NMAM INSTITUTE OF TECHNOLOGY, NITTE

SCHEME OF TEACHING AND EXAMINATION

FOR

M.TECH. SOFTWARE ENGINEERING

AUTONOMOUS SCHEME

Credit based system

AY 2015-2017
NMAM INSTITUTE OF TECHNOLOGY, NITTE
SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH. SOFTWARE ENGINEERING
(AUTONOMOUS SCHEME: 2015-2017)

I Semester (25 Credits)

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Code</th>
<th>Subject</th>
<th>Credits</th>
<th>C.I.A.</th>
<th>SEE</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15SSE101</td>
<td>Software Testing</td>
<td>4+0+1+0</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>15SSE102</td>
<td>Advanced Algorithms</td>
<td>4+0+1+0</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>15SSE103</td>
<td>Advances in Data Base Management Systems</td>
<td>4+0+1+0</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>15SSE104</td>
<td>Computer Systems Performance Analysis</td>
<td>4+0+0+1</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>15SSE11*</td>
<td>Elective-I</td>
<td>4+0+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>15SSE105</td>
<td>Seminar</td>
<td>0+0+0+1</td>
<td>50</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

Total 25  300  250  25

Elective I

15SSE111: Advances in Computer Networks
15SSE112: Distributed Operating Systems
15SSE113: Web Services
15SSE114: Advances in Storage Area Networks
15SSE115: Soft Computing
15SSE116: Supply Chain Management

Note:
- Continuous Internal Assessment (CIA) involves both theory tests and practical evaluation.
- There will be SEE for lab component of Core subjects of I and II Semester.
- Where ever there is a combined theory and lab, students must score minimum passing marks in each of the component.
- SEE question paper will contain 2 questions from each unit. Student must answer one full question from each unit.
- M.Tech(SSE) course is open for Computer Science and Information Science engineering undergraduates
- Self study components under elective subjects, include referring to advanced topics related to the subjects from Technical journals and giving a seminar and a write-up.
### NMAM INSTITUTE OF TECHNOLOGY, NITTE
### SCHEME OF TEACHING AND EXAMINATION FOR
### M.TECH. SOFTWARE ENGINEERING
### (AUTONOMOUS SCHEME: 2015-2017)

#### II Semester

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Code</th>
<th>Subject</th>
<th>Credits</th>
<th>C.I.A.</th>
<th>S.E.E</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lect/Tut/Prac/Self Study</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>15SSE201</td>
<td>Software Project Planning &amp; Management</td>
<td>4+0+1+0</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>15SSE202</td>
<td>Enterprise Application Programming</td>
<td>4+0+1+0</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>15SSE203</td>
<td>Design Patterns</td>
<td>4+0+1+0</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>15SSE204</td>
<td>Software Metrics &amp; Quality Assurance</td>
<td>4+0+0+1</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>15SSE21*</td>
<td>Elective-II</td>
<td>4+0+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>15SSE205</td>
<td>Seminar</td>
<td>0+0+0+1</td>
<td>50</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td>25</td>
<td>300</td>
<td>250</td>
<td>25</td>
</tr>
</tbody>
</table>

#### Elective II
- 15SSE211 Information and Network Security
- 15SSE212 Data Mining & Data Warehousing
- 15SSE213 Information Retrieval
- 15SSE214 Information Storage Management
- 15SSE215 Agile Technologies
- 15SSE216 Service oriented Architecture
- 15SSE217 Distributed Computing
NMAM INSTITUTE OF TECHNOLOGY, NITTE  
SCHEME OF TEACHING AND EXAMINATION FOR  
M.TECH.SOFTWARE ENGINEERING  
(AUTONOMOUS SCHEME: 2015-2017 )

III Semester  
(20 Credits)

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Duration</th>
<th>C.I.A.</th>
<th>EXAM</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15SSE301</td>
<td>Industrial Training/ Mini Project/Certification Exam</td>
<td>Full Time 8 weeks</td>
<td>100</td>
<td>--</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>15SSE302</td>
<td>Seminar +Technical paper writing</td>
<td>--</td>
<td>100</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>15SSE303</td>
<td>Project (Problem Statement, Literature Survey, Initial Design )</td>
<td>Remaining weeks of 3rd sem.</td>
<td>200</td>
<td>--</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td>400</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>

Seminar : Topics should be chosen from IEEE/ACM/Elsevier/Springer/any Refereed Journals /Transactions. Encourage students to convert these seminar topics into a good survey paper or Technical paper.
NMAM INSTITUTE OF TECHNOLOGY, NITTE
SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH.SOFTWARE ENGINEERING
(AUTONOMOUS SCHEME: 2015-2017 )

IV Semester

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Duration Practical/ Field work</th>
<th>C.I.A.</th>
<th>EXAM</th>
<th>Credits</th>
</tr>
</thead>
</table>

TOTAL | -- | 400 | 30 |

GRAND TOTAL from 1st to 4th semester : 100 credits

Note:
*Lab Classes for these Core Subjects are Compulsory (Practical will be Evaluated for 20 marks and Internal assessment for 30 marks). Lab journals should be maintained.
# Seminar : Topics should be chosen from IEEE/ACM/Elsevier/Springer/any Refereed - Journals /Transactions. Encourage students to convert these seminar topics into a good survey paper or Technical paper.

1) Project Phase – I : 6 weeks duration shall be carried out between II and III Semester. Candidates in consultation with guide shall carry out literature survey / visit to Industries to finalize the topic of dissertation.

2) Internship: 24 weeks Duration in 3rd Semester, Evaluation of Marks - Presentation : 25 marks, Report writing and Submission :75 marks and at the end of Internship Viva-Voce Exams shall be conducted for 50 marks.
3) Project Work: 20 weeks duration in IV Semester carries total marks of 250.

4) Project Phase II: 4 days for project work in a week during IV Semester. Evaluation shall be taken during the 8th week of the IV Semester. Total Marks shall be 25.

5) Project Phase – III: Evaluation shall be taken up at the end of the IV Semester for 25 marks. After the Project report is submitted, Project Work Evaluation and Viva-Voce Examination shall be conducted. Total Marks shall be 50+50+100=200 (50 Marks to be evaluated by Internal Guide, 50 Marks to be evaluated by External examiner and 100 for Viva-Voce)

Marks of Evaluation of Project:

I) 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.

