



**Department of Computer Science**  
**Gujarat Vidyapith, Ahmedabad – 14**

**MCA - SEMESTER - V**

**MCA - 501: WEB TECHNOLOGY – II**

**UNIT - I**

**1 CREDIT**

- **Network programming**

Basic of Networking , Client server in Networking , Architecture, Tcp, Udp, Understanding port, TCP and Socket (Client & Server Sockets) , UDP and Datagram, Multicasting, URL,URL Constructor, MalformedURLException, Parsing URL, Reading directly From URL, Connecting URL, Reading From and Writing to URL Connection examples.

- **Object Serialization**

What is Object Serialization & De-Serialization , Dealing with Objects and Object Reference, An example to read /write a Primitive and custom Class, performance issues (Static, Non Static and variant ), SUID, Version controlling . examples.

**UNIT - II**

**1 CREDIT**

- **Reflection**

Examining Classes ,Retrieving Class Objects, Getting the Class Name, Discovering Class Modifiers, Finding Super classes, Identifying the Interfaces Implemented by a Class, Examining Interfaces, Identifying Class Fields, Discovering Class Constructors, Obtaining Method Information. Examples

- **Remote Method Invocation (RMI)**

Introduction of distributed Application , Introduction to RMI, RMI Architecture, Implementing RMI, Unicast Remote Object, Activatable, Passing values and Objects, Callback, Examples



### **UNIT - III**

**1 CREDIT**

- **Internationalization**

Introduction, Internationalization scenario, Rules of I18N, Java and Unicode, The Locale Class, Packaging Locale-Sensitive Data, Input-Output, Resource bundles Class, Readers and Writers, JavaText Framework, Formatting examples using Swing.

- **JAR Archive Files**

Introduction ,Jar, Bundling, Compression, different options to create and extract jar files Backward Compatibility, Portability, Security, JarTool, Applet Tag, The Manifest file, Private Key, Public Key, Jarsigner

- **Security**

Security Models, Encryption and Decryption, Public Key, Private Key and Certificate, Security and Integrity, Security Tool and API, Signed Applet examples.

### **UNIT - IV**

**1 CREDIT**

- **Hibernate**

- Understanding Object/Relational Persistence

- What is persistence?
- The Paradigm Mismatch
- Persistence Layers and Alternatives
- Object/relational Mapping

- Starting a project

- Starting a Hibernate Project
- Reverse Engineering a Legacy Database
- Integration with Java EE service

- Domain Models And Metadata

- The Caveat Emptor application
- Implementing the Domain Model
- Object/relational Mapping Metadata
- Alternative Entity Representation

- Mapping Persistent Classes

- Understanding Entities and Value Types
- Mapping Entities with Identity
- Fine-grained Models and Mappings

- Inheritance and Custom Types

- Mapping Class Inheritance
- The Hibernate Type System
- Creating Custom Mapping Types



**PRACTICAL**

**2 CREDIT**

**LAB – WEB TECHNOLOGY – II**

**Book:-**

- Special Edition Using Java 2 Platform By Joseph L. Weber Pub:- PHI
- Java 2 Programming Black Book By Steven Holzner et al.

\*\*\*\*\* END of MCA – 501 \*\*\*\*\*



**Department of Computer Science**  
**Gujarat Vidyapeeth, Ahmedabad - 14**  
MCA SEMESTER – V  
**MCA-502: COMPUTER NETWORK – II**  
  
**Section - I Computer Network**

**UNIT- I**

**1 CREDIT**

**The Network Layer**

- **Internetworking**
  - Need for Network Layer
  - Internet as a Packet Switching Network
  - Internet as a Connectionless Switching Network
  
- **Addressing**
  - IP4 Addresses
    - Address Space
    - Notations
    - Classful Addressing
    - Subnets
    - Classless Addressing
  - IPV6 Addresses
    - Structure
    - Address Space
  
