BHAGWANT UNIVERSITY

M.TECH (Software Engineering)

I SEMESTER

Subject Code	Name of Subject	Teaching Hours			Credits
		L	T	P	
01 MSE 101	Software Engineering	4	1	0	5
01 MSE 102	Software Architecture	4	1	0	5
01 MSE 103	Optimizing Compilers	4	1	0	5
Elective					
01 MSE 104.1	Advanced Data Structures	4	1	0	5
01 MSE 104.2	High Level System Design And Modeling	4	1	0	5
01 MSE 104.3	Client Server Based IT Solutions	4	1	0	5
Practical					
01 MSE 201	Software Design and Modeling Lab	0	0	3	2
01 MSE 301	Discipline & Extra Curricular activities	0	0	4	1
TOTAL		**16**	**4**	**13**	**23**

II SEMESTER

Subject Code	Name of Subject	Teaching Hours			Credits
		L	T	P	
02 MSE 101	Security Analysis of Software	4	1	0	5
02 MSE 102	Software Verification, Validation and Testing	4	1	0	5
02 MSE 103	Software Quality Management	4	1	0	5
Elective					
02 MSE 104.1	Advanced Data Base Systems	4	1	0	5
02 MSE 104.2	Distributed Operating Systems	4	1	0	5
02 MSE 104.3	Embedded Software and Systems	4	1	0	5
Practical					
02 MSE 201	Software Testing Lab	0	0	3	2
02 MSE 301	Discipline & Extra Curricular activities	0	0	4	1
TOTAL		**16**	**4**	**13**	**23**

III SEMESTER

Subject Code	Name of Subject	Teaching Hours			Credits
		L	T	P	
03 MSE 101	Unified Software Configuration	5	1	0	6
Elective					
03 MSE 102.1	Software Reliability	5	1	0	6
03 MSE 102.2	E-Business	5	1	0	6
03 MSE 102.3	Software Reuse	5	1	0	6
Practical					
03 MSE 201	Dissertation Part-I	4	0	0	4
03 MSE 202	Seminar	2	0	0	2
03 MSE 301	Discipline & Extra Curricular activities	0	0	4	1
TOTAL		**16**	**2**	**4**	**19**

IV SEMESTER

Subject Code	Name of Subject	Teaching Hours			Credits
		L	T	P	
04MSE 201	Dissertation a) Continuous Evaluation b) Project Report c) Viva Voice	5 6 6	0	0	5 6 6
04MSE 301	Discipline & Extra Curricular Activities	0	0	4	1
TOTAL		17	0	4	18

Theory: I Semester

MSE Semester-I
Course/Paper: 01 MSE 101

SOFTWARE ENGINEERING

.

Principles and Motivations
Definitions and need for engineered approach to software development; Software development process models from the points of view of technical development and project management: waterfall, rapid prototyping, incremental development, spiral models, emphasis on computer-assisted environments.

Introduction to Modeling Tools
Basics of object-oriented approach, object-oriented programming and languages, OMT, visual modeling, UML, Rational Rose Tool

Object Modeling and Design
Classes, objects, relationships, key abstractions, common mechanisms, diagrams, class diagrams, advanced classes, advanced relationships, interfaces, types, roles, packages, instances, object diagrams, interactions, use cases, use case diagrams, interaction diagrams, activity diagrams, events and signals, state machines, processes, threads, state chart diagrams, components, deployment, collaborations, patterns and frameworks, component diagrams, systems and models, code generation and reverse engineering.

Software Development Methods
Formal, semi-formal and informal methods; Requirements elicitation, requirements specification; Data, function, and event-based modeling; Some of the popular methodologies such as Yourdon's SAD, SSADM etc; CASE tools-classification, features, strengths and weaknesses; ICASE; CASE standards.

Software Project Management
Principles of software projects management; Organizational and team structure; Project planning; Project initiation and Project termination; Technical, quality, and management plans; Project control; Cost estimation methods - Function points and COCOMO.

