### DEPARTMENT OF SOFTWARE ENGINEERING

**M.Tech. (SOFTWARE ENGINEERING)**

**CURRICULUM – 2013-2014**

<table>
<thead>
<tr>
<th>COURSE CODE</th>
<th>COURSE NAME</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I SEMESTER</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MA2011</td>
<td>Stochastic Processes &amp; Queuing Theory</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2001</td>
<td>Software Engineering</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>SE2002</td>
<td>Pattern Oriented Software Architecture</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>SE2003</td>
<td>Software Project Management</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Elective – I</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>II SEMESTER</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE2004</td>
<td>Agile Software Process</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>SE2005</td>
<td>Software Testing</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>SE2006</td>
<td>Software Process Maturity Model</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Elective– II</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Elective-III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>III SEMESTER</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Elective – IV</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Elective – V</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Elective –VI</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Elective –VII (Interdisciplinary)</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2047</td>
<td>Seminar</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>PROJECT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE2049</td>
<td>Project Work Phase – I</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td><strong>IV SEMESTER</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE2050</td>
<td>Project Work Phase – II</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td><strong>TOTAL CREDITS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL CREDITS TO BE EARNED FOR THE AWARD OF M.TECH DEGREE: 71**

**CONTACT HOUR/CREDIT:**

L: Lecture Hours per week  \( L \)
T: Tutorial Hours per week  \( T \)
P: Practical Hours per week  \( P \)
C: Credit  \( C \)
### ELECTIVES FOR FIRST SEMESTER

<table>
<thead>
<tr>
<th>COURSE CODE</th>
<th>COURSE NAME</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE2101</td>
<td>Database Management System</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2102</td>
<td>Principles of Compiler Design and Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2103</td>
<td>Object Oriented Programming (C++, JAVA)</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2104</td>
<td>Computer Networks &amp; Network Security</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

### ELECTIVES FOR SECOND SEMESTER

<table>
<thead>
<tr>
<th>COURSE CODE</th>
<th>COURSE NAME</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE2105</td>
<td>Distributed Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2106</td>
<td>E-Commerce</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2107</td>
<td>Software Reliability</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2108</td>
<td>Software Agents</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2109</td>
<td>Design Patterns</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2110</td>
<td>Software Measurements and Metrics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2111</td>
<td>Data Warehousing and Data Mining</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2112</td>
<td>Internet Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

### ELECTIVES FOR THIRD SEMESTER

<table>
<thead>
<tr>
<th>COURSE CODE</th>
<th>COURSE NAME</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE2113</td>
<td>Software Quality Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2114</td>
<td>Component Based System Design</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2115</td>
<td>Human Interface System Design</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2116</td>
<td>Service Oriented Architecture</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2117</td>
<td>Software Configuration Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2118</td>
<td>Decision Support Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2119</td>
<td>Web Services</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2120</td>
<td>Language Technologies</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2121</td>
<td>Analysis of Software Artifacts</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SE2122</td>
<td>Cloud Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
### COURSE CODE | COURSE NAME | L | T | P | C
---|---|---|---|---|---
SE2123 | Personal Software Process | 3 | 0 | 0 | 3
SE2124 | Embedded System | 3 | 0 | 0 | 3

### Supportive Course

| COURSE CODE | COURSE NAME | L | T | P | C |
---|---|---|---|---|---|
MA2011 | Stochastic Processes & Queuing Theory | 3 | 0 | 0 | 3

### Inter Disciplinary Electives

| Course code | Course Name | L | T | P | C |
---|---|---|---|---|---|
( Elective is offered in Third Semester) | | 3 | 0 | 0 | 3

**NOTE:**
Students have to register for the courses as per the following guidelines:

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Category</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Core courses</td>
<td>12 (3 courses)</td>
</tr>
<tr>
<td>2.</td>
<td>Program Elective courses</td>
<td>18 (in I to III semesters)</td>
</tr>
<tr>
<td>3.</td>
<td>Interdisciplinary elective courses (any one program elective from other programs)</td>
<td>3 (One course to be taken in Semester I or II or III)</td>
</tr>
<tr>
<td>4.</td>
<td>Supportive courses - mandatory</td>
<td>3 (One course to be taken in Semester I or II or III)</td>
</tr>
<tr>
<td>5.</td>
<td>Seminar</td>
<td>---</td>
</tr>
<tr>
<td>6.</td>
<td>Project work</td>
<td>---</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# SEMESTER-I

<table>
<thead>
<tr>
<th>MA2011</th>
<th>STOCHASTIC PROCESSES &amp; QUEUEING THEORY</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nil</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PURPOSE**

To impart knowledge on probability concepts to study their applications in stochastic processes & queueing theory

**OBJECTIVES**

1. Compute the characteristics of the random variable given the probabilities
2. Understand and apply various distributions
3. Solve cases of different Stochastic processes along with their properties.
4. Use discrete time finite state Markov chains
5. Gain sufficient knowledge in principles of queueing theory

## UNIT I-RANDOM VARIABLES (9 hours)

One dimensional and two dimensional Random Variables – Characteristics of Random Variables : Expectation, Moments.

## UNIT II-THEORETICAL DISTRIBUTIONS (9 hours)

Discrete : Binomial, Poisson, Negative Binomial, Geometric, Uniform Distributions.
Continuous: Uniform, Exponential, Erlang and Gamma, Weibull Distributions.

## UNIT III-STOCHASTIC PROCESSES (9 hours)

Classification of Stochastic Processes – Bernoulli process – Poisson process – Pure birth process – Birth and Death process.

## UNIT IV-MARKOV CHAINS (9 hours)

UNIT V – QUEUING THEORY (9 hours)

Introduction – Characteristics of Markovian Single server and Multi server queuing models [(M/M/1) : (∞ / FIFO), (M/M/1) : (N / FIFO), (M/M/s) : (∞ / FIFO)] – M/G/1 Queuing System – Pollaczek Khinchin formula.

