



SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Sree Sainath Nagar, A. Rangampet – 517 102

M.Tech (Software Engineering)

**Course Structure and Syllabus
(w.e.f 2010-2011)**

DEPARTMENT OF INFORMATION TECHNOLOGY



SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)
Sree Sainath Nagar, A. Rangampet – 517 102

Department of Information Technology

M.Tech (Software Engineering) (w.e.f. 2010-2011)

M.Tech - I Semester

Subject Code	Course	Periods per		Credits	Max. Marks		
		week			Internal	External	Total
		L	P				
10MT10501	Advanced Data Structures and Algorithms	4	-	4	40	60	100
10MT10502	Advanced Software Engineering	4	-	4	40	60	100
10MT10503	Middleware Technologies	4	-	4	40	60	100
10MT10504	Software Metrics	4	-	4	40	60	100
10MT10505	Software Requirements and Estimation	4	-	4	40	60	100
Elective-I							
10MT10506	Software Process Management	4	-	4	40	60	100
10MT10507	Software Project Management						
10MT10508	Virtualization and Cloud Computing						
10MT10509	Advanced Data Structures and Algorithms and Middleware Technologies	-	3	2	25	50	75
TOTAL		24	3	26	265	410	675

Chairman, BOS (I.T)

M.Tech (Software Engineering) (w.e.f. 2010-2011)
II-Semester

Subject Code	Course	Periods per week		Credits	Max. Marks		
		L	P		Internal	External	Total
10MT20501	Service Oriented Architecture	4	-	4	40	60	100
10MT20502	Software Architecture and Design Patterns	4	-	4	40	60	100
10MT20503	Software Reengineering & Reuse	4	-	4	40	60	100
10MT20504	Software Security Engineering	4	-	4	40	60	100
10MT20505	Software Quality Assurance and Testing	4	-	4	40	60	100
Elective-II							
10MT20506	Information Retrieval Systems	4	-	4	40	60	100
10MT20507	Soft Computing						
10MT20508	Software Reliability						
10MT20509	Industrial Visit/Seminar	-	-	2	50	-	50
10MT20510	Software Testing and Design Patterns Lab	-	3	2	25	50	75
TOTAL		24	3	28	315	410	725

M.Tech (Software Engineering) (w.e.f. 2010-2011)
III-Semester

Course	Periods per week		Credits	Max. Marks		
	L	P		Internal	External	Total
Project Phase-I	-	-	4	50	-	50

M.Tech (Software Engineering) (w.e.f. 2010-2011)
IV-Semester

Course	Periods per week		Credits	Max. Marks		
	L	P		Internal	External	Total
Project Phase-II	-	-	12	50	100	150

Chairman, BOS (I.T)

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10501: ADVANCED DATA STRUCTURES AND ALGORITHMS

UNIT I: Basic Data Structures

Review of Arrays, Stacks, Queues, Circular Queues, Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists, Linked stacks and Linked queues, Terminologies and Applications

UNIT II: Algorithm Analysis

Efficiency of algorithms, Apriori Analysis, Asymptotic Notations, Time complexity of an algorithm using O notation, Polynomial Vs Exponential Algorithms, Average, Best, and Worst Case Complexities, Analyzing Recursive Programs.

UNIT III: Trees and Graphs

Introduction, Definition and Basic terminologies of trees and binary trees, Representation of trees and Binary trees, Binary tree Traversals, Threaded binary trees, Graphs-basic concepts, representation and applications, DFS and BFS Traversals.

UNIT IV: Binary Search Trees, AVL Trees and Red – Black Trees

Introduction, Binary Search Trees: Definition, Operations and applications. AVL Trees: Definition, Operations and applications. **Red – Black Trees**: Definition, Operations and applications.

UNIT V: B- Trees, Splay Trees and Hash Tables

B Trees, Splay Trees and its applications. Hash Tables: Introduction, Hash Tables, Hash Functions and its applications.

UNIT VI: Divide – and – Conquer & Greedy Method

General Method, Binary Search, Finding Maximum and Minimum, Quick Sort, Merge sort, Strassen's Matrix Multiplication, Greedy Method- General Method, Minimum Cost Spanning Trees, Single Source Shortest Path.

UNIT VII: Dynamic Programming

General Method, All Pairs Shortest Path, Single Source Shortest Path, 0 / 1 Knapsack problem, Reliability Design, Traveling Sales Person's Problem.

UNIT VIII: Back Tracking and Branch – and – Bound

General Method, 8 – Queen's Problem, Graph Coloring. Branch – and – Bound: The Method, LC Search, Control Abstraction, Bounding, 0 / 1 Knapsack Problem.

TEXT BOOKS:

1. Data Structures and Algorithms by G.A.V. Pai, 2009, TMH.
2. Data structures, Algorithms and Applications in C++, S.Sahni, and University press (India) pvt ltd, 2nd edition, Orient Longman pvt.ltd.

REFERENCE BOOKS:

1. Design and Analysis of Algorithms by E. Horowitz, S. Sahani, 3rd Edition, Galgotia.
2. Data Structures and Algorithms in C++ by Drozdek 2nd Edition, Thomson.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

M.Tech (Software Engineering) (w.e.f 2010-11)

I Year (I-Semester)

10MT10502: ADVANCED SOFTWARE ENGINEERING

UNIT-I

Software and Software Engineering

The Nature of Software, Software characteristics, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths.

Software Paradigms

Prespective Process Models, Specialized Process Models.

UNIT- II

Unified Process Model

The Unified Process, Personal and Team Process Models, Process Technology, Product and Process.

Agile Development

What is Agility? Agility and the Cost of Change, What is an Agile Process? Extreme Programming (XP), Other Agile Process Models, A Tool set for the Agile Process.

UNIT-III

Critical Systems

A simple safety-critical system, System dependability, Availability and reliability, Safety, Security.

Critical systems specification

Risk-driven specification, Safety specification, Security specification Software reliability specification.

Formal Specification

Formal specification in the software process, Sub-system interface specification, Behavioural specification.

UNIT- IV

Software Reuse

The reuse landscape, Design patterns, Generator-based reuse, Application frameworks, Application system reuse.

Component-based Software Engineering

Components and component models, The CBSE process, Component composition.

UNIT-V

Software Testing

System testing, Component testing, Test case design, Test automation.

Software Evolution

Program evolution dynamics, Software maintenance, Evolution Processes, Legacy system evolution.

UNIT-VI

Aspect oriented software engineering: The separation of concerns, Aspects, Join points and pointcuts, Software engineering with aspect, Using AOSD to streamline complex systems development without sacrificing flexibility or scalability.

Service oriented software engineering: Service-based concepts, modeling and documentation, Service discovery and composition, Service-oriented architecture, Services as reusable components, Software development with services.

UNIT-VII

Quality Management

Process and product quality, Quality assurance and standards, Quality Planning, Quality control, Software Measurement and metrics.

Process Improvement

Process and product quality, process classification, Process measurement, Process analysis and modeling, Process change, The CMMI process improvement framework.

UNIT-VIII

User Interface Design

The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebAPP Interface Design, Design Evaluation.

TEXTBOOKS:

1. Software Engineering, by Ian Sommerville, Addison-Wesley, 8th Edition, 2006.
2. Software Engineering, A Practitioner's Approach, by Roger S. Pressman, 7th Edition, 2009.

REFERENCE BOOKS:

1. Using UML: Software Engineering with Objects and Components, by Perdita Stevens, Rob Pooley, Addison-Wesley, 2nd edition, 2006.
2. The Mythical Man-Month : Essays on Software Engineering, by Frederick P., Jr. Brooks, Frederick P. Brooks Jr, Addison-Wesley, 1995.
3. The Future of Software Engineering, edited by Anthony Finkelstein, ACM Press, 2000.
4. Aspect-Oriented Software Development, by Robert E. Filman, Tzilla Elrad, Siobh n Clarke, Mehmet Aksit, Addison-Wesley, 1st edition, 2004.
5. Service-Oriented Software System Engineering : Challenges and Practices, by Zoran Stojanovic, Ajantha Dahanayake, IGI Global, 2005.
6. Software Reuse, by I. Jacobson, M. Griss, and P. Jonsson, ACM Press, 1997

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10503: MIDDLEWARE TECHNOLOGIES

UNIT-I:

CLIENT/SERVER COMPUTING

Building blocks-types of servers-types of Clients-types of middleware-aspects of client/server systems-sizing-scalability-tiered architecture-client/server models-requirements of client/server systems-Distributed objects-benefits-drawbacks-from distributed objects to components

UNIT-II:

COMPONENT TECHNOLOGY

Components- definitions-properties-benefits-components and interfaces- direct and indirect interfaces- versions-interfaces as contracts- callbacks- forms of design levels reuse- connection oriented programming – connectable objects.

UNIT-III:

COMPONENT ARCHITECTURE

Component architecture- component frameworks- composition- data driven, contextual, aspect oriented programming, subject oriented programming, XML components-component development-assembly.

UNIT-IV:

THE MICROSOFT WAY

Component object model- from COM, COM+, DCOM to .NET framework-evolution- web services technologies- XML, WSDL, UDDI, SOAP.

UNIT-V :

COMMON LANGUAGE INFRASTRUCTURE

Common language infrastructure- common language Runtime - .NET framework class library- An Overview of the library, Fundamental Namespaces

UNIT- VI:

ADO.NET- Using .NET Framework Data Providers, Using Connection and Command Objects, Accessing Data with Data Readers, Accessing Data with Data Sets, ASP.NET- enterprise services.

UNIT-VII:

THE SUN WAY

Component variety- applets, servlets, java beans, enterprise beans- EJB architecture- types of beans-characteristics- Building and deploying distributed applications using EJB-java and web services- JXTA and jinni.

UNIT-VIII:

THE OMG WAY

System object model- CORBA timeline- CORBA architecture- ORB- services- facilities-business objects-IIop- transport mechanisms- IDL- Drawbacks of CORBA.

CORBA Component model- POA- CCM components- CCM containers- Meta Object Facility. Comparison of CORBA, .NET and EJB.

TEXT BOOKS:

1. Clemens szyperski, Dominik Gruntz and Stephan Murer, “Component Software Beyond object oriented Programming” third edition, Pearson education, 2004.
2. David Chappell, “Understanding .NET”, Pearson education Inc, 2002.

REFERENCES:

1. Robert Orfali, Dan Harkey, Jeri Edwards, “Client/Server Survival Guide” third edition, John Wiley Inc, 2003.
2. Bill Burke, Richard Monson-Haefel, “Enterprise JavaBeans”, Fifth Edition, O’Reily, 2001.
3. Dan harkey, Robertt Orfali, “ Client/Server programming with JAVA and CORBA”, second edition, wiley& sons Inc, 1999.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10504: SOFTWARE METRICS

UNIT-I: Measurement

Measurement in Everyday Life, Measurement in Software Engineering, Scope of Software Metrics

UNIT-II: Basics Of Measurement

Representational Theory of Measurement, Measurement and Models, Measurement Scales and Scale Types.

UNIT-III: Goal-Based Frame Work for Software Measurement

Classifying Software Measures, Applying Frame Work, Software Measurement Validation

UNIT-IV: Software Metrics Data Collection

Good Data, Definition of Data, Collecting, Storing and Extracting Data.

UNIT-V: Measuring Internal Product Attributes- Size:

Aspects of software size, Length, Reuse, Functionality, Complexity

UNIT-VI: Measuring Internal Product Attributes- Structure:

Types of structural measures, Control-flow structure, Modularity and information flow attributes, Object oriented metrics.

UNIT –VII: Measuring External Product Attributes

Modeling Software Quality, Measuring Aspects of Quality.

UNIT-VIII: Measurement and Management

Planning a Measurement Program: What is metrics plan, why and what, where and when, How, and who.

Measurement in Practice: Success criteria, Measurement in the small, Measurement in the large.

TEXT BOOKS:

1. Fenton, Pfleeger, “Software Metrics: A Rigorous and Practical Approach”, Thomson Computer Press, 1997

REFERENCE BOOKS:

1. Stephen H. Kan: “Metrics & Models in Software Quality Engineering”, Pearson Education, 2 ed, 2003
2. Linda M. Laird and Carol Brennan: “Software Measurement and Estimation - A Practical Approach”, IEEE Computer Science Press and Wiley Inter science, 2006
3. C Ravindranath Pandian: “Software Metrics: A guide to Planning Analysis, and Implementation”, Auerbach Publications, Florida, USA, 2005

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10505: SOFTWARE REQUIREMENTS AND ESTIMATION

UNIT I:

Software Requirements: What And Why

Essential Software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management.

UNIT II:

Software Requirements Engineering

Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality.

UNIT-III:

Software Requirements Management

Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain.

UNIT IV:

Software Estimation

Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation.

UNIT-V:

Size Estimation-Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, Conversion between size measures.

UNIT VI:

Effort, Schedule And Cost Estimation

What is Productivity? Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Putnam Estimation Model, Algorithmic models, Cost Estimation.

Unit VII:

Requirements Management Tools:

Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation.

UNIT-VIII:

Software Estimation Tools:

Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools

TEXT BOOKS:

1. Software Requirements by Karl E. Weigers, Microsoft Press.
2. Software Requirements and Estimation by *Rajesh Naik and Swapna Kishore*, Tata Mc Graw Hill

REFERENCES:

1. Managing Software Requirements, Dean Leffingwell & Don Widrig, Pearson Education, 2003.
2. Mastering the requirements process, second edition, Suzanne Robertson & James Robertson, Pearson Education, 2006.
3. Estimating Software Costs, Second edition, Capers Jones, Tata McGraw-Hill, 2007.
4. Practical Software Estimation, M.A. Parthasarathy, Pearson Education, 2007.
5. Measuring the software process, William A. Florac & Anita D. Carleton, Pearson Education, 1999.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10506: SOFTWARE PROCESS MANAGEMENT

UNIT-I: Software Process Maturity

Software Process Maturity Framework: Software Process Improvement, Process Maturity Levels, People in the optimizing process, the need for the Optimizing Process

The principles of Software Process Change: Process in Perspective, Principles, Misconceptions about software Process, A strategy for implementing software Process Change

Software Process Assessment: Overview, Phases, Principles, Assessment Process and Conduct, Implementation Considerations

UNIT-II: Repeatable Process

Managing Software Organizations: Commitment Discipline, the Management System, Establishing a Project Management System

The Project Plan: Principles, Contents, Size Measures, Estimating, Productivity Factors, Scheduling, Project Tracking, the Development Plan, Planning Models, Final Considerations

UNIT-III: Software Configuration Management-I & Software Quality Assurance

SCM: The need for configuration Management, Software Product Nomenclature, Basic SCM Functions, Baselines, Configuration Management Responsibilities, The need for Automated Tools

SQA: Quality Management, the Role of SQA, Launching the SQA Program, the SQA Plan, SQA Considerations, SQA People, Independent Verification and Validation

UNIT-IV: Software Standards and Software Inspections

Software Standards: Definitions, Reasons for Standards , benefits of Standards, Examples of Major Standards, Establishing Software Standards, Standards Vs Guidelines.

Software Inspections: Types of Reviews, Objectives, Principles, Conduct of Inspections, Training, Reports and Tracking, Other Considerations, Initiating an Inspection Program, Future directions

UNIT-V: Software Testing and SCM Design-II

Software Testing: Definitions, Software Testing Principles, Types of software Tests, Test Planning, Test Development, Test Execution and Reporting, Test Tools and Methods, Real-Time Testing, The Test Organization

SCM Design Phase: Plan, Questions, Support Functions, Requirement Phase, Design Control, The Implementation Phase, Operational data, The Test Phase, SCM for Tools, Configuration Accounting, Software Configuration Audit

UNIT-VI: Defining the Software Process

Software Process: Standards, Definitions, Levels of software Process Models, Prespective and Descriptive Uses of Models, A Software Process Architecture, Critical Software Process issues, Detailed Process Models, Entity Process Models, Process Model Views, Process Guidelines, Establishing and Using a Process Definition

Software Engineering Process Group: Changing the Software Process, The Role of SEPG, Establishing Standards, the Process Database, Technology Insertion Focal Point, Education and Training, Process Consultation, Process Status and Assessment, Establishing the SEPG.

UNIT-VII: Managed Process

Data Gathering and Analysis: Principles, Process, Software Measures, Data Analysis & considerations

Managing Software Quality: Paradigm, Quality Examples and Motivation, Measurement Criteria, Establishing a Software Quality Program, Estimation, Removal Efficiency, Quality Goals and Plans, Tracking and Controlling Software Quality.

UNIT-VIII: Optimizing Process

Defect Prevention: Introduction, Principles, Process Changes for Defect Prevention, Considerations, Management's Role.

Automating the Software Process: Need, What to Automate, Development Environments, Organizational Plans to Automate, Technology Transition, Special Considerations, Productivity, Justification Considerations

TEXT BOOKS:

1. Watt S. Humphrey, "Managing the Software Process", 2003, PEA.

REFERENCE BOOKS:

1. Watts S. Humphrey, "A Discipline for Software Engineering", 2002, PEA
2. Roger S. Pressman, "Software Engineering, A Practitioners Approach", 2003, 5e, TMH.
3. W. S. Humphrey, "Introduction to the Team Software Process", 2002, PEA.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10507: SOFTWARE PROJECT MANAGEMENT

UNIT - I

Conventional Software Management : The waterfall model, conventional software Management performance.