II) 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).
Course Title: Software Testing
Course Code: 15SSE101
Credits(L:T:P): 4:0:1
Type of Course: Lecture & Practical
Core/Elective: Core
Total Contact Hours: 52

COURSE OBJECTIVES
• To explore the basics and goals of software testing.
• To discuss various types of software testing and its techniques
• To list out various tools which can be used for automating the testing process
• To introduce various software quality standards for establishing quality environment
• To discuss various methods and evaluation procedures for improving the quality Models

TOPICS

UNIT-I

10 hours

UNIT-II
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. Data Flow Testing: Definition-Use testing, Slice-based testing, Guidelines and observations. Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. Integration Testing: A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations, Case study.

10 hours

UNIT-III

12 hours

UNIT-IV

10 hours
UNIT-V

Exploratory Testing: The context-driven school, Exploring exploratory testing, Exploring a familiar example, Exploratory and context-driven testing observations. Model-Based Testing: Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing. 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.

10 hours

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.

COURSE OUTCOMES:
Upon Completion of the course, students shall be able to
• Compare and pick out the right type of software testing process for any given real world problem
• Carry out the software testing process in efficient way
• Automate the testing process by using several testing tools
• Establish a quality environment as specified in standards for developing quality software
• Analyze and improve the quality procedures based on the past experience

TEXT BOOKS:

REFERENCE BOOKS:
COURSE OBJECTIVES

• To learn the graph search algorithms.
• To study network flow and linear programming problems.
• To learn the hill climbing and dynamic programming design techniques.
• To develop recursive backtracking algorithms.
• To get an awareness of NP completeness and randomized algorithms.

TOPICS

UNIT-I


10 hours

UNIT-II

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. Polynomials and the FFT: Representation of polynomials; The DFT and FFT; Efficient implementation of FFT.

12 hours

UNIT-III

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.

10 hours

UNIT-IV

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

10 hours

UNIT-V

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

10 hours
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 naive 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 the FFT algorithm efficiently.

COURSE OUTCOMES:
Upon completion of the course, students shall be able to
• Design and apply iterative and recursive algorithms.
• Design and implement optimization algorithms in specific applications.
• Design appropriate shared objects and concurrent objects for applications.
• Implement and apply concurrent linked lists, stacks, and queues.

TEXT BOOKS:

REFERENCE BOOKS:
Course Title: Advances In Database Management Systems
Course Code: 15SSE103
Credits(L:T:P): 4:0:1
Type of Course: Lecture & Practical
Core/Elective: Core
Total Contact Hours:52

COURSE OBJECTIVES:

• To acquire knowledge on parallel and distributed databases and its applications.
• To study the usage and applications of Object Oriented database
• To understand the basic concepts, principles of intelligent databases.
• To understand the advanced topics of data warehousing and mining.
• To learn emerging and advanced data models
• To acquire inquisitive attitude towards research topics in databases.

TOPICS:

UNIT-I
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. Overview of Object-Oriented Concepts – Objects, Encapsulation, Polymorphism, Type and class hierarchies etc.

10 Hours

UNIT-II
Object and Object-Relational Databases: Object Oriented Concepts: Objects, 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.

10 Hours

UNIT- III
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.

10 Hours

UNIT- IV
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.

12 Hours
UNIT-V

**Enhanced Data Models for Some Advanced Applications:** Active database concepts and triggers; Temporal, Spatial, and Deductive Databases – Basic concepts. More Recent Applications: Mobile databases, Multimedia databases, Geographical Information Systems, Genome data management

10 Hours

**ADBMS LABORATORY WORK**

**LABORATORY OBJECTIVES:**
- To acquire practical knowledge on advanced databases and its applications.
- To understand and work on areas like Storage, Retrieval, Multi valued attributes, Triggers and other complex objects, Algorithms etc related to ADBMS.
- To design and implement recent applications database for better interoperability
- Note: The following experiments may be implemented on MySQL/ORACLE or any other suitable RDBMS with support for Object features

1. Develop a database application to demonstrate storing and retrieving of BLOB and CLOB objects.
Write a binary large object (BLOB) to a database as either binary or character (CLOB) data, depending on the type of the field in your data source. To write a BLOB value to the database, issue the appropriate INSERT or UPDATE statement and pass the BLOB value as an input parameter. If your BLOB is stored as text, such as a SQL Server text field, pass the BLOB as a string parameter. If the BLOB is stored in binary format, such as a SQL Server image field, pass an array of type byte as a binary parameter.
Once storing of BLOB and CLOB objects is done, retrieve them and display the results accordingly.

2. Develop a database application to demonstrate the representation of multi valued attributes, and the use of nested tables to represent complex objects. Write suitable queries to demonstrate their use.
Consider Purchase Order Example: This example is based on a typical business activity: managing customer orders. Need to demonstrate how the application might evolve from relational to object-relational, and how you could write it from scratch using a pure object-oriented approach.
Show how to implement the schema -- Implementing the Application under the Relational Model -- using only Oracle's built-in data types. Build an object-oriented application on top of this relational schema using object views.

3. Design and develop a suitable Student Database application by considering appropriate attributes. Couple of attributes to be maintained includes the Attendance of a student in each subject for which he/she has enrolled and Internal Assessment Using TRIGGERS, write active rules to do the following:
Whenever the attendance is updated, check if the attendance is less than 85%; if so, notify the Head of the Department concerned.
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.
Use the following guidelines when designing triggers:
• Use triggers to guarantee that when a specific operation is performed, related actions are performed.
• Use database triggers only for centralized, global operations that should be fired for the triggering statement, regardless of which user or database application issues the statement.
• Do not define triggers that duplicate the functionality already built into Oracle. For example, do not define triggers to enforce data integrity rules that can be easily enforced using declarative integrity constraints.
• Limit the size of triggers (60 lines or fewer is a good guideline). If the logic for your trigger requires much more than 60 lines of PL/SQL code, it is better to include most of the code in a stored procedure, and call the procedure from the trigger.
• Be careful not to create recursive triggers. For example, creating an AFTER UPDATE statement trigger on the EMP table that itself issues an UPDATE statement on EMP causes the trigger to fire recursively until it has run out of memory.

4. Design, develop, and execute a program to implement specific Apriori algorithm for mining association rules. Run the program against any large database available in the public domain and discuss the results.
Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

COURSE OUTCOMES:

Upon completion of the course, the students will be able to
• Select the appropriate high performance database like parallel and distributed database Model and represent the real world data using object oriented database
• Embed the rule set in the database to implement data warehousing of mining
• Choose and design database for recent applications database for better interoperability

TEXT BOOKS:

REFERENCE BOOKS:
COURSE OBJECTIVES:
• To understand the mathematical foundations needed for performance evaluation of computer systems
• To understand the metrics used for performance evaluation
• To understand the analytical modeling of computer systems
• To enable the students to develop new queueing analysis for both simple and complex systems
• To introduce the students to analytical techniques for evaluating scheduling policies

TOPICS:

UNIT-I

10 Hours

UNIT-II

10 Hours

UNIT-III
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.

10 Hours

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

10 Hours

UNIT-V

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 Centers, Hierarchical Decomposition, Limitations of Queuing Theory.

12 Hours

COURSE OUTCOMES:
Upon completion of the course, students shall be able to
• Identify the need for performance evaluation and the metrics used for it
• Implement Little law and other operational laws
• Apply the operational laws to open and closed systems
• Use discrete-time and continuous-time Markov chains to model real world systems
• Develop analytical techniques for evaluating scheduling policies

Text Book:

Reference Books:
Course Title: Advances in Computer Networks

Course Code: 15SSE111

Credits(L:T:P): 4:0:0

Type of Course: Lecture

Total Contact Hours: 50

COURSE OBJECTIVES:

• Become familiar with the basics of Computer Networks
• Learn Network architectures
• Become thorough with design concepts of fundamental protocols

TOPICS

UNIT-I


T1: Chapter 1.1, 1.2, 1.5.1, 1.5.2, 2.1, 2.5  T2: Chapter 4

10 hours

UNIT-II

Internetworking- I: Switching and Bridging, Datagrams, Virtual Circuit Switching, Source Routing, Bridges and LAN Switches, Basic Internetworking (IP), What is an Internetwork ?, Service Model, Global Addresses, Datagram Forwarding in IP, subnetting and classless addressing, Address Translation (ARP), Host Configuration (DHCP), Error Reporting (ICMP), Virtual Networks and Tunnels.

T1: Chapter 3.1, 3.2, 

10 hours

UNIT-III

Internetworking- II: Network as a Graph, Distance Vector (RIP), Link State (OSPF), Metrics, The Global Internet, Routing Areas, Routing among Autonomous systems (BGP), IP Version 6 (IPv6), Mobility and Mobile IP.

T1: Chapter 3.3, 4.1.1, 4.1.3  T2: Chapter 13.1 to 13.18, Chapter 18.

10 hours

UNIT-IV


T1: Chapter 5.1, 5.2.1 to 5.2.8, 6.2, 6.3

10 hours
UNIT-V

**Congestion Control and Resource Allocation:** Congestion-Avoidance Mechanisms, DECbit, Random Early Detection (RED), Source-Based Congestion Avoidance. The Domain Name System (DNS), Electronic Mail (SMTP, POP, IMAP, MIME), World Wide Web (HTTP), Network Management (SNMP)

T1: Chapter 6.4  T2: Chapter 23.1 to 23.16, Chapter 24, 25, Chapter 27.1 to 27.8

**COURSE OUTCOMES:**

Upon completion of the course, students shall be able to

- List and classify network services, protocols and architectures, explain why they are layered.
- Key Internet applications and their protocols, and will apply to develop their own applications (e.g. Client Server applications, Web Services) using the sockets API.
- Explain various congestion control techniques.

**TEXT BOOKS:**

**REFERENCE BOOK:**
Course Title: Distributed Operating Systems  
Course Code: 15SSE112  
Credits(L:T:P): 4:0:0  
Core/Elective: Elective  
Type of Course: Lecture  
Total Contact Hours: 50

COURSE OBJECTIVE:

• To explore distributed systems principles associated with communication, naming, synchronization, distributed file systems, system design, distributed scheduling, and several case studies
• To understand both foundational concepts and well as practical deployments.
• To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
• To gain insight on to the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols

TOPICS

UNIT-I

10 hours

UNIT-II
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.

10 hours

UNIT-III
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.
Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms.

10 hours
UNIT-IV


10 hours

UNIT-V


10 hours

COURSE OUTCOMES:
Upon completion of the course, students shall be able to
• Underlying distributed systems concepts
• Demonstrate an ability to apply theory and techniques to unseen problems.
• Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
• Explore the various resource management techniques for distributed systems

TEXT BOOK:

REFERENCE BOOK:
COURSE OBJECTIVES

- To provide an in-depth knowledge of Web Services.
- To understand the fundamental concepts of Web services.
- To understand the fundamental concepts of WSDL Web Services.
- To design Web service Architecture.
- To Study Building Blocks of Web services.

TOPICS

UNIT-I

Middleware: Understanding the middle ware, RPC and Related Middle ware, TP Monitors, Object Brokers, Message-Oriented Middleware.

10 hours

UNIT-II


10 hours

UNIT-III

Basic Web Services Technology: WSDL Web Services Description Language, UDDI Universal Description Discovery and Integration, Web Services at work interactions between the Specifications, Related Standards.

10 hours

UNIT-IV

Service Coordination Protocols: Infrastructure for Coordination Protocols, WS-Coordination, WS-Transaction, RosettaNet, Other Standards Related to Coordination Protocols.

10 hours

UNIT-V

Service Composition: Basic of Service Composition, A New Chance of Success for Composition, Services Composition Models, Dependencies between Coordination and Composition, BPEL: Business Process Execution Language for Web Services, OutLook, Applicability of the Web Services, Web services as a Problem and a Solution: An Example.

10 hours

COURSE OUTCOMES

Upon completion of the course, students shall be able to

- Bind and unbind services in UDDI.
- Develop WSDL document
- Implement web service client to call public service.
- Implement a service and exposing it as public service.

TEXT BOOK

Course Title: Advances in Storage Area Networks  
Course Code: 15SSE114  
Credits(L:T:P): 4:0:0  
Core/Elective: Elective  
Type of Course: Lecture  
Total Contact Hours: 50

COURSE OBJECTIVES
• To understand the fundamentals of storage centric and server centric systems
• To understand the metrics used for Designing storage area networks
• To understand the RAID concepts
• To enable the students to understand how data centre’s maintain the data with the concepts of backup mainly remote mirroring concepts for both simple and complex systems

TOPICS

UNIT-I

10 hours

UNIT-II
I/O Techniques: The Physical I/O path from the CPU to the Storage System; SCSI; Fibre Channel Protocol Stack; Fibre Channel SAN; IP Storage. Network Attached Storage: The NAS Architecture, The NAS hardware Architecture, The NAS Software Architecture, Network connectivity, NAS as a storage system. File System and NAS: Local File Systems; Network file Systems and file servers; Shared Disk file systems; Comparison of Fibre Channel and NAS.

10 hours

UNIT-III
Storage Virtualization: Definition of Storage virtualization; Implementation Considerations; Storage virtualization on Block or file level; Storage virtualization on various levels of the storage Network; Symmetric and Asymmetric storage virtualization in the Network.

10 hours

UNIT-IV
SAN Architecture and Hardware devices: Overview, Creating a Network for storage; SAN Hardware devices; The Fibre channel switch; Host Bus Adaptors; Putting the storage in SAN; Fabric operation from a Hardware perspective. Software Components of SAN: The switch’s Operating system; Device Drivers; Supporting the switch’s components; Configuration options for SANs.

10 hours
UNIT-V


10 hours

COURSE OUTCOMES:
Upon completion of the course, the students will be able to
• Identify the need for performance evaluation and the metrics used for it
• Apply the techniques used for data maintenance.
• Realize storage virtualization concept,
• Develop techniques for evaluating policies for LUN masking, file systems

TEXT BOOK:
1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, Wiley India, 2013.

REFERENCE BOOKS:
Course Title: Soft Computing  
Course Code: 15SSE115
Credits(L:T:P): 4:0:0  
Core/Elective: Elective
Type of Course: Lecture  
Total Contact Hours: 50

COURSE OBJECTIVES:
- To learn the key aspects of Soft computing
- To know about the components and building block hypothesis of Genetic algorithm.
- To gain insight onto Neuro Fuzzy modeling and control.
- To gain knowledge in machine learning through Support vector machines.

TOPICS

UNIT-I
Introduction to Soft computing: Neural networks, Fuzzy logic, Genetic algorithms, Hybrid systems and its applications. Fundamental concept of ANN, Evolution, basic Model of ANN, Terminologies used in ANN, MP model, Hebb model.
10 hours

UNIT-II
Perceptron Network: Adaptive linear neuron, Multiple adaptive linear neurons, Back propagation Network (Theory, Architecture, Algorithm for training, learning factors, testing and applications of all the above NN models).
10 hours

UNIT-III
Introduction to classical sets and fuzzy sets: Classical relations and fuzzy relations, Membership functions.
10 hours

UNIT-IV
Defuzzification: Fuzzy decision making, and applications.
10 hours

UNIT-V
Genetic algorithms: Introduction, Basic operations, Traditional algorithms, Simple GA General genetic algorithms, The schema theorem, Genetic programming, applications.
10 hours

COURSE OUTCOMES:
Upon completion of the course, the students will be able to
- Implement machine learning through neural networks.
- Design Genetic Algorithm to solve the optimization problem
- Develop a Fuzzy expert system.
- Model Neuro Fuzzy system for clustering and classification.

TEXT BOOKS:
   (Chapters 1, 2, 3(Upto 3.5), 7, 8, 9, 10, 13, 15 (upto 15.6 & 15.9,15,10)
REFERENCE BOOK:
Course Title: Supply Chain Management
Course Code: 15SSE116
Credits(L:T:P): 4:0:0
Core/Elective: Elective
Type of Course: Lecture
Total Contact Hours:50

COURSE OBJECTIVES
- To understand Model of SCM.
- To understand QRM, CPFR.
- To learn Inventory Models and third party logistics.
- To explore Revenue management

TOPICS

UNIT-I

Introduction to Supply Chain Management
Supply chain, objectives, importance, decision phases, process view, competitive and supply chain strategies, achieving strategic fit, supply chain drivers, obstacles, framework, facilities, inventory, transportation, information, sourcing, pricing.

UNIT-II

Designing the supply chain network

UNIT-III

Designing and Planning Transportation Networks
Role of transportation - modes and their performance - transportation infrastructure and policies - design options and their trade-offs - Tailored transportation.

UNIT-IV

Sourcing and Pricing
Sourcing – In-house or Outsource – 3rd and 4th PLs – supplier scoring and assessment, selection – design collaboration – procurement process – sourcing planning and analysis. Pricing and revenue management for multiple customers, perishable products, seasonal demand, bulk and spot contracts.

UNIT-V

Information Technology in the supply chain
COURSE OUTCOMES
Upon completion of the course, the students will be able to
• Discuss SCM Models,
• Design supply chain network
• Design and plan transportation network.
• Implement various Inventory Models and third party logistics.

TEXTBOOKS:

REFERENCE BOOKS:
COURSE OBJECTIVES

- To define and highlight importance of software project management.
- To formulate strategy in managing projects
- To estimate the cost associated with a project
- To plan, schedule and monitor projects for the risk management
- To define the software management metrics

TOPICS

UNIT-I

Metrics: Introduction, The Metrics Roadmap, A Typical Metrics Strategy, What Should you Measure?, Set Targets and track Them, Understanding and Trying to minimize variability, Act on data, People and Organizational issues in Metrics Programs, Common Pitfalls to watch out for in Metrics Programs, Matrices implementation checklists and tools, Software configuration management: Introduction, Some Basic Definitions and terminology, the processes and activities of software configuration management, configuration status accounting, configuration audit, software configuration management in geographically distributed teams, Metrics in software configuration management, software configuration management tools and automation.

10 hours

UNIT-II


12 hours

UNIT-III

Software Requirements gathering: Inputs and start criteria for requirements gathering, Dimensions of requirements gathering, Steps to be followed during requirements gathering, outputs and quality records from the requirements phase, skill sets required during requirements phase, differences for a shrink-wrapped software, challenges during the requirements management phase, Metrics for requirements phase. Estimation: What is Estimation? when and why is Estimation done?, the three phases of Estimation, Estimation methodology, formal models for size Estimation, Translating size Estimate into effort
Estimate, Translating effort Estimates into schedule Estimate, common challenges during Estimation, Metrics for the Estimation processes. Design and Development Phases: Some differences in our chosen approach, salient features of design, evolving an architecture/blueprint, design for reusability, technology choices/constraints, design to standards, design for portability, user interface issues, design for testability, design for diagnose ability, design for maintainability, design for install ability, inter-operability design, challenges during design and development phases, skill sets for design and development, metrics for design and development phases.

10 hours

UNIT-IV

Project management in the testing phase: Introduction, What is testing?, what are the activities that make up testing?, test scheduling and types of tests, people issues in testing, management structures for testing in global teams, metrics for testing phase. Project management in the Maintenance Phase: Introduction, Activities during Maintenance Phase, management issues during Maintenance Phase, Configuration management during Maintenance Phase, skill sets for people in the maintenance phase, estimating size, effort, and people resources for the maintenance phase, advantages of using geographically distributed teams for the maintenance phase, metrics for the maintenance phase.

10 hours

UNIT-V

Globalization issues in project management: Evolution of globalization, challenges in building global teams, Models for the execution of global projects, some effective management techniques for managing global teams. Impact of the internet on project management: Introduction, the effect of internet on project management, managing projects for the internet, Effect on the project management activities. People focused process models: Growing emphasis on people centric models, people capability maturity model(P-CMM), other people focused models in the literature, how does an organization choose the models to use?

10 hours
LABORATORY WORK for Software Project Planning and Management

USE STAR UML TOOL or VISUAL PARADIGM or any other equivalent tool to develop
ATM and Restaurant systems

UML diagrams to be developed are:

1. Use Case Diagram.
2. Class Diagram.
3. Sequence Diagram.
5. State Diagram.
6. Activity Diagram.
7. Component Diagram.
8. Deployment Diagram.

1. ATM SYSTEM

The software to be designed will control a simulated automated teller machine (ATM) having
a magnetic stripe reader for reading an ATM card, a customer console (keyboard and
display) for interaction with the customer, a slot for depositing envelopes, a dispenser for
cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts,
and a key-operated switch to allow an operator to start or stop the machine. The ATM will
communicate with the bank's computer over an appropriate communication link. (The
software on the latter is not part of the requirements for this problem.) The ATM will service
one customer at a time. A customer will be required to insert an ATM card and enter a
personal identification number (PIN) - both of which will be sent to the bank for validation as
part of each transaction. The customer will then be able to perform one or more transactions.
The card will be retained in the machine until the customer indicates that he/she desires no
further transactions, at which point it will be returned - except as noted below. The ATM
must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to
   the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the
   bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting
   of cash and/or checks in an envelope. The customer will enter the amount of the deposit into
   the ATM, subject to manual verification when the envelope is removed from the machine by
   an operator. Approval must be obtained from the bank before physically accepting the
   envelope.
3. A customer must be able to make a transfer of money between any two accounts linked to
   the card.
4. A customer must be able to make a balance inquiry of any account linked to the card.
5. A customer must be able to abort a transaction in progress by pressing the Cancel key
   instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was
allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once
it has been approved. In the case of a deposit, a second message will be sent to the bank
indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back. If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction. The ATM will provide the customer with a printed receipt for each successful transaction.

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

2. Restaurant System

The system is intended to support the day-to-day operations of a restaurant by improving the processes of making reservations and allocating tables to customers. The Restaurant system provides the facilities like:

- Record Booking
- Cancel Booking
- Record Arrival
- Table Transfer

The new system can offer diners eat at the restaurant without making an advance booking, if a free table is available. This is known as Walk-in.

The new system should display the same information as the existing booking sheet and in same format, to make it easy for restaurant staff to transfer, to the new system. When new bookings are recorded or changes made to existing bookings, the display should be immediately updated, so that restaurant staff is working with the latest information available.

COURSE OUTCOMES:

Upon completion of the course, the students will be able to:

- Evaluate a project to develop the scope of work, provide accurate cost estimates and to plan the various activities
- Apply risk management analysis techniques that identify the factors that put a project at risk and to quantify the likely effect of risk on project timescales
- Identify the resources required for a project and to produce a work plan and resource schedule
- Monitor the progress of a project and to assess the risk of slippage, revising targets counteract drift
- Use appropriate metrics to management the software development outcome
• Develop research methods and techniques appropriate to defining, planning and carrying out a research project within your chosen specialist area within the management of software projects.

TEXT BOOK

REFERENCES:
Course Title: Enterprise Application Programming
Course Code: 15SSE202
Credits(L:T:P): 4:0:1
Type of Course: Lecture & Practical
Core/Elective: Core
Total Contact Hours: 52

COURSE OBJECTIVES:
• To gain knowledge about metrics Web Application Development and related terminologies
• To gain knowledge about persistent framework and tools.
• To learn to build solutions using Design Patterns
• To get introduced to latest WEB frameworks

TOPICS

UNIT-I
Web application and java EE 6: Exploring the HTTP Protocol, Introducing web applications, describing web containers, exploring web architecture models, exploring the MVC architecture. Working with servlets 3.0 Exploring the features of java servlet. Exploring new features in servlet 3.0, Exploring the servlet API, explain the servlet life cycle, creating a sample servlet, creating a servlet by using annotation, working with servlet config and servlet context objects, working with the Http servlet request and Http servlet response interfaces, Exploring request delegation and request scope, implementing servlet collaboration.

10 hours

UNIT-II
Handling sessions in servlet 3.0: Describing a session, introducing session tracking, Exploring the session tracking, mechanisms, using the java servlet API for session tracking, creating login application using session tracking. Implementing event handling Introducing events, Introducing event handling, working with the servlet events, developing the online shop web application. Working with java server pages: Introducing JSP technology, Exploring new features of JSP2.1, listing advantages of JSP over java servlet, Exploring the architecture of a JSP page, Describing the life cycle of a JSP page, working with JSP basic tags and implicit objects, working with the action tags in JSP, exploring the JSP unified EL, using functions with EL.

10 hours

UNIT-III
Implementing JSP tag extensions: Exploring the elements of tag extensions, Working with classic tag handlers, Exploring the tag extensions, Working with simple tag handlers. Implementing java server pages standard tag library 1.2: Introducing JSTL, Exploring the tag libraries JSTL, working with the core tag library. Implementing filters: Exploring the need of filters, exploring the working of filters, exploring filters API, configuring a filter, creating a web application using filters, using initializing parameter in filters.