REFERENCES


<table>
<thead>
<tr>
<th>SE2001</th>
<th>SOFTWARE ENGINEERING</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Contact Hours - 75</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

PURPOSE

The main purpose of this course is to impart knowledge on the basic principles of software development life cycle.

INSTRUCTIONAL OBJECTIVES

1. To understand the software life cycle models
2. To understand the importance of the software development process
3. To understand the importance of modeling and modeling languages
4. To design and develop correct and robust software products
5. To understand business requirements pertaining to Software development

UNIT I – INTRODUCTION (9 hours)

UNIT II - SOFTWARE REQUIREMENTS AND ANALYSIS (9 hours)
Requirements Engineering-Establishing the Groundwork-Eliciting Requirements-Developing use cases-Building the requirements model-Negotiating, validating Requirements-Requirements Analysis-Requirements Modeling Strategies.

UNIT III - SOFTWARE DESIGN (9 hours)

UNIT IV - SOFTWARE IMPLEMENTATION (9 hours)
Structured coding Techniques- Coding Styles-Standards and Guidelines-Documentation Guidelines-Modern Programming Language Features: Type checking-User defined data types-Data Abstraction-Exception Handling-Concurrency Mechanism.

UNIT V - SOFTWARE TESTING AND MAINTENANC (9 hours)

LIST OF EXPERIMENTS (30 hours)
Using Rational Rose do the following for a given source code.
1. Understand SRS concept and its documentation.
2. Design a Use Case Diagram for the application.
3. Design an Activity Diagram.
4. Design a Sequence, Collaboration, and Class Diagram.
5. Source Code analysis.
6. Testing the application by Rational Functional Tester.
7. Understand Maintenance activity (change, configuration management) and document it.

REFERENCES

<table>
<thead>
<tr>
<th>SE2002</th>
<th>PATTERN ORIENTED SOFTWARE ARCHITECTURE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 75</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nil</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
The course gives an insight of the most commonly used software architecture and design patterns and their applications

INSTRUCTIONAL OBJECTIVES
1. The students get basic knowledge of patterns and description of patterns
2. To understand basic architectural patterns
3. To get an insight on the design patterns and mining.

UNIT I-INTRODUCTION TO SOFTWARE ARCHITECTURE (9 hours)
UNIT II-DESIGN PATTERNS & PATTERN SYSTEM (9 hours)

UNIT III-COMMUNITY, MINING, CONCURRENT & NETWORKED (9 hours)
Roots – Community – Pattern Mining - Organizing and Indexing – Methods and tools – Algorithm – Data Structures and Patterns – Formalizing Patterns, Concurrent and Networked Objects, Service Access and Configuration Patterns

UNIT IV-EVENT HANDLING & SYNCHRONIZATION PATTERNS (9 hours)

UNIT V-CONCURRENCY & WEAVING PATTERNS (9 hours)
Objects – Active, Monitor, Half- Sync, Async, Leader/ Followers, Threads, Weaving – Individual Patterns, Middleware, Concurrency and Networking, Patterns Language Vs Pattern System. Past, Present and Future of Patterns.

PRACTICAL: The laboratory shall include development of systems by applying the Software Engineering principles and methods for specific applications (30 hours)

REFERENCES
The course gives an insight of the most commonly used software architecture and design patterns and their applications.

**INSTRUCTIONAL OBJECTIVES**

1. To understand Software Project Models and Software Management Concepts.
2. To understand the various methods of Cost Estimation.
3. To Study about Software Quality Management.
4. To Study about Software Metrics.
5. To understand Project Evaluation.

**UNIT I - PROJECT CONCEPTS AND ITS MANAGEMENT** (12 hours)


**UNIT II - COST ESTIMATION** (12 hours)


**UNIT III - SOFTWARE QUALITY MANAGEMENT** (12 hours)


**UNIT IV - SOFTWARE MANAGEMENT AND METRICS** (12 hours)


UNIT V - PROJECT EVALUATION AND EMERGING TRENDS (12 hours)

REFERENCES

<table>
<thead>
<tr>
<th>ELECTIVE - I</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Contact Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Students to choose one Elective course from the list of courses mentioned in the curriculum.
SEMESTER – II

SE2004

AGILE SOFTWARE PROCESS

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hours : 60</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Prerequisites:</td>
<td>Software Engineering Principles</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
This course imparts knowledge to students in the basic concepts of Agile Software Process, methodology and its development.

OBJECTIVES
1. To understand the basic concepts of Agile Software Process.
2. To gain knowledge in the area of various Agile Methodologies.
3. To develop Agile Software Process
4. To know the principles of Agile Testing

UNIT I-INTRODUCTION (12 hours)
Software is new product development – Iterative development – Risk-Driven and Client-Driven iterative planning – Time boxed iterative development – During the iteration, No changes from external stakeholders – Evolutionary and adaptive development - Evolutionary requirements analysis – Early “Top Ten” high-level requirements and skilful analysis – Evolutionary and adaptive planning – Incremental delivery – Evolutionary delivery – The most common mistake – Specific iterative and Evolutionary methods.

UNIT II-AGILE AND ITS SIGNIFICANCE (12 hours)
UNIT III-AGILE METHODOLOGY (12 hours)
Method overview – Lifecycle – Work products, Roles and Practices values –
Common mistakes and misunderstandings – Sample projects – Process mixtures –
Adoption strategies – Fact versus fantasy – Strengths versus “Other” history.

UNIT IV-CASE STUDY (12 hours)
Evo – Practice Tips.

UNIT V-AGILE PRACTICING AND TESTING (12 hours)
Project management – Environment – Requirements – Test – The agile alliances –
The manifesto – Supporting the values – Agile testing – Nine principles and six
concrete practices for testing on agile teams.

REFERENCES
6. www.qualitytree.com

<table>
<thead>
<tr>
<th>SE2005</th>
<th>SOFTWARE TESTING</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 75</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nil</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
The student should develop the basic skills in software testing by implementing various strategies of software testing in their project. They need to bring out the ways and means of controlling and monitoring testing activity.

OBJECTIVES
1. The students learn to apply the testing strategies and methodologies in their projects
2. To understand test management strategies and tools for testing
3. A keen awareness on the open problems in software testing and maintenance
UNIT I-TESTING BASICS  
(9 hours)

UNIT II-TEST CASE DESIGN  
(9 hours)
Introduction to testing design strategies – The smarter tester – Test case design strategies – Using black box approach to test case design – Random testing – Equivalence class partitioning – Boundary value analysis – Other black box test design approaches – Black box testing and COTS – Using white box approach to test design – Test adequacy criteria – Coverage and control flow graphs – Covering code logic – Paths – Their role in white box based test design – Additional white box test design approaches – Evaluating test adequacy criteria.

UNIT III-LEVELS OF TESTING  
(9 hours)
The need for levels of testing – Unit test – Unit test planning – Designing the unit tests – The class as a testable unit – The test harness – Running the unit tests and recording results – Integration tests – Designing integration tests – Integration test planning – System test – The different types – Regression testing – Alpha, beta and acceptance tests.

UNIT IV-TEST MANAGEMENT  
(9 hours)

UNIT V-CONTROLLING AND MONITORING  
(9 hours)

PRACTICAL  
(30 hours)
LIST OF EXPERIMENTS
Using Testing Tool do the following Experiments for the given source code
1. Test Principles and Concepts
2. Test Management
3. Build the Test Environment
4. Test Planning Process
5. Test Design
6. Performing Tests
7. Defect Tracking and Correction
8. Acceptance Testing
9. Status of Testing
10. Test Reporting

REFERENCES

<table>
<thead>
<tr>
<th>SE2006</th>
<th>SOFTWARE PROCESS MATURITY MODEL</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 60</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering Principles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
To know about the software process and Software Process Maturity Models

OBJECTIVES
1. To study about various Software process maturity models
2. To study about how to assess software process
3. To know about the key process areas of the software process
4. To study about software improvement sequences
UNIT I - INTRODUCTION  
(12 hours)  

UNIT II - CMM  
(12 hours)  

UNIT III - CMMI  
(12 hours)  
Evolution of CMMI – CMMI Framework – CMMI for Development – Capability level – Maturity levels – Case Study

UNIT IV - TMM  
(12 hours)  

UNIT V - AGILE MATURITY MODEL  
(12 hours)  

REFERENCES
<table>
<thead>
<tr>
<th>Semester II</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTIVE - II</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Contact Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester III</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTIVE - III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Contact Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SEMESTER III**

<table>
<thead>
<tr>
<th>Semester III</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTIVE – IV</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Contact Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester III</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTIVE – V</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Contact Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester III</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTIVE - VI</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Contact Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester III</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTIVE – VII (Inter Disciplinary)</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Contact Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16
SE2047 | SEMINAR
---|---
L | T | P | C
0 | 0 | 1 | 1

**PURPOSE**
To train the students in preparing and presenting technical topics.

**INSTRUCTIONAL OBJECTIVES**
The student shall be capable of identifying topics of interest related to the program of study and prepare and make presentation before an enlightened audience.

The students are expected to give at least two presentations on their topics of interest which will be assessed by a committee constituted for this purpose. This course is mandatory and a student has to pass the course to become eligible for the award of degree. Marks will be awarded out of 100 and appropriate grades assigned as per the regulations.

<table>
<thead>
<tr>
<th>SE2049</th>
<th>PROJECT WORK PHASE I (III SEMESTER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SE2050</th>
<th>PROJECT WORK PHASE II (IV SEMESTER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**PURPOSE**
To undertake research in an area related to the program of study.

**INSTRUCTIONAL OBJECTIVE**
The student shall be capable of identifying a problem related to the program of study and carry out wholesome research on it leading to findings which will facilitate development of a new/improved product, process for the benefit of the society.

M.Tech projects should be socially relevant and research oriented ones. Each student is expected to do an individual project. The project work is carried out in two phases – Phase I in III semester and Phase II in IV semester. Phase II of the project work shall be in continuation of Phase I only. At the completion of a project the student will submit a project report, which will be evaluated (end semester assessment) by duly appointed examiner(s). This evaluation will be based on the project report and a viva voce examination on the project. The method of assessment for both Phase I and Phase II is shown in the following table:
Assessment | Tool | Weightage
--- | --- | ---
In-semester | I review | 10%
| II review | 15%
| III review | 35%
End semester | Final viva voce examination | 40%

Student will be allowed to appear in the final viva voce examination only if he/she has submitted his/her project work in the form of paper for presentation/publication in a conference/journal and produced the proof of acknowledgement of receipt of paper from the organizers/publishers.

**ELECTIVES FOR FIRST SEMESTER**

<table>
<thead>
<tr>
<th>SE2101</th>
<th>DATABASE MANAGEMENT SYSTEM</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Total Hours – 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prerequisite</td>
<td>Nil</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PURPOSE**
The student should develop skills and understanding in the design methodology for databases and verifying their structural correctness.

**INSTRUCTIONAL OBJECTIVES**

1. To learn the fundamentals of data models and to conceptualize and depict a database system using ER diagram.
2. To make a study of SQL and relational database design.
3. To know the fundamental concepts of transaction processing - concurrency control techniques and recovery procedure.
4. To learn about Object Oriented databases and Web databases.

**UNIT I - INTRODUCTION**

(9 hours)
Overview of file systems and Database Systems - Software Architecture of a typical DBMS - Data Models, Schemas and Instances - ER and EER diagrams and Data Flow Diagrams. Database Administration and Control.

**UNIT II - RELATIONAL MODEL**

(9 hours)
The relational Model – The catalog - Types - Keys - Relational Algebra – Domain Relational Calculus – Tuple Relational Calculus - Fundamental operations –
UNIT III—DATABASE DESIGN (9 hours)
Functional Dependencies — Non-loss Decomposition — Functional Dependencies — First, Second, Third Normal Forms, Dependency Preservation — Boyce / Codd Normal Form— Multi-valued Dependencies and Fourth Normal Form — Join Dependencies and Fifth Normal Form

UNIT IV—TRANSACTIONS (9 hours)

UNIT V—ADVANCED DATABASE SYSTEMS (9 hours)

REFERENCES
Every computer professional should have a basic understanding of how an operating system controls the computing resources and provide services to the users. This course provides an introduction to the operating system functions, design and implementation of a lexical analyzer & parser.

**OBJECTIVES**

1. To apply Structure and functions of OS
2. To apply Processes and Threads, Scheduling algorithms
3. The student needs to understand the Principles of concurrency and Memory management
4. The student will learn the fundamentals of I/O management and File systems
5. To understand, design and implement Lexical analyzer & parser

**UNIT –I - INTRODUCTION TO COMPILER DESIGN** (9 hours)

**UNIT II - PARSER & SYNTAX ANALYSIS** (9 hours)

**UNIT III - INTRODUCTION TO OPERATING SYSTEM** (9 hours)
UNIT IV - CONCURRENCE & MEMORY MANAGEMENT (9 hours)

UNIT V - INPUT/OUTPUT AND FILE SYSTEMS (9 hours)
I/O management and disk scheduling – I/O devices, organization of I/O functions; OS design issues, I/O buffering, disk scheduling, Disk cache. File management – Organization, Directories, File sharing, and Record blocking, secondary storage management.

REFERENCES

<table>
<thead>
<tr>
<th>SE2103</th>
<th>OBJECT ORIENTED PROGRAMMING ( C++ , JAVA)</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours – 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Programming in C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
The course provide insight knowledge about programming language (C++ and JAVA)
INSTRUCTIONAL OBJECTIVES

1. To learn the fundamentals of Object Oriented Programming
2. To understand the concepts of Classes & Objects in C++ and Java
3. To understand the concept of static and dynamic polymorphism in C++ and Java.
4. To understand the concept of streams in C++ and Java.
5. To understand the concept of exception handling technique in C++ and Java.

UNIT I-INTRODUCTION TO OOP (9 hours)
Overview of C++ - classes - structures - union - friend function - friend class - inline function - constructors - static members - scope resolution operator - passing objects to functions - function returning objects - Arrays - pointers - this pointer - references - dynamic memory allocation

UNIT II-OVERLOADING & INHERITANCE (9 hours)
Function overloading - default arguments - overloading constructors - pointers to functions - Operator overloading - member operator function - friend operator function - type conversion - inheritance - types of inheritance - virtual base class - polymorphism - virtual function.

UNIT III-TEMPLATES & EXCEPTION (9 hours)
Class templates and generic classes - function templates and generic functions - exception handling - derived class exception - exception handling functions - Streams - formatted I/O with its class functions and manipulators - creating own manipulators - file I/O - conversion functions - standard template library.

UNIT IV-INTRODUCTION FOR JAVA (9 hours)

UNIT V-JAVA APPLET (9 hours)
Input Stream and Output Stream: Streams- Byte and Character stream- Predefined streams- AWT & Event Handling: Creating User interface with AWT- Applets-
REFERENCES

<table>
<thead>
<tr>
<th>SE2104</th>
<th>COMPUTER NETWORKS &amp; NETWORKS SECURITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
</tr>
<tr>
<td>Total Hours – 45</td>
<td>3</td>
</tr>
<tr>
<td>Prerequisite</td>
<td>Nil</td>
</tr>
</tbody>
</table>

PURPOSE
This course provides an understanding of various principles, protocols and design aspects of computer networking. Also emphasizes about the various security techniques in networks.

INSTRUCTIONAL OBJECTIVES
1. To understand the concepts of data communications and the functions of different layers.
2. To make the students to get familiarized with different protocols and network components.
3. To introduce IEEE standards employed in computer networking.
4. To Apply Encryption techniques and key generation techniques
5. To learn about Authentication, Intrusion, Filtering Analysis and Security measures

UNIT I-COMPUTER NETWORKS & DATA LINK LAYER

UNIT II-NETWORK LAYER  (9 hours)

UNIT III-TRANSPORT & APPLICATION LAYER  (9 hours)

UNIT IV-NETWORK SECURITY AND CRYPTOGRAPHY  (9 hours)

UNIT V-WEB SECURITY  (9 hours)

REFERENCES

ELECTIVES FOR SECOND SEMESTER

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Lect</th>
<th>Tchr</th>
<th>Pract</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE2105</td>
<td>DISTRIBUTED OPERATING SYSTEM</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Hours</th>
<th>45</th>
</tr>
</thead>
</table>

Prerequisite: Operating Systems

PURPOSE
This course provides in-depth knowledge in Advanced Operating System concepts

OBJECTIVES
1. To apply the basic concepts of Operating System principles
2. To emphasizes on Distributed Computing Techniques, Synchronous and Processes
3. The students must fundamentally understand about the Shared Data Access, Files

UNIT I-OVERVIEW OF OPERATING SYSTEMS (9 hours)
Introduction – overview of operating system concepts – Process management and Scheduling, Memory management; partitioning, paging, segmentation, virtual memory, Device and File management Databases.

UNIT II-DISTRIBUTED COMPUTING (9 hours)
UNIT III-SYNCHRONIZATION AND PROCESSES (9 hours)

UNIT IV-SHARED MEMORY AND FILE SYSTEMS (9 hours)
Shared memory: Consistency models – Page based distributed shared memory – Shared variables– Object based distributed shared memory; Distributed File Systems: Design and Implementation

UNIT V-CASE STUDY AMOEBA & MACH (9 hours)

REFERENCES

<table>
<thead>
<tr>
<th>E-COMMERCE TECHNOLOGY</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
The purpose of the course is to impart knowledge on E-Commerce and its various applications

INSTRUCTIONAL OBJECTIVES
1. Student will able to understand the impact of E-Commerce and applications of E-Commerce
2. Student will able to understand E-Commerce framework and business model applications of E-Commerce

3. Student will able to understand e-payment mechanisms

4. Student will able to apply security algorithms

5. Student will able to familiar with marketing and advertising techniques

UNIT I-INTRODUCTION AND APPLICATIONS (9 hours)
Introduction to E-Commerce – Benefits and Impact of E-Commerce - Classification of E-Commerce – Application of E-Commerce Technologies.

UNIT II-E COMMERCE MODELS AND PAYMENT SYSTEMS (9 hours)

UNIT III-ORGANIZATIONAL COMMERCE AND EDI (9 hours)
Electronic Data Interchange – EDI Applications in Business – EDI and E-Commerce– EDI standardization and Implementation – Internet based EDI – Case Study: Indian Customs and Excise Adopts Electronic Data Exchange.

UNIT IV-SECURITY (9 hours)

UNIT V-AGENTS AND MOBILE COMMERCE (9 hours)

REFERENCES
1. Ravi Kalakota and Andrew B Whinston, “Frontiers of Electronic Commerce”, (Chapters 1,2,3,6,10,13) Pearson Education Asia, 2009
SE2107

SOFTWARE RELIABILITY

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Contact Hours – 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Prerequisite: Software Testing, Software Quality Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE

This course will look at professional techniques for understanding, assessing and applying the software reliability models in software development systems.

INSTRUCTIONAL OBJECTIVES

1. To appreciate and understand scientific concepts of Software and Hardware Reliability
2. To apply Software Reliability Growth Models in Software Development
3. To emphasize the Application of Software Reliability Models

UNIT I-INTRODUCTION (9 hours)


UNIT II-SOFTWARE RELIABILITY MODELS (9 hours)

Introduction - Historical Perspective and Implementation, classification, limitations and issues, Exponential Failure Models – Jelinski-moranda model, Poisson, Musa, Exponential models, Weibull Model, Musa-okumoto Model, Bayesinan Model – Littlewood verrall Model, Phase Based Model

UNIT III-PREDICTION ANALYSIS (9 hours)

Model Disagreement and Inaccuracy – Short & Long Term Prediction, Model Accuracy, Analyzing Predictive Accuracy – Outcomes, PLR, U & Y Plot, Errors and Inaccuracy, Recalibration – Detecting Bias, Techniques, Power of Recalibration, Limitations in Present Techniques, Improvements

UNIT IV-THE OPERATIONAL PROFILE (9 hours)

Concepts and Development Procedures – Customer Type, User Type, System Mode, Functional and Operational Profile, Test Selection - Selecting Operations, Regression Test, Special Issues – Indirect Input Variables, Updating, Distributed
system, CASE STUDY - Application of DEFINITY & FASTAR, Power Quality Resource System

UNIT V-TESTING FOR RELIABILITY MEASUREMENT (9 hours)

REFERENCES

<table>
<thead>
<tr>
<th>SE2108</th>
<th>SOFTWARE AGENTS</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering Principles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
This course provides a thorough understanding of agent related system development

INSTRUCTIONAL OBJECTIVES
1. To understand Agent development
2. Gain Knowledge in Multi agent and Intelligent agents
3. To Understand Agents and security
4. Gain Knowledge in Agent Applications

UNIT I-INTRODUCTION (9 hours)
UNIT II - SOCIOLOGICAL AGENTS (9 hours)

UNIT III - INTELLIGENT AUTONOMOUS AGENTS AND COMMUNICATION (9 hours)

UNIT IV - AML (9 hours)

UNIT V - APPLICATIONS OF AGENTS (9 hours)
Multi Agent system: Theory approaches and NASA applications – Agent based control for multi-UAV information collection- Agent based decision support system for Glider pilots – Multi agent system in E- Health Territorial Emergencies – Software Agents for computer network security- Multi-Agent Systems, Ontologies and Negotiation for Dynamic Service Composition in Multi-Organizational Environmental Management.

REFERENCES
### SE2109: DESIGN PATTERNS

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Design , Software Architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### PURPOSE

In software engineering, a design pattern is a general reusable solution to a commonly occurring problem in software design. This course teaches students advanced skills in object-oriented design and programming through learning common design patterns and refactoring software source code.

#### INSTRUCTIONAL OBJECTIVES

1. Understand and be able to apply incremental/iterative development
2. Understand common design patterns
3. Be able to identify appropriate patterns for design problems
4. Be able to evaluate the quality of software source code
5. Be able to refactor badly designed program by properly using design patterns

#### UNIT I-INTRODUCTION TO DESIGN PATTERNS (9 hours)

Design Patterns Arose from Architecture and Anthropology - Architectural to Software Design Patterns - Advantages of Design Patterns - Adapter Pattern - Strategy Pattern - Bridge Pattern - Abstract Factory Pattern

#### UNIT II-NEW PARADIGM OF DESIGN (9 hours)


#### UNIT III-VALUES OF PATTERNS (9 hours)

Observer Pattern - Categories of Patterns - Template Method Pattern - Applying the Template Method to the Case Study - Using Template Method Pattern to Reduce Redundancy

#### UNIT IV-APPLYING DESIGN PATTERNS (9 hours)

Design Patterns: Factories - Singleton Pattern and the Double-Checked Locking Pattern - Applying Singleton Pattern to Case Study. Object Pool Pattern -
UNIT V-CASE STUDIES

REFERENCES

<table>
<thead>
<tr>
<th>SE2110</th>
<th>SOFTWARE MEASUREMENTS AND METRICS</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering Principles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
The purpose of this course is to provide the knowledge about Software Metrics, Essentials of software metrics and practical knowledge to assess software.

INSTRUCTIONAL OBJECTIVES
1. To provide a solid background knowledge about software Metrics.
2. To educate various metrics and models to assess software.
3. To provide hands on experience to use and implement metrics.

UNIT I-THE HISTORY AND EVOLUTION OF SOFTWARE METRICS (9 hours)
Evolution of the software industry and evolution of software measurements – The cost of counting function point metrics – The paradox of reversed productivity for high-Level languages- The Varieties of functional metrics – Variations in
application size and productivity rates – Future Technical Developments in Functional Metrics- Software measures and metrics not based on function points.

