Distributed System

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Objectives:
This course provides different aspects of Distributed System including overview of Middleware System, clock synchronization, coordination and agreement in distribution system. It also highlights the issues of security and failure handling in distributed system.

1. **Introduction to Distributed System** **4 hrs**
   - 1.1. Definition, merits and demerits
   - 1.2. Main characteristics, design goals and challenges
   - 1.3. Models of Distributed system
     - 1.3.1. Architectural
     - 1.3.2. Fundamental
     - 1.3.3. Failure Model

2. **Distributed Objects and Remote Communication** **10 hrs**
   - 2.1. Basic Communication Primitives
   - 2.2. Remote Procedure Call
   - 2.3. Communication between distributed objects
   - 2.4. Remote Method Invocation
   - 2.5. Events and Notifications
   - 2.6. Distributed heterogeneous application and CORBA
   - 2.7. Name Services & Domain Name System
   - 2.8. Distributed File System: SUN-NFS

3. **Time In Distributed System** **6 hrs**
   - 3.1. Physical Clocks
   - 3.2. Clock Synchronization Algorithms (Christians, Berkeley & NTP)
   - 3.3. Logical Clocks (Lamport timestamps, Vector timestamps)
   - 3.4. Causal Ordering of Messages
   - 3.5. Global State and State Recording

4. **Agreement in Distributed System** **6 hrs**
   - 4.1. Distributed Election
     - 4.1.1. The Bully Algorithm
     - 4.1.2. A Ring algorithm
   - 4.2. Algorithms for Mutual Exclusion
     - 4.2.1. Central coordinator Algorithm
     - 4.2.2. Token and non-token based Ricart & Agrawala algorithm
     - 4.2.3. Token Ring Algorithm

5. Replication and Fault Tolerance 10 hrs

5.1. Reasons for replication
5.2. Object replication
5.3. Replication as scaling techniques
5.4. Fault Tolerance Services: Active & Passive Replication
5.5. Failure masking by redundancy
5.6. Agreement in faulty system
5.7. Reliable client server communication
5.8. Distributed commit
5.9. Recovery Approaches

6. Transaction and Concurrency Control 6 hrs

6.1. Distributed transactions
  6.1.1. Flat and nested transactions
  6.1.2. Concurrency control mechanisms
    6.1.2.1. Locks
    6.1.2.2. Optimistic concurrency control
    6.1.2.3. Timestamp ordering
  6.1.3. Distributed deadlocks
  6.1.4. Atomic commit protocols

7. Distributed System Application 3 hrs

7.1. Overview of grid, cloud and cluster computing
7.2. TIB/Rendezvous
7.3. JINI
7.4. WWW

Laboratory:

1. RMI & RPC implementation
2. SUN-NFS implementation
3. Simulation of clock synchronization algorithms
4. Simulation/implementation of election algorithm
5. Case Study of CORBA, JINI, MACH, TIB/Rendezvous

Text Books:

1. Georce Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems Concepts and Design", 4th Edition, Pearson Education
2. A.S. Tanenbaum, M. Van Steen, "Distributed Systems Principles and Paradigms"

Enterprise Application Development

Pre-requisite: Knowledge of Programming. An understanding of HTML, basic knowledge of SQL and a basic knowledge of working at the UNIX or Windows command-line is useful but not required.

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

- It will introduce students to the enterprise application development environment and provides the knowledge about most recent developments in enterprise application frameworks.
- Students will have better understanding of software architecture and various design principles.
- Students will have the understanding of good coding practices, including documentation, testing and builds.
- Students will learn about the web application security vulnerabilities and how to solve those security issues.

Contents

1. Introduction **7 hrs**

1.1Enterprise Application Architecture

1.1.1Platform for Enterprise Solutions: E.g. Java EE, Enterprise Java Bean

1.1.2 Enterprise Architecture Frameworks: E.g. Government enterprise architecture , Zachman Framework.

1.2 Web Technologies Concepts

1.2.1 Protocols

1.2.2 Communication Types

1.2.3 Web Services: Data Exchange Formats: (XML, JSON, RDF), Data Standards & Interoperability(SOAP, WSDL, UDDI, and ebXML)

2. Development Process Management **7 hrs**

2.1 Source code Management

2.2 Continuous Integration

2.3 Software Testing

2.4 Software Documentation (UML Diagram and Tools)

3. Design Patterns **10 hrs**

3.1 Types of Patterns

3.2 Dependency Injection and Inversion of Control.

3.3 Convention vs Configuration.

3.4 Factory pattern.

3.5 Singleton pattern.
3.6 Lazy initialization.

4. Web Application Architecture **7 hrs**

4.1 Layered Architecture for web applications:
- 4.1.1 Presentation Layer
- 4.1.2 Data source Layer
- 4.1.3 Domain Logic & Business Logic.

