SCHOOL
OF
INFORMATION AND COMMUNICATION TECHNOLOGY

COURSE STURCTURE AND DETAILED SYLLABUS

2 YEARS M. TECH. ICT FOR ENGINEERING GRADUATES

SPECIALIZATION:

SOFTWARE ENGINEERING

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

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

SEMESTER – I

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS521	Advanced Software Engineering	3-1-0	4
2	CS523	Advanced Computer Architecture	3-1-0	4
3	CS525	Advanced Data Base Management System	3-1-0	4
4		**Elective-1**	3-0-0	3
5	SS101	Human Values & Buddhist Ethics	2-0-0	2
6	CS585	Advanced Software Engineering Lab	0-0-3	2
7	CS587	Advanced Data Base Management System Lab	0-0-3	2
8	GP521	General Proficiency	------	1
		Total Credits		**22**
		Total Contact Hours	**14-3-6 =**	**23**

Electives - 1		
Sr. No.	**Courses Code**	**Courses**
1	CS441 / CS541	Software Project Management
2	CS443 / CS543	Object-Oriented Software Engineering
3	CS445 / CS545	Information Security
4	CS447 / CS547	Multimedia Techniques
5	CS449 / CS561	Soft Computing
6	CS451 / CS551	Natural Language Processing
7	CS457 / CS557	Machine Translation and Learning
8	EC405	Wireless Mobile Communication
9	EC542	Quality of Services in Networks
10	EC447	Digital Image Processing
11	EC465 / EC555	Principles of VLSI Design
12	EC441	Design with Microcontrollers

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

SEMESTER – II

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 / CS404	Open Source Software Systems	3-0-0	3
4		**Elective-2**	3-0-0	3
5		**Elective-3**	3-0-0	3
6	CS582 / CS482	Open Source Software Systems Lab	0-0-3	2
7	CS592 / CS492	Major Project	0-0-10	5
8	GP532	General Proficiency	------	1
		Total Credits		**24**
		Total Contact Hours	**15-1-13 =**	**29**

Electives (2 & 3)		
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

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

SEMESTER III

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 - 4**	3-0-0	3
4		**Electives - 5**	3-0-0	3
5	CS581/CS681	Software Testing Lab	0-0-3	2
6	CS691/CS591	Dissertation Part - I	0-0-14	7
7	GP631	General Proficiency	------	1
		Total Credits		**22**
		Total Contact Hours	**12-0-17 =**	**29**

Electives (4 & 5)		
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 IV

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 = 90

SOFTWARE ENGINEERING

(SEMESTER - I)

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

ADVANCED SOFTWARE ENGINEERING			
Course Code:	**CS521**	**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, Software Development Life Cycle, software process models, requirement analysis and design, software design process, coding, testing and maintenance.

UNIT II SOFTWARE RE-ENGINEERING AND REVERSE ENGINEERING

Re-engineering concept and approaches, redevelopment vs reengineering, reengineering process, reverse engineering, 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, software re-engineering patterns, patterns based software re-engineering, software re-engineering techniques.

UNIT III OBJECT-ORIENTED SOFTWARE ENGINEERING

OOSE, object-orientation paradigm, object-oriented analysis, use cases, analysis, stereotypes and objects, patterns, object modeling languages, object-oriented design: design stereotypes and objects, design patterns; software design using patterns and framework, object-oriented programming languages, application frameworks, object-oriented case tools, state transition and interaction diagrams.

UNIT IV COMPONENT-BASED SOFTWARE ENGINEERING

Component-Based Software Engineering (CBSE), CBSE vs. object-oriented software engineering, CBSE methodology, CBSE processes, domain engineering, component engineering, component-based software development life cycle, component vs. object, component-oriented programming, component-oriented programming vs. object-oriented programming, overview of component-based technology.

UNIT V ASPECT-ORIENTED SOFTWARE ENGINEERING

Software engineering with aspects, aspects, aspect vs. object, aspect vs. component, join points and pointcuts, separation of concerns, crosscutting concerns, problems caused by scattering and tangling, system development, concepts of Aspect Oriented Programming, inter-type declarations, implementation, comparison to other programming paradigms, Join Point Model, AspectJ Point Model, pointcut designators, aspect weaving.

Text Books:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.

Reference Books:

3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. N. S. Gill, Software Engineering: Software Reliability, Testing and Quality Assurance, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.
5. J. Rumbaugh, M. Blaha, W. Premerlani, Object-Oriented Modeling and Design, PHI, 1991.
6. George T. Heineman, William T. Councill, Component-Based Software Engineering: Putting the Pieces Together, Addision Wesley, 2001.
7. Robert E. Filman, Tzilla Elrad, Siobhán Clarke, Mehmet Aksit, Aspect-Oriented Software Development Addison-Wesley Professional, 2004.

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

ADVANCED COMPUTER ARCHITECTURE			
Course Code:	**CS523**	**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 to parallel processing: parallelism in uniprocessor system, basic uniprocessor architecture, parallel processing mechanism, balancing of sub system bandwidth, multiprogramming and time sharing, parallel computer structures, pipeline computers, array computers, multiprocessor systems, dataflow computer concept, architectural classification scheme: multiplicity of instruction-data streams, serial versus parallel processing, parallelism versus pipelining, parallel processing applications, productive modeling simulation, engineering design and automation.

UNIT II

Principles of pipelining and vector processing: pipelining- an overlapped parallelism, principles of linear pipelining, clock period, efficiency, throughput, classification of pipeline processors, general pipeline and reservation tables.

UNIT III

Principles of designing pipeline processors: effect of branching, data buffering and bussing structures, internal forwarding and register tagging, hazard detection and resolution, job sequencing and collision prevention, reservation and latency analysis, collision free scheduling, state diagram, greedy cycle, pipeline schedule optimization, pipeline throughput, pipeline efficiency.

UNIT IV

Structure and algorithm for array processors: SIMD array processor, SIMD computer organization, inter –PE communication, SIMD interconnection network, static versus dynamic networks, cube interconnection network, shuffle-exchange omega networks, parallel algorithms and SIMD matrix multiplication.

UNIT V

Multiprocessor architecture and scheduling: functional structure, loosely coupled and tightly coupled multiprocessor, deterministic scheduling strategy, deterministic scheduling model, control flow versus data flow computer, data flow graphs and languages.

References Books:

1. Kai Hwang, “Advanced Computer Architecture”, Tata McGrawHill Edition
2. Kai Hwang and Faye A. Briggs, “Computer Architecture and Parallel Processing”, McGraw-Hill International Edition
3. Richard Y. Kain, “Advanced Computer Architecture: a Systems Design”, Prentice Hall.
4. James M. Feldman, Charles T. Retter, “Computer architecture: a designer's Text Based on a generic RISC”, McGraw-Hill
5. Jurij Silc, Borut Robic, Theo Ungerer, “Processor Architecture: From Dataflow to Superscalar and Beyond”, Springer.
6. Hennessy and Patterson, “Computer Architecture: A Quantitative Approach”, Elsevier.
7. Dezso and Sima, “Advanced Computer Architecture”, Pearson.
8. Quinn, “Parallel Computing: Theory & Practice”, TMH.
9. Quinn, “Parallel Programming in C with MPI and Open MP”, TMH

ADVANCED DBMS			
Course Code:	**CS525**	**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 THE EXTENDED ENTITY RELATIONSHIP MODEL AND OBJECT MODEL

ER model revisited, motivation for complex data types, user defined abstract data types and structured types, subclasses , superclasses, inheritance, specialization and generalization, relationship types of degree higher than two, object–oriented concepts, object identity, object structure and type constructors, encapsulation of operations, methods and persistence, type hierarchies and inheritance, type extents and persistent programming languages, OODBMS architecture and storage issues, transactions and concurrency control, examples of ODBMS.

UNIT II OBJECT RELATIONAL AND EXTENDED RELATIONAL DATABASES

Database design for an ORDBMS, nested relations and collections , Storage and access methods, query processing and optimization, overview of SQL3, implementation issues for extended type, Systems comparison of RDBMS, OODBMS, ORDBMS.

UNIT III PARALLEL AND DISTRIBUTED DATABASES AND CLIENT–SERVER ARCHITECTURE

Architectures for parallel databases, parallel query evaluation, parallelizing individual operations, sorting joins, distributed database concepts, data fragmentation, replication and allocation techniques for distributed database design, query processing in distributed databases, concurrency control and recovery in distributed databases, an overview of client–server architecture.

UNIT IV DATABASES ON THE WEB AND SEMI–STRUCTURED DATA

Web interfaces to the web, Overview of XML, structure of XML data, Document Schema, Querying XML data, Storage of XML data, XML applications, semi–structured data model, Implementation issues, Indexes for text data.

UNIT V ENHANCED DATA MODELS FOR ADVANCED APPLICATIONS

Active database concepts, temporal database concepts, spatial databases: concept and architecture, Deductive databases and query processing, mobile databases, Geographic Information Systems (GIS).

Text Books:

1. Elmsari and Navathe, Fundamentals of Database Systems,
2. Ramakrishnan and Gehrke, Database Management Systems,

References Books:

3. Korth, Silberschatz, Sudarshan, Database System Concepts,
4. Rob and Coronel, Database Systems: Design, Implementation and Management,
5. Date and Longman, Introduction to Database Systems,

SOFTWARE ENGINEERING

(ELECTIVES -1)

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

SOFTWARE PROJECT MANAGEMENT			
Course Code:	**CS541/CS441**	**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 PROJECT MANAGEMENT

Scope of project management, project life cycle, software project planning- Step by Step planning, Introduction of project management activities: cost estimation, project scheduling, staffing, software configuration management, quality assurance, project monitoring, risk management, problem with software projects, roles and responsibilities of software project manager.

UNIT II PROJECT EVALUATION AND APPROACHES

Project management and evaluation: Cost benefit analysis, payback period, NPV, ROI, Selection of appropriate project approach: Waterfall model, V-process model, prototyping, spiral model, incremental delivery, iterative process. Capability Maturity Model (CMM).

UNIT III ACTIVITIES PLANNING AND RISK MANAGEMENT

Project scheduling, Project network diagram fundamentals, PERT techniques, Gantt charts, Risk assessment, planning and management, Resource allocation: creating critipaths, scheduling, cost schedules. Monitoring and controlling the projects. Introduction to Microsoft Project.

UNIT IV MONITORING PROJECTS AND CONTRACTS

Monitoring the progress of projects, accessing the risk of slippage, reporting, earned value analysis, control procedures, Managing Contracts: stages needed to acquire software, types of contracts contents of contracts and the evaluation of proposal on the basis of contracts.

UNIT V PROJECT QUALITY AND PEOPLE ISSUES

People: player, team leader, software team, coordination and communication issues; inducting motivating teams, improving efficiency, Software quality and its importance defining, designing and monitoring the software quality.

Reference Books:

1. Software Project Management, Cottrell M. and Hughes B., Tata McGraw Hill, 2006.
2. Software Project Management-A Real-World Guide to Success, Henry J., Addison Wesley, 2009.
3. Effective Software Project Management, Robert K. Wysocki, Wiley India, 2006.
4. Introduction to Software Project Management and Quality Assurance, Ince D., Sharp H. and Woodman M., McGraw Hill, 1993.
5. Project Management, Maylor H., Prentice Hall, 2003.
6. Success in Your Project-A Guide to Student System Development Projects, Weaver P., Prentice Hall, 2004.
7. Managing the Software Process, Humphrey, Watts, Addision-Wesley, 1986.
8. Software Engineering: A Practitioner's Approach, Pressman, Roger, McGraw Hill, 1997.
9. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co. (P) Ltd., New Delhi, 2002.
10. Fundamental of Software Engineering, Rajib Mall, Prentice Hall of India, 2003.
11. Software Engineering Concepts, Richard E. Fairley, Tata McGraw Hill, 1997.
12. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing House, New Delhi 1997.
13. Software Engineering, Ian Sommerville, Pearson Education, 2009.

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

OBJECT-ORIENTED SOFTWARE ENGINEERING			
Course Code:	**CS543/CS443**	**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 SOFTWARE ENGINEERING

OOSE, object-orientation paradigm, object-oriented analysis, basic concepts, use cases, analysis, stereotypes and objects, analysis patterns, object modeling languages, object-oriented design: basic concepts, design stereotypes and objects, design patterns; object-oriented programming: basic concepts, idioms, object-oriented programming languages, application frameworks, object-oriented case tools, state transition and interaction diagrams, testing of object-oriented programs.

UNIT II ADVANCED OBJECT-ORIENTED ANALYSIS AND DESIGN

Frameworks and design patterns, design for reusability, advanced object-oriented programming techniques, design using object-oriented databases and distributed object architectures, design of software agents, project involving object-oriented analysis, design, and implementation.

UNIT III DESIGNING SOFTWARE USING PATTERNS

Process of design, principles, techniques, software architecture, architectural patterns, abstraction-occurrence pattern, hierarchy pattern, player-role pattern, singleton pattern, observe pattern, delegation pattern, adapter pattern, facade pattern, immutable pattern, read only interface pattern, proxy pattern.

UNIT IV OBJCET-ORIENTED METRICS

Measure, metrics and indicators, software measurement, metrics for object-oriented software development environments, characteristic of object-oriented metrics, Chidamber & Kemerer's metrics suite: Weighted Methods Per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Object Classes (CBO), Response For a Class (RFC), Lack of Cohesion in Methods (LCOM), Lorenz and Kidds' metrics, metrics for object-oriented metrics projects: management process, development process, application size, staffing size, scheduling.

UNIT V DESIGN METRICS AND OBJCET-ORIENTED TESTING

Design metrics overview, method size, method internals, class size, class inheritance, method inheritance, class internals, MOOD (Metrics for Object-Oriented Design): Method Hiding Factor (MHF), Attribute Hiding Factor (AHF), Method Inheritance Factor (MIF), Attribute Inheritance Factor (AIF), Polymorphism Factor (PF), Coupling Factor (CF), object-oriented testing, test case design for object-oriented software, testing methods at class level: random testing of object-oriented class; interclass test case design: multiple class testing, test derived from behavior models.

Reference Books:

1. Object-Oriented Software Engineering, Bernd Bruegge, Allen H. Dutoit, PHI, 2003.
2. Object-Oriented Software Engineering, Timothy C. Lethbridge, Robert Laganiere, TMH, 2008.
3. Object-Oriented Modeling and Design, J. Rumbaugh, M. Blaha, W. Premerlani, PHI, 1991.
4. Object-Oriented Design, Grady Booch, 1991.
5. Software Engineering: Practitioner's Approach, Pressman Roger S., TMH, 2004.
6. Software Engineering: Software Reliability, Testing & Quality Assurance, N. S. Gill, KBP, 2002.
7. Fundamental of Software Engineering, Rajib Mall, Prentice Hall of India, 2003.
8. Software Engineering Concepts, Richard E. Fairley, Tata McGraw Hill, 1997.
9. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing, 1997.
10. Software Engineering, Ian Sommerville, Pearson Education, 2009.

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

INFORMATION SECURITY			
Course Code:	**CS545/CS445**	**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

Security problem in computing, threat scenarios, critical infrastructures, security targets and policies, security mechanisms, examples of applications and their different security requirements, multi-lateral security, privacy and data protection, computer misuse legislation, operating system and network security, cyber laws, , hacking , anti hacking solution, case studies of modern antivirus software, Computer Emergency Response Team (CERT) functionality, NIST, Introduce RFC related to security.

UNIT II SECURITY MODELS

Military and Civil Security, vulnerability and threat models, end-end security (COMSEC), link Encryption (TRANSEC), compartments, privacy, authentication, denial of service, no repudiation, private-key and public-key cryptographic algorithms: DES, RSA, SHA, encapsulation, encryption principles, issues in multi-level secure systems, Internet security models: IPv4/IPv6 encapsulation header, digital signature standard,

UNIT III SECURITY POLICIES AND DESIGN GUIDELINES

Policies, policy creation, regularity considerations, and privacy regulations, security: infrastructure and components, design guidelines, authentication: authorization and accounting, physical and logical access control, user authentication: biometric devices. open source software for network security quantum cryptography, Microsoft cryptography toolkit, cryptographic solution using java .

UNIT IV OSI LAYER SECURITY

Secure SNMP, secure routing interoperability: IP Security, virtual networks (DART net/CAIRN), transparent and opaque network services, source masking and hidden channels, techniques for fault detection, isolation and repair, secure network infrastructure services: DNS, NTP, SNMP, privacy Enhanced Mail (PEM), secure binding of multimedia streams, secure RTP, secure RSVP, mobile systems: Address Export and re-use, Secure Session Layer (SSL), Secure Hypertext Transfer Protocol (SHTTP), Time Stamping Protocol (TSP), email security, Firewall platforms, partitioning models and methods,

UNIT V KEY AND CERTIFICATE MANAGEMENT

Secure binding of public and private values: DNS certificates, making and distributing key media: randomization, lifetime issues, key agreement protocols: STS protocol and IETF work orders, key escrow: clipper chip, one-time passwords: schemes based on S/KEY, PKI components and applications, exploiting diversity and redundancy: byzantine generals, time stamping and reliable ordering of events: NTP, consensus and agreement protocols, security in wireless networks, shared secret data authentication: token based/ public key based, session key management: blind key cryptosystems.

Reference Books:

1. Information Security, Principal and practices,Mark Merkow, Jim Breithaupty, Person 2007
2. Cryptography and Network Security: Theory and Practice, Stallings, John Wiley, 2006.
3. Network Security Bible Eric Cole , Ronald L. Krutz,Welley, 2005.
4. Computer Security, Gollmann, D., Wiley Edition, 1999.
5. Cryptography, Theory and Practice, Stinson D., CRC Press, Boca Raton, FA, 1995.
6. Security Engineering: A Guide to Building Dependable Distributed Systems, Anderson R., Wiley, 2nd edition, 2008.
7. Web Security: A Step-by-Step Reference Guide, Stein L., Addison Wesley Longman, Inc., 1998.

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

MULTIMEDIA TECHNIQUES			
Course Code:	**CS547/CS447**	**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

Introduction to multimedia, multimedia information, multimedia objects, multimedia in business and work, convergence of computer, communication and entertainment products, multimedia hardware, Memory & storage devices, Communication devices, Multimedia software's, presentation tools, tools for object generations, video, sound, image capturing, authoring tools, card and page based authoring tools.

UNIT II MULTIMEDIA BUILDING BLOCKS

Text, sound MIDI, digital audio, audio file formats, MIDI under windows environment audio & video capture.

UNIT III DATA COMPRESSION

Huffman coding, Shannon Fano algorithm, Huffman algorithms, adaptive Coding, arithmetic coding Higher order modeling, finite context modeling, dictionary based compression, sliding window compression, LZ77, LZW compression, compression, compression ratio loss less & lossy compression.

UNIT IV SPEECH COMPRESSION & SYNTHESIS

Digital Audio concepts, sampling variables, loss less compression of sound, loss compression & silence compression.