- **Network Layer Protocol: IPV4, IPV6, Address Mapping, ARP, RARP, ICMP**
  - IPV4
    - Datagram
    - Options
    - Fragmentation
  - IPV6
    - Advantage
    - Packet Format
    - Extension Headers
  - Address Mapping
    - Mapping Logical To Physical Address: ARP
    - Mapping Physical To Logical Address: RARP, BOOTP and DHCP
  - ICMP
    - Types of Message
    - Message Format
    - Error Reporting
    - Query
  - IGMP
    - Group Management
    - IGMP Messages
    - Message Format
    - IGMP Operation



- **Network Layer: Delivery, Forwarding and Routing**
  - Introductions to Delivery, Forwarding and Routing
  - Routing Table
  - Unicast Routing Protocols
    - Optimization
    - Interior and Exterior Routing
    - Interior Routing: RIP and OSPF
    - Exterior Routing: BGP Protocol
  - Multicast Routing Protocol
    - Unicast, Multicast and Broadcast
    - Application
    - Multicast Routing
    - Routing Protocol

## **UNIT- II**

**1 CREDIT**

### **Transport Layer**

- Process-to-Process Delivery
  - Multiplexing and Demultiplexing
  - Connectionless Vs. Connection-Oriented Service
  - Reliable Vs. Reliable Service
  - Transmission Control Protocol: TCP
    - TCP Services
    - TCP Features
    - Segment
    - TCP Connection
    - Flow Control
    - Error Control
    - Congestion Control
  - User Datagram Protocol: UDP
    - User Datagram
    - Checksum
    - UDP Operation
    - Use of UDP
- Congestion Control and QoS
  - Data Traffic - Traffic Descriptor, Traffic profiles
  - Congestion – Network Performance
  - Congestion control
    - Open-loop Congestion Control
    - Closed-loop Congestion Control
    - Congestion Control in TCP
  - Quality of Services
    - Flow Characteristics
    - Flow Classes
  - Techniques to Improve QoS
    - Scheduling
    - Traffic Shaping
    - Resource Reservation
    - Admission Control



## Application Layer

- Client-Server Paradigm, Client, Server, Socket Interface, Addressing, Port
- Domain Name System (DNS)
- Electronic Mail, Email Delivery – SMTP, Introduction to POP and IMAP

## SECTION - II Inter Networking with TCP/IP

### UNIT- III

**2 CREDIT**

#### **Internetworking with TCP/IP**

- Introduction and overview.
  - Use of TCP/IP.
  - Designing Applications for a Distributed Environment.
  - Standard and Nonstandard Application Protocols.
  - An example of Standard Application Protocol Use.
  - An example Connection.
- The Client Server Model and Software Design.
  - Introduction.
  - Motivation.
  - Terminology and Concepts.
- Concurrent Processing In Client-Server Software.
  - Introduction.
  - Concurrency in Networks.
  - Concurrency in Servers.
  - Terminology And Concepts.
  - An example Of Concurrent Process Design.
  - Executing New Code.
  - Context Switching and Protocol Software Design.
- Program interface to Protocols.
  - Introduction.
  - Loosely Specified Protocol Software Interface.
  - Interface Functionality.
  - Conceptual Interface Specification.
  - System Calls.
  - Two Basic Approaches To Network Communication.
  - The Basic I/O Functions available in UNIX.
  - Using UNIX I/O with TCP/IP.
- The Socket Interface.
  - Introduction.



- Berkeley Sockets.
- Specifying A Protocol Interface.
- The Socket Abstraction.
- Specifying An Endpoint Address.
- A Generic Address Structure.
- Major System Calls used With Sockets.
- Utility Routines for Integer Conversion.
- Using Socket Calls in a Program.
- Symbolic Constants for Socket Call Parameters.
  
- Algorithms and Issues In Client Software Design.
  - Introduction.
  - Learning Algorithms instated of Details.
  - Client Architecture.
  - Identifying The Location of a Server.
  - Parsing an Address Argument.
  - Looking Up a Domain Name.
  - Looking Up a Well-known Port by Name.
  - Port Numbers and Network byte Order.
  - Looking Up a Protocol by Name.
  - The TCP Client Algorithm.
  - Allocating a Socket.
  - Choosing A Local Protocol Port Number.
  - A Fundamental Problem in Choosing A Local IP Address.
  - Connecting a TCP Socket to a Server.
  - Communicating with the Sever Using TCP.
  - Reading a Response from a TCP Connection.
  - Closing a TCP Connection.
  
- Example Client Software.
  - Introduction.
  - Importance of Small Examples.
  - Hiding Details.
  - An Example Procedure Library for Client Programs.
  - Implementation of Connect TCP.
  - Using the Example Library.
  - The DAYTIME Service.
  - Implementation of a TCP Client For DAYTIME.
  - Reading from TCP Connection.
  - The Echo Service.
  - A TCP Client for the ECHO Service.
  
- Algorithms and Issues in Server Software Design.
  - Introduction.
  - The Conceptual Server Algorithm.
  - Concurrent vs. Iterative Servers.
  - Connection-Oriented Vs. Connectionless Access.
  - Connection-Oriented Severs.
  - Connectionless Servers.



- Failure, Reliability and Statelessness.
  - Optimizing Stateless Servers.
  - Four Basic types of Servers.
  - Request Processing Time.
  - Iterative Servers Algorithms.
  - An Iterative, Connection Oriented Server Algorithms.
  - Binding to a well known Address Using INADDR\_ANY.
  - Placing the Socket in Passive Mode.
  - Accepting Connections and using them.
  - An Iterative, Connectionless Server Algorithm.
  - Forming A Reply Address in a Connectionless Server.
  - Concurrent Server Algorithms.
  - Master and Slave Processes.
  - A Concurrent Connectionless Server Algorithm.
  - A Concurrent Connection-Oriented Server Algorithm.
  - Using Separate as Slaves.
  - Apparent Concurrency Using A Single Process.
  - When to Use each Server type.
  - A Summary of Server types.
  - The Important Problem of Server Deadlock.
  - Alternative Implementations.
- Iterative, Connectionless Servers (UDP)
    - Introduction
    - Creating Passive Socket
    - Process Structure
    - An example of TIME Server
  - Iterative, Connection-Oriented Servers (TCP).
    - Introduction.
    - Allocating A Passive TCP Socket.
    - A Server for the DAYTIME Service.
    - Process Structure.
    - An Example DAYTIME Server.
    - Closing Connections.
    - Connection Termination and Server Vulnerability.
  - Concurrent, Connections- Oriented Servers (TCP)
    - Introduction.
    - Concurrent ECHO.
    - Iterative Vs. Concurrent Implementations.
    - Process Structure.
    - An Example Concurrent ECHO Server.
    - Cleaning of Errant Process.
  - Single Process, Concurrent Servers (TCP).
    - Introduction.
    - Data-driven Processing in a Server.





- Data-driven Processing with a Single Process.
- Process Structure of a Single-Process Server.
- An Example Single-Process ECHO Server.
- Remote Procedure Call Concept (RPC).
  - Introduction.
  - Remote Procedure Call Model.
  - Two Paradigms for Building Distributed Programs.
  - A Conceptual Model for Conventional Procedure Call & Return.
  - An Extension of the Procedural Model.
  - Execution of Conventional Procedure Call & Return.
  - The Procedural Model in Distributed Systems.
  - Analogy between Client-Server and RPC.
  - Distributed Computations as a Program.
  - Sun Microsystems's Remote Procedure Call Definition.
  - Remote Programs and Procedures.
  - Reducing the number of Arguments.
  - Identifying Remote Programs and Procedures.
  - Accommodating Multiple Versions of a Remote Program.
  - Mutual Exclusion for Procedures in a Remote Program.
  - Communications Semantics.
  - At least Once Semantics.
  - RPC Retransmission.
  - Mapping a Remote Program to a Protocol Port.
  - Dynamic Port Mapping.
  - RPC Port Mapper Algorithm.
  - Sun RPC Message Format.
  - Marshaling Arguments for a Remote Procedure.

## **PRACTICAL**

**2 CREDIT**

### **LAB – COMPUTER NETWORK – II**

#### **Text Book:**

**Section-I:** Data Communications and Networking – Behrouz Furouzan (3<sup>rd</sup> Edi) Pub. TMC

**Section-II:** Internetworking with TCP/IP Vol. III. - Douglas E. Comer - PHI.

#### **Reference Book :**

[1] Computer Network – Andrew S, Tenenbaum.

[2] Computer Network – Bhushan Trivedi.

[3] UNIX Network programming - W. Richard Stevens - PHI.

\*\*\*\*\* END MCA – 502\*\*\*\*\*



**Department of Computer Science**  
**Gujarat Vidyapith, Ahmedabad – 14**

**MCA - SEMESTER - V**

**MCA – 503: DATABASE ADMINISTRATION (ORACLE)**

**Prerequisite:**

- 1) Basic knowledge of Database
- 2) Relation database architecture
- 3) Should be aware of relational transactional and concurrency mechanism

**UNIT - I**

**1 CREDIT**

**The Oracle Instance and Database Architecture**

Defining the Instance, Creating the Instance, Understanding the Oracle Instance, Defining the Database, the SYS and SYSTEM Schemas, Understanding the Components of the Database, Understanding Database Segments, Other Database Objects, Creating the Oracle Environment, Designing an Optimal Flexible Architecture, Creating Your First Database

**UNIT - II**

**0.5 CREDIT**

**SQL Plus and PL/SQL Administration**

Administering SQL Plus, Using the SQL Plus COPY Command, Using SQL to Create SQL, Tracing SQL Statements, Oracle Enterprise Manager, Understanding the Enterprise Manager Architecture, Using the Database Administration Tools, Understanding PL/SQL, Understanding the PL/SQL Engine, Adding PL/SQL to Your Toolbox, PL/SQL Language Tutorial, Stored subprograms and packages, Defining Stored Subprograms, Building and Using Stored Programs, Debugging with SHOW ERRORS, Building and Using Packages, Creating a Real-World Example

**UNIT - III**

**1 CREDIT**

**Import-Export & SQL\* Loader, Managing Storage and Lock**

IMPORT/EXPORT, SQL\*Loader, SQL Loader Examples, Conventional and Direct Path Loading, Managing Database Storage, Administering Database Objects, Understanding Database Fragmentation, Managing Rollback Segments, Identifying Storage Problems, Administering Growing Database, Integrity Management  
Locking - Implementing Locks, Analyzing v\$lock, Monitoring Locks on the System, Avoiding Locks: Possible Solutions, Implementing Locks with Latches,



## **UNIT - IV**

**1 CREDIT**

### **Performance tuning fundamentals**

Understanding Why You Tune, Knowing the Tuning Principles, Tuning Goals, Using the Return on Investment Strategy, Revisiting Application Types, Using Oracle Diagnostic Tools

Application Tuning - Understanding the Optimizer, SQL Trace and tkprof, Understanding EXPLAIN PLAN

Tuning Memory - UTLBSTAT/UTLESTAT, Tuning the Shared Pool, Tuning the Database Buffer Cache, Tuning the multithreaded Server (MTS), Tuning Locks, Tuning I/O - Tuning Table-spaces and Data-files, Tuning Blocks and Extents, Tuning Rollback Segments, Tuning Redo Logs, Oracle8 New I/O Features

## **UNIT - V**

**0.5 CREDIT**

### **Supplied Oracle Database package**

About the Supplied Oracle Database Package , Describing Supplied Packages, hands-on with the Oracle-Supplied Packages, DBMS\_APPLICATION\_INFO, DBMS\_DDL, DBMS\_OUTPUT, DBMS\_SESSION, DBMS\_SHARED\_POOL, DBMS\_SPACE, DBMS\_SQL, DBMS\_JOB

## **PRACTICAL**

**2 CREDIT**

### **LAB – DATABASE ADMINISTRATION (ORACLE)**

#### **Book:**

Using Oracle-8 by William Page & Nathan Hughes Publishers – PHI

#### **Reference Book:**

Oracle Unleashed, Oracle Complete Reference, Oracle Bible, Oracle DBA

\*\*\*\*\* END MCA – 503 \*\*\*\*\*



**Department of Computer Science**  
**Gujarat Vidyapeeth, Ahmedabad – 14**

**MCA - SEMESTER - V**

**MCA-504: SOFTWARE ENGINEERING**

**UNIT - I**

**0.25 CREDIT**

**SOFTWARE ENGINEERING AND MODELLING**

THE NATURE OF SOFTWARE, UNIQUE NATURE OF WEBAPPS, THE SOFTWARE ENGINEERING, THE SOFTWARE PROCESS, THE SOFTWARE ENGINEERING PRACTICE