4.2 MVC pattern

5. SOA and RESTFUL Web services **5 hrs**

5.1 Resource-Oriented Architecture
5.2 Resource-Oriented Architecture Analysis and Design
- 5.2.1 Designing Read-Only Resource-Oriented Services
- 5.2.2 Designing Read/Write Resource-Oriented Services

6. Software Security **3 hrs**

6.1 Basic attacks
6.2 State-based attacks
6.3 Dos and Don'ts of client authentication
6.4 Cross-site scripting
6.5 SQL injection

7. Software Development Methodologies **6 hrs**

7.1 Agile
7.2 Test Driven
7.3 Behavior Driven

Lab sessions:

Lab sessions should be conducted to provide students experience with specific technologies and techniques used across many applications.

References

1. Fowler, M., *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2003.
2. E. Jendrock,R. Cervera-Navarro,I. Evans,D. Gollapudi,K. Haase,W. Markito,C. *The Java EE 6 Tutorial*, 3rd edition, Addison-Wesley, 2006.
   (Available online: http://docs.oracle.com/javaee/6 /tutorial/doc/)
3. Anderson, E., Greenspun, P., and Grumet, A.,*Software Engineering for Internet Applications*, 1st ed. MIT Press, 2006.
4. Andrews, M. and Whitaker, J.A., *How to Break Web Software: Functional and Security Testing of Web Applications and Web Services*. Addison-Wesley, 2006.
5. Fowler (2003) UML Distilled: *A Brief Guide to the Standard Object Modeling Language*, 3rd Edition. Addison-Wesley.

Image Processing and Pattern Recognition (3 – 1 - 2)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

- To develop a theoretical foundation of fundamental Digital Image Processing concepts.
- To provide mathematical foundations for digital manipulation of images; image acquisition; preprocessing; segmentation; Fourier domain processing; and compression.
- To gain experience and practical techniques to write programs for digital manipulation of images; image acquisition; preprocessing; segmentation; Fourier domain processing; and compression.

Course Contents:

1. Introduction to Digital Image Processing (4 hrs)

1.1. The origins of Digital Image Processing
1.2. Examples of Fields that Use Digital Image Processing
1.3. Fundamentals Steps in Image Processing
1.4. Elements of Digital Image Processing Systems
1.5. Image Sampling and Quantization
1.6. Some basic relationships like Neighbors, Connectivity, Distance Measures between pixels,
1.7. Elements of Visual Perception

2. Image Enhancement in Spatial Domain (7 hrs)

2.1. Some basic Gray Level Transformations
  2.1.1. Point operations
  2.1.2. Contrast stretching,
  2.1.3. Thresholding,
  2.1.4. Digital negative,
  2.1.5. Intensity level slicing
  2.1.6. Bit Plane Slicing
2.2. Histogram Processing and Equalization
2.3. Enhancement Using Arithmetic and Logic operations
2.4. Basics of Spatial Filters
2.5. Smoothening and Sharpening Spatial Filters
  2.5.1. Averaging
  2.5.2. Median filtering
  2.5.3. Spatial Low Pass
  2.5.4. High pass filtering
  2.5.5. Magnification by replication and interpolation

2.6. Combining Spatial Enhancement Methods

3. Image Enhancement in the Frequency Domain (6 hrs)

3.1. Introduction to Fourier Transform and the frequency Domain
3.2. Computing and Visualizing the 2D DFT
3.3. Smoothing Frequency Domain Filters
3.4. Sharpening Frequency Domain Filters,
3.5. Other Image Transforms
    3.5.1. Hadamard transform
    3.5.2. Haar transform
    3.5.3. Discrete Cosine transform
3.6. Fast Fourier Transform

4. Image Restoration (4 hrs)

4.1. A model of The Image Degradation / Restoration Process,
4.2. Noise Models Restoration in the presence of Noise Only Spatial Filtering
4.3. Periodic Noise Reduction by Frequency Domain Filtering

5. Image Compression (6 hrs)

5.1. Coding Redundancy
    5.1.1. Huffman coding
5.2. Interpixel Redundancy
    5.2.1. Run length Coding
5.3. Psychovisual Redundancy
5.4. Image Compression models
5.5. Lossless and Lossy Compressions
    5.5.1. Predictive coding

6. Introduction to Morphological Image Processing (4 hrs)

6.1. Logic Operations involving binary images
6.2. Dilation and Erosion
6.3. Opening and Closing

7. Image Segmentation (7 hrs)

7.1. Detection of Discontinuities
7.2. Edge linking and boundary detection
7.3. Thresholding
7.4. Region Based Segmentation

8. Representations and Description (3 hrs)

8.1. Introduction to some descriptors
    8.1.1. Chain codes
    8.1.2. Signatures
    8.1.3. Shape Numbers,
    8.1.4. Fourier Descriptors

9. Object Recognition (3 hrs)

9.1. Patterns and pattern classes

9.2. Decision-Theoretic Methods

9.3. Overview of Neural Networks in Image Processing

10. Pattern Recognition (1 hr)

10.1. Overview of pattern recognition

Laboratory:

- Student should write programs related to different image enhancement techniques, image restoration techniques, morphological operations and image segmentation techniques.

Text Book:

Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Prentice Hall of India Pvt. Ltd., 2010.

References:

1. I. Pitas, "Digital Image Processing Algorithms", Prentice Hall, 2009.
2. A. K. Jain, "Fundamental of Digital Image processing", Prentice Hall of India Pvt. Ltd., 2011.
3. K. Castlemann, "Digital image processing", Prentice Hall of India Pvt. Ltd., 2010.
4. R. C. Gonzalez and P. Wintz, "Digital Image Processing", Addison-Wesley Publishing, 2009.
5. P. Monique and M. Dekker, "Fundamentals of Pattern recognition", 2007.
6. M. James, "Pattern recognition", BSP professional books, 2008.

Real-Time Systems (3-0-3)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

- To provide basic theoretical background and profound practical framework for software engineers to design, implement and validate Real-Time systems.
- To provide the knowledge of Real-Time Scheduling issues like Task scheduling, Resource Scheduling and Contentions
- To familiarize the students with the behavior of Real-Time Operating systems, Real-Time Tasks and the POSIX compliance issues of Real-Time Kernels.

Course Contents:

1. Real-Time: Concepts **8 hrs**

1.1 Terminologies:

1.1.1 Systems Concepts

1.1.2 Real-Time Definitions

1.1.2.1 Hard, Soft and Firm Real-Time Systems

1.1.3 Types of Real-Time Tasks

1.1.3.1 Periodic, Aperiodic and Sporadic Tasks

1.1.3 Events and Determinism

1.1.4 CPU Utilization

1.2 Real-Time System Design Issues

1.3 Examples of Real-Time Systems

1.4 Common Misconceptions regarding Real-Time Systems

1.5 Historical Aspects of Real-Time Systems

1.6 Brief Survey of Real-Time Programming Languages

1.6.1 Ada 95, C, C++, C#, Fortran, Java, Occam 2, Special Real-Time Languages

2. Real-Time Specification and Design Techniques **4 hrs**

2.1 Requirements Specification for Real-Time Systems

2.1.1 Z, Finite State Machines, State Charts, Petri Nets

2.2 Recommendations on Specification Approach for Real-Time Systems

2.3 Real-Time System Design Activity

2.3.1 Problems with Structured Analysis and Structured Design in Real-Time Applications

2.3.2 Real-Time Extensions of Structured Analysis and Structured Design

2.3.3 Real-Time Extensions of Unified Modeling Language

3. Real-Time Operating Systems **8 hrs**

3.1 Real-Time Kernels

3.1.1 Nano Kernel and Micro Kernel

3.1.2 Pseudokernels

3.1.3 Interrupt-Driven Systems

    - 3.1.4 Preemptive-Priority Systems
    - 3.1.5 Hybrid Systems
    - 3.1.6 The Task-Control Block Model
        - 3.1.6.1 Task States
        - 3.1.6.2 Task Management
        - 3.1.6.3 Resource Management
- 3.2 Theoretical Foundations of Real-Time Operating Systems
    - 3.2.1 Process Scheduling
        - 3.2.1.1 Task Characteristics of a Real Workload
        - 3.2.1.2 Typical Task Model
    - 3.2.2 Round-Robin Scheduling
    - 3.2.3 Cyclic Executives
    - 3.2.4 Fixed-Priority Scheduling–Rate-Monotonic Approach
    - 3.2.5 Dynamic-Priority Scheduling: Earliest-Deadline–First Approach

4. Inter-task Communication, Synchronization and Memory Management **7 hrs**

- 4.1Intertask Synchronization Mechanisms and Resource Contention
    - 4.1.1 Buffering Data: Time-Relative Buffering, Ring Buffers
    - 4.1.2 Mailboxes
    - 4.1.3 Critical Regions
    - 4.1.4 Semaphores
    - 4.1.5 Resource Contentions and Priority Inversion
        - 4.1.5.1The Priority Inheritance Protocol
        - 4.1.5.2 Priority Ceiling Protocol
- 4.2 Real-Time Memory Management
    - 4.2.1 Process Stack Management
    - 4.2.2 Multiple-Stack Arrangements
    - 4.2.3 Memory Management in the Task-Control-Block Model
    - 4.2.4 Swapping
    - 4.2.5 Overlays
    - 4.2.6 Block or Page Management
    - 4.2.7 Memory Locking
    - 4.2.8 Working Sets
    - 4.2.9 Real-Time Garbage Collection
    - 4.2.10 Contiguous File Systems

5. Performance Analysis And Optimization of Real-Time Systems **6 hrs**

- 5.1 Challenges in Analyzing Real-Time Systems
- 5.2 Performance Analysis
    - 5.2.1 Analysis of Round-Robin Systems
    - 5.2.2 Response-Time Analysis for Fixed-Period Systems
    - 5.2.3 Response-Time Analysis: RMA Example
    - 5.2.4 Analysis of Sporadic and Aperiodic Interrupt Systems
- 5.3 Performance Optimization
    - 5.3.1 Compute at Slowest Cycle
    - 5.3.2 Scaled Numbers
    - 5.3.3 Binary Angular Measure
    - 5.3.4 Optimizing Memory Usage
- 5.4 Analysis of Memory Requirements

5.5 Reducing Memory Utilization
- 5.5.1 Variable Selection
- 5.5.2 Memory Fragmentation

6. Fault-Tolerance and System Integration Tools and Techniques — 6 hrs

6.1 Faults and Fault-Tolerance
- 6.1.1 Spatial Fault-Tolerance
- 6.1.2 Software Black Boxes
- 6.1.3 N-Version Programming
- 6.1.4 Built-In-Test Software

6.2 Systems Integration
- 6.2.1 Goals of System Integration
- 6.2.2 System Unification
- 6.2.3 System Verification
- 6.2.4 System Integration Tools
- 6.2.5 A Simple Integration Strategy
- 6.2.6 Patching
- 6.2.7 The Probe Effect

7. Case Study: Real-time POSIX — 6 hrs

3.5.1 Threads and Pthreads
- 3.5.1.1 Multi-threaded Real-Time Applications
- 3.5.1.2 Thread Creation Mechanism
- 3.5.1.3 Thread Priority, Thread Periodicity

3.5.2 POSIX Mutexes and Condition Variables

3.5.3 POSIX Semaphores and Shared Memory

5.5.4 POSIX Messages

3.5.5 Real-Time POSIX Signals

3.5.6 Clocks and Timers

3.5.7 POSIX Memory Locking

Laboratory Exercise:

Laboratory Exercise comprises of performing analysis, design, development, evaluation of a Real-Time System. It includes building Real-Time (multithreaded) applications using POSIX compliant APIs and running them on a Real-Time Operating System platform (e.g. RT-Linux or RTAI etc).

Text Books:

1. Laplante, Phillip A., Real-Time System Design and Analysis – An Engineer's Hand-Book, 3rd Edition, IEEE Press, IEEE Computer Society Press, PHI, 2004, ISBN: 81-203-1684-3
2. Liu. Jane W.S., Real-Time Systems, PHI, 2009, ISBN:978-81-7758-575-9

Reference Books:

1. Grehan, R., Moote, R., and Cyliax, I., Real-Time Programming – A Guide to 32-bit Embedded Development, Addison-Wesley, 1999, ISBN: 0-201-48550-0
2. Burns, A., and Welling, A., Real-Time System and their Programming Languages – Ada95, Real-Time Java and Real-Time C/POSIX, Third Edition, Addison-Wesley, 2001, ISBN: 0-201-40365-X

Software Testing Verification Validation and QA

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

- To provide basic knowledge of Software Testing.
- To provide the knowledge of Quality assurance in software development process.
- To make familiar with various software testing tools and implementation of tools.

Course Contents:

1. **Introduction To Software Testing** **(2 hrs)**
   - 1.1 Software Testing Background
   - 1.2 Career in Software Testing

2. **Fundamentals of Testing** **(7 hrs)**
   - 2.1 Testing
   - 2.2 Significance of Testing(Software Systems Context, Causes of Software Defects, Role of Testing in Software Development, Maintenance and Operations, Testing and Quality, Ending Test)
   - 2.3 Testing principles
   - 2.4 Fundamental Test process (Test planning and control, Test Analysis and Design, Test Implementation and Execution, Evaluating Exit Criteria and Reporting, Test Closure Activities)
   - 2.5 The Psychology of Testing
   - 2.6 Code of Ethics

3. **Verification and validation.** **(4 hrs)**
   - 3.1 Software Verification: Walkthroughs, Inspections,Audits,Verification Process,Implementation of Verification Activities in Projects
   - 3.2 Validation: Validation of Software Designs Validating the Product Specifications, Validating the Software Product,Testing Different Types of Software Products,Testing Basics,Approaches to Testing,Test Case Design,Test Environment,Testing Scenarios,Product Testing
   - 3.3 Validating preliminary designs through prototyping

4. **Testing Throughout the Software Life Cycle** **(3 hrs)**
   - 4.1 Software Development Models
   - 4.2 Test levels (Unit testing, Integration testing, System testing, Acceptance testing)
   - 4.3 Test types (Functional Testing, Non-functional Testing, Structural Testing, Re-testing and Regression Testing)

5. Static Techniques (4 hrs)

5.1 Static Techniques and the Test Process

5.2 Review Process (Activities of a Formal Review, Roles and Responsibilities, Types of Reviews, Success Factors for Reviews)

5.3 Static Analysis by Tools

6. Test Design Techniques (6 hrs)

6.1 The Test Development Process

6.2 Categories of Test Design Techniques

6.3 Specification-based or Black-box Techniques (Equivalence partitioning, Boundary Value analysis, Decision table testing, State transition testing, Use case testing)

6.4 Structure-based or White-box Techniques (Statement Testing and Coverage, Decision Testing and Coverage, Other Structure-based Techniques)

6.5 Experience-based Techniques

6.6 Choosing Test Techniques

7. Test Management (9 hrs)

7.1 Project Analysis, Understanding Project (In-house projects, Projects for clients), Understanding Technology for better testing, Understanding Risk, Understanding Project Complexity and Size

7.2 Test Organization (Test Organization and Independence, Tasks of the Test Leader and Tester)

7.3 Test Planning and Estimation (Test Planning, Test Planning Activities, Entry Criteria, Exit Criteria, Test Estimation, Test Strategy, Test Approach)

7.4 Test Progress Monitoring and Control (Test Progress Monitoring, Test Reporting, Test Control)

7.5 Configuration Management

7.6 Risk and Testing (Project Risks, Product Risks)

7.7 Incident Management

8. Tool Support for Testing (4 hrs)

8.1 Types of Test Tools (Tools for Testing, Static testing, Management testing, Test specification, Test execution, Performance)

8.2 Effective Use of Tools: Potential Benefits and Risks of Tools Support for Testing, Special Considerations for Some Types of Tools

8.3 Introducing a Tool into an Organization

9. Testing Web and Mobile Application (4 hrs)

9.1 Types of web applications, Impact of Technology and Frameworks on web application testing, Types and Levels of Test

9.2 Under Mobile Platform (General Checklist for mobile application testing, Testing with Emulator VS Testing with Real Device)

10. Quality management. (2 hrs)

10.1Measuring software quality

10.2Software quality standards

Laboratory:

Laboratory exercises should cover the design and development of SQA with related software tools. Laboratory exercises must be designed to develop testing procedure on various test cases and using test scenarios on wide range.

Text Books:

1. Software Testing: Ron Patton *Sams Publishing, Second Edition 2001*
2. Mastering Software Quality Assurance: *Murali Chemuturi, J. Ross Publishing, 2010*