References:
1. Roger Pressman; Software Engineering - A Practitioner's Approach, McGraw Hill, New York.
2. Ian Sommerville; Software Engineering, Addison-Wesley Publishing Company, England
3. Pankaj Jalote; An integrated Approach to Software Engineering, Narosa Publishing House, New Delhi.
4. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Pearson Education, New York.
5. James Rumbaugh, Ivar Jacobson, Grady Booch: The Unified Modeling Language Reference Manual, Addison-Wesley, New YorkGrady Booch, Object-Oriented Analysis and Design, Pearson Education, New York.
6. Terry Quatrani, Visual Modeling with Rational Rose 2000 and UML, Addison- Wesley, New York.
7. G. Schneider, Applying Use Cases: A Practical Guide: Addison- Wesley Object Technology Series, Addison-Wesley, New York.

MSE Semester-I
Course/Paper: 01 MSE 102

SOFTWARE ARCHITECTURE

Software Architecture terms
Component, Relationship, View, Architectural Styles, Frameworks, Patterns, Methodologies, Processes, Functional and Non-functional Properties of Software Architectures

Enabling Techniques for Software Architecture
Abstraction, Encapsulation, Information Hiding, Modularization Separation of Concerns, Coupling and Cohesion, Sufficiency, Completeness and Primitiveness Separation of Policy and Implementation, Separation of Interface and Implementation

Architectural Styles
Pipes and Filters, Data Abstraction and Object-Orientation, Event-Based, Implicit Invocation, Layered Systems, Repositories, Interpreters, Process Control, Heterogeneous Architectures

Software Implementation - development environment facilities
Code generation, reverse engineering, profiling, software libraries, testing and debugging

Software Quality
Changeability, Efficiency, Interoperability, Reliability, Testability, Reusability, Fault tolerant software

References:
1. M. Shaw: Software Architecture Perspectives on an Emerging Discipline, Prentice- Hall.
2. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, Pearson Education Asia.

MSE Semester-I
Course/Paper: 01 MSE 103

OPTIMIZING COMPILERS

Introduction
Optimizing compiler technology, benchmark and designing a computer, Compiler structure front end, building flow graph, dominator optimization, Inter procedural analysis, dependence optimization, global optimization, limiting resources, instruction scheduling, register allocation, rescheduling.

Flow graphs and local optimization
Building flow graph, structure of data, block, instruction and flow graph. Local optimization information, global anticipated and redundancy information, lifetime analysis optimization in building flow graph, encoding pattern matching

Alias analysis
Level of alias analysis, representing modifies relation, building tag table, two kinds of modification and building flow graph , flow graph sensitive information by optimizations.

Dominator based optimization
Adding optimization to renaming process storing information as well as optimization, consult propagation, computing loop invariant temporaries reshaping expression, global value numbering.

Advanced Techniques
Inter procedure analysis, Inlining procedure, cloning procedure, procedure level optimization dependence based transformation loop unrolling.

Global Optimization
Main structure of optimization phase, theory and algorithms, relation between an expression and its operation, Lazy code motion moving load, store, copy instruction

Limiting resources
Design of LIMIT, peephole optimization, computing conflict graph, register renaming and coalescing, reducing register pressure, spill point computations.

References:
1. Robert Morgan, "Building an Optimizing Compiler", by Elsevier Science (USA)
2. Randy Allen and Ken Kennedy, "Optimizing Compilers for Modern Architecture", by Morgan Kaufmann Publishers

MSE Semester-I
Course/Paper: 01 MSE 104.1

ADVANCED DATA STRUCTURES

Advanced data structures: self-adjustment, persistence and multidimensional trees.
Randomized algorithms: Use of probabilistic inequalities in analysis & applications.
Geometric algorithms: Point location, convex hulls and Voronoi diagrams, Arrangements.
Graph algorithms: Matching and Flows.
Approximation algorithms: Use of Linear programming and primal dual, local search heuristics.
Parallel algorithms: Basic techniques for sorting, searching, merging, list ranking in PRAMs and Interconnection networks.