UNIT II-MEASURING SOFTWARE QUALITY (9 hours)

UNIT III-PROCESS METRICS (9 hours)

UNIT IV-MECHANICS OF MEASUREMENT (9 hours)

UNIT V-MEASUREMENTS, METRICS AND INDUSTRY LEADERSHIP (9 hours)
REFERENCES

<table>
<thead>
<tr>
<th>SE2111</th>
<th>DATA WAREHOUSING AND DATA MINING</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>Total Hours - 45</td>
<td>3</td>
</tr>
<tr>
<td>Prerequisite</td>
<td></td>
</tr>
<tr>
<td>Database Management Systems</td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
This course enable to understand the concepts of Data Warehousing and Data Mining.

INSTRUCTIONAL OBJECTIVES
1. To learn the fundamentals of designing a large-scale data warehouse using relational technology.
2. To understand the Data Warehouse and OLAP Technology in Data Mining.
3. To study the Mining Association Rules in Large Databases, Classification and Prediction.
4. To know Cluster Analysis and its Application Trends in Data Mining.

UNIT I-DATA WAREHOUSING AND BUSINESS ANALYSIS (9 hours)

UNIT II-DATA MINING (9 hours)
Frequent Item set Mining Methods – Mining Various Kinds of Association Rules – Association Mining to Correlation Analysis – Constraint-Based Association Mining.

UNIT III-CLASSIFICATION AND PREDICTION
(9 hours)

UNIT IV-CLUSTER ANALYSIS
(9 hours)
Cluster Analysis: - Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Model-Based Clustering Methods – Clustering High-Dimensional Data – Constraint-Based Cluster Analysis – Outlier Analysis

UNIT V-APPLICATIONS OF DATA MINING
(9 hours)

REFERENCES
### UNIT I - BASIC NETWORK AND WEB CONCEPTS (9 hours)
- Internet standards – TCP and UDP protocols – URLs – MIME – CGI – Introduction to SGML

### UNIT II - SCRIPTING LANGUAGE (9 hours)

### UNIT III - AJAX (9 hours)
- Introduction to AJAX on ASP.NET – DHTML – JavaScript and DOM – Data Communication: XML, XSLT, and JSON

### UNIT IV - SILVERLIGHT & FLASH (9 hours)
- Introduction to Silverlight – Silverlight User interface control – Content integration in silverlight application – Network Communication – Working with data-
- Introduction to Flash – working with object editor – Publishing flash files
UNIT V-SERVER SIDE PROGRAMMING


REFERENCES

ELECTIVES FOR THIRD SEMESTER

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE2113</td>
<td>SOFTWARE QUALITY MANAGEMENT</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering Principles, Software Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
This course covers the principles of software development emphasizing processes and activities of quality assurance.

INSTRUCTIONAL OBJECTIVES
1. The student must relate to quality assurance plan
2. The students must apply quality assurance tools & techniques in their project
3. To learn about standards and certifications
4. To describe procedures and work instructions in software organizations
UNIT I-INTRODUCTION (9 hours)

UNIT II-SOFTWARE QUALITY ASSURANCE COMPONENTS IN THE PROJECT LIFE CYCLE (9 hours)

UNIT III-SOFTWARE QUALITY INFRASTRUCTURE COMPONENTS (9 hours)
Procedures and Work Instructions - Supporting Quality Devices - Staff Training, Instructing and Certification - Preventive and Corrective Actions - Configuration Management - Documentation and Quality Records Controls

UNIT IV-SOFTWARE QUALITY MANAGEMENT COMPONENTS (9 hours)
Project Progress Control- Components, Internal & External Participants, Progress control regimes, Computerized tools, Software Quality Metrics – Objective, Classification, Process & Product Metrics, Implementation & Limitation of Software Metrics - Software Quality Costs – Objective, Classification Model of cost, Extended Model and Applications

UNIT V-STANDARDS, CERTIFICATION AND ASSESSMENT (9 hours)

REFERENCE

<table>
<thead>
<tr>
<th>SE2114</th>
<th>COMPONENT BASED SYSTEM DESIGN</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Object Oriented Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PURPOSE**
The course gives an insight of the most commonly used software component technologies and their applications.

**INSTRUCTIONAL OBJECTIVES**
1. To understand the various software components in Java, CORBA, COM and .NET
2. Be able to use the above technologies in solving problems in the development of software
3. To understand the component framework and its development.

**UNIT I-INTRODUCTION**
(9 hours)

**UNIT II-JAVA BASED COMPONENT TECHNOLOGIES**
(9 hours)

**UNIT III-CORBA**
(9 hours)

**UNIT IV-NET BASED COMPONENT TECHNOLOGIES**
(9 hours)
COM - Object reuse - Interfaces and Polymorphism – COM object creation and COM Library – initializing objects, persistence, structural storage, monikers – From COM to distributed COM -Meta-information and Automation – Other COM services- OLE containers and servers - Active X controls - Contextual
Composition and Services - .NET framework -.NET components - Assemblies - Common language Frameworks.

UNIT V-COM OVERVIEW AND COMPONENT FRAMEWORKS (9 hours)

REFERENCES

<table>
<thead>
<tr>
<th>SE2115</th>
<th>HUMAN INTERFACE SYSTEM DESIGN</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering, System Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
This course on user Interface Design provides a basic understanding of interface design and principles

INSTRUCTIONAL OBJECTIVES
1. Students learn about the design process management
2. To understand about Interaction devices and windows strategies
3. To understand about how to Manage Virtual Environments
UNIT I-INTRODUCTION (9 hours)
Goals of System Engineering – Goals of User Interface Design – Motivations of Human factors in Design – High Level Theories – Object-Action Interface Design - Three Principles – Guidelines for Data Display and Data Entry

UNIT II-MANAGING DESIGN PROCESS (9 hours)

UNIT III-MANIPULATION AND VIRTUAL ENVIRONMENTS (9 hours)

UNIT IV-INTERACTION DEVICES (9 hours)