10 hours

UNIT-IV
Persistence Management and Design Patterns: Implementing java persistence using hibernate Introducing hibernate, exploring the architecture of hibernate, downloading hibernate, exploring HQL, understanding hibernate O/R mapping, working with hibernate, Implementing O/R mapping with hibernate. Java EE design patterns: Describing the java EE
application architecture, Introducing a design patterns, discussing the role of design patterns, exploring types of patterns.

10 hours

UNIT-V

**Web Frameworks:** Working with struts 2. Introducing struts 2, understanding actions in struts 2. Working with java server faces 2.0: Introducing JSF, Explaining the features of JSF, Exploring the JSF architecture, describing JSF elements, Exploring the JSF request processing life cycle. Working with spring 3.0: Introducing features of the spring framework, exploring the spring framework architecture, exploring dependency injection & inversion of control, exploring AOP with spring, managing transactions. Securing java EE 6 applications: Introducing security in java EE 6, exploring security mechanisms, implementing security on an application server.

12 hours

LABORATORY WORK

Design Develop and Implement the following modules using any suitable language/tools.

1. Developing the profile management module
   - Design, develop and implement the following theory models using any suitable language tools
     - Implementing logic with servlet.
     - creating the people_employee servlet.
     - creating the employeeobj class.
     - creating the employeedbmethods class.
     - creating the generateid class, creating views.
     - creating the people_insert JSP page.
     - creating the people_search JSP page.
     - creating the people_edit JSP page.
     - creating the people_list JSP page.
     - creating the people_profile JSP page.

2. Developing the recruitment module
   - Registering a new applicant.
     - creating the people_applicant servlet.
     - creating the applicantDBObj class.
     - creating the applicantDBmethods class.
     - creating the generated class.
     - creating an interface for applicant registration.
     - conducting rounds of test.
     - creating the applicant_test_dtl servlet.
     - designing JSP views.
     - working of the recruitment module.

3. Developing the payroll module
   - Updating salary statement,
     - creating the people_payroll servlet.
     - creating the empsal class.
     - creating the employee agreement class.
• creating the payrollbean methods class.
• designing JSP views.
• creating the people_agreement JSP page.
• creating the people_agreement_edit JSP page.
• creating the salary_search.jsp file.
• creating the salary_slip JSP page.

COURSE OUTCOMES:
Upon completion of the course, students shall be able to
• Implement a WEB application.
• Manage deployment configurations are
• Implement Security mechanisms

TEXT BOOK:
1. Kogent learning solution: JAVA SERVER PROGRAMMING JAVA EE6(J2EE 1.6),
Dreamtech press 2014
Course Title: Design Patterns
Credits(L:T:P): 4:0:1
Type of Course: Lecture & Practical

OBJECTIVES:
• To learn how to add functionality to designs while minimizing complexity.
• To learn what code qualities are required to maintain to keep code flexible?
• To understand the common design patterns.
• To explore the appropriate patterns for design problems.

TOPICS

UNIT-I
Introduction: what is a design pattern? Describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern. What is object-oriented development? Key concepts of object oriented design other related concepts, benefits and drawbacks of the paradigm.

10 hours

UNIT-II
Analysis a System: overview of the analysis phase, stage 1: gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain. Design and Implementation, discussions and further reading.

10 hours

UNIT-III
Design Pattern Catalog: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.

10 hours

UNIT-IV
Interactive systems and the MVC architecture: Introduction, The MVC architectural pattern, analyzing a simple drawing program, designing the system, designing of the subsystems, getting into implementation, implementing undo operation, drawing incomplete items, adding a new feature, pattern based solutions.

11 hours

UNIT-V
Designing with Distributed Objects: Client server system, java remote method invocation, implementing an object oriented system on the web (discussions and further reading) a note on input and output, selection statements, loops arrays.

11 hours
Design Patterns Laboratory

OBJECTIVES:
- To understand the need of design pattern to solve problems of different context.
- To understand the common design patterns and to implement them.
- To explore the appropriate patterns for design problems.

Laboratory Work:

Note: Use appropriate tools/language to implement the following experiment:
1. Design, develop and implement an intercepting Filter Pattern which intercepts and intermediates the request received. Implement an HTML to send a request to a server, where the request is intercepted by the filter and the following details are extracted and displayed. Date, Content-Encoding, Content-Length, Content-Location, Content-MD5
2. Design, develop and implement a front controller pattern, which forms the centralized control to handle multiple user request for effectively managing the activities of content retrieval, view management and security service invocation. Implement a java Servlet, which forms the controller and decision maker for the entire application.
3. Implement a data access object pattern to separate the data processing logic from data access logic. Implement the java class that decouples the persistence management from business logic.
4. Implement an object pooling mechanism using N-TON design pattern. Implement a Java class which creates 10 connection objects and forms a pool, and another java class to consume the connection and to persist data into the EIS.
5. Implement the session façade pattern to de-couple the business logic from accessing the third party application object in the distributed environment using EJB.

COURSE OUTCOMES:
Upon completion of the course, the students will be able to
- Design and implement codes with higher performance and lower complexity
- Be aware of code qualities needed to keep code flexible
- Experience core design principles and be able to assess the quality of a design with respect to these principles.
- Capable of applying these principles in the design of object oriented systems.
- Demonstrate an understanding of a range of design patterns. Be capable of comprehending a design presented using this vocabulary.
- Be able to select and apply suitable patterns in specific contexts.

TEXT BOOKS

REFERENCE BOOKS:
Course Title: Software Metrics and Quality Assurance
Course Code: 15SSE204
Credits(L:T:P:S): 4:0:0:1
Core/Elective: Core
Type of Course: Lecture & Self study
Total Contact Hours: 52

COURSE OBJECTIVES:
• To gain basic knowledge about metrics, measurement theory and related terminologies
• To learn measure the quality level of internal and external attributes of the software product
• To introduce the basics of software reliability and to illustrate how to perform planning, executing and testing for software reliability
• To explore various metrics and models of software reliability
• To compare various models of software reliability based on its application

TOPICS

UNIT-I

UNIT-II
Applying The Seven Basic Quality Tools In Software Development: Ishikawa’s Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts, Scatter Diagram, Control Chart, Cause And Effect Diagram. The Rayleigh Model: Reliability Models, The Rayleigh Model Basic Assumptions, Implementation, Reliability And Predictive Validity.