References:
1. Motwani and Raghavan "Randomized Algorithms", Cambridge University Press
2. Preparata and Shamos "Computational Geometry", Springer Verlag
3. Mehlhorn "Data Structures and Algorithms: 1, Searching and Sorting", Springer Verlag EATCP Monograph on Theoretical Computer Science
4. Papadimitrou and Steiglitz "Combinatorial Optimization", Princeton University Press
5. Joseph Ja'Ja' "Introduction to Parallel Algorithms" Addison-Wesley.
6. Vaizirani "Approximation Algorithms", Springer

MSE Semester-I
Course/Paper: 01 MSE 104.2

HIGH LEVEL SYSTEM DESIGN AND MODELING

Introduction
Design Representation of Digital Systems, levels of abstraction, design methodologies, System level methodologies, System specification and design.

Model Taxonomy
State-Oriented models - finite-state machine, Petri net, Hierarchical concurrent finite state machine; Activity-oriented models - Dataflow graph, flow charts; Heterogeneous model - control/data flow graph, Object oriented model, Program-state machine;

Architectural Taxonomy
Application specific architectures - Controller Architecture, Data path architecture, Finite-state machine with data path; Processors - Complex instruction set Computer, Reduced instruction set Computer; Vector machine - Very long instruction word Computer; Parallel processors.

Embedded Systems Specification Requirements Languages
Characteristics of Conceptual models - Concurrency, State Transitions, Hierarchy, Programming Constructors, Behavioral Completion, Communication, Synchronization, Exception handling, Timing; Comparative features of Specification languages - VHDL, Verilog, HardwareC, State-charts, Esterel; Embedded system specification in speccharts.

A Specification example of Telephone answering machine
Specification capture with spec-charts, Sample test bench, Advantage of executable specifications; Strengths of the PSM model - Hierarchy, State transitions, Programming Constructors, Concurrency, Exception handling, Completion.

System Partitioning
Structural versus functional Partitioning. Partitioning issues - Specification extraction level, Granularity, System Component allocation, Metrics and Estimations, Objective functions and closeness functions, Partitioning Algorithm, Output. Basic Partitioning algorithms - Random mapping, Hierarchical clustering, Multistage Clustering, Group Migration, Radio cut and Simulated Annealing.

References:
1. Specification and Design of Embedded Systems by Daniel D. Gajski, PTR Prentice Hall Englewood New Jersey
2. High Level System Modeling : Specification and Design Methodologies edited by Ronald Waxman, Kluwer Academic Publishers

MSE Semester-I
Course/Paper: 01 MSE 104.3

CLIENT-SERVER BASED IT SOLUTIONS

Client Server Computing
Concept of Client-Server Technology, Client-Server Technology and Heterogeneous Computing, Costs and Benefits of Client Server Computing, Implementation and Scalability

Client Server Model and Software Design
Client-Server Model, Motivation, Terminology and Concepts, Applications, Concurrency in Network, Concurrency in Clients, Concurrency in Servers, Context Switching and Protocol Software Design, Advantages of concurrency.

Architecture and Design of Client Server Model
Multitasking with Processes and Threads, Scheduling, Synchronization, Memory, Communications

Algorithms in Client/Server Software Design
TCP Client algorithms, Socket Interface, Programming a UDP Client; The Conceptual Server algorithm, Basic Types of Servers and their comparisons, Interactive Server algorithms, Concurrent Server algorithms, Problem of Server Deadlock

Portable Client/Server Applications
Architecting Portable Application Code, Architecting Platform-Independent Source- Code, Operating System/Communications/File System independent modules, Client Server Applications Architecting using Frameworks

References:
1. Douglas E. Comer, David L; Stevens, Internetworking with TCP/IP: Client-Server Programming and Applications : Vol III , Prentice Hall of India, New Delhi.
2. Jaffrey D. Schqnk; Client Server Applications and architecture, BPB Novell Press, New Delhi
3. Douglas J. Reilly; Client/Server Developers Guide, Addision Wesley Developer's Press, Masschachusetts

Practical: I Semester

MSE Semester-I
Course/Paper: 01 MSE 201

SOFTWARE ENGINEERING LAB

For the instructor: Assign any two projects two a group of exactly two students covering all of the experiments from given experiment list. Each group is required to prepare the following documents for projects assigned to them and develop the software using software engineering methodology.
1. Problem Analysis and Project Planning Thorough study of the problem- identify project scope, infrastructure.
2. Software Requirement Analysis- Describe the individual Phases/modules of the project deliverables.
3. Data Modeling Use work products – data dictionary, use case diagrams and activity diagrams, build and test lass diagrams, sequence diagrams and add interface to class diagrams.
4. Software Developments and Debugging.
5. Software Testing – Prepare test plan, perform validation testing coverage analysis, memory leaks, develop test case hierarchy, Site check and site monitor.
6. Describe: Relevance of CASE tools, high – end and low – end CASE tools, automated support for data dictionaries, DFD, ER diagrams.

S. No.	List of Experiments	Software Required:
1	Course Registration System	Case Tools: Rational Suite, Win runner, Empirix Languages: C/C++/JDK, JSDK, INTERNET EXPLORER UML Front End: VB, VC++, Developer 2000, .NET Back End: Oracle, MS – Access, SQL
2	Quiz System	
3	Online ticket reservation system	
4	Remote computer monitoring	
5	Students marks analyzing system	
6	Expert system to prescribe the medicines for the given symptoms	
7	Platform assignment system for the trains in a railway station	
8	Stock maintenance	
9	Student Marks Analyzing System	
10	Online Ticket Reservation System	
11	Payroll System	
12	Export System	

Theory: II Semester

MSE Semester-II
Course/Paper: 02 MSE 101

SECURITY ANALYSIS OF SOFTWARE

.

Overview of Computer Security
Threats, risks, vulnerabilities, safeguards, attacks, exploits, Information states, Security at the various states of information- processing, storage and transmission; Definition of security based on current state and reachable states, Comprehensive model of security, Confidentiality, integrity and availability, Risk management, corrective action, risk assessment and physical security.

Access Control
Access control matrix, Access control lists, Capabilities, Role-based access, control and Application dependence.

Security Policies
Types of policies, Role of trust, Information states and procedures, Types of access control, Separation of duties, Application dependence, Importance for automated information systems (AIS) and Security planning Confidentiality Policies - Goals and definitions, Bell-LaPadula model and Multi-level security.
Integrity Policies - Goals and definitions, Information states and procedures, Operating system integrity, Biba model and Clark-Wilson model
Hybrid Policies - Chinese wall model and Role-Based Access Control

Authentication
Passwords, Challenge-response, Biometrics, Location, Combinations and Application to access control/authorization

Malicious Logic
Trojan horses, Computer viruses, Computer worms, Logic bombs, Defenses and countermeasures

Auditing
Auditing mechanisms, Auditing system design, Privacy issues, Trails and logs, Access control issues, Application dependence

Intrusion Detection
Principles, Models, Architecture, Organization and Intrusion response

Network Security
Policy development, Network organization, Firewalls, Availability, Access control issues, Attacks anticipation, Traffic analysis, Public vs private

Administrative policies
Purposes, Back-up policies, E-mail security and privacy policies, Wireless policies, FAX security policies, Internet security policies, Incident response policies, Testing and validation policies, Application development control, Facilities management, Copyright management, Licensing management, Biometrics

access management, Software piracy, Law enforcement issues, assisting investigations, Media destruction/ sanitization/ protection, Security planning, Resources misuse or abuse, Documentation and auditing, Review of controls, Policies installment process, Managers endorsement, user obligations, System test and evaluation, Communication with users, Communication with vendors, Software installation and patches

References:

1. Matt Bishop, "Introduction to Computer Security", by Addison Wesley, 2005.
2. Viega, John, Gary McGraw, Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley, Boston, 2002.
3. Oaks, Scott, Java Security, Second edition, O'Reilly & Associates, Inc., Sebastopol, CA, 2001

MSE Semester-II
Course/Paper: 02 MSE 102

SOFTWARE VERIFICATION, VALIDATION AND TESTING

Introduction
Terminology, evolving nature of area

V & V Limitations
Theoretical foundations: impracticality of testing all data, impracticality of testing all paths, no absolute proof of correctness.

Role of V & V in Software Evolution
Types of Products: requirements, specifications, designs, implementations, changes; V&V objectives: correctness, consistency, necessity, sufficiency, performance.

Software V & V Approaches and their Applicability
Software technical reviews; Software testing: levels of testing - module, integration, system, regression; Testing techniques and their applicability-functional testing and analysis, structural testing and analysis, error-oriented testing and analysis, hybrid approaches, integration strategies, transaction flow analysis, stress analysis, failure analysis, concurrency analysis, performance analysis; Proof of correctness; simulation and prototyping; Requirement tracing.

Software V & V Planning
Identification of V & V Goals; Selection of V & V techniques: requirements, specifications, design, implementations, changes; Organizational responsibilities: development organization, independent test organization; software quality assurance; independent V & V contractor; V & V standards; Integrating V & V approaches; Problem tracking; Tracking test activities; Assessment.

References:

1. Boris Beizer; Software Testing Techniques, John Wiley & Dreamtech.
2. William Perry, Effective Methods for Software Testing, John Wiley & Sons, Inc. New York.
3. Boris Beizer; Black-Box Testing-Techniques of Functional Testing of Software and System, John Wiley & Sons, New York.
4. Marc Ropar, Software Testing, McGraw-Hill Book Co., London.
5. Cem Kaner, Jack Falk, Hung; Nguyen Quoc, Testing Computer Software.

MSE Semester-II
Course/Paper: 02 MSE 103

SOFTWARE QUALITY MANAGEMENT

Software Quality Management
Quality control, quality assurance and quality standards with emphasis on ISO 9000; Functions of software QA organization does in a project; interactions with developers; Quality plans, quality assurance towards quality improvement; Role of independent verification & validation; Total quality management; SEI maturity model; Software metrics.

Basics of measurement
Measurement in everyday life, measurement in software engineering, scope of software metrics, representational theory of measurement, measurement and models, measurement scales, meaningfulness in measurement, goal-based framework for software measurement, classifying software measures, determining what to measure, software measurement validation, empirical investigation, types of investigation, planning and conducting investigations.

Software-metrics data collection and analysis
What is good data, how to define the data, how to collect the data, how to store and extract data, analyzing software-measurement data, frequency distributions, various statistical techniques.

Measuring internal product attributes
Measuring size, aspects of software size, length, functionality and complexity, measuring structure, types of structural measures, control-flow structure, modularity and information flow attributes, data structures.

Metrics for object-oriented systems
The intent of object-oriented metrics, distinguishing characteristics of object-oriented metrics, various object-oriented metric suites – LK suite, CK suite and MOOD metrics

Metrics for component-based systems
The intent of component-based metrics, distinguishing characteristics of component based metrics, various component-based metrics

References:
1. Norman E. Fenton and Shari Lawrence Pfleeger; Software Metrics – A Rigorous and Practical Approach, Thomson Asia Pte., Ltd, Singapore.
2. Stephen H. Kan; Metrics and Models in Software Quality Engineering, Addison Wesley, New York.
3. K. H. Möller and D. J. Paulish; Software Metrics - A Practitioner's Guide to Improved Product Development, Chapman and Hall, London.
4. Mark Lorenz and Jeff Kidd; Object-Oriented Software Metrics, Prentice Hall, New York.
5. Mordechai Ben-Manachem, Garry S. Marliss; Software Quality: Producing Practical, Consistent Software, Vikas Publishing House.
6. Alan Gillies; Software Quality, Theory and Management, Chapman and Hall, London.

MSE Semester-II
Course/Paper: 02 MSE -104.1

ADVANCED DATABASE SYSTEMS

Overview of DBMS, concurrency control, failure recovery. Introduction to distributed data base management systems, Semantic Database Models and Systems, Object- Oriented Database Systems, Relational Extensions: Design Techniques, Extension Techniques Object / Relational Systems: Open ODB, Transaction Management, Interface, OSQL, Odapter, Case Study of an ORDBMS, Related Development, Current Product Scenario. Standard for OODBMS Products and Applications: ODM – Standards, ODMG, Smalltalk Binding, SQL, and User Defined ADT in SQL, Routines, ADT Subtypes and Inheritance, Tables, Procedural Facilities, Other Type Constructions, Generic ADT Packages, Language Bindings.