UNIT V-WINDOWS STRATEGIES AND INFORMATION SEARCH (9 hours)
Goals and Designers – Users and their tasks – Object Action Interface Model for Web site Design

REFERENCE

<table>
<thead>
<tr>
<th>SE2116</th>
<th>SERVICE ORIENTED ARCHITECTURE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
To gain understanding about the basic principles of service oriented architecture

INSTRUCTIONAL OBJECTIVES
1. To learn service oriented analysis techniques
2. To learn technology underlying the service design
3. To learn advanced concepts such as service composition, orchestration and Choreography
4. To know about various Web Service specification standards

UNIT I-INTRODUCTION (9 hours)

UNIT II-SERVICE ORIENTED ARCHITECTURE IN WEB SERVICES (9 hours)
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

UNIT III-BUILDING SOA (9 hours)

UNIT IV-SOA PLATFORMS (9 hours)
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).

UNIT V-SOA DESIGN (9 hours)

REFERENCES
SOFTWARE CONFIGURATION MANAGEMENT | L | T | P | C
---|---|---|---|---
Total Hours - 45 | 3 | 0 | 0 | 3
Prerequisite  
Software Quality Management

PURPOSE
This course enables to understand the importance and the benefits of software configuration and change management.

INSTRUCTIONAL OBJECTIVES
1. To learn the changing nature of software and need for change management
2. To study the different phases involved in software configuration management
3. To learn about the SCM plans, audits and reviews
4. To study the various SCM tools and implementation techniques
5. To study the SCM different scenarios and future directions

UNIT I- OVERVIEW TO SOFTWARE CONFIGURATION MANAGEMENT (9 hours)

UNIT II- DIFFERENT PHASES OF SOFTWARE CONFIGURATION MANAGEMENT (9 hours)
Different Phases Of Scm – SCM System design - SCM Plan preparation - SCM Team organization – SCM Infrastructure organization – SCM Team training – Project team training – Configuration identification – Configuration Control – Configuration status accounting – Configuration audits

UNIT III- CONFIGURATION AUDITS AND MANAGEMENT PLANS (9 hours)
When, what and who of auditing - Functional Configuration audit – Physical Configuration audit – Auditing the SCM System – Role of SCM Team in
configuration audits – SCM plan and the incremental approach – SCM Plan and SCM Tools – SCM Organization

**UNIT IV-SOFTWARE CONFIGURATION MANAGEMENT TOOLS AND IMPLEMENTATION** (9 hours)

**UNIT V-TRENDS IN SCM: FUTURE DIRECTIONS** (9 hours)

**REFERENCES**

<table>
<thead>
<tr>
<th>SE2118</th>
<th>DECISION SUPPORT SYSTEMS</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Database Management Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PURPOSE**
The purpose of this course is to impart concepts of decision making, decision processes and its implementation in a Development Environment.
INSTRUCTIONAL OBJECTIVES

1. To appreciate and understand DSS and its Characteristics
2. To apply the Decision Makers and styles
3. To educate about Knowledge management
4. To emphasize on Intelligent DSS
5. To provide a modest experience in Implementation of DSS

UNIT I-INTRODUCTION TO DSS AND DECISION MAKERS (9 hours)
DSS definition- characteristics- History of DSS- Components of DSS- Data and Model Management-DSS knowledge base- user interfaces- DSS user- categories and classes of DSS’s- Decision and Decision Makers : Decision Makers- Decision styles- Decision effectiveness- Hardness of Decisions, CASE STUDY: Mini-Case: Mervyn's Department Store, Mini-Case: Disaster at Tenerife.

UNIT II-DECISION MAKING AND ORGANIZATION DECISION (9 hours)

UNIT III-DECISION PROCESSES (9 hours)

UNIT IV-SYSTEM PERSPECTIVE OF DSS (9 hours)
Perspective of DSS: System – DSS in the context of information system-Information quality issues in DSS design- DSS information system architecture-role of Internet in DSS development and use- Designing and Building DSS: Strategies of DSS Analysis and Design- DSS Developer-tools for DSS development- DSS user Interface Issues
UNIT V-IMPLEMENTATION AND INTEGRATION OF DSS (9 hours)

REFERENCES

<table>
<thead>
<tr>
<th>SE2119</th>
<th>WEB SERVICES</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nil</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
To study and highlight the features of different technologies involved in web services and XML.

INSTRUCTIONAL OBJECTIVES
1. To understand basics in XML Concepts
2. To understand the concepts of web services
3. To gain knowledge in WSDL and UDDI
4. To apply methods for constructing and evaluating architectures

UNIT I-XML TECHNOLOGY FAMILY (9 hours)

UNIT II-ARCHITECTING WEB SERVICES (9 hours)

UNIT III-WEB SERVICES BUILDING BLOCK (9 hours)

UNIT IV-IMPLEMENTING XML IN E – BUSINESS (9 hours)

UNIT V-XML AND CONTENT MANAGEMENT (9 hours)

REFERENCES
This course enables to understand the importance and the benefits of software configuration and change management.

**INSTRUCTIONAL OBJECTIVES**

1. To learn the basic concepts of natural language processing
2. To study the different techniques involved with information retrieval
3. To learn about text mining
4. To explore some the generic issues such as multilingual information retrieval and speech processing.
5. To study the SCM different scenarios and future directions

**UNIT I-INTRODUCTION**


**UNIT II-INFORMATION RETRIEVAL**


**UNIT III-TEXT MINING**

Categorization – Extraction based Categorization- Clustering- Hierarchical Clustering- Document Classification and routing- finding and organizing answers from Text search – use of categories and clusters for organizing retrieval results – Text Categorization and efficient Summarization using Lexical Chains – Pattern Extraction.
UNIT IV-GENERIC ISSUES  (9 hours)
Multilingualism – Multilingual Information Retrieval and Speech processing -
Multimodality – Text and Images – Modality Integration - Transmission and
Storage – Speech coding- Evaluation of systems – Human Factors and user
Acceptability.