UNIT V IMAGES

Multiple monitors, bitmaps, vector drawing, lossy graphic compression, image file formatting animations Images standards, JPEG Compression, Zig Zag coding, multimedia database, content based retrieval for text and images, video: video representation, colors, video compression, MPEG standards, MHEG standard video Streaming on net, video conferencing, multimedia broadcast services, indexing and retrieval of video database, recent development in multimedia.

Reference Books:

1. Tay Vaughan “Multimedia, Making IT Work” Osborne McGraw Hill.
2. Buford “Multimedia Systems” Addison Wesley.
3. Agrawal & Tiwari “Multimedia Systems” Excel.
4. Mark Nelson “Data Compression Book” BPB.
5. David Hillman “Multimedia technology and Applications” Galgotia Publications.
6. Rosch “Multimedia Bible” Sams Publishing.
7. Sleinreitz “Multimedia System” Addison Wesley.
8. James E Skuman “Multimedia in Action” Vikas.

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

SOFT COMPUTING			
Course Code:	**CS449/CS561**	**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 FUZZY LOGIC

Introduction to fuzzy logic, classical and fuzzy sets, overview of fuzzy sets, membership function, fuzzy rule generation, operations on fuzzy sets: compliment, intersection, union, combinations on operations, aggregation operation.

UNIT II FUZZY ARITHMETIC

Fuzzy numbers, linguistic variables, arithmetic operations on intervals & numbers, uncertainty based information, information and uncertainty, no specificity of fuzzy and crisp sets, fuzziness of fuzzy sets.

UNIT III NEURAL NETWORK

Overview of biological neurons, computational neuron, mathematical model of neurons, ANN architecture, single layer and multilayer architectures, activation function, threshold value, self learning and forced learning algorithms, feed forward and feedback architectures.

UNIT IV LEARNING FUNDAMENTALS

Learning paradigms, supervised and unsupervised learning, reinforced learning, ANN
training, algorithms perceptions, training rules, delta, back propagation algorithm, multilayer perception model, Hopfield networks, associative memories, applications of artificial neural networks,

UNIT V GENETIC ALGORITHMS

History of genetic algorithm, terminology of genetic algorithm, biological background, creation of offspring, working principles of genetic algorithms, fitness function, reproduction: Roulette wheel selection, Boltzmann selection, cross over mutation, inversion, deletion, and duplication, generation cycle.

Reference Books:

1. Artificial Neural Networks: An introduction to ANN Theory and Practice, Peteus J. Braspenning, PHI publication, 2005.
2. Fuzzy Logic: A spectrum of Theoretical and Practical issues, Paul P. Wang, pearson publication 2004.
3. Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- Lotfi Asker Zadeh, George J. Kilr, Bo yuan, 2005.
4. Foundations of Fuzzy logic and Soft Computing: 12th International Fuzzy conference proceeding, 2005.
5. Neural Networks Theory, Particia Melin, Oxford University press, 2003
6. Neural Networks Theory and Application, Oscar Castillo, Wiley Eastern publication 2003.

SOFTWARE ENGINEERING

(SEMESTER - II)

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

SOFTWARE ARCHITECTURE AND DESIGN			
Course Code:	**CS532**	**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.

2 Year M. Tech. (ICT) for Engineering 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.

SOFTWARE ENGINEERING

(ELECTIVES - 2 & 3)

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.

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

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 (AOP)

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.

2 Year M. Tech. (ICT) for Engineering 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.

2 Year M. Tech. (ICT) for Engineering 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.

2 Year M. Tech. (ICT) for Engineering 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,

2 Year M. Tech. (ICT) for Engineering 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)

2 Year M. Tech. (ICT) for Engineering 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.

2 Year M. Tech. (ICT) for Engineering 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 ENGINEERING

(ELECTIVES - 4 & 5)

2 Year M. Tech. (ICT) for Engineering 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.

2 Year M. Tech. (ICT) for Engineering 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.

2 Year M. Tech. (ICT) for Engineering 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.

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.

2 Year M. Tech. (ICT) for Engineering 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