Water hammer in pipes.

UNIT - 3 (12 Hours)

Construction Patterns: Introducing Construction – A few construction challenges, summary, Beyound Ordinary Construction; Builder – An Ordinary Builder, Building under Constraints, A Forgiving Builder, Summary; Factory Method – A classic Example (Enumerators), Recognizing FACTORY METHOD, Taking Control of which class to Instantiate, FACTORY METHOD in parallel hierarchies, Summary; Abstract Factory – A Classic Example(GUI Kits), Abstract Factory and Factory Method, Namespaces and Abstract Factories, Summary; Prototype –Prototypes as Factories, Memento Durability, Persisting Mementos across sessions, Summary;

UNIT - 4 (12 Hours)

Operations Patterns: Introducing Operations – Operations and Methods, Signatures, Delegates, Exceptions, Algorithms and Polymorphism, Summary, Beyond Ordinary Operations; Template Method – A Classic Example (Sorting), Completing an Algorithm, TEMPLATE METHOD Hooks, Refactoring to TEMPLATE METHOD, Summary; State – Modeling States, Refactoring to STATE, Making States Constant, Summary; Strategy – Modeling Strategies, Refactoring to STRATEGY, A Classic example (Menu Commands), Using COMMAND to supply a service, COMMAND Hooks, COMMAND in relation to other patterns, Summary; Interpreter – An INTERPRETER Example, Interpreters, Languages and Parsers, Summary;

Recommended Readings:

1. Steven John Metsker; Design Patterns in C#; Addison-Wesley Professional.
2. Eric Gamma, Richard Helm, Ralph Johnson, John Vlissides; Design Patterns, Elements of Reusable Object Oriented Software; Pearson Education.

COMP 7.5.4 PROJECT MANAGEMENT & QUALITY ASSURANCE

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.5.4	Project Management and Quality Assurance	3	1	--	3	100	25	-	-	25	**150**

Course Objectives:

1. To help students to identify key areas of concern over Project Life Cycle (PLC) and use of project management principles across all the phases of PLC.

2. To make students understand the importance and necessity of project plan and how it is helpful to the project manager in monitoring and controlling the various aspects of the project such as schedule, budget, etc.

3. To make students understand the importance of team and how to work as a team member, share best project management practices.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
2. Compare and differentiate organization structures and project structures.
3. Implement a project to manage project schedule, expenses and resources with the application of suitable protect management tools.

UNIT - 1 (10 Hours)

Introduction and Software Project Planning

Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, Project Management Cycle, SPM Objectives, Management Spectrum, SPM

Framework, Software Project Planning, Planning Objectives, Project Plan, Types of project plan, Structure of a Software Project Management Plan, Software project estimation, Estimation methods, Estimation models, Decision process.

UNIT - 2 (12Hours)

Project Organization, Scheduling, Monitoring and Control

Project Elements, Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Life Cycle and Product Life Cycle, Ways to Organize Personnel, Project schedule, Scheduling Objectives, Building the project schedule, Scheduling terminology and techniques,Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts.

Dimensions of Project Monitoring & Control.

Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV),

Cost Performance Index (CPI), Schedule Performance Index (SPI), Interpretation of Earned

Value Indicators, Error Tracking, Software Reviews, Types of Review: Inspections, Deskchecks, Walkthroughs, Code Reviews, Pair Programming.

UNIT - 3 (10 Hours)

Software Quality Assurance and Testing

Testing Objectives, Testing Principles, Test Plans, Test Cases, Types of Testing, Levels of

Testing, Test Strategies, Program Correctness, Program Verification & validation, Testing

Automation & Testing Tools, Concept of Software Quality, Software Quality Attributes,

Software Quality Metrics and Indicators, The SEI Capability Maturity Model CMM), SQA

Activities, Formal SQA Approaches: Proof of correctness, Statistical quality assurance,

Cleanroom process.

UNIT - 4 (10Hours)

Project Management and Project Management Tools

Software Configuration Management: Software Configuration Items and tasks, Baselines, Plan for Change, Change Control, Change Requests Management, Version Control, Risk

Management: Risks and risk types, Risk Breakdown Structure (RBS), Risk Management

Process: Risk identification, Risk analysis, Risk planning, Risk monitoring, Cost Benefit

Analysis, Software Project Management Tools: CASE Tools, Planning and Scheduling Tools,

MS-Project.

Recommended Readings:

1. M. Cotterell; Software Project Management; Tata McGraw-Hill Publication.
2. Royce; Software Project Management; Pearson Education
3. Kieron Conway; Software Project Managemen; Dreamtech Press

4. S. A. Kelkar; Software Project Management; PHI Publication.

COMP 7.5.5 BIG DATA ANALYTICS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.5.5	Big Data Analytics	3	1	--	3	100	25	-	-	25	**150**

Course Objectives:

1. To help students to identify key areas of concern over Big Data.
2. To make students understand the importance and necessity of data analysis tools.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe and determine the purpose and importance of Data Analytics tools.

UNIT - 1 (10 Hours)

Introduction to BigData Platform – Traits of Big data -Challenges of Conventional Systems -Web Data – Evolution Of Analytic Scalability - Analytic Processes and Tools - Analysis vs. Reporting - Modern Data Analytic Tools - Statistical Concepts: Sampling Distributions ReSampling- Statistical Inference - Prediction Error.
Regression Modeling - Multivariate Analysis - Bayesian Modeling - Inference and Bayesian Networks - Support Vector and Kernel Methods - Analysis of Time Series: Linear Systems Analysis - Nonlinear Dynamics - Rule Induction.

UNIT - 2 (10 Hours)

Introduction To Streams Concepts – Stream Data Model and Architecture - Stream Computing -Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream –**Estimating Moments** – Counting Oneness in a Window – Decaying Window - Real time Analytics Platform(RTAP) Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.

UNIT - 3 (10 Hours)

Mining Frequent Itemsets - Market Based Model – Apriori Algorithm – Handling Large Data Sets in Main Memory – Limited Pass Algorithm – Counting Frequent Itemsets in a

Stream –Clustering Techniques – Hierarchical – K-Means – Clustering High Dimensional Data –CLIQUE And PROCLUS – Frequent Pattern based Clustering Methods – Clustering in NonEuclidean Space – Clustering for Streams and Parallelism.

UNIT - 4 **(10 Hours)**

NoSQL Databases:MongoDB,CouchDB
MapReduce – Hadoop, Hive, Pig, MapR
Storage: S3, Hadoop Distributed File System
Servers: EC2, Google App Engine, Elastic Bean Stalk, Heroku
Processing: R, Yahoo! Pipes, Mechanical Turk
Visualizations - Visual Data Analysis Techniques - Interaction Techniques; Systems and Analytics Applications - Analytics using Statistical packages-Approaches to modeling in Analytics – correlation, regression, decision trees, classification, association Intelligence from unstructured information-Text analytics-Understanding of emerging trends and technologies-Industry challenges and application of Analytics

TEXT BOOKS:

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge
University Press, 2012.
3. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley & sons, 2012.
4. Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons, 2007
5. Pete Warden, "Big Data Glossary", O'Reilly, 2011.
6. Jiawei Han, Micheline Kamber "Data Mining Concepts and Techniques", Second Edition,
Elsevier, Reprinted 2008.

COMP 8.1 1DISTRIBUTED OPERATING SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.1	Distributed Operating Systems	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To introduce the basic concepts upon which distributed systems at large and distributed operating systems in particular rely.

2. To understand the design issues, design problems, solutions and performance issues.
3. To present the principles underlying the functioning of distributed systems
4. To provide experience in the implementation of typical algorithms used in distributed systems

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are.
2. List the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions.
3. Recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems.

UNIT -1 (11 Hours)

Introduction to distributed operating systems: What is a distributed system? Goals, Hardware Concepts ,Software Concepts, Design Issues

Communication in distributed systems: Layered Protocols, Asynchronous Transfer Mode Networks, The Client-Server Model, Remote Procedure Call, Group Communication

UNIT - 2 (13 Hours)

Synchronization in Distributed Systems: Clock Synconization, Mutual Exclusion, Election Algorithms, Atomic Transactions, Deadlocks in Distributed Systems

Processes and Processors in Distributed Systems: Threads, System Models

UNIT - 3 (12 Hours)

Processes and Processors in Distributed Systems: Processor Allocation, Scheduling in Distributed Systems, Fault Tolerance

Distributed File Systems: Distributed File System Design, Distributed File System Implementation

UNIT - 4 (12 Hours)

Case Study of Distributed Systems

Case study 1: AMOEBA : Introductionto Amoeba, Objects and capabilities, Process management, Memory management, Communication, The Amoeba Servers

Case study 2: Distributed Computing Environment: Introduction, Threads, RPC, Time Service, Directory Service, Security Service

Recommended Readings:

1. A.S. Tanenbaum; Distributed Operating Systems; Pearson Education; ISBN: 978-81-7758-179-9
2. G. Coulouris, J. Dollimore and T. King Berg;Distributed Systems: Concepts and Design by; Addison Wesley; ISBN:81-7808-462-7
3. M. Singhal and N. G. Shivaratri;Advanced Concepts in Operating Systems; TMH; ISBN:0-07-047268-8
4. A. S. Tanenbaum, Maarten Van Steen; Distributed Systems: Principles and Paradigms; PHI; ISBN: 978-81-203-3498-4
5. William Buchanan; Distributed Systems and Networks; TMH, ISBN: 0-07-058753-1

List of Experiments in Distributed Operating Systems:

1. Socket programming in TCP
2. Socket programming in UDP
3. Remote Method Invocation
4. Clock Synchronization
5. Threads
6. Component Object Model
7. CORBA
8. Distributed Deadlocks
9. Distributed Databases

COMP 8.2 Network Security

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.2	Network Security	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

1. Understand Network Devices functions and configurations hub, switch, tap and routers)

2. Understand Network Security Devices (IDS, Firewall..etc)
3. Understand and analyze network services.
4. Understand and analyze network traffic and protocol
5. Understand network security concepts
6. Understand network intrusions and how to identify them such as
    a. Computer Viruses
    b. Network worms
    c. Botnets

Course Outcomes:

The student after undergoing this course will be able to:

1. Identify infrastructure components and the roles they serve, and design infrastructure including devices, topologies, protocols, systems software, management and security.
2. Analyze performance of enterprise network systems
3. Develop solutions for networking and security problems

UNIT – 1 (12 Hours)

Systems Vulnerability Scanning

Overview of vulnerability scanning, Open Port / Service Identification, Banner / Version Check, Traffic Probe, Vulnerability Probe, Vulnerability Examples, OpenVAS, Metasploit. Networks Vulnerability Scanning –Netcat, Socat, understanding Port and Services tools –Datapipe, Fpipe, WinRelay, Network Reconnaissance – Nmap, THC-Amap and System tools. Network Sniffers and Injection tools – Tcpdump and Windump, Wireshark, Ettercap, Hping Kismet

UNIT –2 (14Hours)

Network Defense tools

Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless VsStateful Firewalls, Network Address Translation (NAT) and Port Forwarding, the basic of Virtual Private Networks, Linux Firewall, Windows Firewall, Snort: Introduction Detection System

Web Application Tools

Scanning for web vulnerabilities tools: Nikto, W3af, HTTP utilities – Curl, OpenSSL and Stunnel, Application Inspection tools – Zed Attack Proxy, Sqlmap. DVWA,

Webgoat, Password Cracking and Brute-Force Tools – John the Ripper, L0htcrack, Pwdump, HTC-Hydra

UNIT – 3 (13 Hours)

Introduction to Cyber Crime and law Cyber Crimes, Types of Cybercrime, Hacking, Attack vectors, Cyberspace and Criminal Behavior, Clarification of Terms, Traditional Problems Associated with Computer Crime, Introduction to Incident Response, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world, A Brief History of the Internet, Recognizing and Defining Computer Crime, Contemporary Crimes, Computers as Targets, Contaminants and Destruction of Data, Indian IT ACT 2000.

UNIT - 4 (13 Hours)

Introduction to Cyber Crime Investigation Firewalls and Packet Filters, password Cracking, Keyloggers and Spyware, Virus and Warms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attack on wireless Networks

Recommended Readings:

1. Anti-Hacker Tool Kit (Indian Edition) by Mike Shema, Publication McGraw Hill.
2. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Nina Godbole and SunitBelpure, Publication Wiley

List of Experiments in Cyber Security

(At least 8 experiments should be conducted.. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. TCP scanning using NMAP
2. Port scanning using NMAP
3. TCP / UDP connectivity using Netcat
4. Network vulnerability using OpenVAS
5. Web application testing using DVWA
6. Manual SQL injection using DVWA
7. XSS using DVWA
8. Automated SQL injection with SqlMap

COMP 8.3.1 OPERATION RESEARCH

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.3.1	Operation Research	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To understand the computer oriented approach in problem solving with the important methods of Operations Researchin solving realistic problems.
2. Learn the types of problems that can be solved by a particular method and to model the problem for solution.
3. To study the models involving optimum decision making.

Course Outcomes:

The student after undergoing this course will be able to:

1. Apply the following techniques in solving reallife problems: Linear Programming, Integer Programming, Dynamic Programming, Branch and Bound Techniques, Sequencing problems, Queuing theory, Network Models.
2. For a given problem will be able to formulate, construct a model, develop a method to solve the model and implement the solution for the problem.

UNIT - 1 (12 Hours)

Introduction: The Beginning and Progress of Operations Research, Classification of problems in Operations Research, Mathematical Modelling in Operations Research.

Linear Programming: Introduction, Formulation of Linear Programming models, Graphic Solution of Linear Programming models, Maximization and Minimization of functions with constraints, Simplex method, Transportation problem, Assignment problem.

UNIT - 2 (14 Hours)

Integer Programming: Introduction, Dual Simplex Method, Implicit Enumeration, Cutting plane technique.

Branch and Bound Technique: Introduction, Branch and Bound Algorithm for Assignment problem, Branch and Bound Algorithm for Travelling Salesman problem , Branch and Bound Algorithm for Integer Programming.

UNIT - 3 (13 Hours)

Dynamic Programming: Introduction,Investment problem, Stage-coach problem, Production Scheduling, Equipment Replacement.

Sequencing problems:Introduction ,Two-Machine sequencing problem, N-job,Three-Machine Sequencing Problem.

UNIT - 4 (13 Hours)

CPM and PERT: Network Representation, Critical Path(CPM) Computations, Time estimates for activities, Critical Path, Probability of completing events on Schedule.
Queuing Theory: Introduction, Notations and Assumptions, Queuing Models with Poisson Input – Exponential service, Queuing Models with Poisson Input –Arbitrary service time.

Recommended Readings:

1. Billey E. Gillett; Introduction to Operations Research: A Computer Oriented Algorithm Approach ; Tata McGraw Hill.
2. H.A. Taha;Operations Research An Introduction;8th Edition ; Pearson Education; 2009
3. Fredericks ,Hiller and Liberman ;Operations Research ; Tata McGraw Hill.
4. J. K. Sharma ;Operations Research – Theory and Applications ; MacMillan India Ltd.
5. P.K.Gupta, D.S Hira; Operations Research; S.Chand 2007.

List of Experiments in Operation Research:
(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Modeling with Linear Programming
2. Simplex Algorithm
3. Artificial Starting Solution
4. Dual Simplex Algorithm
5. Transportation Algorithm
6. Assignment Algorithm
7. Investment Problem using Dynamic Programming
8. Stage Coach Problem
9. N Job, 2-machine Sequencing Problem

COMP 8.3.2 MULTIMEDIA SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.3.2	Multimedia Systems	3	1	2	3 hrs	100	25	-	-	25	**150**

Course Objectives:

1. The aim of this course is to help students develop an understanding of the fundamental principles of multimedia systems and how they are being

developed and applied and also to gain an intuitive understanding of multimedia concepts.

2. In this course, students will be introduced to principles and current technologies of multimedia systems.

Course Outcomes:

The student after undergoing this course will be able to:

1. acquire fundamentals principles of multimedia, including digitization and data compression for non-textual information
2. understand issues in representing, processing, and transmitting multimedia data
3. understand core multimedia technologies and standards

UNIT - 1 (10 Hours)

Introduction: Branch Overlapping Aspects of Multimedia**,** Global Structure
Media and Data Steams: Medium**,** Main properties of a Multimedia System**,** Traditional Data Stream Characteristics, Multimedia Data Stream Characteristics for Continuous Media
Sound/Audio: Basic Sound Concepts Music Speech**,** Music**,** Speech
Image and Graphics: Basic Concepts**,** Computer Image Processing
Video and Animation: Basic concepts**,** Television, Computer-based Animation

UNIT - 2 (10 Hours)

Data Compression: Some Basic Compression Techniques, JPEG **,** H.261**,** MPEG, DVI

Computer Technology: Communication Architecture, Multimedia Workstation Multimedia
Multimedia Operating Systems: Introduction**,** Real time systems **,** File Systems

UNIT - 3 (10 Hours)

Networking Systems : Layers, Protocols and Services**,** Networks**,** LAN**,** MAN**,**WAN
Multimedia Communication Systems : Application Subsystem, Transport Subsystem, Quality of Service and Resource Management
Database Systems: Multimedia Database Management Systems **,** Characteristics of an MDBMS**,** Data Analysis, Data Structure**,** Operations on Data**,** Integration in a Database Model

UNIT - 4 (10 Hours)

User Interfaces: General Design Issues**,** Video at the User Interface **,** Audio at the User Interface **,** User-friendliness as the Primary Goal
Synchronization: Introduction, Notion of synchronization**,** Presentation Requirements, A Reference Model for Multimedia Synchronization**,** Synchronization Specification

Multimedia Applications: Introduction**,** Media Preparation**,** Media Composition, Media Integration, Media Communication, Media Consumption, Media Entertainment

Recommended Readings:

1. Ralf Steinmetz and Klara Nahrstedt; Multimedia: Computing, Communications and Applications; Pearson Education,
2. John F. Koegel Buford ; Multimedia Systems; Pearson Education, ISBN: 81-7808-162- 8
3. Tay Vaughan ; Multimedia: Making it Work; TMH, ISBN: 0-07-047276-93.

List of Experiments in Multimedia Systems:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To Study different Multimedia Formats and Editors
2. To Create An Animation Using JAVA
3. To study different animations Software's
4. To create an Animation using Flash Software
5. To implement Run Length Technique
6. To implement Shortest Seek Time First Algorithm
7. To implement Earliest Deadline First Algorithm
8. To implement Group Sweeping Scheduling Algorithm
9. To Design E-Newspaper Website using Dreamweaver
10. To Study various operations on MDBMS

COMP 8.3.3 BIO INFORMATICS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.3.3	Bio Informatics	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. Aims at providing an introduction to bioinformatics to interpret the rapidly expanding amount of biological information.
2. Discusses the basic concepts of bioinformatics and focuses on how to identify, obtain, establish, maintain and exchange research information in biology.
3. Examine the structure and function genes and proteins through the use of computational analysis, statistics and pattern recognition.
4. To discuss pattern representation, characterization and discovery in proteins.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understandmolecular bioinformatics concepts.
2. Study of Genome analysis and gene mapping.
3. Use of current bioinformatics tools and databases.
4. Study of Dynamic programming for sequence alignment.
5. Concepts of sequence analysis.
6. Analysis, visualization and representation of Molecular Structure.
7. Learn the key methods and tools used in bioinformatics.
8. Applications of bioinformatics in genomics

UNIT -1 (12 Hours)

Bioinformatics – an Introduction: Introduction, Historical Overview and Definition, Bioinformatics, Applications, Major databases in Bioinformatics, Data Management and Analysis, Molecular Biology and Bioinformatics, Central Dogma of Molecular Biology.

Information Search and Data Retrieval: Introduction, Tools for web search, Data Retreival Tools, Data Mining of biological databases.

Genome Analysis and Gene Mapping:Introduction, Genome analysis, Gene Mapping, The Sequence Assembly Problem, Genetic Mapping and Linkage analysis, Physical Maps, Cloning Entire Genome, Genome Sequencing, Applications of Genetic

Maps, Sequence Assembly Tools, Identification of Tools in Contigs, Human Genome Project.

UNIT - 2 (12 Hours)

Sequence Alignment: Introduction, Dot matrices and Hash coding, Dynamic programming in sequence algorithm.

Tools for Similarity Search and Sequence Alignment: Working with FASTA, Working with BLAST, Filtering and Gapped BLAST, FASTA and BLAST algorithm comparison.

Multiple Alignment, Substitution Matrices and Phylogenetic Trees: Multiple sequence alignment, Substitution Matrices, Phylogenetic Trees.

UNIT - 3 (12 Hours)

Protein and DNA Sequence Analysis:Pattern Representation and Characterization, Pattern Discovery and Sequence Classification in Proteins and Nucleic Acids.

Protein Structure Prediction and Protein Folding:Protein Secondary Structure Prediction, Protein tertiary Structure prediction.

Nucleic Acid Structure: RNA structure prediction, DNA Structural Polymorphism.

UNIT - 4 (12 Hours)

Gene Expression and Microarrays:Introduction, Working with DNA microarrays, Clustering Gene Expression Profiles, Data sources and tools for microarrays analysis, Applications – Functional Genomes, Comparative Geneomes, Medical Applications, Microarrays in Pharmaceutical industries, DNA Microarrays.

Protein Classification and Structure Visualisation: Introduction, Overview of protein structure, Protein Structure Visualization, Structure based protein classification, Protein Structure databases, Protein Structure Visualisation Database and Tools, Protein Structure Alignment, Domain Architecture Databases.

Introduction to Drug Discovery: Areas influencing drug discovery, Pharmacogenetics and Pharmacogenimics applications, Analysis of Single Nucleotide Polymorphism, Important parameters in Drug Discovery.

Recommended Readings:

1. S.C. Rastogi, N. Mendiratta, P. Rastogi; Bioinformatics – Methods and Applications; 4th Edition.
2. N. Gautham; Bioinformatics – Databases and Algorithms; Narosa Publication;
3. Jean-Michel Claveriw, CedrocNotredame; Bioinformatics – A Beginner's Guide;
4. Arthur M. Lesk; Introduction to Bioinformatics; OXFORD Publishers (Indian Edition);

5. T.K. Attwood, D J Parry; Introduction to Bioinformatics; Amith Addison Wesley Longman;

List of Experiments in Real Time Systems:

1. Study of biological databases.
2. Analysis of data retrieval and submission tools.
3. Implementation of dynamic programming methods for sequence alignment.
4. Study of BLAST and FASTA.
5. Understanding phylogenetic trees.
6. Clustering techniques for genes.
7. Protein structure visualization tools.
8. SAGE methodology for gene expression patterns.
9. Study of substitution matrices.
10. Analysis of protein structures.

COMP 8.3.4 Storage Area Networks

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.3.4	Storage Area Networks	3	1	2	3	100	25	--	--	25	**150**

Prerequisites: Information Retrieval System, Computer Networks & Cloud Computing

Course Educational Objectives:

The main objective of the course is to expose the students tostorage area network (SAN) infrastructure which facilitates storage consolidation, data sharing, server clustering,. This course focuses on the planning and implementation considerations associated with establishing that SAN infrastructure. Students will also learn the basic concepts and terminology associated with Storage Area Networks (SAN), Network Attached Storage (NAS) &Small Computer System Interface (SCSI)

Course Outcomes:

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

1. Describe the characteristics and components of Storage Area Networks
2. Describe the challenges with Server Centric IT architecture and the advantages of Storage Centric IT architecture.
3. Describe the logical and physical components of storage infrastructure.
4. Describe processes involved in File sharing operations on NAS and SAN
5. Students will demonstrate effective oral and writing communication skills necessary to be effective and to compete at global business environment.
6. Describe the business continuity and disaster recovery in a storage infrastructure.

UNIT 1 (14 hours)

Introduction: Server Centric IT Architecture and its Limitations; Storage – Centric IT Architecture and its advantages. Case study: Replacing a server with Storage Networks The Data Storage and Data Access problem; The Battle for size and access.

Intelligent Disk Subsystems: Architecture of Intelligent Disk Subsystems; Hard disks and Internal I/O Channels; JBOD, Storage virtualization using RAID and different RAID levels; Caching: Acceleration of Hard Disk Access; Intelligent disk subsystems, Availability of disk subsystems.

I/O Techniques: The Physical I/O path from the CPU to the Storage System; SCSI; Fibre Channel Protocol Stack; Fibre Channel SAN; IP Storage.

UNIT 2 (12 hours)

Network Attached Storage: The NAS Architecture, The NAS hardware Architecture, The NAS Software Architecture, Network connectivity, NAS as a storage system.

File System and NAS: Local File Systems; Network file Systems and file servers; Shared Disk file systems; Comparison of fibre Channel and NAS.

UNIT 3 (14 hours)

Storage Virtualization: Definition of Storage virtualization; Implementation Considerations; Storage virtualization on Block or file level; Storage virtualization on various levels of the storage Network; Symmetric and Asymmetric storage virtualization in the Network.

SAN Architecture and Hardware devices: Overview, Creating a Network for storage; SAN Hardware devices; The fibre channel switch; Host Bus Adaptors; Putting the storage in SAN; Fabric operation from a Hardware perspective.

UNIT 4 (12 hours)

Software Components of SAN: The switch's Operating system; Device Drivers; Supporting the switch's components; Configuration options for SANs.

Management: Planning Business Continuity; Managing availability; Managing Serviceability; Capacity planning; Security considerations.

Appropriate required number of case studies/experiments be performed covering the entire syllabus.

Recommended Readings:

1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, Wiley India, 2007.
2. Marc Farley: Storage Networking Fundamentals – An Introduction to Storage Devices, Subsystems, Applications, Management, and File Systems, Cisco Press, 2005.
3. Robert Spalding: "Storage Networks The Complete Reference", Tata McGraw-Hill, 2003.
4. Richard Barker and Paul Massiglia: "Storage Area Network Essentials a CompleteGuide to understanding and Implementing SANs", Wiley India, 2006.

COMP 8.3.5 WEB SERVICES

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.3.5	Web Services	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To learn and understand the various concepts of Web Services.
2. To learn basics of XML which is the basic prerequisite to understand how the different documents of the respective protocols are designed.

3. To learn the different protocols used in web services and their role and importance in designing a web service.

Course Outcomes:

The student after undergoing this course will be able to learn:

1. How information is exchanged between applications within a distributed environment. (SOAP).
2. How the web services are described to the world over internet (WSDL).
3. How the web service is published and made known to the world over the internet. (UDDI).
4. How to explain the conversation pattern that a web service is expecting to engage in. (WSCL)
5. How workflow systems automate business processes. (Workflow).
6. Advantages and Disadvantages of Web Services.
7. Transactions and the transaction protocols used in web service.
8. Security issues in Web Services.

UNIT - 1 (14 Hours)

Web Service and SOA fundamentals: Introduction, Concept of Software as a Service(SaaS), Web services versus Web based applications, Characteristics of Web services, Service interface and implementation, The Service Oriented Architecture(SOA), Quality of service (QoS), Web service interoperability, Web services versus components, RESTful services , Impact and shortcomings of Web services.

Web Services Architecture: Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication, basic steps of implementing web services, developing web services enabled applications.

UNIT - 2 (12 Hours)

Extensible Markup Language (XML): XML Fundamentals. XML, XML Documents, XML Namespaces. XML Schema, Processing XML.
XML Parsing: SAX, COM, JAXB. Xpath, XQuery.

UNIT - 3 (14 Hours)

SOAP: Simple Object Access Protocol, Inter-application communication and wire protocols, SOAP as a messaging protocol, Structure of a SOAP message, SOAP

communication model, Building SOAP Web Services, developing SOAP Web Services using Java, Error handling in SOAP, Advantages and disadvantages of SOAP.

Describing and Discovering Web Services: WSDL in the world of Web Services, Web Services life cycle, anatomy of WSDL definition document, WSDL bindings, WSDL Tools, limitations of WSDL, Service discovery, role of service discovery in a SOA, service discovery mechanisms, UDDI – UDDI Registries, uses of UDDI Registry, Programming with UDDI, UDDI data structures, support for categorization in UDDI Registries, Publishing API, Publishing information to a UDDI Registry, searching information in a UDDI Registry, deleting information in a UDDI Registry, limitations of UDDI.

UNIT - 4 (12 Hours)

Conversations: Web service conversation Language, WSCL Interface component, Relationship between WSCL and WSDL.
Workflow: Business Process Management, Workflow and workflow Management systems, Business Process Execution Language (BPEL).
Security: Everyday Security Basics, Security Is An End-to-End Process, Web Service Security Issues, Types of Security Attacks and Threats, Web Services Security Roadmap, WS-Security.

Recommended Readings:

1. Michael P. Papazoglou; Web Services & SOA: Principles and Technology; Pearson Education , 2/e,.
2. Harvey M.Dietel & Paul J.Dietel ; Web Services: A Technical Introduction; Prentice Hall PTR, ISBN: 0130461350
3. Sandeep Chatterjee, James Webber; Developing Enterprise Web Services – An Architect's Guide; Pearson Education ISBN: 0-13-140160-2.
4. Stephen Potts, Mike Kopack; Sams Teach Yourself Web Services in 24 Hours; Sams Publications ISBN:13:978-0672325151.
5. R. Nagappan, R. Skoczylas, R.P. Sriganesh; Developing Java Web Services; Wiley India.

List of Experiments in Web Services:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To implement XML Schema and File
2. To study and implement XML inheritance.

3. To study and implement SOAP and WSDL.
4. To study and implement DOM.
5. To implement XML encryption
6. To implement XML query
7. Creating web service using JAVA
8. Creating web service using .NET
9. Case study on XPath, XJAXB

COMP 8.4.1 GENETIC ALGORITHMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.4,1	Genetic Algorithms	3	1	2	3	100	25	--	-	25	**150**

Course Objectives:

1. To learn and understand the various concepts of Genetic Algorithms.
2. To learn basics of GA Algorithms and their industrial applications

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the mathematical foundation of Genetic Algorithms,
2. Define fitness function for various events,
3. Implement GA models for applications.

UNIT - 1 (12 Hours)

Introduction to Genetic Algorithms**:** Robustness of traditional optimization and search techniques, Goals of optimization, Similarity Templates. Mathematical Foundations: Fundamental theorem, Schema Processing, Problem solving-2 armed and K armed bandit

problem, Building block hypothesis, Minimal deceptive problem, Similarity templates as hyper planes,

UNIT - 2 (12 Hours)

COMPUTER Implementation Of Genetic Algorithms**,** Data structure, reproduction, crossover and mutation**,** Mapping objective functions to fitness form**,** Fitness scaling, discretization and constraints. Applications Of Genetic Algorithms , DeJong and Function optimization structural optimization via genetic algorithm. Medical image registration with genetic algorithms**,** Iterated prisoner's dilemma problem..

UNIT - 3 (12 Hours)

Advanced Operators And Techniques In Genetic Algorithm Search: Dominance, Diploidy and abeyance, Inversion and other re-ordering operators, Macro operators, niche and special speciation, Multi objective optimization, Knowledge based techniques, Genetic Algorithms and Parallel processors, Genetic Based machine learning, Classifier systems

UNIT - 4 (12 Hours)

Industrial Application Of Genetic Algorithms: Data mining using genetic Algorithms Search in data mining, Genetic algorithms for game playing eg TIC TAC TOE, DNA Sequence Processing using Geneti Algorithms, Clustering using Genetic Algorithm

Genetic Algorithm Performance Analysis with Different Techniques: Sequential Techniques
Statistical Techniques, Data Mining Techniques, Machine Learning Techniques

List of Experiments:

1. Implementation of a procedure that receives two binary strings and a crossing site value, performs simple crossover to return two offspring strings.
2. Implementation of pseudorandom integer generator using specified lower limit and upper limit.
3. Implement a coding routine to implement a floating point code with specified mantissa and exponent.
4. Implementation and testing a routine to perform mutation.
5. Develop a multipoint crossover procedure similar to De Jong's with parameter crossover points.
6. Compare and Contrast alternative scaling schemes.
7. Compare and contrast alternative ranking procedures.
8. Implementation of inversion operator that treats a permutation as a circular string.
9. Implementation of genetic algorithm with diploidy, dominance and the triallelic dominance map.
10. Implementation of Genetic Algorithm Performance Analysis using Different Techniques

TEXT BOOKS:

1. Genetic Algorithms in search, optimization machine leaning - David Goldberg 6th edition , ISBN No-81-7808-130-X
2. Industrial applications of Genetic Algorithms- Charles L Karr and L.Michael Freeman, CRC Press, ISBN No-0-8493-9801-0
3. Data Mining concepts and Techniques - Jiawei Han, Micheline Kamber and Jian Pei, 3 rd Edition.
4. Machine Learning - Tom M. Mitchell - 2nd Edition.

REFERENCE BOOKS

1. Handbook of Genetic Algorithms -Davis, Lawrence, ISBN:0-442-00173-8
2. An Introduction to Genetic Algorithms Melanie Mitchell, ISBN:81-203-1358-5
3. Introduction to Data Mining - **Pang-Ning Tan, Michael Steinbach, Vipin Kumar.**
4. Statistical Techniques for Data Analysis,John K. Taylor, Cheryl Cihon - Second Edition.

COMP 8.4.2 REAL TIME SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.4.2	Real Time Systems	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To introduce students to the concepts, and approaches in the design and analysis of real-time systems.
2. To study issues related to the design and analysis of systems with real-time constraints.
3. To study and analyze scheduling in Real Time Systems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand fundamental principles of real time systems with time and resource limitations
2. Describe the reference model of real time systems.
3. Understand Real-time scheduling and schedulability analysis on uniprocessor systems.
4. Understand the Real Time System model on Multiprocessor and Distributed systems.

UNIT -1 (11 Hours)

Introduction: A Car and Driver Example, Issues in Real Time Computing, Structure of a Real Time system, Task Classes

Hard Versus Soft Real-Time Systems: Jobs and Processors, Release Times, Deadlines and Timing Constraints, Hard and Soft Timing Constraints, Hard Real Time systems, Soft Real Time Systems

A Reference Model of Real Time Systems: Processors and Resources, Temporal Parameters of Real –Time Workload, Period Task Model, Precedence Constraints and Data Dependency, Other Types of Dependencies, Functional Parameters, Resource Parameters of Jobs and Parameters of Resources, Scheduling Hierarchy

Characterizing Real- Time systems and Task: Introduction, Performance Measures for Real-Time Systems, Estimating Program Run Times.

UNIT - 2 (14 Hours)

Clock Driven Scheduling: Notation and Assumptions, Static Timer-Driven Scheduler, General Structure of Cyclic Schedules, Cyclic Executives, Improving the Average Response time of Aperiodic Jobs, Scheduling Sporadic jobs, Practical considerations and Generalizations, Pros and Cons of Clock Driven Scheduling

Priority Driven Scheduling of Periodic Tasks: Static Assumptions, Fixed priority versus Dynamic Priority Algorithms, Maximum Schedulable Utilizations, Optimality of RM and DM algorithms, A schedulability test for Fixed Priority Tasks with Short Response times, Schedulability test for Fixed Priority Tasks with Arbitrary Response times, Sufficient Schedulability conditions for the RM and DM algorithms.

UNIT - 3 (13 Hours)

Scheduling Aperiodic and Sporadic Jobs in Priority Driven Systems: Assumptions and Approaches, Deferrable Servers, Sporadic servers, Constant Utilization, Total Bandwidth and Weighted Fair Queuing Servers, Scheduling of Sporadic jobs.

Resource and Resource Access Control: Assumptions on Resources and their usage, Effects of Resource Contention and Resource Access Control, Nonpreemtive Critical Sections, Basic Priority Inheritance Protocol, Basic Priority Ceiling protocol

UNIT - 4 (11 Hours)

Task Assignment and Scheduling: Task Assignment, Mode Changes

Multiprocessor Scheduling, Resource Access control and Synchronization: Model of Multiprocessor and Distributed systems, Task assignment, Multiprocessor priority ceiling protocol, Elements of Scheduling Algorithms for End to End Periodic tasks, End to End tasks in heterogeneous systems.

Recommended Readings:

1. Jane W. S. Liu; Real-Time Systems; Pearson Education; ISBN: 978-81-7758-575-9
2. C. M. Krishna and K. G. Shin; Real-Time Systems; TMH; ISBN: 0-07-114243-6.
3. Williams Rob; Real Time Systems Development; **ISBN:** 978-81-3121-520-3
4. Alan Burns; Real- Time Systems and Programming Languages; **ISBN:** 0-201-72988-1
5. Laplante P.A.; Real – Time Systems Design and Analysis; **ISBN:** 81-265-0830-2

List of Experiments in Real Time Systems:

1. Thread Programming.
2. Implementation of Clock Driven Scheduler.
3. Implementation of Table Driven cyclic executive.
4. Implementation of cyclic executive with sporadic and aperiodic job scheduling.

5. Implementation of RM Algorithm.
6. Implementation of DM Algorithm.
7. Implementation of EDF Algorithm
8. Implementation of LST Algorithm.
9. Implementation of FIFO Algorithm
10. Implementation of LIFO Algorithm.

COMP 8.4.3 MOBILE COMPUTING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.4.3	Mobile Computing	3	1	2	3	100	25	-	-	25	**150**

Course Objectives:

1. To understand the basic concepts of mobile computing
2. To be familiar with the network protocol stack
3. To learn the basics of mobile telecommunication system
4. To be exposed to Ad-Hoc networks
5. To gain knowledge about different mobile platforms and application development

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the basics of mobile telecommunication system
2. Choose the required functionality at each layer for given application
3. Identify solution for each functionality at each layer
4. Use simulator tools and design Ad hoc networks
5. Develop a mobile application.

UNIT - 1 (11 Hours)

Mobile Computing – Mobile Computing Vs wireless Networking – Mobile Computing Applications – Characteristics of Mobile computing – Structure of Mobile Computing Application. MAC Protocols – Wireless MAC Issues – Fixed Assignment Schemes – Random Assignment Schemes – Reservation Based Schemes.

UNIT - 2 (11 Hours)

Overview of Mobile IP – Features of Mobile IP – Key Mechanism in Mobile IP – route Optimization. Overview of TCP/IP – Architecture of TCP/IP- Adaptation of TCP Window – Improvement in TCP Performance.

UNIT - 3 (11 Hours)

Global System for Mobile Communication (GSM) – General Packet Radio Service (GPRS) –Universal Mobile Telecommunication System (UMTS).

UNIT - 4 (11 Hours)

Ad-Hoc Basic Concepts – Characteristics – Applications – Design Issues – Routing – Essential of Traditional Routing Protocols –Popular Routing Protocols – Vehicular Ad Hoc networks (VANET) – MANET Vs VANET – Security.

Recommended Readings:

1. Prasant Kumar Pattnaik, Rajib Mall; Fundamentals of Mobile Computing; PHI Learning Pvt. Ltd, New Delhi; 2012.
2. Jochen H. Schiller; Mobile Communications - Second Edition; Pearson Education, New Delhi; 2007.
3. Uwe Hansmann, Lothar Merk, Martin S. Nicklons and Thomas Stober; Principles of Mobile Computing; Springer; 2003.

List of Experiments in Mobile Computing:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To implement Aloha.
2. To implement slotted Aloha.
3. To implement CSMA.
4. To implement CSMA/CA.
5. To implement CDMA.
6. Case study on AODV.
7. Case study on DSR.
8. Case study on ABR.

COMP 8.4.4 MACHINE LEARNING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.4.4	Machine Learning	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To understand basic concepts of computer machine vision system,
2. To study the design of algorithms for vision and learning

Course Outcomes:

Upon completion of this class, students should be able to:

1. Identify requirement of a computer learning system
2. Explain the concepts behind learning techniques.
3. Implement the learning algorithms for real time applications.

UNIT - 1 (11 Hours)

Learning problems, Designing a learning system, Issues in machine learning.
Concept Learning, Finding a maximally specific hypothesis, Version Spaces, candidate elimination algorithms, Inductive bias.
Decision Tree Representation, Decision Tree Learning Algorithms, hypothesis space search, Inductive bias and issues in decision tree learning. Evaluating Hypothesis.

,

UNIT - 2 (11 Hours)

Bayesian Learning, Concept learning through Bayes Theorem, Maximum Likelihood and Leasr squared error hypothesis, Minimum Description Length principle, Bayes Optimal classifier, Gibbs Algorithm, Naïve Bayes classifier, Bayesian belief network

UNIT - 3 (11 Hours)

Artificial Neural Networks, Perceptrons, Multilayer neural networks, back propagation algorithm, Instance based learning: k nearest neighbor algorithm, locally weighted regression,

UNIT - 4 (11 Hours)

Theoretical Approaches: inductive Inference , Grammatical Inference. PAC Learning . Complexity of Learning , polynomial learnability , VC-dimension. Instance based learning.
Application of Machine learning to data mining and knowledge discovery

List of Experiments:

1. Implementation of Candidate Elimination algorithm
2. Implementation of Decision Tree classifier
3. Implementation of Naïve Baye's classifier
4. Implementation of Gibb's sampling algorithm
5. Implementation of Artificial Neural Network
6. Implementation of K-Nearest Neighbours algorithm
7. Dataset preparation and feature-engineering
8. Implementation of NLP tasks

Recommended Reading

1: Tom Mitchell, Machine Learning, McGraw Hill Inc, 1997
2 Anthony ,M. and Biggs , N. ,Computational Learning Theory , Cambridge 1992.
3. Ross Q. J : Program for machine learning , Morgan Kaufmann 1997
4. Hastie, Tibshirani, Friedman The elements of Statistical Learning Springer Verlag.
5. Pattern recognition and machine learning by Christopher Bishop, Springer Verlag

COMP 8.4.5 DIGITAL SIGNAL PROCESSING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration(Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.4.5	Digital Signal Processing	3	1	2	3	100	25	-	-	25	**150**

Course Objectives:

1. Understand basic concepts and methodologies in Digital Signal Processing.
2. Understand the fundamental concepts of discrete transforms.
3. Study the applications of Z Transforms.
4. Study of digital filters and their applications.

Course Outcomes

Upon completion of this class, students should be able to:

1. Explain the concepts of signals and systems and the basic operations on them.
2. Analyse the behaviour of periodic and aperiodic signals in frequency domain using the Fourier Series and Fourier Transforms.
3. Describe the concept and characteristics of Z Transforms and its use in the analysis and applications of systems.
4. Explain the techniques of designing of Infinite Impulse Response (IIR) filters and Finite Impulse Response (FIR) filters.

UNIT -1 (12 Hours)

Digital Signal Processing and Its Benefits. Application Areas. Key DSP Operations. Digital Signal Processors. Overview of Real-world Applications of DSP. Telecommunications Applications of DSP. DFT and its Inverse. Properties of the DFT. Computational Complexity of the DFT. The Decimation-in-Time Fast Fourier Transform Algorithm. Inverse Fast Fourier Transform. Implementation of the FFT. Other Discrete Transforms. An Application of the DCT: Image Compression.

UNIT -2 (12 Hours)

Discrete-Time Signals and Systems. The Z-Transform, The Inverse Z-Transform. Properties of the Z-Transform. Some Applications of the Z-Transform in Signal Processing. Correlation and Convolution. Correlation Description. Convolution Description. Implementation of Correlation and Convolution. Application Examples.

UNIT -3 (12 Hours)

Introduction to Digital Filters. Types of Digital Filters: FIR and IIR Filters. Choosing Between FIR and IIR Filters. Filter Design Steps. Introduction. FIR Filter Design. FIR Filter Specifications. FIR Coefficient Calculation Methods. Window Method. The Optimal Method. Frequency Sampling Method. Comparison of the Window, Optimum and Frequency Sampling Methods. Special FIR Filter Design Topics. Realization Structures for FIR Filters. Finite Wordlength Effects in FIR Digital Filters. FIR Implementation Techniques. Design Example. Application Examples of FIR Filters.

UNIT -4 (12 Hours)

Design of Infinite Impulse Response (IIR) Digital Filters: Summary of the Basic Features of IIR Filters. Design Stages for Digital IIR Filters. Performance Specification. Coefficient Calculation Methods for IIR Filters. Pole-Zero Placement Method of Coefficient Calculation. Impulse Variant Method of Coefficient Calculation. Matched Z-Transform (MZT) Method of Coefficient Calculation. Bilinear Z-Transform (BZT) Method of Coefficient Calculation. Use of BZT and Classical Analog Filters to Design IIR Filters. Calculating IIR Filter Coefficients by Mapping S-Plane Poles and Zeros. Using IIR Filter Design Programs. Choice of Coefficient Calculation Methods for IIR Filters. Realization Structures for IIR Digital Filters. Finite Wordlength Effects in IIR Filters. Implementation of IIR Filters. A Detailed Design Example of an IIR Digital Filter.

Recommended Readings:

1. Digital Signal Processing – by Emmanuel C..Ifeachor, &Barrie.W.Jervis, Second edition, Pearson Education / Prentice Hall, 2002.
2. Digital Signal Processing: Principles, Algorithms, and Applications, by John G. Proakis and Dimitris G. Manolakis, Prentice Hall, 1996
3. Discrete-Time Signal Processing, by Alan V. Oppenheim, Ronald W. Schafer, Prentice Hall, ISBN:0-13-216292-X
4. Digital Signal Processing, A Computer Based approach, by S.K. Mitra, Tata McGraw Hill, 1998
5. Digital Signal Processing by Ramesh Babu, Scitech India publications Limited, Fourth Edition,2007

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Defining and plotting Discrete Time Signals
2. Implementation ofLinear Difference Equation.
3. Implementation of Linear Convolution.
4. Implementation of Circular Convolution.
5. Implementation of Overlap Save method of Convolution
6. Implementation of Overlap Add method of Convolution
7. Implementation of Discrete Fourier Transform.
8. Implementation of Inverse Discrete Fourier Transform.
9. Implementation of Z Transform.
10. Implementation of Inverse Z Transform.