UNIT V-APPLICATIONS  (9 hours)
Machine Translation – Transfer Metaphor - Interlingua and Statistical Approaches
- Discourse Processing – Dialog and Conversational Agents – Natural Language
Generation – Surface Realization and Discourse Planning.

REFERENCES
1. Daniel Jurafsky and James H. martin, “Speech and Language Processing”,
2. Ron Cole, J.Mariani, et.al “Survey of the State of the Art in Human Language
3. Michael W. Berry “Survey of Text Mining: Culstering, Classification and
4. Christopher D.Manning and Hinrich Schutze, “Foundations of Statistical

SE2121  ANALYSIS OF SOFTWARE ARTIFACTS  L  T  P  C

<table>
<thead>
<tr>
<th>Total Hours - 45</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Engineering, Software Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
To enhance students software testing and analysis skills.

INSTRUCTIONAL OBJECTIVES
1. To understand the concepts of quality models, frame work and testing
2. To Gain knowledge in analysis of software under various dimensions
3. To Gain knowledge in the area of evaluating an architecture & verification &
   validation of software.

UNIT I-QUALITY MODELS  (9 hours)
Introduction-views on quality-cost of quality-quality models-Statistics and
measurements-Statistics and measurements-Analysis of given source code using
SQALE and Sonar models.
UNIT II-QUALITY FRAMEWORK AND TESTING  (9 hours)
Quality framework characteristics – verification- Measuring test adequacy- overview of black box testing techniques-decision tables-combinatorial testing-classification tree method- white box testing- Random and exploratory.

UNIT III-SOFTWARE ANALYSIS  (9 hours)
Introduction to Static analysis- Static analyzer for finding dynamic programming errors-dataflow testing – procedure to apply data flow testing- examples-performance analysis and verification- Security analysis and verification – Software vulnerabilities and exploitation.

UNIT IV-QUASAR METHOD  (9 hours)
Applying the Design structure matrix to system decomposition and integration problems- achieving Agility through Architecture visibility-Recovering and verifying architecture through design structure matrices.

UNIT V-QUALITY MANAGEMENT  (9 hours)
Project quality management- Essential Testing-Test driven development – guidance for software verification and validation plans-Master test planning.

REFERENCES
<table>
<thead>
<tr>
<th>CLOUD COMPUTING</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PURPOSE**
To provide a comprehensive introduction to cloud computing and about cloud services

**INSTRUCTIONAL OBJECTIVES**
1. To understand Cloud Computing basics and its models.
2. To learn the fundamentals of Data Centers.
3. To understand the Architecture of Data Centers and Design Principles
4. To understand the Security aspects and security framework.

**UNIT I-INTRODUCTION** (9 hours)
Cloud Computing Introduction, From, Collaboration to cloud, Working of cloud computing, pros and cons, benefits, developing cloud computing services, Cloud service development, discovering cloud services.

**UNIT II-CLOUD COMPUTING FOR EVERYONE** (9 hours)
Centralizing email communications, cloud computing for community, collaborating on schedules, collaborating on group projects and events, cloud computing for corporation, mapping schedules managing projects, presenting on road.

**UNIT III-USING CLOUD SERVICES** (9 hours)
Collaborating on calendars, Schedules and task management, exploring on line scheduling and planning, collaborating on event management, collaborating on contact management, collaborating on project management, collaborating on word processing, spreadsheets, and databases.

**UNIT IV-OUTSIDE THE CLOUD** (9 hours)
Evaluating web mail services, Evaluating instant messaging, Evaluating web conference tools, creating groups on social networks, Evaluating on line groupware, collaborating via blogs and wikis
UNIT V-STORING AND SHARING  (9 hours)
Understanding cloud storage, evaluating on line file storage, exploring on line book marking services, exploring on line photo editing applications, exploring photo sharing communities, controlling it with web based desktops.

REFERENCES
6. Grantt Sauls "Introduction to Data Centers", Certified Data Centers Specialist, Tutorial.

<table>
<thead>
<tr>
<th>SE2123</th>
<th>PERSONAL SOFTWARE PROCESS</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours - 45</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Prerequisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering Methodologies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PURPOSE
To learn about how a software professional personally manages the software processes in all aspects

INSTRUCTIONAL OBJECTIVES
1. To study how to manage and track the time for software processes
2. To study how to plan a product and how to measure size of a product
3. To learn how to schedule the process and manage the commitment
4. To learn about software Development process
5. To learn how to estimate the product and process quality.

UNIT I-INTRODUCTION AND TIME MANAGEMENT  (9 hours)

UNIT II-PRODUCT PLANNING AND PRODUCT SIZE (9 hours)

UNIT III-MANGING COMMITMENTS AND SCHEDULES (9 hours)

UNIT IV-SOFTWARE PROCESSES AND QUALITY (9 hours)

UNIT V-PRODUCT AND PROCESS QUALITY (9 hours)

REFERENCES
3. http://www.sei.cmu.edu/library/abstracts/reports/00tr022.cfm
4. http://repository.cmu.edu/cgi/viewcontent.cgi
5. http://dl.acm.org/citation.cfm?id=650271
Purpose
The course aims at introducing basic concepts in Embedded Systems with focus on Embedded System development, Hardware architecture and Embedded Operating System.

Instructional Objectives
1. To understand the basics of Embedded Systems
2. Students get the basic knowledge of Hardware and the Software architecture of any Embedded System
3. To apply the basic productivity and development tools commonly used in Embedded design
4. Students get an insight on various Kernel objects of Embedded operating system.

Unit I-Introduction to Embedded Systems (9 hours)

Unit II-Embedded Processor and Computing Platform (9 hours)
UNIT III-PROCESSES, OPERATING SYSTEMS AND NETWORKS  

UNIT IV-EMBEDDED C PROGRAMMING 

UNIT V-EMBEDDED SYSTEM DEVELOPMENT AND 

REFERENCES
# AMENDMENTS

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Details of Amendment</th>
<th>Effective from</th>
<th>Approval with date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>