Evolution of Software Economics : Software Economics, pragmatic software cost estimation.

UNIT - II

Improving Software Economics : Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.

The old way and the new : The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

UNIT - III

Life cycle phases : Engineering and production stages, inception, Elaboration, construction, transition phases.

Artifacts of the process : The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

UNIT - IV

Model based software architectures : A Management perspective and technical perspective.

Work Flows of the process : Software process workflows, Iteration workflows,

UNIT - V

Checkpoints of the process : Major mile stones, Minor Milestones, Periodic status assessments.

Iterative Process Planning : Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

UNIT - VI

Project Organizations and Responsibilities : Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation : Automation Building blocks, The Project Environment.

UNIT - VII

Project Control and Process instrumentation : The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.

Tailoring the Process : Process discriminants.

UNIT - VIII

Future Software Project Management : Modern Project Profiles, Next generation Software economics, modern process transitions.

Case Study: The command Center Processing and Display system- Replacement (CCPDS-R)

TEXT BOOK :

1. Software Project Management, Walker Royce: Pearson Education, 2005.

REFERENCES :

1. Software Project Management, Bob Hughes and Mike Cotterell: Tata McGraw-Hill Edition.
2. Software Project Management, Joel Henry, Pearson Education.
3. Software Project Management in practice, Pankaj Jalote, Pearson Education.2005.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

M.Tech (Software Engineering) (w.e.f 2010-11)

I Year (I-Semester)

10MT10508: VIRTUALIZATION AND CLOUD COMPUTING

UNIT- I

Introduction to virtualization - Objectives of virtualization, history of virtualization, benefits of virtualized technology, the virtual service desk, what can be virtualized, related forms of computing, cloud computing, software as a service – saas, grid computing, utility computing, virtualization processes, introduction review questions.

UNIT- II

Virtualization Technologies - ubuntu (server edition), altiris, windows, serve, software virtualization, vmware, intel virtualization, red hat virtualization, softgrid application, linux virtualization, desktop, virtualization, hardware virtualization, resource virtualization, processor virtualization, application virtualization,

UNIT- III

Virtualization Technologies - Storage virtualization, virtualization density, para-virtualization, os virtualization, virtualization software, data storage virtualization, intel virtualization technology, thininstall virtualization suite, net framework virtualization, windows virtualization on fedora, storage virtualization technologies, virtualization level, security monitoring and virtualization, oracle virtualization.

UNIT –IV

Virtualization and Storage Management - The heart of cloud computing -virtualization, defining virtualization, why virtualize, what can be virtualized, where does virtualization happen, how does virtualization happen, on the road to storage virtualization, improving availability using virtualization, improving performance through virtualization, improving capacity through virtualization, business value for virtualization.

UNIT – V

Introduction to Cloud Computing – Cloud Introduction and overview- Components, Infrastructure and Services, Why Use Cloud Computing, Benefits and Limitations, Cloud Application Architectures, Cloud Infrastructure Models, Cloud Computing Technology- Hardware & Software Infrastructure.

UNIT- VI

Cloud Computing Architecture- Requirements, Introduction to Cloud Computing Architecture, various kinds of Cloud Computing Architecture, Grid Computing, Transactional Computing, On Demand Computing, and Distributed Computing.

UNIT- VII

Security – Security issues in Cloud Computing - Data Security, Network Security, and Host Security.

UNIT – VIII

Disaster Recovery- Disaster Recovery Planning, Disasters in the Cloud, Disaster Management. Scaling a Cloud Infrastructure- Capacity Planning, Cloud Scale.

Case Studies: Amazon S3, Google APP Engine, IBM Clouds.

TEXT BOOKS:

1. Cloud Computing Virtualization Specialist Complete Certification Kit - Study Guide Book by Ivanka Menken , Gerard Blokdijsk .
2. Cloud Application Architectures Building Applications and Infrastructure in the Cloud
by George Reese, Publisher: O'Reilly Media, Released: April 2009, Press Release.

REFERENCE BOOKS:

1. Cloud Computing: A Practical Approach. Anthony T. Velte, Tobe J. Velte, Robert Elsenpeter. Publication Person Education, 2009.
2. Storage Virtualization: Technologies for Simplifying Data Storage and Management by Tom Clark.
3. Cloud Computing, First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings by Martin Gilje Jaatun, Gansen Zhao and Chunming Rong.
4. Introduction to Cloud Computing: Business & Technology by Timothy Chou

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)**

10MT10509: ADVANCED DATA STRUCTURES & ALGORITHMS AND MIDDLEWARE TECHNOLOGIES LAB

Data Structures Lab

Objectives:

- To make the student learn an object oriented way of solving problems.
- To make the student write ADTS for all data structures.
- Recommended Systems/Software Requirements:
- Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space C++ compiler and STL Recommended.

Week1: Write C++ programs to implement the following using an array.

- a) Stack ADT b) Queue ADT

Week2: Write C++ programs to implement the following using a singly linked list.

- a) Stack ADT b) Queue ADT

Week3: Write C++ programs to implement the deque (double ended queue) ADT using a doubly linked list and an Array.

Week 4: Write a C++ program to perform the following operations:

- a) Insert an element into a binary search tree.
b) Delete an element from a binary search tree.
c) Search for a key element in a binary search tree.

Week 5: Write C++ programs that use recursive functions to traverse the given binary tree in

- a) Preorder b) inorder and c) postorder

Week 6: Write C++ programs that use non-recursive functions to traverse the given binary tree in

- a) Preorder b) Inorder and c) Postorder.

Week 7: Write C++ programs for the implementation of bfs and dfs for a given graph.

Week 8: Write C++ programs for implementing the following sorting methods:

- a) Merge sort b) Heap sort

Week 9: Write a C++ program to perform the following operations

- a) Insertion into a B-tree
b) Deletion from a B-tree

Week 10: Write a C++ program to perform the following operation

- a) Insertion into an AVL - tree
b) Deletion from an AVL - tree

Week 11: Write a C++ program to implement All Pairs Shortest Path.

Week 12: Write a C++ program for implementing 8 Queen's problem.

Middleware Technologies Lab

1. Write an arithmetic operations calculator web application program using ASP.NET
2. Write a client/server arithmetic operations calculator program using java RMI
3. Write a client/server Stock market java program using java RMI
4. Write a client/server address book java program using java RMI.
5. Write a program to create various forms to store customer data, delete customer data, update customer data, retrieve customer data using ASP.NET and MS-ACCESS database.
6. Write a program to create various forms to maintain the Super market using ASP.NET and MS-ACCESS database.
7. Write an address book program using ASP.NET and MS-ACCESS database
8. Write a client/server arithmetic operations calculator program using CORBA.
9. Write a client/server Stock market java program using CORBA.
10. Write a client/server address book java program using CORBA.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20501: SERVICE ORIENTED ARCHITECTURE

UNIT I

SOA and Web Services Fundamentals

Introducing SOA- Fundamental SOA, Common Characteristics of Contemporary SOA, Common tangible benefits of SOA, Common pitfalls of adopting SOA. The Evolution of SOA – An SOA timeline, The continuing evolution of SOA, The roots of SOA.

UNIT II

Web Services and primitive SOA-The Web Services frame work, Services, Service descriptions, Messaging. Web Services and Contemporary SOA (Part I-Activity management and Composition) Message exchange patterns, Service Activity Coordination, Atomic transactions, Business Activities, Orchestration, Choreography.

UNIT III

Web Services and Contemporary SOA (Part-II-Advanced Messaging, Metadata, and Security) Addressing, Reliable messaging, Correlation, Policies, Metadata exchange, Security, Notification and eventing.

UNIT IV

Principles of Service-Oriented

Service – Orientation and the enterprise, Anatomy of SOA, Common Principles of Service – Orientation, interrelation between Principles of Service-Oriented, Service Orientation and Object Orientation, Native Web Services support for Principles of Service-Oriented.

UNIT V

Service Layers

Service-Oriented and Contemporary SOA, Service Layer abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration Scenarios. SOA Delivery Strategies-SOA delivery lifecycle phases, The top-down strategy, The bottom-up strategy, The agile strategy.

UNIT VI

Service Oriented Analysis (Part I-Introduction)-Introduction to Service Oriented Analysis, Benefits of a Business Centric SOA, Deriving Business Services. Service Oriented Analysis (Part-II-Service Modelling)-Service Modelling, Service Modelling guidelines, Classifying Service model logic, Contrasting Service modelling approaches.

UNIT VII

Building SOA (Technology and Design)

Service Oriented Design (Part I-Introduction)-Introduction to Service-Oriented design, WSDL related XML Schema language basics, WSDL language basics, Service interface design tools. Service Oriented Design (Part II-SOA Composition Guidelines)-SOA Composing steps, Considerations for choosing service layers, Considerations for positioning core SOA standards, Considerations for choosing SOA extensions. Service Oriented Design (Part III-Service Design) Service Design overview, Entity-centric business Service Design, Application Service Design, Task-centric business Service Design, Service Design guidelines.

UNIT VIII

Service Oriented Design (Part IV-Business Process Design)-WS-BPEL language basics, WS- Coordination overview, Service Oriented Business process Design. Fundamental WS-* Extensions-WS-Addressing language basics, WS-Reliable Messaging language basics, WS-Policy language basics, WS-Metadata Exchange language basics, WS-Security language basics.

TEXT BOOKS:

1. Service-Oriented Architecture-Concepts, Technology, and Design, Thomas Erl, Pearson Education.
2. Understanding SOA with Web Services, Eric Newcomer, Greg Lomow, Pearson Education.

REFERENCE BOOKS:

1. The Definitive guide to SOA, Jeff Davies & others, Apress, Dreamtech.
2. Java SOA Cook book, E.Hewitt, SPD.
3. SOA in Practice, N.M.Josuttis, SPD.
4. Applied SOA, M.Rosen and others, Wiley India pvt. Ltd.
5. Java Web Services Architecture, J.Mc Govern, and others, Morgan Kaufmann Publishers, Elsevier.
6. SOA for Enterprise Applications, Shankar.K, Wiley India Edition.
7. SOA-Based Enterprise Integration, W.Roshen, TMH.
8. SOA Security, K.Rama Rao, C.Prasad, dreamtech press.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

M.Tech (Software Engineering) (Regulations 2010-11) I Year (II-Semester)

10MT20502: SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

UNIT I

Envisioning Architecture

What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views and the Architecture Business Cycle.

UNIT II

Creating an Architecture

Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT III

Analyzing Architectures

Architecture Evaluation, Architecture design decision making, ATAM, CBAM.

UNIT IV

Moving from one system to many

Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT V

Introduction to Design Patterns

What is Design Pattern, Pattern Description, Organizing catalogs, Role in solving design problems, Selection and Usage.

UNIT VI

Creational and Structural patterns

Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, decorator, façade, flyweight, Proxy.

UNIT VII

Behavioral patterns

Chain of responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

UNIT VIII

Case Studies

Lexi Editor in Design Patterns

A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development.

TEXT BOOKS:

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements&Rick Kazman, Pearson Education,2003.
2. Design Patterns, Erich Gamma, Pearson Education, 1995.

REFERENCE BOOKS:

1. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR,2001
2. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
3. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20503: SOFTWARE REENGINEERING AND REUSE

UNIT I:

Legacy software structure, Software reengineering process model:

Software maintenance:, The maintenance Process, Estimation of maintenance costs, Software change strategies include: Software maintenance, Architectural transformation, Software reengineering.

Legacy software structure and distribution: Ideal structure, Real structure, Layered distribution model, Legacy software distribution, Architectural problems.

Business process reengineering: Business processes, A BPR Model.

UNIT II:

Design Extraction:

Reverse Engineering: Goals of reverse engineering, Reverse engineering process, Reverse engineering to understand processing, Code duplication detection, Reverse engineering to understand data, Reverse engineering user interfaces, Tools for reverse engineering.

Software reengineering and its importance, Goals of reengineering, A software reengineering process model, Software reengineering activities.

UNIT III:

Restructuring (In Traditional context): Code restructuring: Characteristics of unstructured code, Characteristics of structured code, Spaghetti logic, Structured control logic, Restructuring problems, Data restructuring (Data reengineering): Data reengineering process, Data problems, Approaches: Data cleanup, Data extension, Data migration. Tools for restructuring.

UNIT IV:

Refactoring (Restructuring in object oriented context): What is refactoring?, Principles in refactoring: Why should you refactor?, When should you refactor?, Problems with refactoring, Refactoring and design, Refactoring and performance. Refactoring opportunities, Top ten of code bad smells, Different refactorings and their use, Refactoring tools.

UNIT V:

Forward Engineering: What is forward engineering ? Goals of forward engineering, Forward engineering for client/server applications, Forward engineering for object oriented architectures, Forward engineering user interfaces, Tools for forward engineering.

UNIT VI:

Introducing the Reuse-Driven Software Engineering Business: Software Reuse Success Factors, Reuse Driven Software Engineering is a Business.

Architecture Style-I: Object-Oriented Software Engineering, Application and Component Systems

UNIT VII:

Architecture Style-II-Use case Components, Object Components, Layered Architecture

UNIT VIII:

Processes: Application Family Engineering, Component System Engineering, Application System Engineering

TEXT BOOKS:

1. Software Engineering, Ian Sommerville, Addison-Wesley, 6th Edition.
2. Software Engineering, A Practitioner's Approach, Roger S. Pressman, 6th Edition.
3. Software Reuse: Architecture, Process and Organization for Business Success, Ivar Jacobson, Martin Griss, Patrik Jonsson, Pearson Education, 2000.

REFERENCES

1. Software Reengineering, Ed. Robert S. Arnold, IEEE Computer Society, 1993.
2. Software Evolution, Tom Mens, Serge Demeyer, Springer publication company, 2008.
3. Refactoring: Improving the Design of Existing Code, Martin Fowler, K.Beck, J.Brant, W.Opdyke, D.Roberts, Addison- Wesley, NY, 1999.
4. Software Reengineering, Georg Abfalter, VDM Verlag, Germany, 2008.
5. Successful Software Reengineering, Salvatore Valenti, IRM Press, 2002.
6. Logical construction of programs, J.D.Warnier, Van Nostrand-Reinhold, 1974.
7. Tutorial on Software Restructuring, Robert E.Arnold, IEEE Computer Society, 1986.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20504: SOFTWARE SECURITY ENGINEERING

UNIT I:

Why is Security a Software Issue?

Introduction, The problem, Software assurance and software security, Threats to software security, Sources of software insecurity, the benefits of detecting software security defects early, managing secure software development.

UNIT II:

What Makes Software Secure?

Defining properties of secure software, How to influence the security properties of software, How to assert and specify desired security properties.

UNIT III:

Requirements Engineering for Secure Software

The SQUARE process model: Identifying security requirements using the security quality requirements engineering (SQUARE) method, SQUARE sample outputs, Requirements elicitation, Requirements prioritization.

UNIT IV:

Secure Software Architecture and Design

Introduction, Software security practices for architecture and design: Architectural risk analysis. Software security knowledge for architecture and design: Security principles, Security guidelines, and Attack patterns.

UNIT V:

Considerations for Secure Coding and Testing

Introduction, Code analysis, Coding practices, Software security testing, Security testing considerations throughout the SDLC

UNIT VI:

Security and Complexity: System Assembly Challenges

Introduction, Security failures, Functional and attacker perspectives for security analysis, System complexity drivers and security

UNIT VII:

Governance, and Managing for More Secure Software

Governance and security, Adopting an enterprise software security framework, How much security is enough? Security and project management

UNIT VIII:

Security metrics

Defining security metrics, Diagnosing problems and measuring technical security

TEXT BOOKS

1. Software Security Engineering: A Guide for Project Managers-
by Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead,
Pearson Education (India), 2009
2. Security Metrics: Replacing Fear, Uncertainty, and Doubt - by Andrew
Jaquith, Addison-Wesley, 2007.

REFERENCES

1. Software Security: Building Security In, by Gary McGraw , Addison-Wesley, 2006
2. The Art of Software Security Assessment: Identifying and Preventing Software
Vulnerabilities, by Mark Dowd, John McDonald, Justin Schuh, Addison-Wesley,
1st edition, 2006
3. Building Secure Software: How to Avoid Security Problems the Right Way
by John Viega, Gary McGraw, Addison-Wesley, 2001
4. Exploiting Software: How to break code, by G. Hoglund, G. McGraw, Addison Wesley,
2004.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20505:

UNIT I: Software Quality Assurance Framework: What is Quality? Prevention versus Detection, Verification versus Validation, Software Quality Assurance, Components of Software Quality Assurance, **Software Quality Assurance Plan-** Steps to develop and implement a Software Quality Assurance Plan, **Quality Standards:** ISO 9000, CMM, People CMM, CMMI, Malcolm Baldrige National Quality Award, 3 Sigma, 6 Sigma

UNIT II: Software Quality Metrics Overview

Product Quality metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metrics Programs, Collecting Software Engineering Data

UNIT III: The Software Quality Metrics Methodology: Establish Software quality requirements, Identify Software quality metrics, implement the software quality metrics, analyze the software metrics results, validate the software quality metrics, **Software quality indicators, Fundamentals of Measurement theory**

UNIT IV: Building a Software Testing Strategy

Economics of System Development Life Cycle (SDLC) Testing, Testing-an Organizational Issue, Establishing a Testing policy, structured approach to testing, Test Strategy, Testing methodology, Status of software Testing

UNIT V: Establishing a Software Testing Methodology

Testing Purpose, Defects hard to find, reduce the cost of Testing, verification and validation, functional and structural testing, workbench concept, Eight considerations in developing testing methodologies, Testing tactics checklist

UNIT VI: Software Testing Techniques

Black-Box, Boundary value, Bottom-up, Branch coverage, Cause-Effect graphing, CRUD, Database, Exception, Gray-Box, Histograms, Inspections, JADs, Pareto Analysis, Prototyping, Random Testing, Risk-based Testing, Regression Testing, Structured Walkthroughs, Thread Testing, Performance Testing, White-Box Testing

UNIT VII: Software Testing Tools

Taxonomy of Software Testing Tools, Methodology to evaluate automated testing tools, Win runner, QTP, Load Runner, SQA Robot, Silk test, JMeter, JUNIT and Cactus

UNIT VIII: The Eleven-Step Testing Process: Assess Project Management Development Estimate and Status, Develop Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report test results, testing software installation, Test software changes, Evaluate Test Effectiveness

Testing Specialized Systems and Applications

Testing Client/Server Systems, Testing Web-based systems, testing off-the-Shelf Software, Testing a Data Warehouse

TEXT BOOKS:

1. Software Testing and continuous Quality Improvement, by William E.Lewis, Third Edition, An Auerbach Publication
2. Effective Methods for Software Testing, William E. Perry, Second Edition, Wiley India, 2006

REFERENCE BOOKS:

1. Software Testing Techniques, by Bories Beizer, Second Edition, Dreamtech Press
2. Software Testing Tools, K.V.K.K. Prasad, Dream tech press, 2008.
3. Handbook of Software Quality Assurance, Fourth Edition, G.Gordon Schulmeyer, Artech House Publishers
4. Metrics and Models in Software Quality Engineering, by Stephen H. Kan, 2nd Edition, Pearson Education Publication

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20506: INFORMATION RETRIEVAL SYSTEMS

UNIT-I

Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital libraries and Data Warehouses.

UNIT-II

Information Retrieval System Capabilities: Search, Browse, Miscellaneous

UNIT-III

Cataloging and Indexing: Objectives, Indexing Process, Automatic Indexing, Information Extraction.

UNIT-IV

Data Structures: Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure.

UNIT-V

Automatic Indexing: Classes of automatic indexing, Statistical indexing, Natural language, Concept indexing, Hypertext linkages

UNIT-VI

Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters.

UNIT-VII

User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, Weighted searches of Boolean systems, Searching the Internet and hypertext.

Information Visualization: Introduction, Cognition and perception, Information visualization technologies.

UNIT-VIII

Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems.

Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example – TREC results.

TEXTBOOKS :

1. Kowalski, Gerald, Mark T Maybury: Information Retrieval Systems: Theory and Implementation, Kluwer Academic Press, 1997.

REFERENCES :

1. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
2. Modern Information Retrieval By Yates Pearson Education.
3. Information Storage & Retrieval By Robert Korfhage – John Wiley & Sons.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20507: SOFT COMPUTING

UNIT: I

Introduction to intelligent systems and soft computing

Introduction, Intelligent systems, Knowledge-based systems, Knowledge representation and Processing, soft computing

UNIT-II

Fundamentals of Fuzzy Logic Systems

Introduction, Background, Fuzzy sets, Fuzzy logic operations, Generalized fuzzy operations, Implication (if-then), Some definitions, Fuzziness and Fuzzy resolution, Fuzzy relations, Composition and Inference, considerations of fuzzy decision-making.

UNIT-III

Fuzzy Logic Control

Introduction, Background, Basic of Fuzzy control, Defuzzification, Fuzzification, Fuzzy Control Surface, Extensions of Mamdani fuzzy control.

UNIT-IV

Fuzzy Control Architectures

Fuzzy control architectures, Properties of fuzzy control, Robustness and Stability.

UNIT-V

Fundamentals of Artificial Neural Networks

Introduction, Learning and acquisition of knowledge, Features of artificial Neural Networks, Fundamentals of Connectionist Modeling.

Major Classes of Neural Networks

Introduction, The Multilayer Perceptron.

UNIT-VI

Neuro-Fuzzy Systems

Introduction, Background, Architectures of neuron-fuzzy systems, Construction of neuron-fuzzy systems.

UNIT-VII

Evolutionary Computing

Introduction, Overview of evolutionary computing, Genetic algorithms and optimization, The schema theorem: the fundamental theorem of genetic algorithms, Genetic algorithm operators, Integration of genetic algorithms with neural networks, Integration of genetic algorithms with fuzzy logic, Known issues in GAs, Population-based incremental learning, Evolutionary strategies, ES applications.

UNIT-VIII

Applications

Pattern Recognition, Image Processing, Information Retrieval Systems, Share Market Analysis, Soft Computing for Colour Recipe Prediction.

TEXT BOOKS:

1. “Soft Computing and Intelligent Systems Design”, by Fakhreddine.O. Karray and Clarence De Silva, Pearson Education.
2. “Neuro-Fuzzy and Soft Computing”, J.S.R. Jang, C.T. Sun and E.Mizutani, PHI, 2004, Pearson Education 2004.

REFERENCE BOOKS:

1. “Fuzzy Logic with Engineering Applications”, by Timothy J. Ross, McGraw-Hill, 1997.
2. “Computational Intelligence Principles, Techniques and Applications”, by A.Konar, Springer, 2005.
3. “Introduction to Pattern Recognition- Statistical, Structural, Neural and Fuzzy Logic Approaches”, by M.Friedman and Abraham Kandal, World Scientific, 2005.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

**M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (II-Semester)**

10MT20508: SOFTWARE RELIABILITY

UNIT-I:

Introduction

The Need for Reliable Software, Software Reliability Engineering Concepts, Basic definitions, Software practitioners biggest problem, software reliability engineering approach, software reliability engineering process, defining the product

UNIT- II:

The Operational Profile

Reliability concepts, software reliability and hardware reliability, developing operational profiles, applying operational profiles, learning operations and run concepts.

UNIT-III:

Software Reliability Concepts

Defining failure for the product, common measure for all associated systems, setting system failure intensity objectives, determining develop software failure intensity objectives, software reliability strategies, failures, faults and errors, availability, system and component reliabilities and failure intensities, predicting basic failure intensity.

UNIT- IV:

Software Reliability Modeling Survey

Introduction, Historical Perspective and Implementation, Exponential Failure Time Class of Models, Weibull and Gamma Failure Time Class of Models, Infinite Failure Category Models, Bayesian Models, Model Relationship, Software Reliability Prediction in Early Phases of the Life Cycle.

UNIT-V:

Software Metrics for Reliability Assessment

Introduction, Static Program Complexity, Dynamic Program Complexity, Software Complexity and Software Quality, Software Reliability Modeling.

UNIT-VI:

Software Testing and Reliability

Introduction, Overview of Software Testing, Operational profiles, Time/Structure Based Software Reliability Estimation.

UNIT-VII:

Best Practice of SRE

Benefits and approaches of SRE, SRE during requirements phase, SRE during implementation phase, SRE during Maintenance phase.

UNIT VIII:

Neural Networks for Software Reliability

Introduction, Neural Networks, Neural Networks for software reliability, software reliability growth modeling.

TEXT BOOKS

1. Handbook of Software Reliability Engineering Edited by Michael R. Lyu, published by IEEE Computer Society Press and McGraw-Hill Book Company.
2. Software Reliability Engineering John D. Musa, second edition Tata McGraw-Hill.

REFERENCE BOOKS

1. Practical Reliability Engineering, Patric D. T. O connor 4th Edition, John Wesley & Sons, 2003.
2. Fault tolerance principles and Practice, Anderson and PA Lee, PHI, 1981.
3. Fault tolerant computing-Theory and Techniques, Pradhan D K (Ed.): Vol 1 and Vol 2, Prentice hall, 1986.
4. Reliability Engineering E. Balagurusamy, Tata McGrawHill, 1994.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)

M.Tech (Software Engineering) (Regulations 2010-11)
I Year (II-Semester)

10MT20510: SOFTWARE TESTING AND DESIGN PATTERNS LAB

Design Patterns:

1. Write a C++ Program to perform Abstract Factory Design pattern.
2. Write a C++ Program to perform Singleton Design pattern.
3. Write a C++ Program to perform Adapter Design pattern.
4. Write a C++ Program to perform Composite Design pattern.
5. Write a C++ Program to perform Proxy Design pattern.
6. Write a C++ Program to perform Command Design pattern.
7. Write a C++ Program to perform Mediator Design pattern.
8. Write a C++ Program to perform State Design pattern.
9. Write a C++ Program to perform Strategy Design pattern
10. Write a C++ Program to perform Visitor Design pattern.

Lab Experiments Using WinRunner

(I) Testing Flight Reservation Application

Functional specifications:

1. The width of the GUI object “*Name*” in the Flight Reservation window must be 150
2. The height of the GUI object “*Order no*” in the open order window must be 20
3. **The flight schedule frame** of Flight Reservation window should have the following properties
 - a) Height - 283
 - b) Width - 138
 - c) X - 11
 - d) Y - 84
4. **The order information frame** of Flight Reservation window should have the following properties
 - a) Height - 283
 - b) Width - 252
 - c) X - 155
 - d) Y - 84
5. It is sufficient to verify **X or Y** Axis, both properties are not necessary

(II) Testing standard Calculator Application

- 1) Test the inverse operation (inverse of 4,5 using 1/x button)

Test Case	Expected Output
4 , 1/x	0.25
5 , 1/x	0.2

- 2) Test the operations “square root of -6 and square root of 4 ”

Test Case	Expected Output
-6 , sqrt	Err: “Invalid Input for function”
4 , sqrt	2

- 3) Clear the display after performing some arithmetic operations

Test Case	Expected Output
4+5=9 , C	Clears the display
5*3=15 , C	Clears the display

- 4) Test the multiplication of two numbers

Test Case	Expected Output
1.2 * 3	3.6
15 * 3	45

- 5) Test the Division of two numbers

Test Case	Expected Output
5/2.0	2.5
4/2	2

- 6) Test the operations “7+8-9” and “15*3-10”

Test Case	Expected Output
7+8-9	6
15*3-10	35

- 7) Test the operations “2% of 600” and “8% of 700”

Test Case	Expected Output
600 * 2%	12
700 * 8%	56

- 8) Test the Buttons “MR” and “MS”

Test Case	Expected Output
2 , MS , C , MR	2
5*3=15, MS , C , MR	15

- 9) Test the Buttons “M+” and “MR”

Test Case	Expected Output
MC,2,M+,3,M+,C,MR	5
MC,9,M+,5,M+,C,MR	14

- 10) Test the Buttons “MC” and “Backspace”

Test Case	Expected Output
9,MS,C,MR,MC,C,MR	0
4*5=20,MS,C,MR,MC,C,MR	0
856,backspace(3 times)	0

(II) Testing Employee Database Application

- 1) Test the EMPDB Application with suitable operations
- 2) Implementation of *Checkpoints*
- 3) Implementation of *Call* Function

Lab Experiments Using Quick Test Professional

- 1) Testing Caluculator,EMP DB,Flight Reservation Applications
- 2) Implementation of Checkpoints Using QTP
- 3) Implementation of Data Driven Testing in QTP
- 4) Testing a Web Application Using QTP
- 5) Synchronization of test cases in QTP