VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR M.TECH. SOFTWARE ENGINEERING

I Semester

Subject Code	Name of the Subject	Teaching hours/week		Duration of Exam in Hours	Marks for		Total Marks
		Lecture	Practical / Field Work / Assignment/ Tutorials		I.A.	Exam	
12SSE11	Software Engineering	4	2#	3	50	100	150
12SSE12	Advanced Algorithms	4	2*	3	50	100	150
12SSE13	Advances in Database Management Systems	4	2*	3	50	100	150
12SSE14	Web Services	4	2#	3	50	100	150
12SSE15x	Elective-I	4	2	3	50	100	150
12SSE16	Seminar	--	3	--	50	--	50
	Total	**20**	**13**	**15**	**300**	**500**	**800**

Elective – I

12SSE151 – Computer Graphics and visualization
12SSE152 – Computer Systems Performance Analysis
12SSE153 – Cloud Computing

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR M.TECH. SOFTWARE ENGINEERING

II Semester

Subject Code	Name of the Subject	Teaching hours/week		Duration of Exam in Hours	Marks for		Total Marks
		Lecture	Practical / Field Work / Assignment / Tutorials		I.A.	Exam	
12SSE21	Software Project Management	4	2*	3	50	100	150
12SSE22	Metrics and Models in Software Quality Engineering	4	2*	3	50	100	150
12SSE23	Software Testing	4	2#	3	50	100	150
12SSE24	Fault-Tolerant Systems	4	2#	3	50	100	150
12SSE25x	Elective-II	4	2	3	50	100	150
	**Project Phase-I(6 week Duration)						
12SSE26	Seminar	--	3	--	50	--	50
	Total	**20**	**13**	**15**	**300**	**500**	**800**

Elective – II
12SSE251 – Mobile Computing
12SSE252 – Distributed Systems
12SSE253 – Web Engineering
12SSE254 – Service Oriented Architecture
**** Between the II Semester and III Semester after availing a vocation of 2 weeks.**

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR M.TECH. SOFTWARE ENGINEERING

III Semester

Subject Code	Subject	No. of Hrs./Week		Duration of Exam in Hours	Marks for		Total Marks
		Lecture	Field Work / Assignment / Tutorials		I.A.	Exam	
12SSE31	Software Architectures	4	--	3	50	100	150
12SSE32x	Elective-III	4	2	3	50	100	150
12SSE33x	Elective-IV	4	2	3	50	100	150
12SSE34	Project Phase-II		$				
12SSE35	Evaluation of Project Phase – I	–	3	–	50	–	50
	Total	**12**	**07**	**09**	**200**	**300**	**500**

Elective – III
12SSE321 Soft Computing
12SSE322 Information Retrieval
12SSE323 Multimedia Communications

Elective – IV
12SSE331 Distributed Operating Systems
12SSE332 Multicore architecture & Programming
12SSE333 Embedded Computing Systems

$ 3 Days Course work and 3 days for Project work

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR M.TECH. SOFTWARE ENGINEERING

IV Semester

Course Code	Subject	No. of Hrs./Week		Duration of the Exam in Hours	Marks for		Total Marks
		Lecture	Practical / Field Work		I.A.	Exam	
12SSE41	Evaluation of Project Phase – II	-	-	-	50	-	50
12SSE42	Evaluation of Project work – III	-	-	-	50	-	50
12SSE43	Project work evaluation and Viva-voce	-	-	3	–	100+100	200
	Total	**-**	**-**	**03**	**100**	**200**	**300**
Grand Total (I to IV Sem.) : 2400							

Note: Project work shall be continuously evaluated for phase I, phase II and after completion of the project.

Note:

* Lab Classes for any two core subjects are compulsory (practical will be evaluated for 20 marks and internal assessment for 30 marks. Lab journals should be maintained).

For the remaining two core subjects, it can be field work, assignment, tutorials.

1) Project Phase – I : 6 weeks duration shall be carried out between II and III Semesters. Candidates in consultation with the guides shall carryout literature survey / visit to Industries to finalise the topic of dissertation. Evaluation of the same shall be taken up during beginning of III Semester. Total Marks shall be 50. Colleges have to send the synopsis after Phase – I.

2) Project Phase – II : 16 weeks duration. 3 days for project work in a week during III Semester. Evaluation shall be taken during the first two weeks of the IV Semester. Total Marks shall be 50.

3) Project Phase – III : 24 weeks duration in IV Semester. Evaluation shall be taken up during the middle of IV Semester. Total Marks shall be 50. At the end of the Semester Project Work Evaluation and Viva-Voce Examinations shall be conducted. Total Marks shall be 50 + 50 + 100 = 200 (50 marks for guide, 50 marks for external and 100 for viva-voce).

Marks of Evaluation of Project:

- The Marks of Project Phase – I shall be sent to the University along with III Semester I.A. Marks of other subjects.
- The I.A. Marks of Project Phase – II & III shall be sent to the University along with Project Work report at the end of the Semester.

4) During the final viva, students have to submit all the reports.

5) The Project Valuation and Viva-Voce will be conducted by a committee consisting of the following:

a) Head of the Department (Chairman)

b) Guide

c) Two Examiners appointed by the university. (out of two external examiners at least one should be present).

M. Tech Software engineering

I SEMESTER

SOFTWARE ENGINEERING

Subject Code: 12SSE11	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Introduction and Review of Software Process Models : FAQs about Software Engineering; Professional and ethical responsibility;Software process models; Process iteration; Process activities; Computer-Aided Software Engineering.

2. Rapid Software Development, Software Reuse: Agile methods; Extreme programming; Rapid application development. Reuse landscape; Design patterns; Generator-based reuse; Application frameworks; Application system reuse.

3. CBSE: Components and component models; Component-Based Software Engineering (CBSE).

4. Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution.

5. Verification and Validation: Planning verification and validation; Software inspections; System testing; Component testing; Test case design; Test automation.

6. Critical Systems, Specifications of Critical Systems: What are critical systems? Examples; System dependability, availability and reliability. Risk-driven specification; Safety specification; Security specification; Software reliability specification.

7. Critical Systems Development, Validation: Dependable processes; Dependable programming; Fault tolerance and fault tolerant architectures. Reliability validation; Safety assurance; Security assessment; Safety and dependability cases.

8. Distributed Systems Architecture: Multiprocessor architectures; Client-Server architectures; Distributed object architectures; Inter-Organizational distributed computing.

9. Real-Time Software Design: Real-time systems; System design; Monitoring and control systems; Data acquisition systems.

TEXT BOOK:

1. Ian Sommerville: Software Engineering, 8th Edition, Pearson Education, 2007.

REFERENCE BOOKS:

1. Roger.S.Pressman: Software Engineering-A Practitioners approach, 7th Edition, McGraw-Hill, 2007.

ADVANCED ALGORITHMS

Subject Code: 12SSE12`	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Review of Analysis Techniques: Growth of Functions: Asymptotic notations; Standard notations and common functions; Recurrences and Solution of Recurrence equations- The substitution method, The recurrence – tree method, The master method; Amortized Analysis: Aggregate, Accounting and Potential Methods.

2. Graph Algorithms: Bellman - Ford Algorithm; Single source shortest paths in a DAG; Johnson's Algorithm for sparse graphs; Flow networks and Ford-Fulkerson method; Maximum bipartite matching.

3. Internet Algorithms: Search engines; Ranking web pages; Hashing; Caching, content delivery, and consistent hashing.

4. Number -Theoretic Algorithms: Elementary notions; GCD; Modular Arithmetic; Solving modular linear equations; The Chinese remainder theorem; Powers of an element; RSA cryptosystem; Primality testing; Integer factorization.

5. String-Matching Algorithms: Naïve string Matching; Rabin - Karp algorithm; String matching with finite automata; Knuth-Morris-Pratt algorithm; Boyer – Moore algorithms.

6. Probabilistic and Randomized Algorithms: Probabilistic algorithms; Randomizing deterministic algorithms, Monte Carlo and Las Vegas algorithms; Probabilistic numeric algorithms.

Laboratory Work:

1. Design, develop, and run a program in any language to implement the Bellman-Ford algorithm and determine its performance.

2. Design, develop, and run a program in any language to implement Johnson's algorithm and determine its performance.

3. Design, develop, and run a program in any language to implement a Monte Carlo algorithm to test the primality of a given integer and determine its performance.

4. Design, develop, and run a program in any language to solve the string matching problem using naïve approach and the KMP algorithm and compare their performances.

5. Design, develop, and run a program in any language to solve modular linear equations.

6. Design, develop, and run a program in any language to implement a Page Ranking algorithm.

TEXT BOOKS:

1. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3rd Edition, Prentice-Hall of India, 2010.

2. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning, 2002.

REFERENCE BOOKS:

1. Ellis Horowitz, Sartaj Sahni, S.Rajasekharan: Fundamentals of Computer Algorithms, 2nd Edition, Universities press, 2007.

Advances in Database Management Systems

Subject Code: 12SSE13 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. **Review of Relational Data Model and Relational Database Constraints:** Relational model concepts; Relational model constraints and relational database schemas; Update operations, transactions and dealing with constraint violations.

2. **Object and Object-Relational Databases:** Overview of Object-Oriented Concepts – Objects, Encapsulation, Type and class hierarchies, complex objects; Object model of ODMG, Object definition Language ODL; Object Query Language OQL; Overview of C++ language binding; Conceptual design of Object database. Overview of object relational features of SQL; Object-relational features of Oracle; Implementation and related issues for extended type systems; The nested relational model.

3. **Parallel and Distributed Databases:** Architectures for parallel databases; Parallel query evaluation; Parallelizing individual operations; Parallel query optimizations; Introduction to distributed databases; Distributed DBMS architectures; Storing data in a Distributed DBMS; Distributed catalog

management; Distributed Query processing; Updating distributed data; Distributed transactions; Distributed Concurrency control and Recovery.

4. **Data Warehousing, Decision Support and Data Mining:** Introduction to decision support; OLAP, multidimensional model; Window queries in SQL; Finding answers quickly; Implementation techniques for OLAP; Data Warehousing; Views and Decision support; View materialization; Maintaining materialized views. Introduction to Data Mining; Counting co-occurrences; Mining for rules; Tree-structured rules; Clustering; Similarity search over sequences; Incremental mining and data streams; Additional data mining tasks.

5. **More Recent Applications:** Web Database Programming using PHP, Deductive databases; Mobile databases; Geographical Information Systems; Genome data management- basic concepts.

Laboratory Work:

(The following tasks can be implemented on Oracle or any other suitable RDBMS with support for Object features)

1. Demonstrate object relational features of SQL.

2. Demonstrate object relational features of SQL.
3. Develop a database application to demonstrate storing and retrieving of BLOB and CLOB objects.

4. Develop a database application to demonstrate the representation of multivalued attributes, and the use of nested tables to represent complex objects. Write suitable queries to demonstrate their use.

5. Design and develop a suitable Student Database application. One of the attributes to me maintained is the attendance of a student in each subject for which he/she has enrolled. Using TRIGGERS, write active rules to do the following:

   a. Whenever the attendance is updated, check if the attendance is less than 85%; if so, notify the Head of the Department concerned.

   b. Whenever, the marks in an Internal Assessment Test are entered, check if the marks are less than 40%; if so, notify the Head of the Department concerned.

6. Design, develop, and execute a program in a language of your choice to implement any one algorithm for mining association rules. Run the program against any large database available in the public domain and discuss the results.

TEXT BOOKS:

1. Elmasri and Navathe: Fundamentals of Database Systems, Pearson Education, 2007.

2. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw-Hill, 2003.

REFERENCE BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan: Database System Concepts, 6th Edition, McGraw Hill, 2010.

WEB SERVICES

Subject Code: 12SSE14	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Introduction: The basics of Web Services; An example; Next generation of the Web; Interacting with Web Services; The Technology of Web Services; XML for business collaboration: ebXML; Web Services versus other technologies; Additional technologies.

2. XML: An example; Instance and schema; Processing XML documents; Namespaces; Transformation; XML specifications and information.

3. WSDL: Basics; WSDL elements; The extensible WSDL framework; Importing WSDL elements; WSDL- Related Namespaces; Extensions for binding to SOAP.

4. SOAP: Example; The SOAP specifications; SOAP message processing; SOAP use of Namespaces; Changes in the V1.2 draft; SOAP Multipart MIME; Attachments; SOAP I the context of existing systems; Future directions.

5. UDDI Registry: The UDDI organization; The concepts underlying UDDI; How UDDI works? UDDI SOAP APIs; Usage scenarios; Using WSDL with UDDI; UDDI for private use; UDDI support for SOAP, Complex business relationships, and UNICODE.

6. EBXML: Overview of ebXML; ebXML specifications.

7. Implementation: Implementation architectures; Major implementation streams; .NET; J2EE Application Servers.

8. Some More Issues: Transaction Management; Security; Practical considerations; Future standards.

TEXT BOOKS:

1. Eric Newcomer: Understanding Web Services XML, WSDL, SOAP, and UDDI, Pearson , 2002.

2. James McGovern et al: Java Web Services Architecture, Elsevier, 2003.

REFERENCE BOOKS:

1. Aaron E. Walsh: UDDI, SOAP, and WSDL – The Web Services Specification Reference Book, Prentice Hall PTR, 2000.

2. Relevant web Sites.

COMPUTER GRAPHICS AND VISUALIZATION

Subject Code: 12SSE151 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction: Applications of computer graphics; A graphics system; Images: Physical and synthetic; Imaging systems; The synthetic camera model; The programmer's interface; Graphics architectures; Programmable pipelines; Performance characteristics. Graphics Programming: The Sierpinski gasket; Programming two-dimensional applications.

2. The OpenGL: The OpenGL API; Primitives and attributes; Color; Viewing; Control functions; The Gasket program; Polygons and recursion; The three-dimensional gasket; Plotting implicit functions.

3. Input and Interaction: Interaction; Input devices; Clients and servers; Display lists; Display lists and modeling; Programming event-driven input; Menus; Picking; A simple CAD program; Building interactive models; Animating interactive programs; Design of interactive programs; Logic operations.

4. Geometric Objects and Transformations: Scalars, points, and vectors; Three-dimensional primitives; Coordinate systems and frames; Modeling a colored cube; Affine transformations; Rotation, translation and scaling. Transformations in homogeneous coordinates; Concatenation of transformations; OpenGL transformation matrices; Interfaces to three-dimensional applications; Quaternions.

5. Viewing: Classical and computer viewing; Viewing with a computer; Positioning of the camera; Simple projections; Projections in OpenGL; Hidden-surface removal; Interactive mesh displays; Parallel-projection matrices; Perspective-projection matrices; Projections and shadows.

6. Lighting and Shading: Light and matter; Light sources; The Phong lighting model; Computation of vectors; Polygonal shading; Approximation of a sphere by recursive subdivisions; Light sources in OpenGL; Specification of materials in OpenGL; Shading of the sphere model; Global illumination.

7. Curves and surfaces: Representation of curves and surfaces; Design criteria; Parametric cubic polynomial curves; Interpolation; Hermite curves and surfaces; Bezier curves and surfaces; Cubic B-Splines; General B-Splines; Rendering curves and surfaces; Curves and surfaces in OpenGL.

TEXT BOOK:

1. Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th Edition, Pearson, 2009.

REFERENCE BOOKS:

1. Donald Hearn and Pauline Baker: Computer Graphics- OpenGL Version, 2nd Edition, Pearson, 2004.

2. F.S. Hill,Jr.: "Computer Graphics Using OpenGL",2nd Edition, Pearson, 2001.

3. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Computer Graphics, Addison-wesley 1997.

COMPUTER SYSTEMS PERFORMANCE ANALYSIS

Subject Code: 12SSE152	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Introduction: The art of Performance Evaluation; Common Mistakes in Performance Evaluation, A Systematic Approach to Performance Evaluation, Selecting an Evaluation Technique, Selecting Performance Metrics, Commonly used Performance Metrics, Utility Classification of Performance Metrics, Setting Performance Requirements.

2. Workloads, Workload Selection and Characterization: Types of Work loads, addition instructions, Instruction mixes, Kernels; Synthetic programs, Application benchmarks, Popular benchmarks. Work load Selection: Services exercised, level of detail; Representativeness; Timeliness, Other considerations in workload selection. Work load characterization Techniques: Terminology; Averaging, Specifying dispersion, Single Parameter Histograms, Multi Parameter Histograms, Principle Component Analysis, Markov Models, Clustering.

3. Monitors, Program Execution Monitors and Accounting Logs: Monitors: Terminology and classification; Software and hardware monitors, Software versus hardware monitors, Firmware and hybrid monitors, Distributed System Monitors, Program Execution Monitors and Accounting Logs, Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and Interpretation of Accounting log data, Using accounting logs to answer commonly asked questions.

4. Capacity Planning and Benchmarking: Steps in capacity planning and management; Problems in Capacity Planning; Common Mistakes in Benchmarking; Benchmarking Games; Load Drivers; Remote-Terminal Emulation; Components of an RTE; Limitations of RTEs.

5. Experimental Design and Analysis: Introduction: Terminology, Common mistakes in experiments, Types of experimental designs, 2k Factorial Designs, Concepts, Computation of effects, Sign table method for computing effects; Allocation of variance; General 2k Factorial Designs, General full factorial designs with k factors: Model, Analysis of a General Design, Informal Methods.

6. Queuing Models: Introduction: Queuing Notation; Rules for all Queues; Little's Law, Types of Stochastic Process. Analysis of Single Queue: Birth-Death Processes; M/M/1 Queue; M/M/m Queue; M/M/m/B Queue with finite buffers; Results for other M/M/1 Queuing Systems. Queuing Networks: Open and Closed Queuing Networks; Product form networks, queuing Network models of Computer Systems. Operational Laws: Utilization Law; Forced Flow Law; Little's Law; General Response Time Law; Interactive Response Time Law; Bottleneck Analysis; Mean Value Analysis and Related Techniques; Analysis of Open Queuing Networks; Mean Value Analysis; Approximate MVA; Balanced Job Bounds; Convolution Algorithm, Distribution of Jobs in a System, Convolution Algorithm for Computing G(N), Computing Performance using G(N), Timesharing Systems, Hierarchical Decomposition of Large Queuing Networks: Load Dependent Service Centres, Hierarchichal Decomposition, Limitations of Queuing Theory.

TEXT BOOK:

1. Raj Jain: The Art of Computer Systems Performance Analysis, John Wiley and Sons, 2007.

REFERENCE BOOKS:

1. Paul J Fortier, Howard E Michel: computer Systems Performance Evaluation and prediction, Elsevier, 2003.

2. Trivedi K S: Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd Edition, Wiley India, 2001.

Cloud Computing

Subject Code: 12SSE153 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. **Introduction :** Business and IT perspective, Cloud and virtualization, Cloud services requirements, cloud and dynamic infrastructure, cloud computing characteristics, cloud adoption.
2. **Cloud models :** Cloud characteristics, Measured Service, Cloud models, security in a public cloud, public verses private clouds, cloud infrastructure self service.
3. **Cloud at a service :** Gamut of cloud solutions, principal technologies, cloud strategy, cloud design and implementation using SOA, Conceptual cloud model, cloud service demand.
4. **Cloud solutions :** Cloud ecosystem, cloud business process management, cloud service management, cloud stack, computing on demand, cloud sourcing.
5. **Cloud offerings :** Cloud analytics, Testing under cloud, information security, virtual desktop infrastructure, Storage cloud.

6. **Cloud management :** Resiliency, Provisioning, Asset management, cloud governance, high availability and disaster recovery, charging models, usage reporting, billing and metering.
7. **Cloud virtualization technology :** Virtualization defined, virtualization benefits, server virtualization, virtualization for x86 architecture, Hypervisor management software, Logical partitioning, VIO server, Virtual infrastructure requirements**.** Storage virtualization, storage area networks, network attached storage, cloud server virtualization, virtualized data center.
8. **Cloud and SOA :** SOA journey to infrastructure, SOA and cloud, SOA defined, SOA defined, SOA and IAAS, SOA based cloud infrastructure steps, SOA business and IT services.

TEXT BOOKS:

1. Cloud Computing by Dr. Kumar Saurabh, Wiley India, 2011.

Reference Books

1. Michael Miller, Cloud Computing: Web based applications that change the way you work and collaborate online, Que publishing , August 2009
2. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On Demand computing applications and data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008.

II SEMESTER

SOFTWARE PROJECT MANAGEMENT

Subject Code: 12SSE21	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Agile development: What is agile? Agility and cost of change; What is an agile process? Extreme programming; Other agile process models.

2. Web Application Design: Web application design quality; Design quality and design pyramid; Interface design; Aesthetic design; Content design; Architecture design; Navigation design; Component-level design; Object-oriented hypermedia design method.

3. Formal Modeling and verification: The cleanroom strategy; Functional specification; Cleanroom design; Cleanroom testing; Formal methods: Concepts; Applying mathematical notation for formal specification; Formal specification languages.

4. Software Project Management: The management spectrum; The management of people, product, process and project; The W5HH Principle; Critical practices.

5. Estimation for Software Projects: Software project estimation; Decomposition techniques, Examples; Empirical estimation models; Estimation for Object-Oriented projects; Specialized estimation techniques; The make / buy decision.

6. Software Project Scheduling: Basic concepts and principles of project scheduling; Defining task set and task network; Scheduling; Earned value analysis.

7. Risk Management: Reactive versus proactive strategies; Software risks; risk identification; Risk projection; Risk refinement; Risk mitigation, monitoring and management; The RMMM plan.

8. Maintenance and Reengineering: Software maintenance; Software supportability; Reengineering; Business process reengineering; Software reengineering; Reverse engineering; Restructuring; Forward engineering; The economics of reengineering.

9. Software Process Improvement (SPI): Approaches to SPI; Maturity models; The SPI process; The CMMI; The People CMM; Other SPI frameworks: SPICE, Bootstrap, PSP and TSP, ISO; SPI return on investment.

10. Software Configuration Management (SCM): Basic concepts; SCM repository; The SCM process; Configuration management for web applications; SCM standards.

11. Product Metrics: A framework for product metrics; Metrics for requirements model, design

model, source code, testing and maintenance; Design metrics for web applications.

12. Process and Project Metrics: Basic concepts; Software measurement; Metrics for software quality; Integrating metrics within the software process; Metrics for small organizations; Establishing a software metrics program.

TEXT BOOKS:

1. Roger S. Pressman: Software Engineering: A Practitioner's Approach, 7th Edition, McGraw Hill, 2007.

REFERENCE BOOKS:

1. Ian Sommerville: Software Engineering, 8th Edition, Pearson, 2007.

METRICS AND MODELS IN SOFTWARE QUALITY ENGINEERING

Subject Code: 12SSE22 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction: Quality: Popular views; Quality: Professional views; Software quality; Total quality management.

2. Overview of Software Quality Metrics: Product quality metrics; In-process quality metrics; Metrics for software maintenance; Examples of metrics programs; Collecting software engineering data.

3. Applying the 7 Basic Quality Tools in Software Development: Ishikawa's seven basic tools; Checklist; Pareto diagram; Histogram; Run charts; Scatter diagram; Control chart; Cause-and-effect diagram; Relations diagram.

4. Defect Removal Effectiveness: Review; A closer look at defect removal effectiveness; Defect removal effectiveness and quality planning; Cost effectiveness of phase defect removal; Defect removal effectiveness and process maturity level.

5. The Rayleigh Model: Reliability models; The Rayleigh model; Basic assumptions; Reliability and predictive validity.

6. Exponential Distribution and Reliability Growth Models: The exponential model; Reliability growth models; Model assumptions; Criteria for model evaluation; Modeling process; Test compression factor; Estimating the distribution of total defects over time.

7. Quality Management Models: The Rayleigh model framework; The code integration pattern; The PTR submodel; The PTR arrival / backlog projection model; Reliability growth models; Criteria for model evaluation; In-process metrics and reports; Orthogonal defect classification.

8. In-Process Metrics for Software Testing: In-process metrics for software testing; In-process metrics and quality management; Possible metrics for acceptance testing to evaluate vendor-developed software; When is the product good enough to ship?

9. Metrics and Lessons Learned for Object-Oriented Projects: Object-oriented concepts and constructs; Design and complexity metrics; Productivity metrics; Quality and quality management metrics; Lessons learned for OO projects.

10. Availability Metrics: Definition and measurements of system availability; Reliability, availability, and defect rate; Collecting customer outage data for quality improvement; In-process metrics foroutage and availability.

11. Measuring and Analyzing Customer Satisfaction: Customer satisfaction surveys; Analyzig satisfaction data; Satisfaction with Company; How good is good enough?

12. Conducting In-Process Quality Assessments: The preparation phase; The evaluation phase; The summarization phase; Recommendations and risk mitigation.

Laboratory Work:

1. Design, develop and execute a program in a language of your choice to determine phase-wise effectiveness metrics from the matrix of defect data organized as Defect Origin by Where Found. Experiment with different sets of simulated data or data available from public domains and discuss the impact early defect removal efforts on software quality.

2. Design, develop and execute a program in a language of your choice to implement the Rayleigh model, plot the graph, and to estimate the latent error rate using the model. Experiment with different sets of simulated data or data available from public domains and discuss the results.

3. Design, develop and execute a program in a language of your choice to implement the Jelinski-Moranda model, plot the graph, and to estimate the software reliability using the model. Experiment with different sets of simulated data or data available from public domains and discuss the results.

4. Design, develop and execute a program in a language of your choice to implement the Musa-Okumoto model, plot the graph, and to estimate the software reliability using the model. Experiment with different sets of simulated data or data available from public domains and discuss the results.

5. Design, develop and execute a program in a language of your choice to implement the Delayed S model, plot the graph, and to estimate the software reliability using the model. Experiment with different sets of simulated data or data available from public domains and discuss the results.

TEXT BOOKS:

1. Stephan H. Kan: Metrics and Models in Software Quality Engineering, 2nd Edition, Pearson, 2003.

REFERENCE BOOKS:

1. Fenton N. E. , S. L. Pfleeger: Software Metrics: A Rigorous Approach, 2nd Edition, Cengage Learning, 1997.

2. Jeff Tian: Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, John Wiley and Sons Inc., 2005.

SOFTWARE TESTING

Subject Code: 12SSE23 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Basics of Software Testing and Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. Examples: Generalized pseudocode, The triangle problem, The NextDate function, The commission problem, The SATM (Simple Automatic Teller Machine) problem.

2. Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the NextDate function, Test cases for the commission problem, Guidelines and observations.

3. Data Flow Testing: Definition-Use testing, Slice-based testing, Guidelines and observations.

4. Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing.

5. Integration Testing: A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations, Case study.

6. System Testing: Threads, Basic concepts for requirements specification, Finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example.

7. Interaction Testing: Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing,.

8. Issues in Object-Oriented Testing: Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism, Levels of object-oriented testing, GUI testing, Dataflow testing for object-oriented software, Examples.

9. Class Testing: Methods as units, Classes as units.

10. Object-Oriented Integration Testing: UML support for integration testing, MM-paths for object-oriented software, A framework for object-oriented dataflow integration testing.

10. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program.

11. Object-Oriented System Testing: Currency converter UML description, UML-based system testing, Statechart-based system testing.

12. Exploratory Testing: The context-driven school, Exploring exploratory testing, Exploring a familiar example, Exploratory and context-driven testing observations.

13. Model-Based Testing: Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing.

14. Test-Driven Development: Test-then-code cycles, Automated test execution, Java and JUnit example, Remaining questions, Pros, cons, and open questions of TDD, Retrospective on MDD versus TDD.

15. A Closer Look at All Pairs Testing: The all-pairs technique, A closer look at NIST study, Appropriate applications for all pairs testing, Recommendations for all pairs testing.

16. Software Testing Excellence: Craftsmanship, Best practice of software testing, Top 10 best practices for software testing excellence, Mapping best practices to diverse projects.

Laboratory Work:

1. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive at least 10 different test cases, execute these test cases and discuss the test results.

2. Design, develop, code and run the program in any suitable language to solve the NextDate problem. Analyze it from the perspective of decision table-based testing, derive at least 10 different test cases, execute these test cases and discuss the test results.

3. Design, develop, code and run the program in any suitable object-oriented language to solve the calendar problem. Analyze it from the perspective of OO testing, derive test cases to test the method that increment the date and the method that increments the month., execute these test cases and discuss the test results.

4. Design, develop, code and run the program in any suitable object-oriented language to solve the currency converter problem. Analyze it from the perspective of use case-based system testing, derive appropriate system test cases., execute these test cases and discuss the test results.

TEXT BOOKS:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2008.

REFERENCE BOOKS:

1. Aditya P Mathur: Foundations of Software Testing, Pearson, 2008.

2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, John Wiley & Sons, 2008.

3. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 2nd Edition, Pearson, 2007.

FAULT-TOLERANT SYSTEMS

Subject Code: 12SSE24 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction: Fault classification; Types of Redundancy; Basic measures of FaultTolerance.

2. Hardware Fault Tolerance: The rate of hardware failures; Failure rate, Reliability, and Mean Time To Failure; Canonical and Resilient Structures; Other Reliability Evaluation Techniques; Fault-Tolerance – Processor-Level techniques; Byzantine Failures.

3. Information Redundancy: Coding; Resilient Disk Systems; Data Replication; Algorithm-Based Fault Tolerance.

4. Fault-Tolerant Networks: Measures of Resilience; Common Network Topologies and Their Resilience; Fault-Tolerant Routing.

5. Software Fault Tolerance: Acceptance Tests; Single-Version Fault Tolerance; N-Version Programming; Recovery Block Approach; Preconditions, Postconditions, and Assertions; Exception Handling; Software Reliability Models; Fault-Tolerant Remote Procedure Calls.

6. Checkpointing: What is Checkpointing? Checkpoint Level; Optimal Checkpointing – An Analytical Model; Cache-Aided Rollback Error Recovery; Checkpointing in Distributed Systems; Checkpointing in Shared Memory Systems; Checkpointing in Real-Time Systems; Other uses of Checkpointing.

7. Defect Tolerance in VLSI Circuits: Manufacturing Defects and Circuit Faults; Probability of Failure and Critical Areas; Basic Yield Models; Yield Enhancement through Redundancy.

8. Fault Detection in Cryptographic Systems: Overview of Ciphers; Security Attacks through Fault Injection; Countermeasures.

9. Case Studies: Non-Stop Systems; Stratus Systems; Cassini Command and Data Sub-System; IBM G5; IBM Sysplex; Itanium.

TEXT BOOKS:

1. Israel Koren, C. Mani Krishna: Fault-Tolerant Systems, Elsevier, 2007.

REFERENCE BOOKS:

1. D. K. Pradhan (Ed): Fault Tolerant Computer Systems Design, Prentice Hall, 1996.

2. K. S. Trivedi: Probability, Statistics with Reliability, Queuing and Computer Science Applications, John Wiley, 2002.

MOBILE COMPUTING

Subject Code: 12SSE251 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Overview: Mobile communications; Mobile computing; Mobile computing architecture; Mobile devices; Mobile system networks; Data dissemination; Mobility management; Mobile phones, Digital Music Players, Handheld Pocket Computers, Handheld Devices, Operating Systems, Smart Systems, Limitations of Mobile Devices, Automotive Systems.

2. GSM and Similar Architectures: GSM – Services and System Architectures, Radio Interfaces, Protocols, Localization, Calling, Handover, General Packet Radio Service, High-speed circuit-switched data, DECT.

3. Wireless Medium Access Control and CDMA – based Communication: Medium Access Control, Introduction to CDMA – based Systems, OFDM.

4. Mobile IP Network Layer: IP and Mobile IP Network Layers Packet Delivery and Handover Management, Registration, Tunneling and Encapsulation, Route Optimization, Dynamic Host Configuration Protocol.

5. Mobile Transport Layer: Indirect TCP, Snooping TCP, Mobile TCP, Other Methods of TCP – layer Transmission for Mobile Networks.

6. Databases: Database Hoarding Techniques, Data Caching, Client – Server Computing and Adaptation, Transactional Models, Query Processing, Data Recovery Process, Issues relating to Quality of Service.

7. Data Dissemination and Broadcasting Systems: Communication Asymmetry, Classification of Data – Delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing Techniques, Digital Audio Broadcasting, Digital video Broadcasting.

8. Data Synchronization in Mobile Computing Systems: Synchronization, Synchronization Protocols, SyncML – Synchronization Language for Mobile Computing, Synchronized Multimedia Markup Language (SMIL).

9. Mobile Devices, Server and Management: Mobile agent, Application Server, Gateways, Portals, Service Discovery, Device Management, Mobile File Systems, Security.

10. Wireless LAN, Mobile Internet Connectivity and Personal Area Network: Wireless LAN (WiFi) Architecture and Protocol Layers, WAP 1.1 and WAP 2.0 Architectures, Bluetooth – enabled Devices Network, Zigbee.

11. Mobile Application languages – XML, Java, J2ME and JavaCard: Introduction, XML, JAVA, Java 2 Micro Edition (J2ME), JavaCard.

12. Mobile Operating Systems: Operating System, PalmOS, Windows CE, Symbian OS, Linux for Mobile devices.

TEXT BOOK:

1. Raj Kamal: Mobile Computing, Oxford University Press, 2007.

REFERENCES:

1. Asoke Talukder, Roopa R Yavaga: Mobile Computing – Technology, Applications and Service Creation, 2nd Edition, Tata McGraw Hill, 2009.

2. Reza B'Far: Mobile Computing Principles – Designing and Developing Mobile Applications with UML and XML, Cambridge University press, 5th Edition, 2006.

3. Schiller: Mobile Communication, Pearson, 2004.

DISTRIBUTED SYSTEMS

Subject Code: 12SSE252 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. **Characterization of Distributed Systems and System Models:** Introduction, Examples of distributed systems, Resource sharing and the Web, Challenges, Architectural models, Fundamental models.

2. **Networking and Internetworking:** Types of Networks, Networks principles, Internet protocols

3. **Interprocess Communication:** Introduction, The API for the Internet protocols, External data representation and marshalling, Client -Server communication, Group communication, Case study: Interprocess communication in UNIX

4. **Distributed Objects and Remote Invocation:** Communication between distributed objects, Remote procedure call, events and notifications

5. **Operating System Support and Security:** The Operating system layer, protection, processes and threads, communication and invocation , operating system architecture

6. **Transactions and Concurrency Control:** Transactions, nested transactions, locks, optimistic concurrency control, timestamp ordering, comparison of methods for concurrency control

7. **Distributed Shared Memory:** Design and Implementation issues, sequential consistency and Ivy.

TEXT BOOKS:

1. George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems, Concept and Design, 3rd Edition, Pearson Education, 2005.

REFERENCE BOOKS:

1. Sukumar Ghosh: Distributed Systems, An Algorithmic Approach, Chapman &Hall / CRC, 2007.

2. Pradeep K. Sinha: Distributed Operating Systems, Concepts and Design, PHI, 2007.

3. Randy Chow, Theodore Johnson: Distributed Operating Systems and Algorithm Analysis, Pearson, 2009.

WEB ENGINEERING

Subject Code: 12SSE253 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction: Motivation, Categories of web applications, Characteristics of web applications.

2. Requirements Engineering: Introduction, Fundamentals, RE specifics in web engineering, Principles of RE for web applications, Adapting RE methods to web application development, Outlook.

3. Modeling Web Application: Introduction, Fundamentals, Modeling specifics in web engineering, Modeling requirements, Content modeling, Hypertext modeling, Presentation modeling, Customization modeling, Methods and tools, Outlook.

4. Web Application Architectures: Introduction, Fundamentals, Specifics of web application architectures, Components of a generic web application architecture, Layered architectures, Data-aspect architectures.

5. Technology-Aware Web Application Design: Introduction, Web design from an evolutionary perspective, Presentation design, Interaction design, Functional design, Outlook.

6. Technologies for Web Applications: Introduction, Fundamentals, Client/Server communication on the web, Client side technologies, Document-specific technologies, Server-side technologies, Outlook.

7. Testing Web Applications: Introduction, Fundamentals, Testing specifics in web engineering, Test approaches, Test scheme, Test methods and techniques, Test automation, Outlook.

8. Operation and Maintenance of Web Applications: Introduction, Challenges following the launch of a web application, Content management, Usage analysis, Outlook.

9. Web Project Management: From software project management to web project management, Challenges in web project management, Managing web teams, Managing the development process of a web application, Outlook.

10. The Web Application Development Process: Motivation, Fundamentals, Requirements for a web application development process, Analysis of the rational unified process, Analysis of extreme programming, Outlook.

11. Usability of Web Applications: Motivation, What is usability? What characterizes the usability of web applications? Design guidelines, Web usability engineering methods, Web usability engineering trends, Outlook.

12. Performance of Web Applications: Introduction, What is performance? What characterizes performance of web applications, System definition and indicators, Characterizing the work load, Analytical techniques, Representing and interpreting results, Performance optimization methods, Outlook.

13. Security for web Applications: Introduction, Aspects of security, Encryption, digital signatures, and certificates, Secure Client/Server interaction, Client security issues, Service provider security issues, Outlook.

14. The Semantic Web: Fundamentals of the semantic web, Technological concepts, Specifics of semantic web applications, Tools, Outlook.

TEXT BOOK:

1. Gerti Kappel, Birgit Proll, SiegfriedReich, Werner Retschitzegeer (Editors): Web Engineering, Wiley India, 2007.

REFERENCE BOOKS:

1. Roger Pressman, David Lowe: Web Engineering: A Practitioner's Approach, McGraw Hill, 2008.

SERVICE ORIENTED ARCHITECTURE

Subject Code: 12SSE254 **I.A. Marks : 50**
Hours/Week : 04 **Exam Hours: 03**
Total Hours : 52 **Exam Marks: 100**

1. Roots of SOA – Characteristics of SOA - Comparing SOA to client-server and distributed internet architectures – Anatomy of SOA- How components in an SOA interrelate - Principles of service orientation.

2. Web services – Service descriptions – Messaging with SOAP –Message exchange Patterns – Coordination –Atomic Transactions – Business activities – Orchestration – Choreography - Service layer abstraction – Application Service Layer – Business Service Layer – Orchestration Service Layer.

3. Service oriented analysis – Business-centric SOA – Deriving business services- service modeling - Service Oriented Design – WSDL basics – SOAP basics – SOA composition guidelines – Entity-centric business service design – Application service design – Taskcentric business service design.

4. SOA platform basics – SOA support in J2EE – Java API for XML-based web services (JAX-WS) - Java architecture for XML binding (JAXB) – Java API for XML Registries (JAXR) - Java API for XML based RPC (JAX-RPC)- Web Services Interoperability Technologies (WSIT) - SOA support in .NET – Common Language Runtime - ASP.NET web forms – ASP.NET web services – Web Services Enhancements (WSE).

5. WS-BPEL basics – WS-Coordination overview - WS-Choreography, WS-Policy, WSSecurity.

TEXT BOOKS:

1. Thomas Erl, “Service-Oriented Architecture: Concepts, Technology, and Design”, Pearson Education, 2005.

REFERENCES:

1. Thomas Erl, “SOA Principles of Service Design “(The Prentice Hall Service-Oriented Computing Series from Thomas Erl), 2005.

2. Newcomer, Lomow, “Understanding SOA with Web Services”, Pearson Education, 2005.

3. Sandeep Chatterjee, James Webber, “Developing Enterprise Web Services, An Architect’s Guide”, Pearson Education, 2005.

4. Dan Woods and Thomas Mattern, “Enterprise SOA Designing IT for Business Innovation” O’REILLY, First Edition, 2006

Semester III

SOFTWARE ARCHITECTURES

Subject Code: 12SSE31	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Review of Basic Concepts: What is a pattern? What makes a pattern? Pattern Categories; Relationships between patterns; Pattern description; Patterns and software architecture; What software architecture is and what it is not; Other points of view; Architectural patterns, reference models and reference architectures; Importance of software architecture; Architectural structures and views.

2. Designing the Architecture: Architecture in the life cycle; Designing the architecture; Forming the team structure; Creating a skeletal system.

3. Reconstructing Software Architectures: Introduction; Informal extraction; Database construction; View fusion; Reconstruction; Examples.

4. Software Product Lines: Introduction; What makes software product lines work? Scoping; Architectures for product lines; What makes software product lines difficult?

5. Building Systems from Off-the-Shelf Components: Impact of components on architecture; Architectural mismatch; Component-based design as search; ASEILM example.

6. Some Design Patterns: Introduction; Management: Command processor, View handler; Communication: Forwarder-Receiver, Client-Dispatcher-Receiver, Publisher-Subscriber.

7. Pattern Systems: What is a Pattern System? Pattern classification; Pattern selection; Pattern systems as implementation guidelines; The evolution of pattern systems.

8. Case Studies: Key Word In Context; Instrumentation Software; Mobile Robotics; Cruise Control; The World Wide Web: A case study in interoperability; J2ee / EJB: A case study in industry-standard computing infrastructure.

TEXT BOOKS:

1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson Education, 2003.

2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007.

3. Mary Shaw and David Garlan: Software Architecture-Perspectives on an Emerging Discipline, PHI Learning, 2007.

REFERENCE BOOKS:

1. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable Object-Oriented Software, Pearson Education, 1995.

2. Web site for Patterns: http://www.hillside.net/patterns/

SOFT COMPUTING

Subject Code : 12SSE321 IA Marks : 50
No of Lecture Hrs/Week : 4 Exam hours : 3
Total No of Lecture Hours : 52 Exam Marks : 100

1. Neural Networks: History, overview of biological Neuro-system, Mathematical Models of Neurons, ANN architecture, Learning rules, Learning Paradigms-Supervised, Unsupervised and reinforcement Learning, ANN training Algorithmsperceptions, Training rules, Delta, Back Propagation Algorithm, Multilayer Perceptron Model, Hopfield Networks, Associative Memories, Applications of Artificial Neural Networks.

2. Fuzzy Logic: Introduction to Fuzzy Logic, Classical and Fuzzy Sets: Overview of Classical Sets, Membership Function, Fuzzy rule generation.

3. Operations on Fuzzy Sets, Fuzzy Arithmetic, Fuzzy Logic, Uncertainty based Information: Compliment, Intersections, Unions, Combinations of Operations, Aggregation Operations .Fuzzy Numbers, Linguistic Variables, Arithmetic Operations on Intervals & Numbers, Lattice of Fuzzy Numbers, Fuzzy Equations. Classical Logic, Multivalued Logics, Fuzzy Propositions, Fuzzy Qualifiers, Linguistic Hedges. Information & Uncertainty, Nonspecificity of Fuzzy & Crisp Sets, Fuzziness of Fuzzy Sets..

4. Introduction of Neuro-Fuzzy Systems: Architecture of Neuro Fuzzy Networks, Applications of Fuzzy Logic: Medicine, Economics etc.

5. Genetic Algorithms: An Overview, GA in problem solving, Implementation of GA.

TEXT BOOKS:

1. Anderson J.A.: An Introduction to Neural Networks, PHI, 1999.

2. Hertz J. Krogh, R.G. Palmer: Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.

3. G.J. Klir & B. Yuan: Fuzzy Sets & Fuzzy Logic, PHI, 1995.

4. Melanie Mitchell: An Introduction to Genetic Algorithm, PHI, 1998.

INFORMATION RETRIEVAL

Subject Code: 12SSE322 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction: Motivation, Basic concepts, Past, present, and future, The retrieval process.

2. Modeling: Introduction, A taxonomy of information retrieval models, Retrieval: Adhoc and filtering, A formal characterization of IR models, Classic information retrieval, Alternative set theoretic models, Alternative algebraic models, Alternative probabilistic models, Structured text retrieval models, Models for browsing.

3. Retrieval Evaluation: Introduction, Retrieval performance evaluation, Reference collections.

4. Query Languages: Introduction, keyword-based querying, Pattern matching, Structural queries, Query protocols.

5. Query Operations: Introduction, User relevance feedback, Automatic local analysis, Automatic global analysis.

6. Text and Multimedia Languages and Properties: Introduction, Metadata, Text, Markup languages, Multimedia.

7. Text Operations: Introduction, Document preprocessing, Document clustering, Text compression, Comparing text compression techniques.

8. Indexing and Searching: Introduction; Inverted Files; Other indices for text; Boolean queries; Sequential searching; Pattern matching; Structural queries; Compression.

9. Parallel and Distributed IR: Introduction, Parallel IR, Distributed IR.

10. User Interfaces and Visualization: Introduction, Human-Computer interaction, The information access process, Starting pints, Query specification, Context, Using relevance judgments, Interface support for the search process.

11. Searching the Web: Introduction, Challenges, Characterizing the web, Search engines, Browsing, Metasearchers, Finding the needle in the haystack, Searching using hyperlinks.

TEXT BOOKS:

1. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: Modern Information Retrieval, Pearson, 1999.

REFERENCE BOOKS:

1. David A. Grossman, Ophir Frieder: Information Retrieval Algorithms and Heuristics, 2nd Edition, Springer, 2004.

2. William B. Frakes, Ricardo Baeza-Yates (Editors): Information Retrieval Data Structures and Algorithms, Pearson Education, 1992.

MULTIMEDIA COMMUNICATIONS

Subject Code: 12SSE323 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction to Multimedia Communications: Introduction, Human communication model, Evolution and convergence, Technology framework, Standardization framework.

2. Framework for Multimedia Standardization: Introduction, Standardization activities, Standards to build a new global information infrastructure, Standardization processes on multimedia communications, ITU-T mediacom2004 framework for multimedia, ISO/IEC MPEG-21 multimedia framework, IETF multimedia Internet standards.

3. Application Layer: Introduction, ITU applications, MPEG applications, Mobile servers and applications, Universal multimedia access.

4. Middleware Layer: Introduction to middleware for multimedia, Media coding, Media Streaming, Infrastructure for multimedia content distribution.

5. Network Layer: Introduction, QoS in Network Multimedia Systems.

TEXT BOOKS:

1. K.R. Rao, Zoran S. Bojkovic, Dragorad A. Milovanovic: Introduction to Multimedia Communications – Applications, Middleware, Networking, Wiley India, 2006.

REFERENCE BOOKS:

1. Fred Halsall: Multimedia Communications – Applications, Networks, Protocols, and Standards, Pearson, 2001.

2. Nalin K Sharad: Multimedia information Networking, PHI Learning, 2002.

3. Ralf Steinmetz, Klara Narstedt: Multimedia Fundamentals: Volume 1-Media Coding and Content Processing, 2nd Edition, PHI Learning, 2003.

4. Prabhat K. Andleigh, Kiran Thakrar: Multimedia Systems Design, PHI, Learning, 2003.

DISTRIBUTED OPERATING SYSTEMS

Subject Code: 12SSE331	I.A. Marks : 50
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

1. Fundamentals: What is Distributed Computing Systems? Evolution of Distributed Computing System; Distributed Computing System Models; What is Distributed Operating System? Issues in Designing a Distributed Operating System; Introduction to Distributed Computing Environment (DCE).

2. Message Passing: Introduction, Desirable features of a Good Message Passing System, Issues in IPC by Message Passing, Synchronization, Buffering, Multi-datagram Messages, Encoding and Decoding of Message Data, Process Addressing, Failure Handling, Group Communication, Case Study: 4.3 BSD UNIX IPC Mechanism.

3. Remote Procedure Calls: Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC.

4. Distributed Shared Memory: Introduction, General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM.

5. Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms.

6.Resource Management: Introduction, Desirable Features of a Good Global Scheduling Algorithm, Task Assignment Approach, Load – Balancing Approach, Load – Sharing Approach

7. Process Management: Introduction, Process Migration, Threads.

8. Distributed File Systems: Introduction, Desirable Features of a Good Distributed File System, File models, File–Accessing Models, File – Sharing Semantics, File – Caching Schemes, File Replication, Fault Tolerance, Atomic Transactions, Design Principles.

TEXT BOOK:

1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.

REFERENCE BOOK:

1 Andrew S. Tanenbaum: Distributed Operating Systems, Pearson Education, 2002.

MULTICORE ARCHITECTURE AND PROGRAMMING

Subject Code: 12SSE332 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction to Multi-core Architecture

Motivation for Concurrency in software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-core Architectures from Hyper- Threading Technology, Multi-threading on Single-Core versus Multi-Core Platforms Understanding Performance, Amdahl's Law, Growing Returns: Gustafson's Law.

2. System Overview of Threading

Defining Threads, System View of Threads, Threading above the Operating System, Threads inside the OS, Threads inside the Hardware, What Happens When a Thread Is Created, Application

Programming Models and Threading, Virtual Environment: VMs and Platforms, Runtime Virtualization, System Virtualization.

3. Fundamental Concepts of Parallel Programming

Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Challenges You'll Face, Parallel Programming Patterns, A Motivating Problem: Error Diffusion, Analysis of the Error Diffusion Algorithm, An Alternate Approach: Parallel Error Diffusion, Other Alternatives.

4. Threading and Parallel Programming Constructs

Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control- based Concepts, Fence, Barrier, Implementation-dependent Threading Features.

5. Threading APIs

Threading APls for Microsoft Windows, Win32/MFC Thread APls, Threading APls for Microsoft. NET Framework, Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads, Managing Threads, Thread Synchronization, Signaling, Compilation and Linking.

6. OpenMP: A Portable Solution for Threading

Challenges in Threading a Loop, Loop-carried Dependence, Data-race Conditions, Managing Shared and Private Data, Loop Scheduling and Portioning, Effective Use of Reductions, Minimizing Threading Overhead, Work-sharing Sections, Performance-oriented Programming, Using Barrier and No wait, Interleaving Single-thread and Multi-thread Execution, Data Copy-in and Copy-out, Protecting Updates of Shared Variables, Intel Task queuing Extension to OpenMP, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, performance.

7. Solutions to Common Parallel Programming Problems

Too Many Threads, Data Races, Deadlocks, and Live Locks, Deadlock, Heavily Contended Locks, Priority Inversion, Solutions for Heavily Contended Locks, Non-blocking Algorithms, ABA Problem, Cache Line Ping-ponging, Memory Reclamation Problem, Recommendations, Thread-safe Functions and Libraries, Memory Issues, Bandwidth, Working in the Cache, Memory Contention, Cache-related Issues, False Sharing, Memory Consistency, Current IA-32 Architecture, Itanium Architecture, High-level Languages, Avoiding Pipeline Stalls on IA-32,Data Organization for High Performance.

Text Book

1. Multicore Programming , Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts , Intel Press , 2006

EMBEDDED COMPUTING SYSTEMS

Subject Code: 12SSE333 I.A. Marks : 50
Hours/Week : 04 Exam Hours: 03
Total Hours : 52 Exam Marks: 100

1. Introduction to Embedded Systems: Embedded systems; Processor embedded into a system; Embedded hardware units and devices in a system; Embedded software in a system; Examples of embedded systems; Embedded System-on-Chip (SoC) and use of VLSI circuit design technology; Complex systems design and processors; Design process in embedded system. Formalization of system design; Design process and design examples; Classification of embedded systems; Skills required for an embedded system designer.

2. Devices: I/O types and examples; Serial communication devices; Parallel device ports; Sophisticated interfacing features in device ports. Wireless devices; Timer and counting devices; Watchdog timer; Real time clock.

3. Communication Buses for Device Networks: Networked embedded systems; Serial bus communication protocols; Parallel bus device protocols; Internet enabled systems; Wireless and mobile system protocols.

4. Device Drivers and Interrupts Service Mechanism: Device access without interrupts; ISR concept; Interrupt sources; Interrupt servicing mechanism; Multiple interrupts; Context and the periods for context-switching, interrupt latency and deadline; Classification of processors' interrupt service mechanism from context-saving angle; Direct memory access; Device drivers programming.

5. Program Modeling Concepts, Processes, Threads, and Tasks: Program models; DFG models; State machine programming models for event controlled program flow; Modeling of multiprocessor systems. Multiple processes in an application; Multiple threads in an application; Tasks and task states; Task and data; Distinctions between functions, ISRs and tasks.

6. Real-time Operating systems: Operating System services; Process management; Timer functions; Event functions; Memory management; Device, file and I/O sub-systems management; Interrupt routines in RTOS environment and handling of interrupt source calls. Real-Time Operating Systems; Basic design using an RTOS; RTOS task scheduling models, interrupt latency and response times of the tasks as performance metrics; OS security issues.

7. Embedded Software Development, Tools: Introduction; Host and target machines; Linking and locating software; Getting embedded software in to the target system; Issues in hardware software design and co-design; Testing on host machine; Simulators; Laboratory tools.

TEXT BOOKS:

1 Rajkamal: Embedded Systems Architecture, Programming and Design, 2nd Edition, Tata McGraw Hill, 2008.

REFERENCE BOOKS:

1 Wayne Wolf: Computers as Components Principles of Embedded Computer System Design, 2nd Edition, Elsevier, 2008.

2. Steve Heath: Embedded Systems Design, 2nd Edition, Elsevier, 2003.

4 Dr. K.V.K.K. Prasad: Embedded/Real-Time Systems: Concepts, Design and Programming – The Ultimate Reference, Dreamtech Press/Wiley India, 2007.

4. Michael J.Point: Embedded C, Pearson, 2002.