UNIT-III

UNIT-IV
UNIT-V


11 hours

COURSE OUTCOMES:
Upon completion of the course, students shall be able to
• Identify and apply various software metrics, which determines the quality level of software
• Identify and evaluate the quality level of internal and external attributes of the software product
• Compare and Pick out the right reliability model for evaluating the software
• Evaluate the reliability of any given software product
• Design new metrics and reliability models for evaluating the quality level of the software based on the requirement

TEXT BOOK

REFERENCES:
COURSE OBJECTIVES:
• To understand the fundamentals of Cryptography
• To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
• To understand the various key distribution and management schemes.
• To understand how to deploy encryption techniques to secure data in transit across data networks
• To design security applications in the field of Information technology

UNIT-I
Classical Encryption Techniques: Symmetric Cipher Model, Cryptography, Cryptanalysis and Brute-Force Attack, Substitution Techniques, Caesar Cipher, Monoalphabetic Cipher, Playfair Cipher, Hill Cipher, Polyalphabetic Cipher, One Time Pad. Block Ciphers and the data encryption standard: Traditional block Cipher structure, stream Ciphers and block Ciphers, Motivation for the feistel Cipher structure, the feistel Cipher, The data encryption standard, DES encryption, DES decryption, A DES example, results, the avalanche effect, the strength of DES, the use of 56-Bit Keys, the nature of the DES algorithm, timing attacks, Block cipher design principles, number of rounds, design of function F, key schedule algorithm.

UNIT-II
Public-Key Cryptography and RSA: Principles of public-key cryptosystems. Public-key cryptosystems. Applications for public-key cryptosystems, requirements for public-key cryptosystems, public-key cryptanalysis. The RSA algorithm, description of the algorithm, computational aspects, the security of RSA: Other Public-Key Cryptosystems: DiffieHELLMAN key exchange, The algorithm, key exchange protocols, man in the middle attack, Elgamal Cryptographic systems, Elliptic curve arithmetic, abelian groups, elliptic curves over real numbers, elliptic curves over Zp, elliptic curves over GF(2m), Elliptic curve cryptography, Analog of Diffie-Hellman key exchange, Elliptic curve encryption/ decryption, security of Elliptic curve cryptography, Pseudorandom number generation based on an asymmetric cipher, PRNG based on RSA.

UNIT-III
Key Management and Distribution: Symmetric key distribution using Symmetric encryption, A key distribution scenario, Hierarchical key control, session key lifetime, a transparent key control scheme, Decentralized key control, controlling key usage, Symmetric key distribution using asymmetric encryption, simple secret key distribution, secret key distribution with confidentiality and authentication, A hybrid scheme, distribution of public keys, public announcement of public keys, publicly available directory, public key authority, public keys certificates, X-509 certificates, Certificates, X509 version 3, public key infrastructure. User Authentication: Remote user Authentication principles, Mutual Authentication, one way Authentication, remote user Authentication using Symmetric encryption, Mutual Authentication, one way Authentication, Kerberos, Motivation, Kerberos version 4, Kerberos version 5, Remote user Authentication using Asymmetric encryption.
Mutual Authentication, one way Authentication, federated identity management, identity management, identity federation, personal identity verification.

10 hours

UNIT-IV


10 hours

UNIT-V


10 hours

COURSE OUTCOMES:
Upon Completion of the course, the students will be able to

• Implement basic security algorithms required by any computing system
• Analyze the vulnerabilities in any computing system and hence be able to design a security solution.
• Analyze the possible security attacks in complex real time systems and their effective counter measures
• Identify the security issues in the network and resolve it.

Text Books:

Reference
Course Title: Data Mining & Data Warehousing
Course Code: 15SSE212
Credits(L:T:P): 4:0:0
Core/Elective: Elective
Type of Course: Lecture
Total Contact Hours: 50

COURSE OBJECTIVES:
• To expose the students to the concepts of Data warehousing Architecture and Implementation
• To Understand Data mining principles and techniques and Introduce DM as a cutting edge business intelligence
• To learn to use association rule mining for handling large data
• To understand the concept of classification for the retrieval purposes
• To know the clustering techniques in details for better organization and retrieval of data

TOPICS

UNIT-I
Introduction: What is a Data Warehouse?, A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Data cube Technology, From Data warehousing to Data Mining, Data Mining Functionalities, Data cleaning, Data Integration and Transformation, Data Reduction.
10 hours

UNIT-II
Data Mining Primitives, Languages And System Architectures: Data Mining primitives, Presentation and Visualization of Discovered patterns, A Data Mining Query Language.
MINING ASSOCIATION RULES IN LARGE DATA BASES: Association Rule Mining
Single –Dimensional Boolean Association Rules From Transactional Databases, Mining Multilevel Association Rules from Transactional Databases.
10 hours

UNIT-III
Classification and Prediction: Issues regarding Classification and Prediction, classification by Decision tree induction, Bayesian classification, Classification by back propagation, Classification Based on the concepts from association rule mining. Other classification methods, prediction.
10 hours

UNIT-IV
Cluster Analysis: What is Cluster Analysis? Types of data in cluster Analysis: a Categorization of Major Clustering Methods, Partitioning Methods, And Hierarchical methods, Density-Based Methods, Model-Based Clustering Methods:
Statistical Approach, Neural Network Approach Outliner Analysis.
10 hours

UNIT-V
Applications and Trends In Data Mining: Data mining application, Data mining system Products research Prototypes, Additional Themes on Data Mining, Data Mining and Intelligent Query Answering, Tends in Data Mining.
10 hours
COURSE OUTCOMES:
Upon completion of the course, students shall be able to
• Store voluminous data for online processing
• Preprocess the data for mining applications
• Apply the association rules for mining the data
• Design and deploy appropriate classification techniques
• Cluster the high dimensional data for better organization of the data
• Discover the knowledge imbibed in the high dimensional system

Text Books:
Course Title: Information Retrieval
Course Code: 15SSE213
Credits(L:T:P): 4:0:0
Core/Elective: Elective
Type of Course: Lecture
Total Contact Hours: 50

COURSE OBJECTIVES:
• To understand the basics of Information Retrieval with pertinence to modeling, query operations and indexing
• To get an understanding of machine learning techniques for text classification and clustering
• To understand the various applications of Information Retrieval giving emphasis to Multimedia IR, Web Search
• To understand the concepts of queries specification judgment and search engines

TOPICS

UNIT-I

UNIT-II

UNIT-III

UNIT-IV
Indexing and Searching: Introduction, Inverted Files, Other indices for text, Boolean queries, Sequential searching, Pattern matching, Structural queries, Compression. Parallel and Distributed IR: Introduction, Parallel IR, Distributed IR

UNIT-V
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. Searching the Web: Introduction, Challenges, Characterizing the web, Search engines, Browsing, Meta searchers, Finding the needle in the haystack, Searching using hyperlinks.

