SCHOOL
OF
INFORMATION AND COMMUNICATION TECHNOLOGY

COURSE STURCTURE AND DETAILED SYLLABUS

3 YEARS M. TECH. ICT FOR SCIENCE GRADUATES

SPECIALIZATION:

SOFTWARE ENGINEERING

GAUTAM BUDDHA UNIVERSITY
GAUTAM BUDH NAGAR, GREATER NOIDA
2011-2012

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SEMESTER – I

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	MA431	Engineering Mathematics	3-1-0	4
2	EC431	Digital Electronics	3-0-0	3
3	CS431 / CS203	Operating System	3-0-0	3
4	CS433 / CS202	Software Engineering	3-0-0	3
5	CS435	Problem Solving Using C	3-1-0	4
6	SS431	Technical Communication	2-0-0	2
7	EC483 / EC281	Digital Electronics Lab	0-0-2	2
8	CS483	Programming Lab	0-0-2	2
9	GP431	General Proficiency	------	1
		Total Credits		**24**
		Total Contact Hours	**17-2-4 =**	**23**

SEMESTER – II

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	EC442	Communication Engineering	3-0-0	3
2	CS434	Object-Oriented Programming with JAVA	3-0-0	3
3	CS436 / CS206	Data Base Management System	3-0-0	3
4	CS438 / CS309	Computer Organization and Architecture	3-0-0	3
5	EC444 / EC205	Signal and Systems	3-1-0	4
6	EC446 / EC306	Microelectronics Engineering	3-0-0	3
7	CS484	Object-Oriented Programming with Java Lab	0-0-2	2
8	EC480	Design Lab	0-0-2	2
9	GP432	General Proficiency	------	1
		Total Credits		**24**
		Total Contact Hours	**18-1-4 =**	**23**

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SEMESTER – III

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS531	Data Structures and Algorithm Design	3-1-0	4
2	CS533	Principles of Compiler Design	3-0-0	3
3	EC553	Data Communication	3-0-0	3
4	EC555 / EC465	Principles of VLSI Design	3-0-0	3
5	EC557	Cellular Mobile Communication	3-0-0	3
6	EC559 / EC304	Microprocessors and Interfacing	3-0-0	3
7	CS583	Data Structures and Algorithms Design Lab	0-0-2	2
8	EC587 / EC384	Microprocessor and Interfacing Lab	0-0-2	2
9	GP531	General Proficiency	------	1
		Total Credits		**24**
		Total Contact Hours	**18-1-4 =**	**23**

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SEMESTER IV

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	MA402	Simulation & Modeling	3-1-0	4
2	CS532	Software Architecture and Design	3-0-0	3
3	CS534	Open Source Software Systems	3-0-0	3
4		**Elective-1**	3-0-0	3
5		**Elective-2**	3-0-0	3
6	CS582	Open Source Software Systems Lab	0-0-2	2
7	CS592	Major Project	0-0-10	5
8	GP532	General Proficiency	------	1
		Total Credits		**24**
		Total Contact Hours	**15-1-12 =**	**28**

Electives (1 & 2)		
Sr. No.	**Courses Code**	**Courses**
1	CS542 / CS442	Component-Based Software Engineering
2	CS544 / CS444	Aspect-Oriented Software Engineering
3	CS546 / CS446	Software Re-Engineering
4	CS548 / CS448	Software Reusability
5	CS550 / CS450	Web-Based Software Engineering
6	CS552 / CS452	Software Agents

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SEMESTER V

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS631	Software Testing	3-0-0	3
2	CS633	Research Techniques in ICT	3-0-0	3
3		**Electives – 3**	3-0-0	3
4		**Electives – 4**	3-0-0	3
5	CS581	Software Testing Lab	0-0-2	2
6	CS691	Dissertation Part - I	0-0-14	7
7	GP631	General Proficiency	------	1
		Total Credits		**22**
		Total Contact Hours	**12-0-16 =**	**28**

Electives (3 & 4)		
Sr. No.	**Courses Code**	**Courses**
1	CS641 / CS545	Software Measurement and Estimation
2	CS643 / CS547	Software Reliability and Fault Tolerant Systems
3	CS645 / CS549	Software Quality Assurance and Engineering
4	CS647 / CS553	Software Maintenance
5	CS649 / CS555	Software Performance

SEMESTER VI

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS690	Dissertation Part – II	------	21
2	GP632	General Proficiency	------	1
		Total Credits		**22**

GRAND TOTAL CREDITS = 140

SOFTWARE ENGINEERING

(SEMESTER - I)

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

ENGINEERING MATHEMATICS			
Course Code:	**MA431**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I ORDINARARY DIFFERENTIAL EQUATIONS

First order differential equations: Separable equations, exact differential equations, integrating factors, linear first order differential equations, variation of parameters, existence and uniqueness of solutions, homogenous linear equations of the second order, homogenous second order equations with constant coefficients, general solutions, homogenous linear equations of arbitrary order with constant coefficients, nonhomogenous linear equations.

UNIT II LAPLACE TRASFORM AND ITS APPLICATIONS

Laplace transform, linearity, Laplace transforms of derivatives and integrals, existence theorem, differentiation and integration of transforms, unit step function and Dirac Delta function, Laplace transform of periodic functions, inverse transform, solution of Initi as value problems.

UNIT III MATRICES

Algebra of matrices, rank of a matrices, Cramer's rule, homogeneous and non homogenous system of linear equations, Gauss's elimination method, eigenvalues and eigenvectors, reduction to diagonal form, Quadratic forms.

UNIT IV FOURIER SERIES AND PARTIAL DIFFERENTIAL EQUATIONS

Periodic functions, Fourier series, Euler's formulas, even and odd functions, functions having arbitrary period, half range expansions, determinations of Fourier coefficients without integration, methods for determining coefficients, Fourier integrals, orthogonal functions, Sturm-Liouville problem, vibrating string, one-dimensional wave equation, Method of separation of variables, D'Alemberts's solution of wave equation, one-dimensional heat flow, vibrating membrane, two-dimensional wave equation, Laplace's equation,

UNIT V COMPLEX VARIABLES

Function of complex variables, analytic functions, Cauchy-Riemann equations, Laplace's equation, line integral in the complex plane, Poles and singularities, Cauchy's integral theorem, Cauchy's integral formula, The derivative of an analytic function, Taylor series, Laurent series and applications, Cauchy residue theorem and its applications to evaluating real integrals..

Reference Books:

1. Advanced Engineering Mathematics, Erwin Kreyszig, 9th Edition, John Wiley, 2006
2. Complex Variables and Applications, Ruel V.Churchill, James Ward Brown, 8th edition, McGraw Hill, 2008.
3. Applied Complex Variables, John W Dettman, Dover Publications New York, 1984
4. Calculus and Analytic Geometry, George B. Thomas, Ross Finney, 9th edition, Narosa, 2003.
5. Introduction to Linear Algebra, Strang Gilbert, 4th edition, Wellesley Cambridge Press, 2009.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

DIGITAL ELECTRONICS			
Course Code:	**EC431**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Number systems & codes, Binary arithmetic Boolean algebra and switching function. Minimization of switching function, concept of prime implecant etc. Karnaugh's map method, Quine & McCluskey's method, cases with don't care terms and multiple outputs switching function. Logic gates, NAND, NOR realization of switching function; half-adder half-subtractor full-adders full-subtractor circuits. Series & parallel addition and BCD adders, look-ahead carry generator.

UNIT II

Linear wave shaping circuits, Bistable, monostable & astable multivibrators, Schmitt trigger circuits .Introduction to D/A converters. Various types of Analog to Digital & Digital to Analog converters sample & hold circuits and V-F converters.

UNIT III

Logic families: RTL, DTL, all types of TTL circuits, ECL, 12 L and PMOS, NMOS & CMOS logic etc. Gated flip-flops and gated multivibrators etc; Interfacing between TTL to MOS, vice-versa.

UNIT IV

Introduction to shift registers / ring counters synchronous & asynchronous counters and designing of combinational circuits like code converters & counters etc.

UNIT V

Semiconductor memories & designing with ROM and PLA: Decoders Encoders multiplexers & demultiplexers**.**

Text Books:

1. Tocci, "Digital Systems Principles & Applications".
2. M. Mano, "Digital Logic & Computer Design", (PHI).

Reference Books:

3. John F. Wakerly, Digital Design: Principles & Practices, Pearson Education.2003
4. Richard F.Tinder, Engineering Digital Design, 2/e, Harcourt India Private Ltd., 2001
5. William I. Fletcher, An Engineering Approach to Digital Design, Pearson Education
6. William H.Gothmann, Digital Electronics: An Introduction to Theory and Practice, Eastern Economy Edition, Prentice-Hall of India Private Limited, New Delhi. 2001.
7. Jacob Millman & Herbert Taub,Pulse,Digitaland Switching Waveforms,13th Reprint,Tata McGraw Hill Publishing Company Ltd.,NewDelhi, 1999

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

OPERATING SYSTEM			
Course Code:	**CS431/CS203**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO OPERATING SYSTEM

Importance of operating systems, basic concepts and terminology about operating system, memory management functions, processor management functions, device management functions, information management functions.

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm evaluation process synchronization, synchronization hardware, semaphores, classical problem of synchronization, monitors and atomic transaction deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY MANAGEMENT

Memory management, logical versus physical address space, swapping, contiguous allocation, paging, segmentation, demand paging, page replacement, page replacement algorithms, allocation of frames, thrashing, demand segmentation.

UNIT IV STORAGE MANAGEMENT

File concept, directory structure, protection, file-system structure, allocation method, free-space management, directory implementation.

UNIT V I/O SYSTEMS

I/O hardware, Application of I/O interface, Overview of Kernel I/O subsystem, three types of I/O systems, memory based I/O, I/O based I/O, peripheral based I/O.

Reference Books:

1. Galvin, Wiley, Operating Systems Concepts, 8th edition, 2009.
2. James L Peterson, Operating Systems Concept, John Wiley & Sons Inc, the 6Rev edition, 2007.
3. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
4. Stallings William, Operating Systems, PHI, New Delhi, 1997.
5. Madnick and Donavon, Operating Systems, McGraw Hill, International edition, 1978.
6. S. Tanenbaum Modern Operating Systems, , Pearson Education, 3rd edition, 2007.
7. Nutt, Operating System, Pearson Education, 2009.
8. S. Tanenbaum, Distributed Operating Systems, Prentice Hall, 2nd edition, 2007.
9. M. Singhal & N. Shivaratri, Advanced Concepts in Operating Systems, McGraw Hill, 2003.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE ENGINEERING			
Course Code:	**CS433/CS202**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering: definitions, role of software engineering, planning a software project, defining the problem, developing a solution strategy, planning the development process, software engineering process paradigms, principles of software engineering, software engineering activities.

UNIT II REQUIREMENT ANALYSIS AND DESIGN

Software Requirement Specification (SRS): Introduction, need of SRS, significance, characteristics of SRS, Structure of SRS, IEEE standards for SRS design, functional and non-functional requirements, Requirement gathering and analysis, requirement engineering and management.

UNIT III SOFTWARE DESIGN PROCESS

Software Design: Introduction, design process activities: architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design modular approach, top-down design, bottom-up design, design methods: data-flow model: data flow diagram, entity-relation-attribute model: E-R diagram, structural model: structure charts, context diagrams, object models: use case modeling, use case diagrams, sequence diagrams, cohesion and coupling.

UNIT IV SOFTWARE LIFE CYCLE MODELS

Software Development Life Cycle (SDLC), SDLC models, waterfall model and its variations, prototype model, iterative enhancement model, spiral model, RAD model, comparison of these models, software development teams, software development environments, validation and traceability, maintenance, prototyping requirements, Software project management.

UNIT V SOFTWARE TESTING AND MAINTENANCE

Testing Methods: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing, thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, Evolution of software products, economics of maintenance, category of software maintenance, Role of product development life cycle, deployment model, adaptive maintenance, corrective maintenance, perfective maintenance, enhancement request, proactive defect prevention, problem reporting, problem resolution, software maintenance from customers' perspective, maintenance standard: IEEE-1219, ISO-12207.

Reference Books:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.
3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.

PROBLEM SOLVING USING C			
Course Code:	**CS435**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3 +1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION

Introduction to problem solving, concept of algorithm, design of algorithms, iterative versus recursive style, top down design, program verification, efficiency of algorithms, concept of recursion, some simple example to illustrate these concepts like finding the GCD of two numbers, swapping two variables, summation of n given numbers, generation of Fibonocci sequence, reversing a given number, base conversion.

UNIT II INTRODUCTION TO C

C character set, delimiters, C keywords, identifiers, constants, variables, rules, type conversion, priority of operators and their clubbing, comma and conditional operator, arithmetic operators, relational operators, logical operators, bitwise operators, input in C, formatted and unformatted functions, library functions.

UNIT III MORE ABOUT C

If statement, if--- else statement, various forms of if-nested if-break statement, continue statement, go to statement, switch statement, nested switch statement, for statement, while statement do while statement to while statement, arrays, working with string and standard functions.

UNIT IV ADVANCED CONCEPTS OF C

Introduction to pointers, pointer declaration, arithmetic operations with pointers, pointers and arrays, pointers and two-dimensional arrays, array of pointers, pointers to pointers, pointers and strings, void pointers, function definition and declaration, proto types of functions, call by value and reference, functions returning more values, function as an argument, function with operators, function and decision statements, function and loop statements, function with arrays and pointers, recursion, pointer to function, storage classes.

UNIT V PROBLEM SOLVING

Reversal of an array, removal of duplicates in an ordered array, partitioning of an array, finding the Kth smallest of an element of an array, finding the longest monotone subsequence of an array, linear search, binary search, hash searching, bubble sort, merge sort, quick sort, insertion sort, selection sort, text processing, towers of Hanoi problem using recursion.

ADDITIONALS IN C: preprocessor directives, structures and unions, bit wise operators, files, command line arguments, dynamic memory allegation, graphics in C.

Text Books:

1 Ashok N.Kamthane, Programming with ANSI and Turbo C, Pearson Education, New Delhi.
2. R.G. Dromey, How to Solve it by computer, Prentice Hall of India Ltd, New Delhi.

Reference Books:

3. N.G. Venkateshmurthy, Programming techniques through C, Pearson Education, New Delhi.
4. Byron s Gottfried, Programming with C, Schaum's Outline series, Tata McGraw Hill New Delhi.
5. Jacqueline A.Jones & Keith Harrow, C programming with problem solving, Dreamtech publications, New Delhi.

DIGITAL ELECTRONICS LAB			
Course Code:	**EC483/EC281**	**Credits:**	**2**
No. of Practical (Hrs/Week):	**2**		
Total No. of Lab Sessions:	**15**	**End Sem. Exam Hours:**	**2**

Programs/Experiments List:

1. Verify the truth table of AND Gate.
2. Verify the truth table of OR and NOT gates.
3. Verify the truth table of NAND , NOR Gates.
4. Verify the truth table of and EX-OR Gate.
5. Design a combinational circuit to realize the function f(ABC)=A(B+C) using NAND gates.
6. Design a half adder using NOR gates only
7. Design full adder
8. Design a given size of Mux using gates.
9. Verify RS and JK flip flops
10. Verify D and T Flip flops
11. Design a up/down 3-bit counter
12. Design a 3 bit shift register.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

PROGRAMMING LAB			
Course Code:	**CS483**	**Credits:**	**2**
No. of Practical (Hrs/Week):	**2**		
Total No. of Lab Sessions:	**15**	**End Sem. Exam Hours:**	**2**

Programs/Experiments List:

1. Write a C Code to implement each of the following:
   Variable, constant, arithmetic operator, relational operator, logical operator, assignment operator, increment & decrement operator, conditional operator, bitwise operator.

2. Write a C Code to implement each of the following:
   Decision statement, loops statement and branch statements

3. Write a C Code to implement each of the following:
   Array: Single and Two dimensional arrays

4. Write C Code to generate the following output with the help of two dimensional array.
   7 14 21 28 35 42 49 56 63 70 Sum = 385
   5 10 15 20 25 30 35 40 45 50 Sum = 275
   3 6 9 12 15 18 21 24 27 30 Sum = 165

5. Write C Code to implement each of the following.
   Matrix Multiplication, Matrix Addition

6. Write a C Code to implement each of the following:
   Strings, Standard library string functions and array of pointers to strings

7. Write a C Code to implement each of the following sorting:
   Bubble Sort, Selection Sort, Insertion Sort, Merge Sort

8. Write a C Code to implement each of the following searching:
   Linear search, Binary search and Hash searching

9. Write a C Code to implement each of the following:
   Function: Implementation of function with call by values and call by reference

10. Write a C Code to implement each of the following:
    Pointers: Pointers declaration, array of pointers, pointers to pointers, pointers and strings

11. Write a C Code to implement recursion and tower of Hanoi problem using recursion.

12. Write a C Code to implement each of the following:
    Structure, Array of structure

13. Write a C Code to implement each of the following:
    Preprocessor, Macro Expansion and File Inclusion

14. Write a C Code to implement the following:
    File Handling
15 Develop a mini projects in C.

SOFTWARE ENGINEERING

(SEMESTER - II)

COMMUNICATION ENGINEERING			
Course Code:	**EC442**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SIGNALS AND ITS REPRESENTATION

Review of Fourier transform, convention, signal transmission though linear systems, signal distortion in transmission, poley wiener criteria, bandwidth and rise time, energy and power signals, spectral density and persevall's theorem for energy of power signals, Hilbert transform.

UNIT II LINEAR MODULATION AND EXPONENTIAL MODULATION

Linear modulation: definition, necessity of modulation, principle of amplitude modulation, generation and detection of AM, Side bands, generation and detection of side bands, comparison of various AM systems, FDM, Synchronous detection, exponential modulation: definitions and relationship between PM and FM frequency deviation, Bessells function, spectrum and transmission BW of FM signals, NBFM, WBFM, phaser diagram of FM signal, multi tone FM, Generation and detection of FM Non linear effects in FM systems, comparison of AM and FM systems, TDM.

UNIT III RADIO TRANSMITTER AND RECEIVERS

Different types of AM and FM transmitters and receivers, AM and FM standard broadcast transmitter and receivers, image rejection, mixer. Noise: classification and sources of noise, Noise calculations for single and cascaded stages, SNR, SNR in DSB, SSB, VSB, AM and FM systems, pre-emphasis and De-emphasis Sampling theorem, quantization, PCM, Companding intesymbol interface, eye patterns, Delta modulation, Adaptive delta modulation, DPCM, SIN performance of PCM and delta modulation, bandwidth of PCM and delta modulation.

UNIT IV DIGITAL MODULATION TECHNIQUES

ASK, BPSK, QPSK, M-ray PSK, DPSK, BFSK, M-ray FSK, Duobinary signalling baseband signal receiver, Probability of error, Optimum filter, Matched filter, Coherent and non-coherent detection, bit error rate, random signals, random variables and processes, cumnlative distribution function, probability density function, average value, variance, standard deviation moment and moment,] generating function, characteristics function, Tchebycheffs inequality, Binary, Poisson and Gaussian distributions, other distributions, central limit theorem.

UNIT V INFORMATION THEORY AND CODING

Unit of information, average information, joint and conditional entropy, mutual information, channel capacity efficiency, BBS and BEC, Shannon's Theorem, Shannon-Hartely theorem, bandwidth - SIN ratio trade-off. Coding separable codes, Prefix property, Coding efficiency, Source coding, Shannon - Fano code, Huffman code, Error connection codes, FEC and ARQ, Hamming distance, Minimum distance, Channel coding, block code, cyclic code, convolutional code.

Text Books:

1. S Haykin, Communication System, John Willey and Sons.
2. Taub Schilling, Principle of Communication, TMH.
3. B.P. Lathi, Modern Digital and Analog Communication System, Oxford Press.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

OBJECT-ORIENTED PROGRAMMING WITH JAVA			
Course Code:	**CS434**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I OBJECT-ORIENTED PROGRAMMING

Concept of object-oriented programming (OOP), benefits of OOP, application of OOP, Java history, Java features, Java streaming, Java and Internet, Java contribution to Internet: Java applets, security, portability; Java environment, Java library, Java program structure, Java program, Java Virtual Machine (JVM) architecture, Just In Time compiler (JIT), data type, variables and arrays, operators, control statements, object-oriented paradigms; abstraction, encapsulation, inheritance, polymorphism, Java class and OOP implementation

UNIT II DATA TYPE, OPERATORS AND CONTROL STATEMENT

Data types, Java key words, identifiers, constants, variables, declaration and scope of the variable, symbolic constant, type casting, arithmetic operator, relational operator, logical operator, assignment operator, increment and decrement operator, conditional operator, bitwise operator, ?: operator, arithmetic expressions, expressions, type conversions in expressions, mathematical functions, more data types: arrays, strings, vectors, wrappers classes, program control statements: decision making and branching: if, if….else, else….if, else if ladder, switch, decision making and looping: while, do….while, for.

UNIT III CLASSES, OBJECTS AND METHODS

Java class libraries, class fundamentals, object, methods, adding variables, add methods, creating objects, accessing class members, constructors, methods overloading, static members, nesting of methods, inheritance: extending a class, overriding methods, final variables and methods, final classes, finalizer methods, abstract methods and classes, visibility control, exception handling fundamental.

UNIT IV INTERFACES AND PACKAGES

Interfaces, extending interfaces, implementing interfaces, interfaces references, accessing interface variable, creating queue interface, variable in interfaces, packages, finding a packages and classpath, package and member access, Java API package, system package, naming conventions, creating package, accessing a package, adding a class to a package, hiding classes,

UNIT V MULTITHREADING AND APPLET PROGRAMMING

Multithreading programming: creating threads, thread class and runnable interface extending the thread class, stopping and blocking a thread, life cycle of a thread, thread methods, thread exceptions, thread priority, synchronization, thread communication using notify(), wait(), and notify all(), applet programming : applet basic, applets architecture, a complete applet skeleton, building applets code, applets life cycle, creating a executable applet, designing a web page, applets tag, passing parameters to applets, applets and HTML.

Test Books:

1. E. Balagurusawamy, Programming with JAVA, Tata McGraw Hill, 1998.

Reference Books:

2. Herbert Schildt, JAVA Beginner's guide, Tata McGraw Hill, 2007.
3. Deitel & Deitel, Java How to Program, Prentice-Hall, 1999.
4. Herbert Schildt, The Complete Reference JAVA 2, 7th Edition, Tata McGraw Hill, 2009.
5. Ken Arnold, James Gosling, The Java Programming Language, Addison-Wesley, 1996.
6. Peter Coffee, How to Program Java, Ziff-Davis Press, 1996.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

DATABASE MANAGEMENT SYSTEM			
Course Code:	**CS436/CS206**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I DATA BASE SYSTEM

Data base system vs. file system, view of data, data abstraction, instances and schemas, data models, ER model, relational model, database languages, DDL, DML, database access for applications programs, data base users and administrator, transaction management, data base system structure, storage manager, query processor, history of data base systems, data base design and ER diagrams, beyond ER design entities, attributes and entity sets, relationships and relationship sets, additional features of ER model, concept design with the ER model, and conceptual design for large enterprises.

UNIT II RELATIONAL MODEL

Introduction to the relational model, integrity constraint over relations, enforcing integrity constraints, querying relational data, and logical data base design, destroying /altering tables and views. relational algebra and calculus: relational algebra, selection and projection set operations, renaming, joins, division, relational calculus, tuple relational calculus, domain relational calculus, expressive power of algebra and calculus.

UNIT III BASIC SQL QUERY

Examples of basic SQL queries, nested queries, correlated nested queries set, comparison operators, aggregative operators, NULL values, comparison using null values, logical connectivity's, AND, OR and NOTR, impact on SQL constructs, outer joins, disallowing NULL values, complex integrity constraints in SQL triggers and active data bases.

UNIT IV SCHEMA REFINEMENT

Problems caused by redundancy, decompositions, problem related to decomposition, reasoning about FDS, FIRST, SECOND, THIRD normal form, BCNF, forth normal form, lossless join decomposition, dependency preserving decomposition, schema refinement in data base design, multi valued dependencies.

UNIT V OVERVIEW OF TRANSACTION MANAGEMENT

ACID properties, transactions and schedules, concurrent execution of transaction, lock based concurrency control, performance locking, and transaction support in SQL, crash recovery, concurrency control, Serializability and recoverability, lock management, lock conversions, dealing with dead locks, specialized locking techniques, concurrency without locking, crash recovery: ARIES, log, other recovery related structures, the write, ahead log protocol, check pointing, recovering from a system crash, media recovery, other approaches and interaction with concurrency control.

References Books:

1. Elmasri Navrate, Data Base Management System, Pearson Education, 2008.
2. Raghurama Krishnan, Johannes Gehrke, Data Base Management Systems, TMH, 3rd edition, 2008.
3. C. J. Date, Introduction to Database Systems, Pearson Education, 2009.
4. Silberschatz, Korth, Database System Concepts, McGraw hill, 5^{th} edition, 2005.
5. Rob, Coronel & Thomson, Database Systems Design: Implementation and Management, 2009.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

COMPUTER ORGANIZATION AND ARCHITECTURE			
Course Code:	**CS438/CS309**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I COMPUTER ARITHMETIC AND NUMBER SYSTEM

Number representation, number system, fixed and floating point number representation, arithmetic algorithms (addition, subtraction, booth multiplication).

UNIT II REGISTER TRANSFER AND MICROOPERATION

Register transfer language, bus and memory transfers, bus architecture, bus arbitration, arithmetic logic, shift microoperation, arithmetic logic shift unit, design of fast address.

UNIT III PROCESSOR DESIGN

Processor organization: general register organization, stack organization, addressing mode, instruction format, data transfer & manipulations, program control, reduced instruction set computer.

UNIT IV INPUT-OUTPUT ORGANIZATION

I/O Interface, synchronous and asynchronous data transfer, strobe, handshaking schemes, modes of transfer, interrupts & interrupt handling, direct memory access, input-output processor.

UNIT V MEMORY ORGANIZATION

Memory hierarchy, main memory (RAM and ROM Chips), organization of 2D and $2^{1/2}$D, auxiliary memory, cache memory, virtual memory, memory management hardware.

Text Books:

1. Patterson, Computer Organisation and Design, Elsevier Pub. 2009
2. John P Hays, “Computer Organization”, McGraw Hill

References Books:

3. William Stalling, “ Computer Organization”, PHI
4. Vravice,Hamacher & Zaky, “Computer Organization”, TMH
5. Mano,” Computer System Architecture”, PHI
6. Tannenbaum,” Structured Computer Organization’, PHI
7. P Pal Chaudhry, ‘Computer Organization & Design’, PHI

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SIGNAL AND SYSTEMS			
Course Code:	**EC444/EC205**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I LTI SYSTEMS

Continuous time and discrete time signals, even and odd signals, elementary continuous time and discrete time signals, classification of signals, causality; stability, time invariance, linearity, continuous time and discrete time LTI Systems, convolution integral and convolution sum, properties of LTI Systems, differential and difference equations, singularity functions.

UNIT II ANALYSIS OF PERIODIC SIGNALS

Fourier series representation of CTPS, convergence of FS, Properties of CTFS, Fourier series representation of DTFS, Fourier series and LTI Systems, filtering, RC low pass and high pass filters. recursive and non recursive Discrete Time filters, sampling theorem, sampling of continuous time signal with impulse train and zero order hold, reconstruction, aliasing, discrete-time processing of continuous time signals, digital differentiator, sampling of discrete time signals, decimation and Interpolation.

UNIT III ANALYSIS OF APERIODIC SIGNALS

Continuous Time Fourier Transform (CTFT), convergence of FT, properties of CTFT, Discrete Time Fourier Transform (DTFT), properties of DTFT, system characterized by Linear constant co-efficient differential equations, magnitude and phase spectrum, group delay, time domain and frequency domain aspects of ideal non-ideal filters, first order and second order continuous time and discrete time systems.

UNIT IV LAPLACE TRANSFORM

Laplace transform, region of convergence for Laplace transform, inverse Laplace transform, geometric evaluation of Fourier transform from pole zero plot, first order, second order and all pass systems. Properties of Laplace transform, analysis and characterization of LTI systems using the Laplace transform. Causality, stability, differential equations, Butterworth and Chebychev filters, unilateral Laplace transform, its properties and uses.

UNIT V Z-TRANSFORM

Basic principles of z-transform, z-transform definition, region of convergence, properties of ROC, properties of z-transform, Poles and Zeros, inverse z-transform using Contour integration, Residue Theorem, power series expansion and partial fraction expansion, relationship between z-transform and Fourier transform.

Reference Books:

1. Oppenheim Willsky and Nawab, Signals and Systems, PHI.
2. Simon Haykin, Signals and Systems, John Wiley.
3. Taub and Schilling, Principles of Communication Systems, TMH.
4. Dungan F R, Electronic Communication Systems, Thomas-Delmar.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

MICROELECTRONICS ENGINEERING			
Course Code:	**EC446/EC306**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I MOSFETS

Device Structure and physical operation, V-I Characteristics, MOSFET circuits at DC, biasing in MOS amplifier circuits, small signal operation and models, MOSFET as an amplifier and as a switch, biasing in MOS amplifier circuits, small signal operation, MOSFET internal capacitances and high frequency mode, CMOS digital logic inverter, detection type MOSFET.

UNIT II SINGLE STAGE IC AMPLIFIER

IC Design philosophy, comparison of MOSFET and BJT, current sources, current mirrors and current steering circuits, high frequency response, CS and CF amplifiers, CG and CB amplifiers, cascade amplifiers, CS and CE amplifiers with source (emitter) degeneration source and emitter followers, some useful transfer parings, current mirrors with improved performance.

UNIT III DIFFERENTIAL AND MULTISTAGE AMPLIFIERS

MOS differential pair, small signal operation of MOS differential pair, the BJT differences pair, other non-ideal characteristics and differential pair, differential amplifier with active loads, frequency response and differential amplifiers, multistage amplifier.

UNIT IV FEEDBACK AND OPERATIONAL AMPLIFIERS

General Feedback structure, properties of negative feedback, basic feedback topologies, loop gain, Stability problem, effect of feedback on amplifier poles, stability analysis by Bode plots, frequency compensation, two stage CMOS Op-amp, folded cascade CMOS op-amp, 741 op-amps, data converters, A-D and D-A converters.

UNIT V DIGITAL CMOS CIRCUITS

Overview, design and performance analysis of CMOS inverter, logic gate circuits, pass-transistor logic. dynamic logic circuits.

Text Book:

1. Microelectronic Circuits, Adel Sedra and K.C. Smith, 5th Edition, Oxford Uni. Press, 2004.
2. Microelectronic Circuit Design, Richard C. Jaeger and Blalock, 3rd Edition, TMH 2007

References Books:

3. Fundamentals of Microelectronics, Behzad Razavi, John Wiley, 2008

OBJECT-ORIENTED PROGRAMMING WITH JAVA LAB			
Course Code:	**CS484**	**Credits:**	**2**
No. of Practical (Hrs/Week):	**2**		
Total No. of Lab Sessions:	**15**	**End Sem. Exam Hours:**	**2**

Programs/Experiments List:

1. Write a separate Java Code to implement each of the following:
   Class, Command Line Argument, how to enter value through keyboard

2. Write a separate Java Code to implement each of the following data types:
   Variable, Constant, Arrays, Strings, Vectors, Wrappers Classes, Type Casting

3. Write a separate Java Code to implement each of the following operators:
   Arithmetic operator, Relational operator, Logical operator, Assignment operator, Increment & Decrement operator, Conditional operator, Bitwise operator, ?: operator

4. Write a separate Java Code to implement each of the following control statements:
   Decision statement, Loops statement and Branch statements

5. Write a separate Java Code to implement each of the following sorting:
   Bubble Sort, Selection Sort, Insertion Sort, Merge Sort

6. Write a separate Java Code to implement each of the following:
   Class, Object, Constructors, Method, Method Overloading and Method Overriding

7. Write a separate Java Code to implement each of the following:
   Final variable, final class, final method, abstract class, abstract method and concrete method

8. Write a separate Java Code to implement each of the following OOPs concepts:
   Abstraction, Polymorphism, Encapsulation, Inheritance
9. Write a separate Java Code to implement each of the following:
   Exception handling with Try, Catch, Throw, Throws, Finally
   Multiple catch statement with the following exceptions :
   ArithmeticException, ArrayOutOfBoundsException and ArrayStoreException

10. Write a separate Java Code to implement each of the following:
    Visibility Controls: Private, Public and Protected

11. Write a separate Java Code to implement each of the following:
    Interface, extending and implementing interface

12. Write a separate Java Code to implement each of the following:
    Multithreading: Create thread with thread class and runnable interface, thread priorities, synchronization

13. Write a separate Java Code to implement each of the following:

Packages : Create package A with following methods and import this package A into another Java program to show the result of methods of package A.

i) First method: Factorial number with the help of recursion.

ii) Second method: Fibonacci Series

iii) Third Method: Generate first 10 prime numbers and show the sum of first 10 prime numbers.

14. Write Java Code to generate the following output on applet with the help of two dimensional array and show the result with the help of HTML file.

7 14 21 28 35 42 49 56 63 70	Sum = 385
5 10 15 20 25 30 35 40 45 50	Sum = 275
3 6 9 12 15 18 21 24 27 30	Sum = 165

15. Write a Java Code to design the following web page with the help of applet and HTML.

School of Information and Communication Technology
GAUTAM BUDDHA UNIVERSITY
GREATER NOIDA

- **Student Name:**
- **Enrollment Number:**
- **Programme Name:**
- **Semester:**
- **Course Name:**
- **E-Mail ID:**
- **Mobile Number:**
- **Blood Group:**

DESIGN LAB			
Course Code:	**CS480**	**Credits:**	**2**
No. of Practical (Hrs/Week):	**2**		
Total No. of Lab Sessions:	**15**	**End Sem. Exam Hours:**	**2**

Programs/Experiments List:

1. Characterization of JFET.
2. Characterization of MOSFET
3. Design of Inverter Using MOSFET.
4. Design of an Inverting Amplifier using OP-AMP.
5. Design of noninverting Amplifier using OP-Amp.
6. Design of an integrater using Op-Amp.
7. Design of Astable Multivibrator using 555 Timer.
8. Design of Schmitt trigger using 555 timer
9. Design an 8-bit A/D Converter.
10. Design an 8-bit D/A Converter.
11. Verify the truth table of AND, OR and NOT gates.
12. Verify the truth table of NAND , NOR and EX-OR Gate.
13. Design a combinational circuit to realize the function f(ABC)=A(B+C) using NAND gates only.
14. Design a half adder using NOR gates only
15. Design full adder
16. Design a given size of Mux using gates.
17. Verify RS and JK flip flops
18. Verify D and T Flip flops
19. Design a up/down 3-bit counter
20. Design a 3 bit shift register.

SOFTWARE ENGINEERING

(SEMESTER - III)

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

DATA STRUCTURES AND ALGORITHM DESIGN			
Course Code:	**CS531**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION

Basic Terminology, Elementary Data Organization, Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big-Oh, Time-Space trade-off.

Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations.

Linked lists: Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal

UNIT II STACKS AND QUEUE:

Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Queue, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Dequeue and Priority Queue.

UNIT III TREES

Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm, Convex hull , Searching. Greedy methods. Binary Search Trees(BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees, Red-Black trees, Binomial Heaps, Fibonacci Heaps, Fast Fourier Transform, String Matching, Theory of NP-completeness, Approximation algorithms and Randomized algorithms,

UNIT IV GRAPH

Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Graph Traversal : Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.

UNIT V SEARCHING AND SORTING

Searching : Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting. Dynamic programming with examples such as Kanpsack, All pair shortest paths – Warshal's and Floyd's algorithms, Resource allocation problem.

Text books:

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein "Data Structures Using C and C++" , PHI
2. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.

Reference Books:

3. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publication

4. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill
5. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education
6. Lipschutz, "Data Structures" Schaum's Outline Series, TMH

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

PRINCIPLES OF COMPILER DESIGN			
Course Code:	**CS533**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO COMPILER

Introduction to compiler, phases and passes, bootstrapping, finite state machines and regular expressions and their applications to lexical analysis, optimization of DFA-based pattern matchers implementation of lexical analyzers, lexical-analyzer generator, LEX-compiler, formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC, syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

UNIT II PARSING TECHNIQUE

Parsers, shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic construction of efficient parsers: LR parsers, the canonical collection of LR(0) items, constructing SLR parsing tables, constructing canonical LR parsing tables, constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III SYNTAX-DIRECTED TRANSLATION

Syntax-directed translation schemes, implementation of syntax directed translators, intermediate code, postfix notation, parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser, more about translation: array references in arithmetic expressions, procedures call, declarations and case statements.

UNIT IV SYMBOL TABLES

Data structure for symbols tables, representing scope information, run-time administration: implementation of simple stack allocation scheme, storage allocation in block structured language, Error detection & recovery: lexical phase errors, syntactic phase errors, semantic errors.

UNIT V CODE GENERATION

Design issues, the target language. addresses in the target code, basic blocks and flow graphs, optimization of basic blocks, code generator. code optimization: machine-independent optimizations, loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, global data-flow analysis

Reference Books:

1. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education
2. V Raghvan, " Principles of Compiler Design", TMH
3. Kenneth Louden," Compiler Construction", Cengage Learning.
4. Charles Fischer and Ricard LeBlanc," Crafting a Compiler with C", Pearson

DATA COMMUNICATION			
Course Code:	**EC553**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO DATA COMMUNICATION AND NETWORKS

Data communication, networks, physical structures; different topologies, categories of networks: LAN, MAN, WAN, interconnection of networks, Internet, protocols and standards, standards organizations. network models, layered tasks, OSI model, different layers in OSI model, TCP/IP protocol suite; different layers, addressing, physical, logical, port and specific addresses, analog and digital, digital signals-Bit length, digital signal as a composite analog signal, transmission of digital signals, data rate limits-noiseless channel, noisy channel.

UNIT II PHYSICAL LAYER

Digital-to-digital conversion, line coding, line coding scheme, block coding, scrambling, multiplexing, frequency division, wavelength division, synchronous time division, statistical time division multiplexing, circuit switched networks, three phases, efficiency, delay, datagram networks, routing table, efficiency, delay, datagram networks in the internet. virtual circuit networks, addressing, three phases, efficiency, delay, circuit switched technology in WANs, structure of circuit and packet switches, dial-up modems, digital subscriber line - ADSL, ADSL Lite, HDSL, SDSL, VDSL, cable TV for data transfer- bandwidth, sharing, CM and CMTS, data transmission schemes.

UNIT III DATA LINK LAYER

Introduction, types of errors, redundancy, detection vs correction, forward error correction vs retransmission, modular arithmetic, block coding, error detection, error correction, hamming distance, minimum hamming distance, linear block codes, cyclic codes, cyclic redundancy check, hardware implementation, polynomials, cyclic code analysis, advantages, checksum, framing, fixed and variable, size, flow and error control, protocols, noiseless channels, simplest and stop-and-wait protocols, noisy channels, stop-and-wait automatic repeat request, go-back-n automatic repeat request, selective repeat automatic repeat request.

UNIT IV MEDIUM ACCESS

Random Access- ALOHA, Carrier Sense Multiple Access (CSMA), Carrier Sense Multiple Access with Collision Detection (CSMA/CD), Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), controlled access-reservation, polling, token passing, channelization, Frequency Division Multiple Access (FDMA), Time-Division Multiple Access (TDMA), Code-Division Multiple Access (CDMA), IEEE standards, standard ethernet, changes in the standard, fast ethernet, gigabit ethernet, IEEE 802.11-Architecture, MAC sub layer, addressing mechanism, physical layer, Bluetooth, architecture, radio Layer, baseband Layer, L2CAP.

UNIT V CONNECTING LANS

Connecting devices, passive hubs, repeaters, active hubs, bridges, two-layer switches, three-layer switches, gateway, backbone networks-bus, star, connecting remote LANs, Virtual LANs -Membership, configuration, communication between switches, network layer, logical addressing - IPv4Addresses- address space, notation, classful addressing, classless addressing, Network Address Translation (NAT). IPv6 addresses - structure and address space, internetworking - need for network layer, internet as a datagram network, internet as a connectionless network, IPv4, datagram, fragmentation, checksum, options, IPv6 advantages, packet format, extension headers, transition from IPv4 to IPv6, address mapping- logical to physical address, physical to logical address, routing, delivery forwarding techniques and processes, routing table, unicast routing protocols, optimization, inter domain, intra domain, distance vector, link state and path vector,

routing, Multicast routing protocol, unicast, multicast and broadcast, applications, multicast routing and routing protocols.

Text Books:

1. B. A. Forouzan and Sophia Chung Fegan: Data Communications and Networking, 4th Ed, TMH.
2. W. Tomasi: Introduction to Data Communications and Networking, Pearson Education.
3. A. S. Tanenbaum: Computer Networks, Pearson Education.
4. W. Stalling: Data and Computer Communication, Pearson Education.

PRINCIPLE OF VLSI DESIGN			
Course Code:	**EC555/EC465**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I MOS TRANSISTOR THEORY AND PROCESS TECHNOLOGY

NMOS and PMOS transistors, threshold volt age- Body effect- Design equations- second order effects. MOS models and small signal AC characteristics, basic CMOS technology.

UNIT II INVERTERS AND LOGIC GATES

NMOS and CMOS inverters, stick diagram, inverter ratio, DC and transient characteristics , switching times, Super buffers, driving large capacitance loads, CMOS logic structures, transmission gates, static CMOS design, dynamic CMOS design.

UNIT III CIRCUIT CHARACTERIZATION AND PERFORMANCE ESTIMATION

Resistance estimation, capacitance estimation, inductance, switching characteristics, transistor sizing, power dissipation and design margining, charge sharing, scaling.

UNIT IV VLSI SYSTEM COMPONENTS CIRCUITS & SYSTEM LEVEL PHYSICAL DESIGN

Multiplexers, decoders, comparators, priority encoders, shift registers, arithmetic circuits, ripple carry adders, carry look ahead adders, high-speed adders, multipliers, physical design, delay modeling ,cross talk, floor planning, power distribution, clock distribution.

UNIT V VERILOG HARDWARE DESCRIPTION LANGUAGE

Overview of digital design with Verilog HDL, hierarchical modeling concepts, modules and port definitions, gate level modelling, data flow modeling, behavioral modeling, task & functions, test bench.

Text Books:

1. Neil H.E. Weste and Kamran Eshraghian, Principles of CMOS VLSI Design, Pearson Education ASIA, 2nd edition, 2000.
2. Pucknell, “Basic VLSI Design”, Prentice Hall of India Publication, 1995.

Reference Books:

3. John P.Uyemura “Introduction to VLSI Circuits and Systems”, John Wiley & Sons, Inc., 2002.
4. Samir Palnitkar, “Verilog HDL”, Pearson Education, 2nd Edition, 2004.
5. Eugene D.Fabricius, Introduction to VLSI Design McGraw Hill International Editions,1990.
6. J.Bhasker, B.S.Publications, “A Verilog HDL Primer”, 2nd Edition, 2001.
7. Wayne Wolf “Modern VLSI Design System on chip. Pearson Education.2002.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

CELLULAR MOBILE COMMUNICATION			
Course Code:	**EC557**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Introduction to cellular mobile system, basic cellular system, performance criteria, uniqueness of mobile radio environment, operation of cellular systems, planning and cellular system, elements of cellular radio system design: general description of problem, concept of frequency channels, co-channel interface reduction factor, hand off, cell splitting, consideration of the components of cellular systems.

UNIT II

Interface: introduction to co-channel interface, real time co-channel interface, co-channel measurement, design of antenna system, antenna parameter and their effects, diversity receiver non co-channel interface different types.

UNIT III

Cell coverage for signal and traffic: general introduction, obtaining the mobile point-to-point mode propagation over water or flat open area, foliage loss, propagation near in distance, long distance propagation, point-to-point prediction model-characteristics, cell site, antenna heights and signal coverage cells, Mobile-to-mobile propagation.

UNIT IV

Cell site antennas and mobile antennas: antennas at cell site, mobile antennas, frequency management and channel assignment: frequency management, fixed channels assignment, non-fixed channel assignment, traffic and channel assignment.

UNIT V

Digital cellular system: GSM, architecture, layer modeling, transmission, GSM channels, multiple process, CDMA, terms, power limits & control modulation characteristics, call processing, hand off.

Text Books:

1. Lee, Cellular and Mobile Communication, McGraw Hill.
2. Faher Kamilo., Wireless Digital Communication, PHI.

MICROPROCESSORS AND INTERFACING			
Course Code:	**EC559/EC304**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Introduction to 8085 microprocessor: pin diagram, architecture, programming model, instruction set, and classification of instruction set, instruction and data format, timing diagram of instructions, basic concept of programming, addressing modes of 8085 microprocessors.

UNIT II

Microprocessor 8086 architecture, BIU and EU, registers, pin diagram, memory addressing, clock generator 8284, buffers and latches, maximum and minimum modes.

UNIT III

Addressing Modes, Instruction set of 8086, assembly language programming, assemblers, procedures, macros, interrupts, 8086 based multiprocessor systems

UNIT IV

Interfacing Chips- IC 8155 (Static Ram with I/O Ports and Timer), 8755 (EPROM with I/O Ports), 8251A (USART), 8255A (Programmable Peripheral Interface), 8253/8254 (Programmable Interval Timer/Counter), 8257 (DMA Controller), 8259A (Programmable Interrupt Controller).

UNIT V

The 8051 architecture Microprocessor and Microcontroller, Comparison of microprocessors and microcontrollers Microcontroller survey, microcontrollers of different word length, make and features, selection criteria for microcontroller ,8051 microcontroller hardware, I/O pins and internal architecture internal RAM, ROM organization, I/O port circuits ,connecting external memory, addressing modes, instruction set and assembly language programming.

Text Books:

1. A. K. Ray : Advanced Microprocessors and Interfacing, 2nd edition,TMH
2. Mazidi and Mazidi: The 8051 Microcontroller and Embedded Systems, Pearson Education

References:

3. B. B. Brey: The Intel Microprocessors, Architecture, Programming and Interfacing, Pearson Education.
4. Liu Gibson: Microcomputer Systems: The 8086/8088 Family- Architecture, Programming and Design, PHI
5. D. V. Hall: Microprocessors and Interfacing, TMH.
6. Ayala Kenneth:- The 8051 microcontroller, Third Edition, Cengage Learning
7. A. V. Deshmukh: Microcontroller (Theory and Application), TMH.
8. Raj Kamal: Embedded Systems- Architecture, Programming and Design, TMH, New Delhi.
9. R S Gaonkar, Microprocessor, Architecture, Programming, and Applications with the 8085, Penram International Publication, 5/e
10. P.K. Ghosh and P. R. Sridhar, 0000 to 8085 Introduction to microprocessor for Engineers and Scientists, PHI, 2/e

DATA STRUCTURE AND ALGORITHM DESIGN LAB			
Course Code:	**CS583**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**		
Total No. of Lectures:	**15**	**End Sem Exam Hours:**	**2**

Programs/Experiments List:

1. Write a program that implements tower of hanoii.

2. Write a program that implements insertion sort, Selection Sort.

3. Write a program that implements heap sort.

4. Write a program that implements quick sort.

5. Write a program that implements merge sort.

6. Write a program that implements binary search.

7. Write a program that implements Prim's algorithm.

8. Write a program that implements Kruskal's algorithm.

9. Write a program that implements make a change using greedy.

10. Write a program that implements knapsack using greedy.

11. Write a program that implements Dijkstra's algorithm.

12. Write a program that implements Longest Common Subsequence.

13. Write a program that implements N-queen Problem.

14. Write a program that implements knapsack using backtracking

15. Write a program that implements make a change using dynamic.

16. Write a program that implements All pair shortest path problem

MICROPROCESSOR AND INTERFACING LAB			
Course Code:	**EC587/EC384**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**		
Total No. of Lectures:	**15**	**End Sem Exam Hours:**	**2**

Programs/Experiments List:

1. Addition of two 8-bit numbers, result 8-bit.
2. Addition of two 8-bit numbers, result 16-bit.
3. Subtraction of two 8-bit numbers.
4. Addition of two 16-bit numbers.
5. Multiplication of two 8-bit numbers.
6. Division of two 8-bit numbers.
7. 2's Compliment of a 8-bit number.
8. Arrange the array in ascending order.
9. Arrange the array in descending order.
10. Moving the block of data from one memory location to another memory location.
11. Largest number in an array.
12. Smallest number in an array.
13. BCD to HEX conversion.
14. HEX to BCD conversion.
15. HEX to ASCII conversion.
16. ASCII to HEX conversion.
17. Square of a number using lookup table method.
18. Interfacing of 8255.
19. Interfacing of 8253/8354.
20. Interfacing of 8237/8257.
21. Interfacing of 8259.
22. Interfacing of 8251.

SOFTWARE ENGINEERING

(SEMESTER - IV)

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE ARCHITECTURE AND DESIGN			
Course Code:	**CS532/CS402**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE ARCHITECTURE

Foundations of software architecture, software life cycle architectural styles, quality attributes, architectural patterns, pipes and filters, layered systems, repositories, frameworks, patterns, methodologies, processes and process control, functional and non-functional properties of software architectures, heterogeneous architectures, virtual machine architecture, data flow architecture, service oriented architecture.

UNIT II DESIGN FUNDAMENTALS AND METHODOLOGIES

Nature of design process: objectives, building modules, constructs, design qualities, assessing the design, design viewpoints for software, design strategies: top down and bottom up, organizational methods and design, Jackson structural programming, Jackson system development, models for software architecture

UNIT III SOFTWARE ARCHITECTURE DESIGN

Architectural design and mapping, architecture design patterns, module architecture view,styles of the module view type, execution architecture view, code architecture view, component-and-connector viewtype, styles of component-and-connector viewtype, allocation viewtype and styles, object-oriented architecture, user interface architecture, quantified design space, formalizing architectural description language, first class connectors, tools for architectural design: Unicon, A4; exploiting style in architectural design, architectural interconnection.

UNIT IV INTERACTION ORIENTED SOFTWARE ARCHITECTURE and Design

Model-View-Controller (MVC), Presentation-Abstraction-Control (PAC) architecture, distributed architecture: client server architecture, multi-tire, service-oriented architecture (SOA). Design principles, traditional approach to design, Structured Analysis Design Technique (SADT), Structures System Analysis and Design Method (SSADM), user interface design; human factor, human computer interaction, interface design guide lines, standards, object-oriented analysis and design.

UNIT V PATTERNS

Design patterns, creational patterns, access control patterns, service variation patterns, service extension patterns, archetypes patterns, model driven architecture with archetype patterns, literate modeling, Customer Relationship Management (CRM) archetype pattern, product archetype pattern, quantity archetype pattern, rule archetype pattern, layering, organizing domain logic, mapping to relational databases, web presentation, domain logic patterns, data source architectural patterns, object-relational behavioral patterns, object-relational structural patterns, object-relational metadata mapping patterns, web presentation patterns, distribution patterns, offline concurrency patterns.

Text Books:

1. Software Architecture Perspectives on an Emerging Discipline, M. Shaw Prentice-Hall, 1996.
2. Software Architecture Design: Methodology and Styles, Lixin Tao, Xiang Fu and Kai Qian, Stipes Publishing L.L.C., 2006.
3. Software Architecture in Practice, Len Bass, Paul Clements, Rick Kazman, Pearson Education Asia, 2003.

References Books:

4. Software Design, David Budgen, Addison-Wesley, 1994.
5. Software Engineering, Pressman R.S, McGraw Hill Inc., 1996.

6. Structured System Analysis and Design methods Application and Context, Ed Downs, Peter Clare, Jan Coe, Prentice Hall, 1998.
7. Design Patterns for Object-Oriented Software Development, Wolfgang Pree, Addison- Wesley, 1995.
8. Software Architecture Resource website. http://www2.umassd.edu/SECenter/SAResources.html.
9. Essential Software Architecture, Ian Gorton Springer, 2006.
10. Pattern-Oriented Software Architecture, Frank Buschmann, Hans Rohnert, Kevin Henney, Douglas C. Schmidt, Wiley, 2004.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

OPEN SOURCE SOFTWARE SYSTEM			
Course Code:	**CS534/CS404**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I OPEN SOURCE SOFTWARE

Open Source Software (OSS), history, philosophy and ethics of open source software, Pernes' principle, open source software development methodology, open source vs. closed source, open source software vs. free software, open source software vs. source available, Windows and Linux, open source development environment, methods and models, standards, open source standards, benefits of open standards, standard setting organizations and processes, project management via open source and open standard, OSS in e-government.

UNIT II OPEN SOURCE TECHNOLOGY

Open source technology and platform, Operating system: Linux, Berkeley Software Distribution; web server: Apache; communication servers: send mail, jabber; application and messaging server: JBoss, Zope, Zend, concurrent version control system, sub-version revision control system, and distributed revision control system.

UNIT III OPEN SOURCE LANGUAGES

Ruby, Ruby and object-orientation, data, expressions and flow control, class, object and modules, project and libraries, developing a basic Ruby application, PHP, configure environment, PHP basic, functions, arrays, object-oriented PHP, error and exception handling, MYSQL, PostgreSQL.

UNIT IV OPEN SOURCE SOFTWARE APPLICATIONS AND FRAMEWORK

Open source desktop applications: graphical desktop, web browsers, office suite; Wiki software, LAMP application, web server and database server application, OSS management tools: taskjuggler, dotProject.net, rapid web application development framework: Ruby on Rail, Model-View-Controller model, Don't Repeat Yourself principle.

UNIT V OPEN SOURCE IN THE ENTERPRISE

Nature of open source, leadership in open source software life cycle, comparison in the risks of commercial and open source software, measuring the maturity of open source, designing an open source strategy, open source licenses, comparison of open source licenses, open source empowerment.

Text Books:

1. Paul Kavanagh, Open Source Software: Implementation and Management, Digital Press, 2004.
2. W. Jason Gilmore, Beginning PHP and MySQL, Apress, 2010.
3. Timothy Fisher, Ruby on Rail, Apress, 2009.

Reference Books:

4. Dan Woods, Open Source for the Enterprise: Managing Risks, Reaping Rewards, O'Reilly, 2005.
5. James Lee, Brent Ware,Open Source Web Development with LAMP, Pearson Education, 2008.
6. Steven Weber, The Success of Open Source, Harvard University Press, 2004.
7. Peter Cooper, Beginning Ruby, Apress, 2007.

OPEN SOURCE SOFTWARE SYSTEM LAB			
Course Code:	**CS582**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**		
Total No. of Lectures:	**15**	**End Sem Exam Hours:**	**2**

Programs/Experiments List:

1. Study, analysis and compare various open source software according to the open source standards.
2. Download any open source software application written in JAVA with source code and modify the source code to add extra functionality.
3. Develop a small application with the help of database (MySQL and PostGre SQL) for any domain in JAVA technology.
4. Learn the working and installation of the following tools:
   Open3, Enhydra, Jboss, Zope, Zend.
5. Learn the working and installation process of the following Operating Systems and deliver a presentation: Linux, Fedora, ubuntu, CentOS and Boss.
6. Rapid web application development framework: Ruby on Rail.
7. Learn the working and installation process of the following Open Source Software Management Tools : Taskjuggler and dotProject.net
8. Introduction to MySQL, an exercise on data types in MySQL & Data Definition Language.
9. Exercise on Data Manipulation Language and transaction control commands using MySQL.
10. Exercise on Types of Data Constraints using MySQL.
11. Exercise on Joins (single-table or multi-table) and using normalization using MySQL.
12. Exercise on group-by clause and date arithmetic using MySQL.
13. Exercise on different functions (aggregate, math and string) using MySQL.
14. Introduction to PL/SQL, Control Structures, Procedures and Functions, View using MySQL.
15. Exercise on different types of sub queries using MySQL.

SOFTWARE ENGINEERING

(ELECTIVES - 1 & 2)

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

COMPONENT-BASED SOFTWARE ENGINEERING			
Course Code:	**CS542/CS442**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO CBSE

Component-Based Software Engineering (CBSE), CBSE vs. Object-Oriented Software Engineering, CBSE methodology, CBSE processes, domain engineering, component engineering, component-based software life cycle, component vs. object, CBSE project management, measurement and metrics for CBSE, challenge CBSE, advantages and disadvantages of CBSE, economics of CBSE.

UNIT II COMPONENT-ORIENTED PROGRAMMING

Component-oriented programming, object-oriented programming to component-oriented programming, component-oriented programming vs. object-oriented programming, principle and infrastructure of component-oriented programming, component-oriented programming with Java Bean.

UNIT III COMPONENT AND COMPONENT MODEL

Component, component technology, software component, specification of software component, component architecture, component framework, component interface, component abstraction, component services, components model, component selection, component adaptability, component certification, component composition, component and interface modeling, domain modeling, patterns and frameworks.

UNIT IV COMPONENT-BASED DESIGN AND REUSE

Principles of component design and reuse, design prototyping, design production, design refactoring, design documentation, component-based software reuse, reusable component, component-based reuse metrics.

UNIT V COMPONENT TECHNOLOGIES

Component technologies: Component Object Model (COM), Distributed Component Object Model (DCOM), Common Object Requesting Broker Architecture (CORBA), Enterprise Java Beans (EJB) and Remote Method Invocation (RMI).

Text Books:

1. George T. Heineman, William T. Councill, Component-Based Software Engineering: Putting the Pieces Together, Addision Wesley, 2001.
2. Andy Ju An Wang, Kai Qian, Component-Oriented Programming, Willey Interscience, 2005

Reference Books:

3. Clemens Szyperski, Component Software: Beyond Object-Oriented Programming, Addison Wesley, 1997.
4. Alan W. Brown, Component-Based Software Engineering, Wiley-IEEE Computer Society, 1996.
5. Sudha Sadasivam, Component-Based Technology, G. Willy, 2008.
6. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
7. N. S. Gill, Software Engineering: Software Reliability, Testing and Quality Assurance, Khanna Book Publishing Co. (P) Ltd., New Delhi, 2002.

ASPECT-ORIENTED SOFTWARE ENGINEERING			
Course Code:	**CS544/CS444**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I ASPECT-ORIENTED SOFTWARE ENGINEERING

Software engineering with aspects, aspect-oriented software evolution, aspects, aspect vs. object, aspect vs. component, join points and pointcuts, separation of concerns, crosscutting concerns, problems caused by scattering and tangling, system development, system maintenance and evolution, adoption risks.

UNIT II ASPECT-ORIENTED PROGRAMMING

Concepts of AOP, inter-type declarations, implementation, comparison to other programming paradigms, nature of aspect-orientation, concepts and terminology, Join Point Model, AspectJ Point Model, pointcut designators, advice bodies, inter-type declarations, aspect weaving, comparison with object-oriented programming.

UNIT III ASPECT-ORIENTED REQUIREMENT ENGINEERING

Aspect-oriented requirements engineering and process, aspect-oriented requirements notations, aspect-oriented requirements tool support, adoption and integration of aspect-oriented requirements engineering, and assessment/evaluation of aspect-oriented requirements.

UNIT IV ASPECT-ORIENTED SOFTWARE ARCHITECTURE

Aspect-oriented software architecture and process, aspect-oriented architecture notations, aspect-oriented architecture tool support, adoption and integration of aspect-oriented architecture, and assessment/evaluation of aspect-oriented architecture.

UNIT V ASPECT-ORIENTED MODELING AND DESIGN

Aspect-Oriented Modeling, AOM approach, aspect model, aspect-oriented design, aspect-oriented design process, aspect-oriented design notations, aspect-oriented design tool support, adoption and integration of aspect-oriented design, and assessment/evaluation of aspect-oriented design, AspectJ, Aspect Werkz, Hyper/J, Java Aspect Component.

Reference Books:

1. Aspect-Oriented Software Development, Robert E. Filman, Tzilla Elrad, Siobhán Clarke, Mehmet Aksit, Addison-Wesley Professional, 2004.
2. Aspect-Oriented Software Development with Use Cases, Ivar Jacobson, Addison-Wesley Object Technology Series, 2005.
3. Aspect-Oriented Analysis and Design: The Theme Approach, Siobhán Clarke, Addison-Wesley Object Technology Series, 2005.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE RE-ENGINEERING			
Course Code:	**CS546/CS446**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO RE-ENGINEERING

Re-engineering concept and approaches, growing problems and maintaining software, redevelopment vs reengineering, reengineering process and methods, Reengineering economics, BPR process, Lehman's law, Pitfalls of reengineering, Technology for reengineering.

UNIT II REVERSE ENGINEERING

Function abstraction, data abstraction, Process abstraction, levels of reverse engineering: re-documentation, design recovery, specification recovery, conditions for reverse engineering, supporting techniques: forward engineering, restructuring, re-engineering, benefits of reverse engineering.

UNIT III SOURCE CODE TRANSLATION AND DATA REENGINEERING

Need of source code translation: hardware platform update, organizational policy change, lack of software support, understandable software code, testing and maintainable, detection of duplicate code. Data reengineering and Migration, Documentation.

UNIT IV HYBRID RE-ENGINEERING

Hybrid re-engineering tracks: translation of existing code, Commercial of the Shelf (COTS), custom code, hybrid re-engineering approach, risks of hybrid re-engineering: schedule, functionality, cost, quality, interface and interoperability, and benefits of hybrid re-engineering.

UNIT V SOFTWARE RE-ENGINEERING PATTERNS AND TECHNIQUES

Software re-engineering patterns, patterns based software re-engineering, object-oriented re-engineering patterns & technique, design patterns, testing patterns, software re-engineering techniques: restructuring, refactoring and data re-engineering, forward re-engineering, Clean room approach, tools support for re-engineering.

Reference Books:

1. Software Re-engineering, Robert S. Arnold, IEEE Computer Society Press, 1993.
2. Reversing: Secrets of Reverse Engineering, Eldad Eilam, John Wiley and Sons, 2005.
3. Software Engineering, James F. Peters, Witold Pedrycz, Willey, 2008.
4. Object-Oriented Reengineering Patterns, Serge Demeyer, Morgan Kaufmann, 2003.
5. Introduction to Software Engineering, Ronald Leach, CRC, 1999.
6. Component-Based Software Engineering: Alan W. Brown, Wiley-IEEE Computer Society, 1996.
7. Software Engineering: A Practitioner's Approach, Pressman, Roger, McGraw Hill, 1997.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE REUSABILITY			
Course Code:	**CS548/CS448**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I REUSABILITY

Motivation for reuse, reuse data element, reuse requirements, reuse activities, reuse design, reuse source code, reuse interface, reuse development plan, reuse test plan, reuse test case, reuse driven organizations, design patterns, generators based reuse, application framework, managing a reuse project, characteristics of reuse project, adapting a project to reuse, reuse tools, cost effective techniques for reuse, composition-based and generation-based reuse techniques.

UNIT II APPLICATIONS AND COMPONENT SYSTEM

Reuse object-oriented software engineering model, component applications, facades control access to component systems, facades and component systems, components specialization for reuse, variability and its mechanism, reuse of variable components to build application systems.

UNIT III OBJECT COMPONENTS

Object models, reusing analysis and design components, expressing variability in object model components, subsystem components group related classes, reusable design and implementation, components packaging and documenting object components. Reuse development processes, develop for reuse, develop with reuse, testing of reusable components, object-oriented components techniques and life cycles, object-oriented development for reuse, detailed design for reuse, implementation for reuse, Verification and Validation.

UNIT IV COMOPONENT SYSTEM ENGINEERING

Building flexible component systems, analyzing requirements focusing on variability, performing robustness analysis, designing and testing the component system, packaging of component system for reuse, principles of component design and reuse, design prototyping, Component-Based Software Reuse (CBSR), component develop for reuse, develop with reuse, testing of reusable component, reuse metrics, challenges in CBSR.

UNIT V APPLICATION SYSTEM ENGINEERING (ASE)

Application system reuse, building application systems from reusable components, analyzing requirements, performing robustness analysis for flexible application systems, designing, implementing, testing the application system, packaging application system.

Reference Books:

1. Software Reuse: Architecture, Process and Organization for Business Success, Ivar Jacobson, Martin Griss, Patrik Jonsson, Pearson Education, 2009.
2. Software Reuse: A Hoilstic Approach, Even-Andre Karisson, John Wiley and Sons, 1996.
3. Software Reuse Techniques: Additional Reuse to the Systems Development Process, Karma McClure, Prentice Hall, 1997.
4. Software Reusability, Wilhelm Schafer, Ruben Prieto-Diaz, Masao Matsumoto, Prentice Hall, 1993.
5. Component-Based Software Engineering: Alan W. Brown, Wiley-IEEE Computer Society, 1996.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

WEB-BASED SOFTWARE ENGINEERING			
Course Code:	**CS550/CS450**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Introduction, analysis, architectural design, design patterns, formulation, interface design, navigation, design, project, management, quality, attributes, structures, testing, WebApp, attributes, WebApp categories, WebE process, WebE team.

UNIT II

Attributes of web based applications, quality attributes, technologies, web application quality, the web process, framework for the web, web process model

UNIT III

Formulating and analyzing web based system, design for web based applications, testing web based applications, management issue, project management.

UNIT IV

Reviews general architectures for web application and technology-aware application designs, discusses the concepts and techniques for engineering and evaluating user interfaces appropriate for a web application's intended audience.

UNIT V

Explores the interaction between users and the application's user interface, special attention will be paid to web technologies and standards available for audiences with special needs.

Related research papers reading as suggested by subject Teacher and their analysis.

Reference Books:

1. Web Engineering: A Practitioner's Approach by Pressman and Lowe which considers the Web engineering process in its entirety.
2. Web Engineering: Principles and Techniques [Paperback] by Woojong Suh
3. Roger Pressman.S., " Software Engineering : A Practitioner's Approach ", (3rd Edition), McGraw Hill,

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE AGENTS			
Course Code:	**CS552/CS452**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE AGENTS PARADIAGM

Software agent, history, theoretical foundations for software agents, agent programming, agent programming paradigms, agent vs. object, aglet, mobile agents, agent frameworks, agent reasoning, agent applications.

UNIT II AGENT TYPOLOGY

Software agents: collaborative agents, interface agents, mobile agents, information agents, reactive agents, hybrid agents, heterogeneous agent system, smart agents.

UNIT III MULTIAGENT SYSTEMS

Multiagent system, interaction between agents, reactive agents, cognitive agents, interaction protocols, agent coordination, agent negotiation, agent cooperation, agent organization, self- interested agents in e-commerce applications.

UNIT IV INTELLIGENT SOFTWARE AGENTS

Design and implementation of intelligent agents: reactive, deliberative, planning, interface agents, agent communication languages, agent knowledge representation, agent adaptability, mobile agent applications, languages & tools for design, implementation of intelligent agents.

UNIT V AGENTS AND SECURITY

Agent security issues, mobile agents security, protecting agents against malicious hosts, untrusted agent, black box security, authentication for agents, security issues for aglets.

Reference Books:

1. Constructing Intelligent Agents with JAVA, Bigus & Bigus, Wiley, 1997.
2. Software Agents, Bradshaw, MIT Press, 2000.
3. Artificial Intelligence: A Modern Approach, von Stuart J. Russell, Peter Norvig, Prentice Hall, 1994.
4. Intelligent Software Agents, Richard Murch, Tony Johnson, Prentice Hall, 2000.

SOFTWARE ENGINEERING

(SEMESTER - III)

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE TESTING			
Course Code:	**CS631/CS501**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE TESTING

Essentials of software testing, testing methodology, testing and debugging, software and hardware testing, verification and validation, correctness vs. reliability, challenges in software testing, software testing life cycle (STLC), flow graphs and path testing, transaction flow testing, data flow testing, database testing, web-based testing.

UNIT II SOFTWARE TESTING TECHNIQUES

testing levels: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing; thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, performance testing, ad hoc testing, smoke testing,exhaustive testing, structural testing, mutation testing; Testing Maturity Model (TMM), verification process, defect tracking, severity and priority, defects, fault, failure, bug, bug life cycle, bug report and bug reporting tools.

UNIT III TEST METRICS AND MEASUREMENT

Purpose of test plan, test plan design, test script, test cases, test management, test case specification, executing test cases, test result analysis. Metrics and measurement, project metrics: effort variance , schedule variance, effort distribution across phases, progress metrics: test defect metrics, development defect metrics, productivity metrics, defect density, defect leakage ratio, Residual Defect Density (RDD), test phase effectiveness, test reports.

UNIT IV SOFTWARE VERIFICATION AND VALIDATION

Verification, methods of verification, types of review on the basis of stage, reviews in testing life cycle, coverage in verification, concerns of verification, validation, coverage in validation, management of Verification and Validation (V & V), software development V & V activities.

UNIT V SOFTWARE TESTING TOOLS

Manual vs. automated testing, functionality and regression testing tool: Win Runner ; load and performance testing tool: Load Runner; web based application testing tool: Quick Test Professional (QTP); Rational Seed Tool for requirement analysis to testing and project management.

Reference Books:

1. Effective Methods for Software Testing, William E. Perry, John Wiley and Sons, 2002.
2. Effective Software Testing: 50 Specific Ways to Improve Your Testing, Dustin, Pearson Education, 2002.
3. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing House, New Delhi 1997.
4. The Art of Software Testing, Glenford J.Myers, John Wiley & Sons, 1979.
5. Software Testing: A Craftman's Approach, P. C. Jorgensen, CRC Press, 1995.
6. Software Testing Techniques, Boris Beizer, Dreamtech, 2006.
7. Software Testing: Principles and Practices, Srinivasan Desikan, Gopalaswamy Ramesh, Pearson Education, 2008.
8. Software Testing, Aditya P. Mathur, Pearson Education, 2008.
9. Software Testing: Principle, Techniques and Tools, M. G. Limaye, Tata McGraw Hill, 2009.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

RESEARCH TECHNIQUES IN ICT			
Course Code:	**CS633/CS503**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO RESEARCH TECHNIQUES

Meaning of research, objectives of research, motivation in research, types of research (Introduction to experimental test bed, algorithmic research, simulation research, mathematical modeling approach), characteristics and prerequisites of research, significance of research, research process, sources of research problem, criteria of identifying the problem, necessity of defining the problem, errors in selecting research problem, technique involved in defining the problem, report and paper writing.

UNIT II DATA ANALYSIS AND STATISTICAL TECHNIQUES

Data and their analyses, quantitative methods and techniques, Measure of central tendency, measures of variation, frequency distribution, analysis of variance, methods, Correlation analysis, regression analysis, time series and forecasting, introduction to discriminant analysis, factor analysis, cluster analysis, conjoint analysis, probability distribution, binomial distribution, poisson distribution, uniform distribution, exponential distribution, and normal distribution, sampling methods, test of hypothesis.

UNIT III MATHEMATICAM MODELING

Steps of modeling, operations research models like queuing theory, stochastic processes, application of models, conceptual framework development and validation techniques, optimization techniques.

UNIT IV ALGORITHMIC RESEARCH

Algorithmic research problems, types of algorithmic research, types of solution procedure, steps of development of algorithm, steps of algorithmic research, design of experiments,

UNIT V SIMULATION AND SOFT COMPUTING TECHNIQUES

Introduction to soft computing, artificial neural network, genetic algorithm, fuzzy logic and their applications, tools of soft computing, need for simulation, types of simulation, simulation language, fitting the problem to simulation study, simulation models, output analysis, data simulation packages like MATLAB, NS2, ANSYS, Cadence.

Reference Books:

1. Research Methodologies, R. Panneerselvam, Prentice Hall, 2007.
2. Research in Education, Best John V. and James V Kahn, Wiley eastern, 2005.
3. Elements of Educational Research, Sukhia, S.P., P.V. Mehrotra, and R.N. Mehrotra, PHI publication, 2003.
4. Methodology of Research Education, K. Setia, lEEE publication, 2004.
5. Research methodology, Methods and Techniques, Kothari, C.R., 2000.

SOFTWARE TESTING LAB			
Course Code:	**CS581**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**		
Total No. of Lectures:	**15**	**End Sem Exam Hours:**	**2**

Programs/Experiments List:

1. Introduction to various phases of SDLC Implementation.
2. Create the Requirement and Design document using IEEE format.
3. Design and Manage the Test Plan using Rational Test Manager.
4. Write a program to design the Test Plan.
5. Write a program to calculate the Cyclomatic Complexity.
6. Design Test Cases using Rational Test Manager.
7. Develop and Execute Manual Test.
8. Implement Data-Driven Testing.
9. Create and modify various types of Verification points.
10. Determine Test Results and list out the Test Case Distribution Report.

SOFTWARE ENGINEERING

(ELECTIVES - 3 & 4)

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE MEASUREMENT AND ESTIMATION			
Course Code:	**CS641/CS545**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE MEASUREMENTS THEORY

Fundamentals of software measurement, measurement scale, scope of software, need for measurement, type of measurement process, measures of central tendency and variability, validation, validity and reliability of measurement, empirical investigation, planning experiments, challenges of software measurement, measurement models, data collection, analysis methods, statistical methods, measurement life cycle.

UNIT II MEASURING SOFTWARE SYSTEM

Measuring size of software, physical measurement of software, measuring functionality, measuring complexity, structural complexity, conceptual complexity, estimating efforts.

UNIT III MEASUREMENT AND METRICS

Software metrics, Design metrics: method size, method internals, class size, class inheritance, method inheritance, class internals, class externals; software quality metrics: product quality, process quality, metrics for software, maintenance; metrics for reliability prediction, measuring the costs of defect removal, evaluating defect prevention methods, Rayleigh Model, Problem Tracking Report (PTR) model, reliability growth model, model evaluation, orthogonal defect classification, case studies of metrics programs.

UNIT IV SOFTWARE QUALITY MODELS AND RELIABILITY

Software quality models: Boehm's model, McCall's model, ISO 9126 model, Basic software quality metrics, Quality management models, Measuring customer satisfaction , Software Quality Assurance (SQA), Defects, Faults vs. failures, Defect Projection Techniques and Models, Software Reliability Measurement and Prediction, The Cost of Reliability, Software Reliability Theory, reliability models, failure arrival rate, response time, response time measurements, availability measuring progress, case studies of software quality.

UNIT V SOFTWARE COST ESTIMATION

Software estimation methodologies and models, combining estimates, estimating Issues, software cost factors, cost estimation, software cost estimation techniques, staffing-level estimation, estimating software maintenance costs, Cost estimation Constructive Cost Model (COCOMO), Function Point (FP) model, Common Software Measurement International Consortium (COSMICS), Full Function point (FFP) approach, software estimation crisis, case studies

References Books:

1. Software Measurement and Estimation: A Practical Approach, Linda M. Laird, M. Carol Brennan, Willy, 2006.
2. Software Metrics, N. E. Fentar and S. L. Pflieger, International Thomson Computer Press, 1997.
3. Metric and Models in Software Quality Engineering, Stephen H. Kin, Addision Wesley, 1995.
4. Measuring Software Process, William A. Florac and Areitor D. Carletow, Addison - Wesley, 1995.
5. Estimating Software Costs: Bringing Realism to Estimating, Capers Jones, Tata M. Hill, 2007.
6. Applied Software Measurement: Global Analysis of Productivity and Quality, Capers Jones, Tata McGRAW Hill, 2008.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE RELIABILITY AND FAULT TOLERANT SYSTEMS			
Course Code:	**CS643/CS547**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE RELIABILITY

Measures of software reliability, Mean Time To Failure (MTTF), Mean Time Between Failure (MTBF), Mean Time To Recovery (MTTR), availability, maintainability, Musa's operational profiles and type-1 uncertainty, defect removal and type-2 uncertainty, reliability stability and reliability growth, hardware reliability vs. software reliability, failure probability density function and reliability function, Reliability prediction , reliability metrics.

UNIT II DEVELOPMENT OF RELIABLE SOFTWARE

Reliable software, defect prevention, detection and removal, design for robustness, verification & validation, stabilization of requirements, design, code and test artifacts, active and passive fault detection, fault handling and correction, exceptions, survivability, reliability models, software availability model.

UNIT III FAULT TOLERANCE IN HARDWARE SYSTEMS

Fault classification, fault tolerance attributes and system structure, fault prevention, anticipated and unanticipated fault, test generation for digital systems, combinational logic network, Boolean difference method, test generation for sequential circuits, fault simulation, application of hardware fault tolerance in developing fault tolerant software systems.

UNIT IV SOFTWARE AND HARDWARE FAULT TOLERANCE

Software and hardware faults, failure intensity function, characterization of fault injection, detection and correction, techniques for prediction of remaining faults and fault injection, classification tree analysis, code coverage, coding technique, fault tolerant & self checking, fail safe circuits, synchronous and asynchronous fail safe circuits.

UNIT V FAULT TOLERANT SOFTWARE

Concept of N-version programming (NVP) and methods, recovery block, acceptance tests, fault trees, validation of fault tolerant systems, security, fault tolerance in wireless/mobile networks and Internet.

Reference Books:

1. Software Reliability Engineering, John D. Musa, Tata McGRAW Hill, 2005.
2. Fault-Tolerant Computer System Design, D.K. Pradhan, 2003.
3. Design and Analysis of Fault-Tolerant Digital Systems, B. W. Johnson, Addison-Wesley, 1989.
4. Fault-Tolerant Computing, Theory & Techniques, D.K. Pradhan, Prentice Hall, 1986.
5. Reliable Computer Systems: Design and Evaluation, D. P. Siewiorek and R. S. Swartz, Digital Press, 1992.
6. Probability and Statistics with Reliability, Queueing and Computer Science application, K.S.Trivedi, Prentice Hall, 1982.
7. Fault Tolerant Principles and Practice, Anderson and Lee, PHI, 1989.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE QUALITY ASSURANCE AND ENGINEERING			
Course Code:	**CS645/CS549**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE QUALITY AND ENGINEERING

Quality concepts and productivity relationship, software quality factors, software quality costs, Total Quality Management (TQM), continuous improvement cycle: Plan, Do, Check and Act (PDCA), quality policy, cost of quality, quality engineering, quality planning: goal setting and strategy formation, assessment and improvement.

UNIT II SOFTWARE QUALITY ASSURANCE (SQA)

Components of SQA, classification, defect detection, defect prevention, defect reduction, defect containment, QA activities in software processes, verification and validation, software review, inspection, formal verification, statistical software quality approach.

UNIT III COMPONENTS MEASUREMENT WITH REFERENCE TO SQA

Metrics, product quality metrics, process quality metrics, metrics for software maintenance, quality tools for quality control, test management and organizational structures, Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI), ISO 9000, quality and quality management metrics, Deming's Principle, SQA team formation

UNIT IV QUALITY MANAGEMENT MODEL

Integrating quality activities in project life cycle, reviews, software testing, strategies and implementation, Computer-Aided Software Engineering (CASE) tools, The Rayleigh model framework, code integration pattern, Problem Tracking Report (PTR), reliability growth model, Service Quality, Kano Model, Customer retention, continuous process improvement, Juran's Trilogy, TQM principles, Kaizen Technique, Statistical Quality Assurance, Mc call quality factors

UNIT V SOFTWARE QUALITY ASSURANCE BEYOND TESTING

Defect prevention and process improvement, root cause analysis for defect prevention, software inspection, inspection related activities, fault tolerance and failure containment, comparing quality assurance techniques and activities.

Reference Books:

1. Metrics and Models in Software Quality Engineering, Stephan H. Kan, Pearson Education, 2007.
2. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing House, New Delhi 1997.
3. Making Sense of Software Quality Assurance, Raghav J. Nandyal, Tata McGRAW Hill, 2007.
4. Software Quality Assurance: A Practitioner Approach, Kaman Malik, Praveen Chaudhary, Tata McGRAW Hill, 2008.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE MAINTENANCE			
Course Code:	**CS647/CS553**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO SOFTWARE MAINTENANCE

Evolution of software products, economics of maintenance, characteristics of software maintenance, product development life cycle, different type of software products, deployment model, adaptive maintenance, enhancement request, proactive defect prevention, maintenance process, problem reporting, problem resolution, fix distribution, software maintenance from customers' perspective, category of software maintenance.

UNIT II REVERSE ENGINEERING

Function abstraction, data abstraction, and process abstraction, levels of reverse engineering: re-documentation, design recovery, specification recovery, conditions for reverse engineering, supporting techniques: forward engineering, restructuring, re-engineering, benefits of reverse engineering.

UNIT III CONFIGURATION MANAGEMENT

Software configuration management process, patches, configuration management in global development teams, baseline, software configuration items, identification of objects in software configuration, version control, change control, configuration audit, status reporting, software configuration management standards, metrics for maintenance in configuration management. traditional process model: Code and Fix Model, Waterfall Model, Spiral Model; maintenance process model: Quick Fix Model, Boehm's Model, Osbornes' Model, Iterative Enhancement Model; process maturity model: Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI).

UNIT IV MAINTENANCE AND OTHER LIFE CYCLE ACTIVITIES

Design and maintenance, programming & maintenance, debugging and maintenance, testing and maintenance, maintenance management, maintenance management functions: planning, organizing, staffing, leading, controlling; maintenance management organizations: functional organization, project organization, matrix organization.

UNIT V MAINTENANCE MEASURES

Importance of integrity in measurement, software measure and metrics, objective of software measurement: evaluation, control, assessment, improvement, prediction, maintenance measures: size, complexity; quality: product and process quality, understandability and maintainability, impact analysis in creating maintainable system.

Reference Books:

1. Software Maintenance: Concept and Practice, Penny Grubb, Aramstrong A. Takang, International Thompson Publishing Inc., 1996.
2. Software Maintenance, Gopalaswamy Ramesh, Ramesh Bhattiprolu, Tata McGraw Hill, 2009.
3. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.
4. Software Engineering: Practitioner's Approach, Pressman Roger S., McGraw-Hill Inc., 2004.
5. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Jeff Tian, John Wiley and Sons Inc., and IEEE Computer Society Press, 2005.
6. Software Quality Assurance, Daniel Gain, Pearson Education, 2009.

3 Year M. Tech. (ICT) for Science Graduates **Effective from: 2011 -2012**

SOFTWARE PERFORMANCE			
Course Code:	**CS649/CS555**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE PERFORMANCE

Software and performance, repository and scalability, importance of performance, consequences and cause of performance failure, reactive and proactive performance management, software performance Engineering (SPE), modeling strategy and models, SPE for object-oriented system.

UNIT II SOFTWARE PERFORMANCE ENGINEERING PROCESS

SPE Process for object oriented system, performance solutions, principles, patterns, implementation solutions,

UNIT III SOFTWARE EXECUTION MODELS

Software Execution Models (SEM), representing of SEM, execution graphs and graph restrictions, model solution, basic solution algorithms, analysis procedure, execution graph from sequence diagrams.

UNIT IV SYSTEM EXECUTION MODELS

System Execution Models, system model basics, performance metrics, system models for SPE, advanced system models, Schedulability.

UNIT V SOFTWARE PERFORMANCE ENGINEERING

SPE Data Collection, SPE data requirement, key performance scenarios, performance objectives, execution environment software resource requirements, data gathering issues, performance walkthrough, resource estimation techniques.

Reference Books:

1. Performance Solution : A Practical Guide to Creating Responsive, Scalable Software, Addison-Wesley Professional.
2. Performance prototyping: a simulation methodology for software performance engineering, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=218462&tag=1