- AGILE DEVELOPMENT
  - WHAT IS AGILITY?
  - AGILITY AND COST OF CHANGE
  - WHAT IS AN AGILE PROCESS?
    - Agility principles
    - The Politics of Agile Development
    - Human factors
  
- MODELING
  - REQUIREMENTS MODELING FOR WEBAPPS
    - How Much Analysis is Enough?
    - Requirements Modeling Input and output
    - Different Models for WebApps
    - Navigation Modeling

**UNIT - II**

**1 CREDIT**

**DESIGN CONCEPTS**

- DESIGN WITHIN THE CONTEXT OF SOFTWARE ENGINEERING
- THE DESIGN PROCESS
  - Software Quality guideline and Attributes
  - The Evolution of Software Design
  
- DESIGN CONCEPTS
  - Abstraction, Architecture, Patterns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring, Object Oriented Design Concepts, Design Classes



- THE DESIGN MODEL
  - Data, Architectural, Interface, Component-level, Deployment-level Design Elements

## **ARCHITECTURAL DESIGN**

- SOFTWARE ARCHITECTURE
  - Architectural Descriptions and Decisions
- ARCHITECTURAL GENRES
- ARCHITECTURAL STYLES
  - Different Architectural Styles
  - Architectural Patterns
- Architectural Design
  - Representing The System in Context
  - Architectural Patterns
  - Refining the Architecture into Components
  - Describing Instantiations of the System
- ASSESSING ALTERNATIVE ARCHITECTURAL DESIGNS
  - An Architecture Trade-Off Analysis method
  - Architectural Complexity
  - Architectural Description Languages
- ARCHITECTURAL MAPPING USING DATAFLOW
  - Transform Mapping

## **COMPONENT - LEVEL DESIGN**

- WHAT IS A COMPONENT?
  - An Object-Oriented View, The Traditional View, A Process-Related View
- DESIGNING CLASS-BASED COMPONENTS
  - Basic Design Principles
  - Component-Level Design guidelines, Cohesion, Coupling
- CONDUCTING COMPONENT-LEVEL DESIGN
- COMPONENT-LEVEL DESIGN FOR WEBAPPS
  - Content and Functional Design at Component Level
- DESIGNING TRADITIONAL COMPONENTS
  - Different Design Notation
- COMPONENT-BASED DEVELOPMENT
  - Domain Engineering
  - Component Qualification, Adaptation and Composition
  - Analysis and Design for Reuse
  - Classifying and Retrieving Components

## **USER INTERFACE DESIGN**

- THE GOLDEN RULES OF DESIGN



- USER INTERFACE ANALYSIS & DESIGN
- INTERFACE ANALYSIS
  - User Analysis
  - Task Analysis and Modeling
  - Analysis of Display Content
  - Analysis of the Work Environment
- INTERFACE DESIGN STEPS
  - Design Patterns and Design Issues
- WEBAPPS INTERFACE DESIGN
  - Interface Design Principles and Guidelines
  - Interface Design Workflow for WebApps
- DESIGN EVALUATION

### **WEBAPP DESIGN**

- WEBAPP DESIGN QUALITY AND GOALS
- A DESIGN PYRAMID FOR WEBAPPS
- WEBAPPS INTERFACE DESIGN
- AESTHETIC DESIGN
  - Layout Issues
  - Graphic Design Issues
- CONTENT DESIGN
  - Content Objects
  - Content Design Issues
- ARCHITECTURAL DESIGN
  - Content Architecture
  - WebApp Architecture
- NAVIGATION DESIGN
  - Navigation Semantics
  - Navigation Syntax
- COMPONENT-LEVEL DESIGN
- OBJECT ORIENTED HYPERMEDIA DESIGN MODEL (OOHDM)
  - Conceptual Design for OOHDM
  - Navigational Design for OOHDM
  - Abstract Interface Design and Implementation

### **UNIT - III**

**1.5 CREDIT**

### **QUALITY MANAGEMENT**

- WHAT IS QUALITY
- SOFTWARE QUALITY
  - Gravin's Quality Dimensions
  - McCall's Quality Factors, ISO 9126 Quality Factors, Targeted Quality Factors
  - The Transition to a Quantitative View



- THE SOFTWARE QUALITY DILEMMA
  - Cost of Quality
  - Risks
  - Negligence and Liability
  - Quality and Security
  - The Impact of Management Actions
- ACHIEVING SOFTWARE QUALITY
  - Software Engineering Methods
  - Project management Techniques
  - Quality Control
  - Quality Assurance

### **REVIEW TECHNIQUES**

- COST IMPACT OF SOFTWARE DEFECTS
- DEFECT AMPLIFICATION AND REMOVAL
- REVIEW METRICS AND THEIR USE
  - Analyzing Metrics
  - Cost Effectiveness of Reviews
- REVIEWS: A FORMALITY SPECTRUM
- INFORMAL REVIEWS
- FORMAL TECHNICAL REVIEWS

### **SOFTWARE QUALITY ASSURANCE**

- BACKGROUND ISSUES
- ELEMENTS OF SOFTWARE QUALITY ASSURANCE
- SQA TASKS, GOALS AND METRICS
  - SQA Tasks
  - Goals, Attributes and Metrics
- FORMAL APPROACHES TO SQA
- STATISTICAL SOFTWARE QUALITY ASSURANCE
  - A Generic Example
  - Six Sigma for Software Engineering
- SOFTWARE RELIABILITY
  - Measures of Reliability and Availability
  - Software Safety
- THE ISO 9000 QUALITY STANDARDS
- THE SQA PLAN

### **SOFTWARE TESTING STRATEGIES**

- A STRATEGIC APPROACH TO SOFTWARE TESTING
  - Verification and Validation
  - Organizing for Software Testing
  - Criteria for Completion of Testing



- STRATEGIC ISSUES
- TEST STRATEGIES FOR CONVENTIONAL SOFTWARE
  - Unit Testing
  - Integration Testing
- TEST STRATEGIES FOR OBJECT-ORIENTED SOFTWARE
  - Unit Testing in OO Context
  - Integration Testing in OO Context
- TEST STRATEGIES FOR WEBAPPS
- VALIDATION TESTING
- SYSTEM TESTING
  - Recovery Testing
  - Security Testing
  - Stress Testing
  - Performance Testing
  - Deployment Testing
- THE ART OF DEBUGGING

### **TESTING CONVENTIONAL APPLICATIONS**

- SOFTWARE TESTING FUNDAMENTALS
- INTERNAL AND EXTERNAL VIEWS OF TESTING
- WHITE-BOX TESTING
  - Basis Path Testing
  - Control Structure Testing
- BLACK-BOX TESTING
  - Graph-Based Testing Methods
  - Equivalence Partitioning
  - Boundary Value Analysis
  - Orthogonal Array Testing
- MODEL-BASED TESTING
- TESTING FOR SPECIALIZED ENVIRONMENTS, ARCHI & APPLI
  - Testing GUIs
  - Testing of Client/Server Architectures
  - Testing Documentation and Help Facilities
  - Testing for Real-time Systems

### **TESTING OBJECT-ORIENTED APPLICATIONS**

- BROADENING THE VIEW OF TESTING
- TESTING OOA AND OOD MODELS
  - Correctness of OOA and OOD Models
  - Consistency of Object-Oriented Models
- OBJECT-ORIENTED TESTING STRATEGIES
  - Unit Testing in the OO Context
  - Integration Testing in the OO Context
  - Validation Testing in an OO Context





- OBJECT-ORIENTED TESTING METHODS
  - The Test Case Design Implications of OO Concepts
  - Applicability of Conventional Test Case Design Methods
  - Fault-Based Testing
  - Test Cases & Class Hierarchy
  - Scenario-Based Testing
  - Testing Surface Structure & Deep Structure
- TESTING METHODS APPLICABLE AT THE CLASS LEVEL
  - Random Testing for OO Classes
  - Partition Testing at the Class level
- INTERCLASS TEST CASE DESIGN
  - Multiple Class testing
  - Tests Derived from Behavior Models

## **TESTING WEB APPLICATIONS**

- TESTING CONCEPTS FOR WEBAPPS
  - Dimensions of Quality
  - Errors within a WebApps Environment
  - Testing Strategy and Test Planning
- THE TESTING PROCESS - AN OVERVIEW
- CONTENT TESTING
  - Content Testing
  - Database Testing
- USER INTERFACE TESTING
- COMPONENT-LEVEL TESTING
- NAVIGATION TESTING
  - Testing Navigation Syntax and Semantics
- CONFIGURATION TESTING
  - Server-side Issues
  - Client-side Issues
- SECURITY TESTING
- PERFORMANCE TESTING
  - Performance Testing Objectives
  - Load Testing
  - Stress Testing

## **SOFTWARE CONFIGURATION MANAGEMENT**

- SOFTWARE CONFIGURATION MANAGEMENT
  - AN SCM Scenario
  - Elements of a Configuration Management System
  - Baselines
  - Software Configuration Items
- CONFIGURATION MANAGEMENT FOR WEBAPPS
  - Dominant Issues



- WebApps Configuration Objects
- Content Management
- Change Management
- Version Control
- Auditing and Reporting

## **PRODUCT METRICS**

- A FRAMEWOTRK FOR PRODUCT METRICS
  - Measures, Metrics & Indicators
  - The Attributes of Effective Software Metrics
- METRICS FOR REQUIREMENT MODEL
  - Function-Based Metrics
  - Metrics for Specification Quality
- DESIGN METRICS FOR WEBAPPS

## **UNIT - IV**

**1.25 CREDIT**

## **MANAGING SOFTWARE PROJECTS**

### **PROJECT MANAGEMENT CONCEPTS**

- THE MANAGEMENT SPECTRUM
- The People, Product, ProcessThe Project PEOPLE
  - The Stakeholders, Team Leaders
  - The Software Team
  - Agile Teams
  - Coordination and Communication Issues
- THE PRODUCT
  - Software Scope
  - Problem Decomposition
- THE PROCESS
  - Melding the Product and the process
  - Process Decomposition
- THE PROJECT
- THE W<sup>5</sup>HH PRINCIPLE

### **SOFTWARE PROJECT MANAGEMENT TOOL**

- MS-PROJECT

### **PROCESS AND PROJECTS METRICS**

- METRICS IN THE PROCESS AND PROJECT DOMAINS
  - Process Metrics and Software Process Improvement
  - Project Metrics
- SOFTWARE MEASURMENT



- Size-Oriented Metrics
- Function-Oriented Metrics
- Reconciling LOC & FP Metrics
- Object-oriented Metrics
- Use-Case Oriented metrics
- WebApp Project Metrics
- METRICS FOR SOFTWARE QUALITY
  - Measuring Quality
  - Defect Removal Efficiency
- INTEGRATING METRICS WITHIN THE SOFTWARE PROCESS
  - Establishing A Baseline

### **ESTIMATION FOR SOFTWARE PROJECTS**

- THE PROJECT PLANNING PROCESS
- SOFTWARE SCOPE & FEASIBILITY
- RESOURCES
  - Human Resources
  - Reusable Software Resources
  - Environmental Resources
- SOFTWARE PROJECT ESTIMATION
- DECOMPOSITION TECHNIQUES
  - Software Sizing
  - Problem-Based Estimation
  - Process-Based Estimation
  - Estimation with Use Cases
  - Reconciling Estimates
- EMPIRICAL ESTIMATION MODELS
  - The Structure of Estimation Models
  - The COCOMO II Model
  - The Software Equation
- ESTIMATION FOR OBJECT-ORIENTED PROJECTS
- SPECIALIZED ESTIMATION TECHNIQUES
  - Estimation for Agile Development
  - Estimation for WebApps Projects
- THE MAKE-BUY DECISION
  - Creating a Decision Tree
  - Outsourcing

### **RISK MANAGEMENT**

- REACTIVE VS. PROACTIVE RISK STRATEGIES
- SOFTWARE RISKS
- RISK IDENTIFICATION
  - Assessing Overall Project Risk
  - Risk Components and Drivers



- RISK PROJECTION
  - Developing a Risk Table
  - Assessing Risk Impact
- RISK REFINEMENT
- RISK MITIGATION MONITORING AND MANAGEMENT
- THE RMMM PLAN

**Text Books:**

Software Engineering – A Practitioner’s Approach  
Publication. McGraw-Hill International Edition  
Author. Roger S. Pressman (Seventh Edition)

**Reference Books:**

- Software Engineering - Publication. Printice\_Hall India  
Author. Ian Summarville
- Software Engineering - Author. Pankaj Jalote

\*\*\*\*\* END MCA – 504 \*\*\*\*\*

**Department of Computer Science**  
**Gujarat Vidyapith, Ahmedabad – 14**

**MCA SEMESTER - V**

**MCA-505: ENTERPRISE RESOURCE PLANNING AND E-COMMERCE**

**Objective:** To Introduce the Use and Implementation of ERP and E-Commerce

**UNIT - I**

**1 CREDIT**

- ENTERPRISE RESOURCE PLANNING
  - Introduction
  - ERP Structure
  - Business Engineering
  - Process Reengineering
  - Business Modeling for ERP
  - ERP Implementation
  - Implementation Methodology
  - ERP and the Competitive Advantage

**UNIT - II**

**1 CREDIT**



- ERP DOMAIN STUDY
  - Overview
  - ERP Product Study
  - IFS
  - Baan
  - SAP
- ERP IMPLEMENTATION CASE STUDY
  - Overview
  - ERP Implementation Case Study of different companies

### **UNIT - III**

**1 CREDIT**

- ELECTRONIC COMMERCE
  - Introduction
- Evolution of Business
  - Business and its Transactions through Internet
  - Types of Business done through Electronic Media
  - B2B, B2C, C2C, B2G, C2G
- Methods for Safe and Reliable E Business
  - Cryptography
  - Private Key Cryptography
  - Public Key Cryptography
  - Digital Signature
  - Message Digest
- Protocol
  - SSL, SHTTP, SET

### **UNIT - IV**

**1 CREDIT**

- DIFFERENT PAYMENT METHODS FOR E COMMERCE
- Electronic Payment Methods
- Third Party Payment Method
  - Cyber Cash
  - Virtual Internet Payment System
  - Payment through Credit Card
- Luhn algorithm for Credit Card Validation
- Digital Currency and its Introduction
  - DigiCash
- Micro-Payment Methods

#### **Text Book:**

- Enterprise Resource Planning - Concepts and Practice  
Author: Vinod Kumar Garg and N.K. Venkitakrishnan,  
Publication PHI

#### **Reference Books:**

- PUBLIC-KEY CRYPTOGRAPHY Theory and Practice



Author: Abhijit Das and C.E. Veni Madhavan  
Publication: Pearson Education

- Cryptography and Network Security Principles and Practice  
Author: William Stallings  
Publication: Pearson Education
- Electronic Commerce A Managerial Perspective  
Author: Efraim Turban, Jae Lee, David King & H.Michael Chung  
Publication: Pearson Education
- Electronic Commerce: Security, Risk Management & control  
Author: Greenstein and Feinman  
Publication: Tata McGrawHill

\*\*\*\*\* END OF MCA – 505 \*\*\*\*\*



**Department of Computer Science**  
**Gujarat Vidyapith, Ahmedabad – 14**

**MCA SEMESTER - V**

**MCA-506: OPEN ERP (OPTIONAL)**

**OPEN-ERP**

**3 CREDIT (Extra)**

**Functional and Technical Training\***

100% Internal evaluation (100% આંતરિક મૂલ્યાંકન)

**Books**

\*\*\*\*\***End of MCA-506**\*\*\*\*\*

---

**GRAMJIVAM PADYATRA**

**2 CREDIT**

**COMMUNITY LIVING (SAMUHJIVAN)**

**GRADE**