References:

1. C S R Prabhu,"Object Oriented Data Base Systems" approaches and Architectures,PHI,
2. F. H. Lochousky, DC Tsichritzis"DBMS" NewYork Academic Press.
3. F. H. Lochousky, DC Tsichritzis"Data Models" PHI.
4. C.J.DATE "Introduction to Data Base to Management System" Addison Wesley.
5. N. Goodman, V. Hadzilacos "Concurrency Control and Recovery in Data Base System" Addison Wesley.

MSE Semester-II
Course/Paper: 02 MSE -104.2

DISTRIBUTED OPERATING SYSTEMS

Architecture of distributed systems
Motivation, System Architecture types, Distributed Operating Systems, Issues in Distributed Operating Systems, Communication Network, and Communication Primitives.

Theoretical Foundations
Inherent Limitation of a Distributed Systems, Lamport's Logical Clocks, Vector Clocks, Casual Ordering of messages, Global State, Cuts of a Distributed Comutation, Termination Detection.

Distributed Mutual Exclusion
The classification of Mutual Exclusion Algorithm, requirements of Mutual Exclusion Algorithm, How to measure performance, a simple solution to Distributed Mutual Exclusion, Non token based Algorithm, Lamports Algorithm, Ricart-Agrawala Algorithm, Maekawas Algorithm, A generalized Non Token based Algorithm, Token based Algorithm, Suzuki-Kasami broadcast algorithm, singhals Heuristic Algorithm, Raymonds tree based Algorithms, A comparative performance analysis.

Distributed Deadlock Detection
Preliminaries, Deadlock handling Strategies in Distributed Systems, Issues in Deadlock detection and resolution, Control Organization for Distributed Deadlock Detection, Centralized Deadlock Detection Algorithm, Distributed Deadlock Detection Algorithm, Hierarchical Deadlock Detection Algorithms.

Agreement Protocols
The System Model, A classification agreement problems, solutions to the Byzantine Agreement problem, Application of Agreement Algorithms.

Distributed File Systems
Architecture, Mechanisms for building Distributed File Systems, Design issues, Case studies – The SUN network file system, coda, and the x-kernal logical file system.

References:

1. Mukesh Singhal, Niranjan Shivratri; Advanced Concepts in Operating Systems. McGraw-Hill, New York.
2. George Coulouris, Jean Dollimore and Tim Kindberg; Distributed Systems Concepts and Design, Addison-Wesley, Masschachusetts
3. Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, New York.

MSE Semester-II
Course/Paper: 02 MSE -104.3

EMBEDDED SOFTWARE AND SYSTEMS

Introduction
Real time systems, Characterizing real time systems, software development process

Requirements
Project planning and requirements process, requirements elicitation and structured analysis, object-oriented analysis, formal methods.

Architecture
Architecture, Architectural Properties, Real Time Architecture.\

Design
Real-time systems, Temporal Techniques.

Design – Scheduling
Tasks, Timing and Scheduling, Rate Monotonic Scheduling, Rate Monotonic Analysis

Design Systems
Concurrent Systems, Distributed systems.

Verification and Validation
Verification and Validation, Risk and Failure Analysis, Real-Time Operating Systems. Real time Languages

References:

1. Alan, C. Shaw; Real-Time Systems and Software , John Wiley & Sons, New York.
2. Philip Laplante; Real-time systems design and analysis, an engineer's handbook, IEEE Computer Society Press , New York.

3. J.E. Cooling; Software Design for Real-time Systems , Chapman and Hall, New York.
4. Hassan Gomaa; Software Design Methods for Concurrent and Real-time Systems, Addison-Wesley, Masschachusetts
5. Krishna M. Kavi; Real-time systems, abstractions, languages and design methodologies, IEEE Computer Society Press, New York.

Practical: II Semester

MSE Semester-II
Course/Paper: 02 MSE 201

SOFTWARE TESTING LAB

The student will submit a synopsis at the beginning of the semester for the approval to the University project committee in a specified format. The student will have to present the progress of the work through seminars and progress report. A report must be submitted to the University for Evaluation Purpose at the end of the semester in a specified format.

1. To learn to use the testing tools to carry out the functional testing, load/stress testing.
2. To learn to use the following (or similar) automated testing tools to automate testing:
    a) Win Runner/QTP for functional testing.
    b) Load Runner for Load/Stress testing.
    c) Test Director for test management.

Theory: III Semester

MSE Semester-III
Course/Paper: 03 MSE 101

UNIFIED SOFTWARE CONFIGURATION MANAGEMENT

Software Configuration Management
SCM best practices, SCM tools and process, Dyeing with changing project requirements.

Overview of the Unified Change Management Model
UCM, ClearCase, UCM process overview, defining the Implementation Model, The UCM baseline and Change Model.

Functional Overview of Objects
The Repository, Versioned Object Base, Workspaces, Component Management, Process, Building, Clearmake, Derived Objects, Configuration records

Establishing the Initial SCM Environment
ClearCase Architecture Basics, Defining the Implementation Model, Creating the VOBs, Baseline promotion levels Project Management in ClearCase

Coordinating Multiple Project Teams and Other Scenarios
Organizing large Multi project development efforts, Coordinating cooperating projects, Independent components, Shared components, Multiple Parallel release, Using UCM without Activity-based SCM.

Development Using the UCM Model
A Developer's perspective of UCM, joining a project, making changes, delivering changes to the project, Rebasing your development stream, Dealing with conflicting changes.

Integration, Build and Release
Software Integration, Isolation and integration, Building and Baselining, Staging and release

References:
1. Brian A. White, Software Configuration Management Strategies and Rational Clear Case Addison Wesley, New York.
2. Roger S. Pressman, Software Engineering a Practitioner's Approach, McGraw-Hill, New York.
3. James Rumbaugh, Ivar Jacobson and Grady Booch, The Unified Modeling Language Reference Manual, Addison Wesley, New York.

MSE Semester-III
Course/Paper: 03 MSE 102.1

SOFTWARE RELIABILITY

Software Reliability
Basic Ideas of Software Reliability, Computation of software reliability, Classes ofsoftware reliability Models.

Time Dependent Software Reliability Models
Time between failure reliability Models, Fault Counting Reliability Models

Time Independent Software Reliability Models
Fault injection model of Software Reliability, Input Domain Reliability Model, Orthogonal defect classification, Software availability Models

Software Reliability Modeling
A general procedure for reliability modeling

References:

1. Hoang Pham, Software Reliability, Springer Verlag, New York.
2. Jhon D. Musa, Software Reliability Engineered Testing, Mc. Graw Hill , New York.
3. Doron Reled, Software Reliability Methods, Springer Verlag, New York
4. R. Ramakumar, Reliability Engineering: Fundamentals and Applications, Prentice Hall, New Delhi.

MSE Semester-III
Course/Paper: 03 MSE -102.2

E-BUSINESS

Overview of e-Business
Linking today's Business with tomorrow's technology, e-Business means structural transformation, business design for e-Business, challenge traditional definitions of value, value in terms of customer experience, e-Business communities, major business trends.

Constructing e-Business Design
Self-diagnosis as a first step of e-Business design, Reversing the value chain as a second step of e-Business design, Choosing a narrow focus as a third step of e-Business design – service excellence, operational excellence, continuous innovation excellence; Case studies.

Constructing e-Business Architecture
Issues of application integration, Cross-functional integrated applications, Integrating applications clusters into an e-Business architecture, Aligning the e-Business design with application integration.

Customer Relation Management (CRM)
Why CRM?, Defining CRM, New CRM architecture, Supporting requirements of the next-generation CRM infrastructure, Challenges in CRM implementation, Next generation CRM trends, Manager's roadmap for building a CRM infrastructure.

Selling-Chain Management
Deriving forces for Selling-Chain management, Managing the order acquisition process, Case study of CISCO's Selling-Chain management, Elements of Selling-Chain infrastructure.

Enterprise Resource Planning (ERP)
What is ERP?, Why ERP?, Enterprise architecture planning, ERP usage in the real World, ERP implementation, Future of ERP applications.

Supply Chain Management (SCM)
Defining SCM, Basics of Internet-Enabled SCM, e-Supply chain fusion, Manager's roadmap for SCM.

e-Procurement
Purchasing versus procurement, Operating resource procurement, Case study of open resource procurement at Microsoft, e-Procurement chain management, Next generation integrated procurement applications, Elements of Buy-Side e-Procurement solutions, Elements of Sell-Side e-Procurement solutions, Manager's roadmap for e-Procurement.

Knowledge-Tone Applications
Why knowledge applications and what is it?, Emerging classes of knowledge-tone applications, knowledge-tone usage in the real World, Elements of knowledge-tone architectural framework, Data warehousing, Online analytical processing (OLAP), Roadmap to knowledge-tone framework.

Developing the e-Business Design
Challenges of e-Business strategy creation, Roadmap to moving your company into e-Business.

Translating e-Business Strategy into Action
The overall process – translating strategy into action, e-Business blueprint creation, Basic steps of e-Business blueprint planning, Key elements of a business case, e-Business project planning checklist, Why e-Business initiatives fail?

References:
1. Ravi Kalakota and Marcia Robinson; e-Business- Roadmap for Success; Pearson Education Asia Pte Ltd, Tecmedia, New Delhi.
2. H. Albert Napier, Philip J. Judd, Ollie Rivers, Stuart W. Wagner; Creating a Winning E-Business; Vikas Publishing House Pvt. Ltd., New Delhi.

MSE Semester-III
Course/Paper: 03 MSE -102.3

SOFTWARE REUSE

.

Introduction
Software Reuse and Software Engineering, Concepts and Terms, Software Reuse products, Software Reuse processes, Software Reuse paradigms.

State of the Art and the Practice
Software Reuse Management, Software Reuse Techniques, Aspects of Software Reuse, Organizational Aspects, Technical Aspects and Economic Aspects.

Programming Paradigm and Reusability
Usability Attributes, Representation and Modeling Paradigms, Abstraction and Composition in development paradigm.

Object-Oriented Domain Engineering
Abstraction and Parameterization Techniques, Composition Techniques in Object Orientation.

Application Engineering: Component Storage and Retrieval, Reusable Asset Integration.

Software Reuse Technologies
Component Based Software Engineering, COTS based development, Software Reuse Metrics, Tools for Reusability.

References:
1. Reuse Based Software Engineering Techniques, Organization and Measurement by Hafedh Mili, Ali Mili, Sherif Yacoub and Edward Addy, John Wiley & Sons Inc
2. The Three Rs of Software Automation: Re-engineering, Repository, Reusability by Carma McClure, Prentice Hall New Jersey
3. McClure, Carma L. Software reuse techniques : adding reuse to the system development process / : Prentice Hall
4. Poulin, Jeffrey S. Measuring software reuse : principles, practices, and economic models / Jeffrey S. Poulin. Reading, Mass. : Addison-Wesley

Practical: III Semester

MSE Semester-III
Course/Paper: 03 MSE 201

DISSERTATION

The student will submit a synopsis at the beginning of the semester for the approval from the University project committee in a specified format. Synopsis must be submitted within a two weeks. The first defense, for the dissertation work, should be held with in a one month. Dissertation Report must be submitted in a specified format to the University for Evaluation Purpose.

MSE Semester III
Course/Paper: 03MSE202

SEMINAR

OBJECTIVE
The students are to select one technical topic related its branch for Seminar. The student is to submit the synopsis for assessment and approval. Progress for preparation of the seminar topic would be continuously assessed from time to time. Two periods per week are to be allotted and students are expected to present the seminar Progress. A faculty guide is to be allotted and he / she will guide and monitor the progress of the student and maintain the attendance.

Students have to give a final presentation for 15 minutes on his topic. Students are encouraged to use various teaching aids such as over head projectors, power point presentation and demonstrative models. This will enable them to gain confidence in facing the placement interviews

Theory: IV Semester

MSE Semester-IV
Course/Paper: 04 MSE 101

DISSERTATION

The student will submit a synopsis at the beginning of the semester for the approval from the University project committee in a specified format. Synopsis must be submitted within a two weeks. The first defense, for the dissertation work, should be held with in a one month. Dissertation Report must be submitted in a specified format to the University for Evaluation Purpose.