10 hours
COURSE OUTCOMES:
Upon completion of the course, the students will be able to
- Build an Information Retrieval system using the available tools
- Identify and design the various components of an Information Retrieval system
- Apply machine learning techniques to text classification and clustering which is used for efficient Information Retrieval
- Analyze the Web content structure
- Design an efficient search engine

TEXT BOOKS:

REFERENCE BOOKS:
Course Title: Information Storage Management
Course Code: 15SSE214
Credits(L:T:P): 4:0:0
Core/Elective: Elective
Type of Course: Lecture
Total Contact Hours: 50

COURSE OBJECTIVES:
- To outline basic terminology and components in information storage and retrieval systems
- To compare and contrast information retrieval models and internal mechanisms such as Boolean, Probability, and Vector Space Models
- To describe current trends in information retrieval such as information visualization.
- To understand Backup process and securing of management storage infrastructure

TOPICS

UNIT-I
Introduction to Information Storage: Information Storage, Evolution of Storage Architecture, Data center Infrastructure, Virtualization and cloud computing, Data Center Environment: Application, Database Management System(DBMS), Host(compute), Connectivity, Storage, Disk Drive Components, Disk Drive Performance, Host Access to Data, Direct-Attached Storage, Storage Design Based On Application, Disk Native Command Queuing, Introduction to Flash Drives, Concept in Practice: VMware ESXi, Data Protection: RAID: RAID Implementation Methods, RAID Array Components, RAID Techniques, RAID Levels, RAID Impact on Disk Performance, RAID Comparison, Hot Spares.

10 hours

UNIT-II

10 hours

UNIT-III

10 hours
UNIT-IV
Backup and Archive: Backup Purpose, Backup Considerations, Backup Granularity, Recovery Considerations, Backup Methods, Backup Architecture, Backup and Restore Operation, Backup Topologies, Backup in NAS Environments, Backup Targets, Data Deduplication for Backup, Backup in Virtualized Environments, Data Archive, Archiving Solution Architecture, Concepts in Practice: EMC Networker, EMC Avamar, and EMC Data domain.

10 hours

UNIT-V

10 hours

COURSE OUTCOMES
After completion of the course, the student shall be able to
- Recognize the role and use are technology in business systems and operations
- Identify and describe organizational structure and business processes within these
- Implement information systems in industry.
- Choose backup method and replication method.
- Provide securing of management storage infrastructure.

TEXT BOOK:
1. EMC2: Information Storage and Management, Willey India 2013.

REFERENCE BOOKS:
4. Additional resource material on www.emc.com/resource-library/resource-library.esp
Course Title: Agile Technologies
Course Code: 15SSE215
Credits(L:T:P): 4:0:0
Core/Elective: Elective
Type of Course: Lecture
Total Contact Hours: 50

COURSE OBJECTIVES
• To understand how an iterative, incremental development process leads to faster delivery of more useful software
• To understand the essence of agile development methods
• To understand the principles and practices of extreme programming
• To understand the roles of prototyping in the software process
• To understand the concept of Mastering Agility

TOPICS

UNIT-I
Why Agile?: Understanding Success, Beyond Deadlines, The Importance of Organizational Success, Enter Agility, How to Be Agile?: Agile Methods, Don’t Make Your Own Method, The Road to Mastery, Find a Mentor.

10 hours

UNIT-II

10 hours

UNIT-III

10 hours

UNIT-IV

10 hours
UNIT-V

10 hours

COURSE OUTCOMES
Upon completion of the course, students shall be able to
• Understand the XP Lifecycle, XP Concepts, Adopting XP
• Work on Pair Programming, Root-Cause Analysis, Retrospectives, Planning, Incremental Requirements, Customer Tests
• Implement Concepts to Eliminate Waste

TEXT BOOK:

REFERENCE BOOKS:
Course Title: Service Oriented Architecture
Course Code: 15SSE216
Credits(L:T:P): 4:0:0
Type of Course: Lecture
Core/Elective: Elective
Total Contact Hours: 50

COURSE OBJECTIVES:
• To understand various architecture for application development
• To understand the importance of SOA in Application Integration
• To learn web service and SOA related tools
• To Learn implementation details of SOA
• To understand varies case studies

TOPICS

UNIT-I

UNIT-II

UNIT-III

UNIT-IV

UNIT-V
COURSE OUTCOMES:
After successful completion of the course, the student shall be able to
• Compare different IT architecture
• Analyze and design of SOA based applications
• Implement web service and realize of SOA
• Implement REST full services
• Design and implement of SOA based Application Integration using BPEL

TEXT BOOK:

REFERENCE BOOKS:
Course Title: Distributed Computing  
Course Code: 15SSE217  
Credits(L:T:P): 4:0:0  
Type of Course: Lecture  
Core/Elective: Elective  
Total Contact Hours: 50

COURSE OBJECTIVES:
- To learn Basic Concepts of DSM, Hardware DSM
- To understand File Sharing, DFS Implementation, Replication in DFS,
- To explore the concepts of Cryptography, Secure channels, Access control
- To understand real-time Distributed Operating System and emerging trends in Distributed Computing

TOPICS

UNIT-I

10 hours

UNIT-II

10 hours

UNIT-III

10 hours

UNIT-IV

10 hours

UNIT-V

10 hours
COURSE OUTCOMES:

Upon completion of the course, students shall be able to
- Explain distributed systems concepts
- Demonstrate an ability to apply theory and techniques to unseen problems.
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Explore the various resource management techniques for distributed systems

TEXT BOOK:

REFERENCE BOOK: