## VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

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

#### I Semester

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

**Elective – I**

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

---

1
## II Semester

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Teaching hours/week</th>
<th>Duration of Exam in Hours</th>
<th>Marks for Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>12SSE21</td>
<td>Software Project Management</td>
<td>4</td>
<td>2* 3</td>
<td>50 100 150</td>
</tr>
<tr>
<td>12SSE22</td>
<td>Metrics and Models in Software Quality Engineering</td>
<td>4</td>
<td>2* 3</td>
<td>50 100 150</td>
</tr>
<tr>
<td>12SSE23</td>
<td>Software Testing</td>
<td>4</td>
<td>2# 3</td>
<td>50 100 150</td>
</tr>
<tr>
<td>12SSE24</td>
<td>Fault-Tolerant Systems</td>
<td>4</td>
<td>2# 3</td>
<td>50 100 150</td>
</tr>
<tr>
<td>12SSE25x</td>
<td>Elective-II</td>
<td>4</td>
<td>2 3</td>
<td>50 100 150</td>
</tr>
<tr>
<td></td>
<td><strong>Project Phase-I(6 week Duration)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12SSE26</td>
<td>Seminar</td>
<td>--</td>
<td>3</td>
<td>50 -- 50</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td>20</td>
<td>13 15</td>
<td>300 500 800</td>
</tr>
</tbody>
</table>

### 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.**
### III Semester

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject</th>
<th>No. of Hrs./Week</th>
<th>Duration of Exam in Hours</th>
<th>Marks for</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lecture</td>
<td>Field Work / Assignment / Tutorials</td>
<td>I.A.</td>
<td>Exam</td>
</tr>
<tr>
<td>12SSE31</td>
<td>Software Architectures</td>
<td>4</td>
<td>--</td>
<td>3</td>
<td>50</td>
</tr>
<tr>
<td>12SSE32x</td>
<td>Elective-III</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>50</td>
</tr>
<tr>
<td>12SSE33x</td>
<td>Elective-IV</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>50</td>
</tr>
<tr>
<td>12SSE34</td>
<td>Project Phase-II</td>
<td></td>
<td>$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12SSE35</td>
<td>Evaluation of Project Phase – I</td>
<td>–</td>
<td>3</td>
<td>–</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>12</td>
<td>07</td>
<td>09</td>
<td>200</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Subject</th>
<th>No. of Hrs./Week</th>
<th>Duration of the Exam in Hours</th>
<th>Marks for</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lecture</td>
<td>Practical / Field Work</td>
<td>I.A.</td>
<td>Exam</td>
</tr>
<tr>
<td>12SSE41</td>
<td>Evaluation of Project Phase – II</td>
<td>-</td>
<td>-</td>
<td>50</td>
<td>-</td>
</tr>
<tr>
<td>12SSE42</td>
<td>Evaluation of Project work – III</td>
<td>-</td>
<td>-</td>
<td>50</td>
<td>-</td>
</tr>
<tr>
<td>12SSE43</td>
<td>Project work evaluation and Viva-voce</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>100+100</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>-</td>
<td>03</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

| 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).
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.


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:

REFERENCE BOOKS:

ADVANCED ALGORITHMS

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


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:


REFERENCE BOOKS:


Advances in Database Management Systems

Subject Code: 12SSE13
Hours/Week : 04
Total Hours : 52

I.A. Marks : 50
Exam Hours: 03
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:

REFERENCE BOOKS:

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 in 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.

TEXT BOOKS:


REFERENCE BOOKS:


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:


REFERENCE BOOKS:


COMPUTER SYSTEMS PERFORMANCE ANALYSIS

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


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.


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:**


**REFERENCE BOOKS:**


---

### Cloud Computing

<table>
<thead>
<tr>
<th>Subject Code: 12SSE153</th>
<th>I.A. Marks : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week : 04</td>
<td>Exam Hours: 03</td>
</tr>
<tr>
<td>Total Hours : 52</td>
<td>Exam Marks: 100</td>
</tr>
</tbody>
</table>

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
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:**

**REFERENCE BOOKS:**

**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 for outage and availability.

11. **Measuring and Analyzing Customer Satisfaction**: Customer satisfaction surveys; Analyzing 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:**


**REFERENCE BOOKS:**

SOFTWARE TESTING

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


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.


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. 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.


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:


REFERENCE BOOKS:


**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:**


**REFERENCE BOOKS:**

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.


**TEXT BOOK:**


**REFERENCES:**


**DISTRIBUTED SYSTEMS**

Subject Code: 12SSE252

| I.A. Marks : 50 |
| Exam Hours: 03 |
| Exam Marks: 100 |

| Hours/Week : 04 |
| Total Hours : 52 |


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

22
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:**

**REFERENCE BOOKS:**
WEB ENGINEERING

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


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


**TEXT BOOK:**


**REFERENCE BOOKS:**


TEXT BOOKS:


REFERENCES:


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:**


**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


TEXT BOOKS:

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.


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 points, 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:

REFERENCE BOOKS:


MULTIMEDIA COMMUNICATIONS

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


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.


**TEXT BOOKS:**


**REFERENCE BOOKS:**


**DISTRIBUTED OPERATING SYSTEMS**

Subject Code: 12SSE331

<table>
<thead>
<tr>
<th>I.A. Marks</th>
<th>Exam Hours</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>03</td>
<td>100</td>
</tr>
</tbody>
</table>


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.


**TEXT BOOK:**


**REFERENCE BOOK:**


**MULTICORE ARCHITECTURE AND PROGRAMMING**

Subject Code: 12SSE332
Hours/Week : 04
Total Hours : 52

I.A. Marks : 50
Exam Hours: 03
Exam Marks: 100

1. **Introduction to Multi-core Architecture**


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
3. **Fundamental Concepts of Parallel Programming**


4. **Threading and Parallel Programming Constructs**


5. **Threading APIs**

   Threading APIs for Microsoft Windows, Win32/MFC Thread APIs, Threading APIs 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**


7. **Solutions to Common Parallel Programming Problems**


**Text Book**

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 into the target system; Issues in hardware software design and co-design; Testing on host machine; Simulators; Laboratory tools.

TEXT BOOKS:

REFERENCE BOOKS: