



SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)

Sree Sainath Nagar, Tirupati

Department of Information Technology

Supporting Document for 1.1.3

Courses having focus on
Employability/ Entrepreneurship/ skill Development

Program: M.Tech.- Software Engineering

Regulations : SVEC-14

The Courses (with course outcomes) under SVEC-14 Regulations which focus on ***employability/ entrepreneurship/ skill development*** are highlighted with the following colours.

Skill

Employability

Entrepreneurship

M.Tech. (SE) I Semester
(14MT10504) DATA STRUCTURES AND ALGORITHMS

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: A course on "Computer Programming".

COURSE DESCRIPTION:

Linear data structures, algorithms analysis; Searching & sorting, and non-linear data structures; Hashing; Algorithms design techniques – Divide & Conquer and Greedy method; Dynamic programming, backtracking and branch & bound.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge on:
 - Stacks, Queues and Lists
 - Trees and Graphs,
 - Algorithm design paradigms such as Divide & Conquer, Greedy method and Dynamic programming
 - Backtracking and Branch & Bound techniques.
2. Analyze and determine efficiency of algorithms using space and time complexities.
3. Identify efficient algorithm design techniques and apply Stack, Queue, Tree, and Graph data structures to solve real world computer applications.

DETAILED SYLLABUS

UNIT-I: BASIC DATA STRUCTURES AND ALGORITHM ANALYSIS

(Periods:12)

Basic Data Structures: Review of Arrays, Stacks, Queues, Circular Queues, Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists, Terminologies and Applications

Algorithm Analysis: Efficiency of algorithms, Apriori Analysis, Asymptotic Notations, Polynomial Vs Exponential Algorithms, Average, Best and Worst Case Complexities, Analyzing Recursive Programs.

UNIT-II: SEARCHING, SORTING, TREES AND GRAPHS **(Periods:13)**

Searching and Sorting: Linear Search, Binary Search, Fibonacci Search, Bubble Sort, Insertion sort, Selection Sort, Radix Sort.

Trees: Introduction, Definition and Basic terminologies of trees and binary trees, Representation of trees, Binary Tree Traversals, Threaded binary trees.

Graphs: Basic concepts, Representation and Applications, DFS and BFS Traversals.

UNIT-III: BINARY SEARCH TREES, AVL TREES, B- TREES AND HASH TABLES (Periods:12)

Binary Search Trees: Introduction, Definition, Operations and applications.

AVL Trees: Definition, Operations and applications.

B-Trees: Definition, Operations and applications.

Hash Tables: Introduction, Hash Tables, Hash Functions and its applications.

UNIT-IV: DIVIDE AND CONQUER & GREEDY METHOD (Periods:10)

Divide and Conquer: General Method, Binary Search, Finding Maximum and Minimum, Quick Sort, Merge sort, Polyphase Merge Sort, Cascade Merge Sort, Strassen's Matrix Multiplication.

Greedy Method: General Method, Minimum Cost Spanning Trees and Single Source Shortest Path.

UNIT-V: DYNAMIC PROGRAMMING, BACK TRACKING & BRANCH AND BOUND (Periods:12)

Dynamic Programming: General Method, All Pairs Shortest Path, Single Source Shortest Path, 0 / 1 Knapsack problem.

Back Tracking: General Method, 8 – Queen's Problem, Graph Coloring.

Branch and Bound: The Method, LC Search, Control Abstraction, Bounding, 0/1 Knapsack Problem.

Total Periods:59

TEXT BOOKS:

1. G. A. V. Pai, "Data Structures and Algorithms: Concepts, Techniques and Applications," 1st Edition, Tata McGraw Hill, 2008.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms," 2nd Edition, Universities Press (India) Pvt. Ltd, 2008.

REFERENCE BOOKS:

1. D. Samanta, "Classic Data Structures," 1st Edition, PHI Publication, 2008.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++," 3rd Edition, Pearson Education, 2007.
3. Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, "Data Structures using C and C++," 2nd Edition, Pearson Education, 2006.
4. Sartaj Sahni, "Data structures, Algorithms and Applications in C++," 2nd Edition, Universities press (India) Pvt. Ltd., 2005.

M.Tech. (SE) I Semester

(14MT12501) SOFTWARE DEVELOPMENT METHODOLOGIES

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: A course on "Software Engineering".

COURSE DESCRIPTION:

Software process models; Requirements analysis and modeling; Design techniques; Implementation and modern programming language features; Testing and maintenance of software.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain in-depth knowledge in: Software Paradigms, Agile Development, Software Reuse and Testing.
2. Analyze, design, develop and maintain software systems.
3. Apply Software Engineering Principles and Practices to develop and implement software systems.
4. Demonstrate skills in applying software development methodologies and management principles for effective implementation of software projects.

DETAILED SYLLABUS

UNIT-I: INTRODUCTION

(Periods:13)

Software Engineering: Software Process-Generic process model, Prescriptive process model, specialized, unified process. Agile development-Agile Process, Extreme Programming, Adaptive Software Development(ASD), Scrum, Dynamic Systems Development Method(DSDM), Crystal, Feature Driven Development(FDD), Lean Software Development(LSD), Agile Modeling(AM), Agile Unified Process (AUP). Software Engineering Knowledge-core Principles, Principles that guide each framework Activity.

UNIT-II: SOFTWARE REQUIREMENTS AND ANALYSIS

(Periods:11)

Establishing the Groundwork, Eliciting Requirements, Developing use cases, Building the requirements model, Negotiating, Validating Requirements. Requirements Analysis, Requirements Modeling Strategies.

UNIT-III: SOFTWARE DESIGN

(Periods:13)

Design Process, Design concepts - Abstraction, Architecture, Patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring, Object Oriented Design Concepts, Design Classes. Design Model- Data, Architectural, Interface, Component, Deployment Level Design Elements. Design Techniques- Stepwise Refinement, Level of Abstraction, Structure Design, Integrated Top-Down Development, Jackson Structured Programming, Summary of Design Techniques.

UNIT-IV: SOFTWARE IMPLEMENTATION (Periods:10)

Implementation Issues: Structured Coding Techniques, Coding Styles, Standards and Guidelines, Documentation Guidelines.

Modern Programming Language Features: Type checking, User Defined Data Types, Data Abstraction, Exception Handling, Concurrency Mechanism.

UNIT-V: SOFTWARE TESTING AND MAINTENANCE (Periods:12)

Testing: Strategic Approach to Software Testing, Strategic Issues, Testing Strategies for Conventional Software, Object Oriented Software and Web Apps, Validating Testing, System Testing, Art of Debugging.

Maintenance: Software Maintenance, Software Supportability, Reengineering, Business Process Reengineering, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering, Economics of Reengineering.

Total Periods:59

TEXT BOOKS:

1. Roger S. Pressman, "*Software Engineering – A Practitioner's Approach*," 7th Edition, Tata McGraw-Hill, 2009.
2. Richard Fairley, "*Software Engineering Concepts*," Tata McGraw Hill, 2008.

REFERENCE BOOKS:

1. Ian Sommerville, "*Software Engineering*," 7th Edition, Pearson Education Asia, 2007.
2. Shari Lwarence Pfleeger, Joanne M. Atlee, "*Software Engineering Theory and Practice*," 3rd Edition, Pearson Education, 2006.

M.Tech. (SE) I Semester
(14MT12502) SOFTWARE PROCESS AND PROJECT
MANAGEMENT

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: A course on "Software Engineering"

COURSE DESCRIPTION:

Software process maturity levels; Framework of various activities used in software process- software cost estimation, software standards, inspections, SCM-I & SCM-II, defect prevention; Project management-planning, artifacts of the process, process automation, modern processes transitions and Case studies.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Acquire knowledge of software process and project management activities.
2. Perform analysis of project management techniques for deploying software systems quickly.
3. Skills to solve problems in modern software process and project management.

DETAILED SYLLABUS

UNIT-I: SOFTWARE PROCESS MATURITY AND THE REPEATABLE PROCESS (Periods:11)

A Software Maturity Framework, The Principles of Software Process Change, Software Process Assessment, The Initial Process, Managing Software Organizations, The Project Plan, SCM-I.

UNIT-II: THE DEFINED, MANAGED, OPTIMIZING PROCESS AND RISK MANAGEMENT (Periods:13)

Software Standards, Inspections, SCM-II, Defining the Software Process, SEPG, Data Gathering and Analysis, Managing Software Quality, Defect Prevention, Automating the Software Process, Risk Management – Introduction, The nature of risk, Managing risk, Risk identification, Risk analysis, Reducing the risks, Evaluating risks to the schedule, Calculating the z- values.

UNIT-III: SOFTWARE MANAGEMENT RENAISSANCE AND PROCESS FRAMEWORK (Periods:12)

Conventional Software Management, Evolution of Software Economics, Improving Software Economics, The old way and the new, Life Cycle Phases, Artifacts of the Process, Workflows of the Process, Checkpoints of the Process.

UNIT-IV: SOFTWARE MANAGEMENT DISCIPLINES (Periods:14)

Iterative Process Planning, Project Organizations and Responsibilities, Process Automation, Project control and Process Instrumentation, Tailoring the Process.

UNIT-V: NEXT GENERATION SOFTWARE ECONOMICS AND CASE STUDIES (Periods:11)

Modern Project Profiles, Next Generation Software Economics, Modern Process Transitions.

Case studies: The command Center Processing and Display System-Replacement (CCPDS-R), Process Improvement and Mapping to the CMM.

Total Periods:46

TEXT BOOKS:

1. Watts S. Humphrey, "*Managing the Software Process*," 1st Edition, Pearson Education, 2009.
2. Walker Royce, "*Software Project Management*," 1st Edition, Pearson Education, 2005.

REFERENCES:

1. Bob Hughes and Mike Cotterel, "*Software Project Management*," 4th Edition, Tata McGraw- Hill, 2006.
2. Joel Henry, "*Software Project Management*," 1st Edition, Pearson Education, 2004.
3. Roger S. Pressman, "*Software Engineering - A Practitioner's Approach*," 6th Edition, McGraw-Hill, International Edition, 2001.

M.Tech. (SE) I Semester

(14MT12503) SOFTWARE REQUIREMENTS AND ESTIMATION

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: Courses on "Software Engineering" and "Object Oriented Analysis & Design".

COURSE DESCRIPTION:

Requirements elicitation, analysis, modeling, and specification; Requirements management; Components of software estimation and Estimation management tools.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge on:
 - Requirements Engineering and Management
 - Estimation of software size, effort, schedule and cost.
2. Analyze software projects critically and develop a quality software product using Software Requirements and Estimation.
3. Acquire problem solving skills to assess the project risks and create prototypes to manage the risks.

DETAILED SYLLABUS

UNIT-I: SOFTWARE REQUIREMENTS (Periods:15)

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

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-II: SOFTWARE REQUIREMENTS MANAGEMENT (Periods:10)

Requirements management, Principles and Practices, Requirements Attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain.

UNIT-III: SOFTWARE ESTIMATION (Periods:12)

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

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

UNIT-IV: EFFORT, SCHEDULE AND COST ESTIMATION (Periods:10)

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

UNIT-V: REQUIREMENTS MANAGEMENT TOOLS (Periods:10)

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

SOFTWARE ESTIMATION TOOLS

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

Total Periods:57

TEXT BOOKS:

1. Karl E. Weigers, "*Software Requirements*," 2nd Edition, Microsoft Press, 2013.
2. Rajesh Naik and Swapna Kishore, "*Software Requirements and Estimation*," Tata McGraw Hill, 2001.

REFERENCE BOOKS:

1. Dean Leffing well and Don Widrig, "*Managing Software Requirements*," Pearson Education, 2003.
2. Suzanne Robertson and James Robertson, "*Mastering the Requirements Process*," 2nd Edition, Pearson Education, 2006.
3. Capers Jones, "*Estimating Software Costs*," 2nd Edition, Tata McGraw-Hill, 2007.
4. M.A. Parthasarathy, "*Practical Software Estimation*," 1st Edition, Pearson Education, 2007.

M.Tech. (SE) I Semester
(14MT12504) WEB TECHNOLOGIES

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: A course on "Object Oriented Programming".

COURSE DESCRIPTION:

Web Technologies- HTML5, CSS, JavaScript, JQuery; Open source server-side scripting language- PHP; MySQL database concepts; and AJAX.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Acquire knowledge in various web page design technologies.
2. Analyze the functionality of client and server side components to validate web pages.
3. Gain skills in dynamic design concepts to make web pages more interactive, scalable and user friendly web applications.

DETAILED SYLLABUS

UNIT-I: HTML5 AND CSS3 (Periods:14)

HTML5: Overview of HTML and XHTML, HTML5 - Introduction, HTML5 Document Structure, Creating Editable Content, Checking Spelling Mistakes, Exploring Custom Data Attributes, Microdata, Client-Side Storage, Drag and Drop Feature, ARIA Accessibility, Offline Web Applications, Web Communications, Cross-Document Messaging and Desktop Notifications, 2D and 3D Graphics. **CSS3:** Introduction, Features of CSS3, Syntax of CSS, Exploring CSS selectors, Inserting CSS in HTML Document, State of CSS3.

UNIT-II: JAVASCRIPT AND JQUERY (Periods:12)

JavaScript: Overview of JavaScript, JavaScript Functions, Events, Image Maps and Animations, JavaScript Objects. **JQuery:** Fundamentals of JQuery, JQuery Selectors, JQuery Methods to Access HTML Attributes and Traversing, JQuery Manipulators, Events and Effects.

UNIT-III: INTRODUCTION TO PHP (Periods:10)

Introduction, Data Types, Variables, Constants, Expressions, String Interpolation, Control Structures, Functions, Arrays, Embedding PHP Code in Web Pages, Object Oriented PHP.

UNIT-IV: PHP AND MYSQL (Periods:12)

PHP and Web Forms, Sending Form Data to a Server, Authenticating Users with PHP, Session Handlers, PHP with MySQL, Interacting with the Database, Database Transactions.

UNIT-V: AJAX**(Periods:12)**

Exploring Different Web Technologies, Exploring AJAX, Creating a Sample AJAX Application, Displaying Date and Time using AJAX, Creating the XMLHttpRequest Object, Reading a File Synchronously and Asynchronously, Reading Response Headers, Loading List Boxes Dynamically using XMLHttpRequest Object, JQuery with AJAX, Validating a Field using AJAX and PHP.

Total Periods:60**TEXT BOOKS:**

1. Kogent Learning Solutions Inc, "*HTML 5 Black Book: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP and JQuery*," 1st Edition, Dreamtech Press, 2011.
2. W. Jason Gilmore, "*Beginning PHP and MySQL*," 4th Edition, APress, 2011.

REFERENCE BOOKS:

1. Andrea Tarr, "*PHP and MySQL*," 1st Edition, Willy India, 2012.
2. Thomas A. Powell, "*The Complete Reference: HTML and CSS*," 5th Edition, Tata McGraw Hill, 2010.
3. Steve Suehring, Tim Converse and Joyce Park, "*PHP6 and MySQL*," 1st Edition, Willy India, 2009.
4. P. J. Deitel and H. M. Deitel, "*Internet & World Wide Web How to Program*," 4th Edition, Pearson, 2009.

M.Tech. (SE) I Semester
(14MT12505) DISTRIBUTED DATABASES

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: A Course on "Database Management Systems"

COURSE DESCRIPTION:

Distributed database Design framework; Query processing and query optimization techniques; Distributed transactions management and reliability; Database administration activities; Case studies on Distributed database systems.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain in-depth knowledge in distributed database architecture and recovery from failures.
2. Perform analysis on translating global queries into fragment queries to respond quickly for user queries.
3. Design and develop techniques to avoid concurrency problems in distributed databases.

DETAILED SYLLABUS

UNIT-I: INTRODUCTION, PRINCIPLES & DISTRIBUTED DATABASE DESIGN (Periods:13)

Features of Distributed databases Versus Centralized databases, Why Distributed Databases?, DDBMSs, Level of Distributed Transparency, Reference Architecture, Types of Data Fragmentation, Distribution Transparency, Access primitives, Integrity constraints, A framework for Distributed Database Design, The Design of Database Fragmentation, The allocation of fragments.

UNIT-II: TRANSLATION OF GLOBAL QUERIES TO FRAGMENT QUERIES & OPTIMIZATION OF ACCESS STRATEGIES (Periods:13)

Equivalence Transformations for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parameter Queries, A Frame Work for Query Optimization, Join Queries, General Queries.

UNIT-III: THE MANAGEMENT OF DISTRIBUTED TRANSACTIONS AND CONCURRENCY CONTROL (Periods:13)

A Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrence Control for Distributed Transactions, Architectural aspects of distributed transactions, Foundations of distributed concurrency control, Deadlocks, Timestamps and Optimistic methods.

UNIT-IV: RELIABILITY & DISTRIBUTED DATABASE ADMINISTRATION (Periods:10)

Basic Concepts, Non-blocking Commitment Protocols, Reliability and Concurrency Control, Determining a Consistent View of Network, Detection and

Resolution of Inconsistency, Check points and Cold Restart, Catalog Management, Authorization and Protection.

UNIT-V: DISTRIBUTED DATABASE SYSTEMS: COMMERCIAL SYSTEMS

(Periods:10)

Tandem's ENCOMPASS Distributed Database system, IBM's Inter System Communication, Distributed-INGRES, POREL, Problems of Heterogeneous Distributed Databases, Brief study of MULTIBASE.

Total Periods:59

TEXT BOOKS:

1. Stefano Ceri and Giuseppe Pelagatti, "*Distributed Databases: Principles & Systems*," McGraw-Hill, 1985.
2. M. Tamer Ozsu and Patrick Valduriez, "*Principles of Distributed Database Systems*," 2nd Edition, Pearson Education, 2006.

REFERENCE BOOKS:

1. Ramez Elmasri, S. B. Navathe, D.V.L.N. Somayajulu and S. K. Gupta "*Fundamentals of Database Systems*", 4th Edition, Pearson Education, 2006.
2. C. J. Date, A. Kannan and S. Swaminathan "*An Introduction to Database Systems*", 8th Edition, Pearson Education, 2006.

M.Tech. (SE) I Semester
(14MT12506) MACHINE LEARNING

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: - NIL -

COURSE DESCRIPTION:

Machine learning fundamentals, applications; Multivariate methods, Bayesian networks, Decision tree learning; Support Vector Machines, Statistical learning methods, Unsupervised learning; Kernel Machines; Combining Multiple Learners and Reinforcement learning.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Acquire knowledge on supervised, unsupervised, and reinforcement machine learning techniques.
2. Analyze, design and develop applications such as Fraud detection, automation of medical diagnosis and games.
3. Apply Machine Learning approaches to solve complexity level of pattern recognition systems.

DETAILED SYLLABUS

UNIT-I: INTRODUCTION

(Periods:12)

What is Machine Learning, Examples of Machine Learning Applications. Supervised Learning: Learning a Class from Examples, Vapnik-Chervonenkis (VC) Dimension, Probably Approximately Correct (PAC) Learning, Noise, Learning Multiple Classes, Regression, Model Selection and Generalization and Dimensions of a Supervised Machine Learning Algorithm. Bayesian Decision Theory: Classification, Losses and Risks, Discriminant Functions, Utility Theory and Association Rules.

UNIT-II: MULTIVARIATE METHODS

(Periods:12)

Multivariate Data, Parameter Estimation, Estimation of Missing Values, Multivariate Normal Distribution, Multivariate Classification, Tuning Complexity, Discrete Features, Multivariate Regression. Dimensionality Reduction: Subset Selection, Principal Components Analysis, Factor Analysis, Multidimensional Scaling, Linear Discriminant Analysis, Isomap, Locally Linear Embedding. Clustering: Mixture Densities, k-Means Clustering, Expectation-Maximization Algorithm, Mixtures of Latent Variable Models, Supervised Learning after Clustering, Hierarchical Clustering, Choosing the Number of Clusters.

UNIT-III: DECISION TREES

(Periods:12)

Univariate Trees, Pruning, Rule Extraction from Trees, Learning Rules from Data, Multivariate Trees. Linear Discrimination: Generalizing the Linear Model, Geometry of the Linear Discriminant, Pairwise Separation, Parametric Discrimination Revisited, Gradient Descent, Logistic Discrimination, and Discrimination by Regression. Multilayer Perceptrons: The Perceptron, Training a Perceptron, Learning Boolean Functions, Multilayer Perceptrons, MLP as a Universal Approximator, Backpropagation Algorithm, Training Procedures,

Tuning the Network Size, Bayesian View of Learning, Dimensionality Reduction, Learning Time.

UNIT-IV: KERNEL MACHINES

(Periods:12)

Optimal Separating Hyperplane, The Nonseparable Case: Soft Margin Hyperplane, v-SVM, Kernel Trick, Vectorial Kernels, Defining Kernels, Multiple Kernel Learning, Multiclass Kernel Machines, Kernel Machines for Regression, One-Class Kernel Machines, Kernel Dimensionality Reduction. Bayesian Estimation: Estimating the Parameter of a Distribution, Bayesian Estimation of the Parameters of a Function, Gaussian Processes. Hidden Markov Models: Discrete Markov Processes, Hidden Markov Models, Three Basic Problems of HMMs, Evaluation Problem, Finding the State Sequence, Learning Model Parameters, Continuous Observations, The HMM with Input, Model Selection in HMM.

UNIT-V: COMBINING MULTIPLE LEARNERS

(Periods:11)

Rationale, Generating Diverse Learners, Model Combination Schemes, Voting, Error-Correcting Output Codes, Bagging, Boosting, Mixture of Experts Revisited, Stacked Generalization, Fine-Tuning an Ensemble, Cascading. Reinforcement Learning: Single State Case: K-Armed Bandit, Elements of Reinforcement Learning, Model-Based Learning, Temporal Difference Learning, Generalization, Partially Observable States.

Total Periods:59

TEXT BOOK:

1. Ethem Alpaydin, "*Introduction to Machine Learning*," 2nd Edition, MIT Press, 2009.

REFERENCE BOOKS:

1. Tom M. Mitchell, "*Machine Learning*," McGraw-Hill, 2013
2. Christopher M. Bishop, "*Pattern Recognition and Machine Learning*," 2nd Edition, Springer-Verilog, 2006.
3. Kevin P. Murphy, "*Machine Learning: A Probabilistic Perspective*," MIT Press, 2012.

M.Tech. (SE) I Semester
(14MT12507) USER INTERFACE DESIGN

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: A Course on "Software Engineering".

COURSE DESCRIPTION:

Characteristics & principles of User Interface Design; Requirement analysis- direct & indirect methods; Design- using Formatting menus & windows; Design-using Text boxes, multimedia and Windows layout.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Understand the concepts of user interfaces and related business functions.
2. Analyze user requirements and design interfaces appropriately.
3. Apply the theoretical design principles to design and evaluate user interfaces.

DETAILED SYLLABUS

UNIT-I: INTRODUCTION

(Periods:09)

Human Computer Interface, Characteristics of Graphics Interface, Direct Manipulation Graphical System, Web User Interface – Popularity, Characteristics and Principles. User Interface Design Process – Obstacles, Usability, Human Characteristics in Design, Human Interaction Speed.

UNIT-II: HUMAN COMPUTER INTERACTION

(Periods:13)

Business Functions, Requirement Analysis – Direct & Indirect Methods, Basic Business Functions, Design Standards, System Timings, Human Consideration in Screen Design, Structures of Menus, Functions of Menus, Contents of Menu.

UNIT-III: FORMATTING MENUS AND WINDOWS

(Periods:13)

Formatting, Phrasing the Menu, Selecting Menu Choice, Navigating Menus, Graphical Menus.

Windows: Characteristics, Components - Presentation Styles, Types, Managements, Organizations – Operations, Web Systems, Device– Based Controls Characteristics, Screen – Based Controls, Operate Control.

UNIT-IV: TEXT BOXES AND MULTIMEDIA

(Periods:15)

Text Boxes – Selection Control, Combination Control, Custom Control.

Multimedia: Text For Web Pages, Effective Feedback, Guidance and Assistance, Internationalization– Accessibility, Icons, Image, Multimedia, Coloring.

UNIT-V: WINDOWS LAYOUT

(Periods:07)

Prototypes, Kinds of Tests, Retest, Information Search, Visualization, Hypermedia, WWW– Software Tools.

Total Periods:57

TEXT BOOKS:

1. Wilbert O. Galitz, *"The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques,"* 2nd Edition, Wiley Publishing Inc., 2007.

REFERENCE BOOKS:

1. Ben Shneiderman, *"Designing the User Interface,"* 3rd Edition, Pearson Education, 1998.
2. Alan Dix, et.al, *"Human Computer Interaction,"* 3rd Edition, Pearson, 2003
3. J. Prece, Y. Rogers and H. Sharp, *"Interaction Design,"* 3rd Edition, Wiley, 2011.
4. Soren Lauesen, *"User Interface Design: A Software Engineering Perspective,"* Addison-Wesley, 2005.

M.Tech. (SE) I Semester
(14MT12508) CLOUD COMPUTING

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: Courses on “Operating Systems” and “Computer Networks”

COURSE DESCRIPTION:

Virtualization, Cloud architecture, assessing the value proposition; Understanding services & applications, capacity planning; Platform as a Service case studies- Google & Amazon Services; Cloud management and their applications- Microsoft cloud services, cloud security.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Acquire knowledge on services and types of infrastructural models in Cloud.
2. Identify the Cloud Architectures in developing the Web Applications.
3. Gain problem solving skills to assess & improve quality attributes, security, privacy concerns and performance in Cloud Computing.

DETAILED SYLLABUS

UNIT-I: VIRTUALIZATION (Periods:10)

Introduction to Virtualization, Objectives of Virtualization, Benefits of Virtualized Technology, Adding guest Operating System. Introduction to Virtualization Technologies - Ubuntu, VMware, Microsoft Hyper-V.

UNIT-II: DEFINING CLOUD (Periods:12)

Defining Cloud Computing, Cloud Types - The NIST model, The Cloud Cube Model, Deployment models, Service models, Examining the Characteristics of Cloud Computing, Paradigm shift, Benefits of cloud computing, Disadvantages of cloud computing, Assessing the Role of Open Standards.

Assessing the Value Proposition: Measuring the Cloud's Value, Early adopters and new applications, The laws of cloudonomics, Cloud computing obstacles, Behavioral factors relating to cloud adoption, Measuring cloud computing costs, Avoiding Capital Expenditures, Right-sizing, Computing the Total Cost of Ownership, Specifying Service Level Agreements, Defining Licensing Models.

Understanding Cloud Architecture: Exploring the Cloud Computing Stack, Composability, Infrastructure, Platforms, Virtual Appliances, Communication Protocols, Applications, Connecting to the Cloud, The Jolicloud Netbook OS, Chromium OS: The Browser as an Operating System.

UNIT-III: UNDERSTANDING SERVICES AND APPLICATIONS BY TYPE (Periods:12)

Defining Infrastructure as a Service (IaaS), IaaS workloads, Pods, aggregation, and silos, Defining Platform as a Service (PaaS), Defining Software as a Service (SaaS), SaaS characteristics, Open SaaS and SOA, Salesforce.com and CRM SaaS, Defining Identity as a Service (IDaaS), what is an identity? Net-

worked identity service classes, Identity system codes of conduct, IDaaS interoperability, Defining Compliance as a Service (CaaS).

Understanding Abstraction and Virtualization: Using Virtualization Technologies, Load Balancing and Virtualization, Advanced load balancing, The Google Cloud Understanding Hypervisors, Virtual machine types, VMware vSphere, Understanding Machine Imaging, Porting Applications, The Simple Cloud API, AppZero Virtual Application Appliance.

Capacity Planning: Capacity Planning, Defining Baseline and Metrics, Baseline measurements, System metrics, Load testing, Resource ceilings, Server and instance types, Network Capacity, Scaling.

UNIT-IV: EXPLORING PLATFORM AS A SERVICE (Periods:13)

Defining Services, Salesforce.com versus Force.com, SaaS versus PaaS, Application development, Using PaaS Application Frameworks, Drupal, Eccentex AppBase 3.0, LongJump, Squarespace, WaveMaker, Wolf Frameworks.

Using Google Web Services: Exploring Google Applications, Surveying the Google Application Portfolio, Indexed search, The dark Web, Aggregation and disintermediation, Productivity applications and services, Enterprise offerings, AdWords, Google Analytics, Google Translate, Exploring the Google Toolkit, The Google APIs, Working with the Google App Engine.

Using Amazon Web Services: Understanding Amazon Web Services, Amazon Web Service Components and Services, Working with the Elastic Compute Cloud (EC2), Amazon Machine Images, Pricing models, System images and software, Creating an account and instance on EC2, Working with Amazon Storage Systems, Amazon Simple Storage System (S3), Amazon Elastic Block Store (EBS), CloudFront, Understanding Amazon Database Services, Amazon SimpleDB, Amazon Relational Database Service (RDS), Choosing a database for AWS.

UNIT-V: USING MICROSOFT CLOUD SERVICES (Periods:12)

Exploring Microsoft Cloud Services, Defining the Windows Azure Platform, The software plus services approach, The Azure Platform, The Windows Azure service, Windows Azure AppFabric, Azure Content Delivery Network, SQL Azure, Windows Azure pricing, Windows Live services, Using Windows Live, Windows Live Essentials, Windows Live Home, windows live for mobile.

Managing the Cloud: Administrating the Clouds, Management responsibilities, Lifecycle Management, Cloud Management Products, Emerging Cloud Management Standards, DMTF Cloud Management standards, Cloud Commons and SMI.

Understanding Cloud Security: Securing the Cloud, The security boundary, Security service boundary, Security mapping, Securing Data, Brokered cloud storage access, Storage location and tenancy, Encryption, Auditing and compliance, Establishing Identity and Presence, Identity protocol standards, Windows Azure identity standards.

Total Periods:59

TEXT BOOKS:

1. Barrie Sosinsky, "*Cloud Computing Bible*," 1st Edition, Wiley India Pvt Ltd, 2011.
2. Ivanka Menken and Ivanka Menken, "*Cloud Computing Virtualization Specialist Complete Certification Kit - Study Guide Book*," 1st Edition, Emereo Pty. Ltd., 2009.

REFERENCE BOOKS:

1. Anthony T. Velte, Toby J. Velte Robert Elsenpeter, "*Cloud Computing: A Practical Approach*," 1st Edition, Tata Mc Graw Hill, 2010.
2. John W. Rittinghouse and James F. Ransome, "*Cloud Computing Implementation, Management and Security*," 1st Edition, CRC Press, Taylor & Francis Group, 2010.
3. George Reese, "*Cloud Application Architectures*," 1st Edition, O'Reilly Publishers, 2010.
4. David S. Linthicum, "*Cloud Computing and SOA Convergence in your Enterprise*," 1st Edition, Addison- Wesley, 2010.

M. Tech. – I Semester
(14MT10310) RESEARCH METHODOLOGY
(Common to all M. Tech. Programmes)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	3	--	--	3

PRE-REQUISITES: --

COURSE DESCRIPTION:

Fundamentals of research work - research problem and design; Data collection, Analysis and hypothesis; Statistics in Research; Interpretation and Report Writing.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to

1. Demonstrate knowledge on research approaches, research process and data collection.
2. Identify and analyze research problem.
3. Solve the research problems using statistical methods.
4. Carryout literature survey and apply good research methodologies for the development of scientific/ technological knowledge in one or more domains of engineering.
5. Learn, select and apply modern engineering tools to complex engineering activities.
6. Write effective research reports.

DETAILED SYLLABUS:

UNIT-I: INTRODUCTION TO RESEARCH METHODOLOGY (Periods:07)

Objectives and Motivation of Research, Types of Research, Research Approaches, Research Process, Criteria of good Research.

UNIT-II: RESEARCH PROBLEM AND DESIGN (Periods:09)

Defining and Formulating the Research Problem, Problem Selection, Necessity of Defining the Problem, Techniques involved in Defining a Problem. Features of Good Design, Research Design Concepts, Different Research Designs.

UNIT-III: DATA COLLECTION, ANALYSIS, AND HYPOTHESIS (Periods:09)

Different Methods of Data Collection, Processing Operations, Types of Analysis, Basic Concepts of Testing of Hypothesis, Hypothesis Testing Procedure.

UNIT-IV: STATISTICS IN RESEARCH**(Periods:09)**

Review of Statistical Techniques - Mean, Median, Mode, Geometric and Harmonic Mean, Standard Deviation, Measure of Asymmetry. Normal Distribution, Chi-Square Test as a Test of Goodness of Fit.

UNIT-V: INTERPRETATION AND REPORT WRITING**(Periods:06)**

Interpretation – Techniques and Precautions. Report Writing – Significance, Stages, Layout. Types of reports, Precautions in Writing Reports.

Total Periods: 40**TEXT BOOK:**

1. C.R. Kothari, *Research Methodology: Methods and Techniques*, New Age International Publishers, New Delhi, 2nd Revised Edition, 2004.

REFERENCE BOOKS:

1. Ranjit Kumar, *Research Methodology: A step-by-step guide for beginners*, Sage South Asia, 3rd ed., 2011.
2. R. Panneerselvam, *Research Methodology*, PHI learning Pvt. Ltd., 2009

M.Tech. (SE) I Semester

(14MT12521) DATA STRUCTURES AND WEB TECHNOLOGIES LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
25	50	75	-	-	4	2

PRE-REQUISITES: A course on "Object Oriented Programming".

COURSE DESCRIPTION:

Implementation of various linear and non-linear data structures using C++; Design and development of the interactive dynamic web sites using web technologies-HTML5, CSS3, JavaScript, JQuery, PHP and AJAX.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Understanding of:
 - Linear data structures: lists, stacks and queues
 - Non-linear data structures: trees and graphs
 - Dynamic programming and algorithms design techniques
 - Acquire practical knowledge on various web page design technologies like HTML5, CSS3, JavaScript, JQuery and AJAX.
2. Gain skills in developing effective and efficient algorithms and data structures to solve real-world problems and to develop web pages which are more dynamic, interactive, scalable and user friendly.
3. Design more efficient client-server applications using modern web design tools.

LIST OF EXPERIMENTS:

Data Structures:

Week-1: Write C++ programs to implement the following using an array.

- a) Stack ADT
- b) Queue ADT

Week-2: Write C++ programs to implement the following using a singly linked list.

- a) Stack ADT
- b) Queue ADT

Week-3: Write C++ programs to implement the circular queue ADT using a doubly linked list and an Array.

Week-4: Write C++ programs for the implementation of BFS and DFS for a given graph.

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

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

Week-6: 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-7: Write a C++ program to perform the following operations

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

Week-8: Write C++ programs for implementing the following using Divide and Conquer:

- a) Binary search
- b) Merge sort
- c) Quick sort

Week-9: Write C++ programs for implementing the following using Greedy method:

- a) Kruskal's algorithm
- b) Prim's algorithm

Week-10: Write a C++ program to implement following using Dynamic programming

- a) All Pairs Shortest Path.
- b) Single Source Shortest Path

Week-11: Write a C++ program for implementing following

- a) 8 Queen's problem using back tracking
- b) 0/1 Knapsack problem using Branch and Bound

Web Technologies:

1. Design a web page with the following features using HTML5 and CSS3
 - a. Displaying of images with Custom animated effects
 - b. Playing of selected video from the list of loaded videos
2.
 - a. Design a web page to store user information like user name and mobile number using local storage concept.
 - b. Design a web page to store employee information including Name, ID, Department, Salary and Address on a client's machine using a real SQL database.
3. Design a web page that:
 - a. Shows the animated text in increasing and decreasing font size
 - b. Changes the size of the area in a web page using DIV tag
 - c. Hides and Shows elements on web page.
4. Design a web page that allows the user to edit the content of a web page element and to change the position of elements using drag and drop features of HTML5.
5. Design a PHP page that reads User Name and Favorite Color from the html form and then displays user name in green color and sets user favorite color as a background color to the web page.
6. Write a PHP code to read the user id and password entered in the Login form and authenticate with the values (user id and passwords) available in the cookie. If he/she is a valid user (i.e., user id and password match) you should welcome him/her by user id otherwise you should display a message stating that he/she is not an authorized user. **Note:** Assume four user id's user1, user2, user3 and user4 and their passwords pwd1, pwd2, pwd3 and pwd4 respectively. Create four cookies on four user id's and passwords and then validate.
7.
 - a) Design a PHP page that reads user details from the registration form and store user information into a MySQL database.
 - b) Write a PHP code to authenticate users by reading user id and password from the Login form. Compare User id and password values with user id's and passwords stored at database. If he/she is a valid user (i.e., user id and password match) you should welcome him/her by name (first name + last name), otherwise make that as unauthorized access.

8. Design a web page to update some part of the web page content using XML
Http Request object.

REFERENCE BOOKS:

1. Sartaj Sahni "*Data structures, Algorithms and Applications in C++*," 2nd Edition, Universities press (India) Pvt. Ltd, 2005
2. Adam Drozdek "*Data Structures and Algorithms in C++*," 4th Edition, Delmar Cengage Learning, 2012.
3. Kogent Learning Solutions Inc, "*HTML 5 Black Book: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP and JQuery*," 1st Edition, Dreamtech Press, 2011.
4. W. Jason Gilmore, "*Beginning PHP and MySQL*," 4th Edition, APress, 2011.

M.Tech. (SE) II Semester
(14MT22501) SERVICE ORIENTED ARCHITECTURE

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: Courses on "Software Engineering" and "Web Technologies".

COURSE DESCRIPTION:

Introduction to SOA, Web services & Primitive SOA; WS extensions; Principles of SOA, Service Layers, Delivery strategies; Service Modeling; Service and Business process design- Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Web Services- Business Process Execution Language (WS-BPEL).

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge on:
 - Fundamental of web services.
 - Principles, services and policies of service orientation.
2. Analyze complex business process critically in identifying appropriate service model logic.
3. Gain skills on Technologies: XML, WSDL, BPEL related to SOA.

DETAILED SYLLABUS

UNIT-I: SOA AND WEB SERVICES FUNDAMENTALS (Periods:10)

Introducing SOA: Fundamental SOA, Common Characteristics of Contemporary SOA, Common tangible benefits of SOA, Common pitfalls of adopting SOA.

Web Services and Primitive SOA: The Web Services frame work, Services, Service descriptions, Messaging.

UNIT-II: SOA AND WS EXTENSIONS (Periods:12)

Web Services and Contemporary SOA (Part-I Activity Management and Composition): Message Exchange Patterns, Service Activity, Coordination, Atomic transactions, Business Activities, Orchestration, Choreography.

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

UNIT-III: PRINCIPLES, SERVICE LAYERS AND PLANNING (Periods:12)

Principles of Service-Oriented: Anatomy of SOA, Common principles of Service Orientation, Service Orientation and Object Orientation.

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-IV: BUILDING SOA (ANALYSIS)**(Periods:09)**

Introduction: Introduction to Service Oriented Analysis, Benefits of a Business Centric SOA, Deriving Business Services.

Service Modeling: Service Modeling, Service Modeling guidelines, Classifying Service model logic, Contrasting Service modeling approaches.

UNIT-V: BUILDING SOA (DESIGN)**(Periods:13)**

Introduction: Introduction to Service-Oriented design, WSDL related XML Schema language basics, WSDL language basics, SOAP language basics, Service interface design tools.

Service Design: Service Design overview, Entity-centric business Service Design, Application Service Design, Task-centric business Service Design, Service Design guidelines.

Business Process Design: WS-BPEL language basics, WS- Coordination overview, Service Oriented Business process Design.

Total Periods:56**TEXT BOOK:**

1. Thomas Erl, "*Service-Oriented Architecture - Concepts, Technology and Design*," Pearson, 2008.

REFERENCE BOOKS:

1. Shankar Kambhampaty, "*Service Oriented Architecture for Enterprise and Cloud Applications*," Wiley- India, 2012
2. Eric Newcomer and Greg Lomow, "*Understanding SOA with Web Services*," Pearson Education, 2007.
3. M. Rosen et al., "*Applied SOA*," Wiley India Pvt. Ltd, 2009.

M.Tech. (SE) II Semester
(14MT22502) SOFTWARE ARCHITECTURE AND DESIGN
PATTERNS

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: Courses on “Software Development Methodologies,” and “Object Oriented Programming”.

COURSE DESCRIPTION:

Software Architecture and Architecture Business Cycle(ABC); Understanding the requirements, designing, documenting, and reconstructing software architectures; Evaluating the architectures and Moving From one System to Many; Fundamentals of Design Patterns, Creational, Structural and Behavioral Patterns.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge in:
 - i. Software architecture styles and business life cycle.
 - ii. Various design issues and patterns.
2. Analyze and identify architectural styles and patterns to solve software design problems.
3. Apply appropriate software pattern to solve problems in object oriented software design process.

DETAILED SYLLABUS

UNIT I: ENVISIONING ARCHITECTURE (Periods:12)

Introduction to Software Architecture (SA) and Architecture Business Cycle (ABC):

What is Software Architecture, What Software Architecture is and What it isn't, Architectural Structures and Views, What makes a good Architecture, Why is SA important, Where do Architectures come from, Software Processes and the ABC, Architectural Patterns, Reference Model and Reference Architecture.

Pattern System: What is Pattern System, Pattern Classification, Pattern Selection, Introduction to Architectural Patterns, Pipes & Filter, Model-View-Controller.

UNIT II: CREATING AN ARCHITECTURE (Periods:14)

Understanding the Requirements: Functionality and Architecture, Architecture and Quality Attributes (QA), System QAs, QA Scenarios in Practice, Business and Architecture Qualities, Achieving Qualities.

Designing the Architecture: Architecture in the Life Cycle, Attribute Driven Design (ADD).

Documenting Software Architectures: Uses of Architectural Documentation, Views. **Reconstructing Software Architectures:**

Information Extraction, Database Construction, View Fusion and Reconstruction.

UNIT III: ANALYZING ARCHITECTURES AND MOVING FROM ONE SYSTEM TO MANY (Periods:09)

Evaluating the Architecture: The ATAM, The CBAM.

Moving From one System to Many: Software Product Lines, Building Systems from off the shelf components, Software Architecture in the future.

UNIT IV: INTRODUCTION TO DESIGN PATTERN AND CREATIONAL PATTERNS (Periods:09)

Introduction to Design Patterns: What is Design Pattern(DP), Design Patterns in Smalltalk MVC, Describing DPs, The Catalog of DPs, Organizing the Catalog, How DP Solve Design Problem, How to Select and Use of a DP.

Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype.

UNIT V: STRUCTURAL AND BEHAVIORAL PATTERNS (Periods:11)

Structural Patterns: Adapter, Composite, Decorator, Flyweight.

Behavioral Patterns: Command, Iterator, Mediator, Observer, State.

Total Periods:55

TEXT BOOKS:

1. Len Bass, Paul Clements and Rick Kazman, "*Software Architecture in Practice*," 2nd Edition, Addison-Wesley, 2003.
2. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides "*Design Patterns: Elements of Reusable Object-Oriented Software*," Pearson Education, 1995.

REFERENCE BOOKS:

1. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad and Michael Stal, "*Pattern-Oriented Software Architecture: A System of Pattern*," Volume 1, John Wiley & Sons, 2001.
2. Mary Shaw and David Garlan, "*Software Architecture: Perspectives on an Emerging Discipline*," Prentice Hall, 1996.
3. Eric Freeman and Elisabeth Freeman, "*Head First Design patterns*," O'REILLY, 2004.
4. Richard N. Taylor, N. Medvidovic and Eric M. Dashofy, "*Software Architecture: Foundations, Theory, and Practice*," Wiley, 2006.

M.Tech. (SE) II Semester

(14MT22503) SOFTWARE MEASUREMENT AND METRICS

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: A course on "Software Engineering".

COURSE DESCRIPTION:

Software measurement theory; Models of software engineering measurement; Software products metrics, software process metrics; Measuring & management and Software quality metrics.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge in:
 - Data collection and measures.
 - Software quality.
2. Acquire skills in analyzing project implementation risks and make decisions appropriately to develop product.
3. Apply software metrics to measure the parameters of a project.

DETAILED SYLLABUS

UNIT-I: FUNDAMENTALS OF MEASUREMENT (Periods:10)

Measurement and Basics of Measurement - Measurement in Everyday Life, Measurement in Software Engineering, Scope of Software Metrics, Representational Theory of Measurement, Measurement and Models, Measurement Scales and Scale Types.

UNIT-II: ANALYSIS OF MEASUREMENT (Periods:11)

Goal-Based Frame Work for Software Measurement - Classifying Software Measures, Determining what to measure, Applying Frame Work, Software Measurement Validation

Software Metrics Data Collection - Good Data, Definition of Data, Collecting, Storing and Extracting Data.

UNIT-III: PRODUCT ATTRIBUTES (Periods:12)

Measuring Internal Product Attributes – Size - Aspects of software size, Length, Reuse, Functionality, Complexity.

Structure - Types of structural measures, Control-flow structure, Modularity and information flow attributes.

UNIT-IV: MEASUREMENT AND MANAGEMENT (Periods:11)

Measuring External Product Attributes: Modeling Software Quality, Measuring Aspects of Quality. Object-Oriented Metrics: Object-Oriented Concepts and Constructs, Design and Complexity metrics, Productivity Metrics, Quality and Quality Management Metrics, Lessons Learned for OO Projects.

UNIT-V: QUALITY METRICS (Periods:11)

Software Quality Metrics Overview - Product Quality Metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metrics Programs,

Collecting Software Engineering Data, Applying the Seven Basic Quality Tools in Software Development.

Total Periods:55

TEXT BOOKS:

1. Fenton, Pfleeger, "*Software Metrics*," 2nd Edition, Thomson, 2005.
2. Stephen H. Kan, "*Metrics and Models in Software Quality Engineering*," 2nd Edition, Addison Wesley, 2011.

REFERENCE BOOKS:

1. Linda M. Laird and Carol Brennan, "Software Measurement and Estimation - A Practical Approach," IEEE Computer Science Press and Wiley Inter Science, 2006.
2. C Ravindranath Pandian: "Software Metrics: A guide to Planning Analysis and Implementation," Auerbach Publications, 2005.

M.Tech. (SE) II Semester
(14MT22504) SOFTWARE SECURITY ENGINEERING

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: Courses on "Software Engineering," and "Network Security".

COURSE DESCRIPTION:

Importance of security in software; Requirements engineering for secure software; Secure software architecture & design, secure coding & testing; System assembly challenges; and Governance & managing.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge on security issues in:
 - Requirement Engineering
 - Architecture and Design
 - Coding and Testing
 - System Assembling
2. Analyze complex software projects to describe security risks and mitigation techniques.
3. Acquire skills to solve the different security risks arising at different levels of software development life cycle.

DETAILED SYLLABUS

UNIT-I: IMPORTANCE OF SECURITY IN SOFTWARE (Periods:12)

Security a Software Issue: Introduction, The problem, Software Assurance and Software Security, Threats to software security, Sources of software insecurity, Benefits of detecting software security defects early, Managing secure software development.

What Makes Software Secure: Introduction, Properties of Secure Software, Influencing the security properties of software, Asserting and specifying the desired security properties.

UNIT-II: REQUIREMENTS ENGINEERING (Periods:10)

Requirements Engineering for Secure Software: Introduction, Misuse and abuse cases, the SQUARE process Model, SQUARE sample outputs, Requirements elicitation, Requirements prioritization.

UNIT-III: SECURITY PRINCIPLES IN SDLC (Periods:12)

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.

Secure Coding and Testing: Introduction, Code analysis, Software Security testing, Security testing considerations throughout the SDLC.

UNIT-IV: SECURITY AND COMPLEXITY**(Periods:10)**

System Assembly Challenges: Introduction, Security failures, Functional and attacker perspectives for security analysis in web services and identity management, system complexity drivers and security.

UNIT-V: GOVERNANCE AND MANAGING**(Periods:12)**

Governance and Managing for more Secure Software: Introduction, Governance and security, adopting an enterprise software security framework, how much security is enough, Security and Project Management, Maturity of Practice.

Total Periods:56**TEXTBOOK:**

1. Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead, "*Security Engineering: A Guide for Project Managers*," Pearson Education, 2009.

REFERENCE BOOKS:

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

M.Tech. (SE) II Semester
(14MT22505) SOFTWARE TESTING

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: Courses on "Software Engineering," and "Software Measurements and Metrics".

COURSE DESCRIPTION:

Software testing principles, functional & structural testing; Integration, regression & internationalization testing; Ad-hoc testing, testing of object oriented systems, usability & accessibility testing; Software test management & metrics; Software test automation.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain in-depth knowledge in software testing methodologies, and test strategies.
2. Analyze contemporary issues in software testing and design best test strategies.
3. Acquire problem solving skills to ensure quality software development to meet the industry standards.

DETAILED SYLLABUS

UNIT-I: SOFTWARE TESTING CONCEPTS – I (Periods:12)

A Perspective on Testing: Definitions, Test cases, Identifying Test cases, Error & Fault Taxonomies, Levels of Testing.

Functional Testing: Boundary Value Testing - Boundary Value Analysis, Test case Examples, Random Testing, Guidelines for Boundary Value Testing. Equivalence Class Testing - Equivalence Classes, Test case Examples, Guidelines and Observations. Decision Table based Testing - Decision Tables, Test case Examples, Guidelines and Observations.

Structural Testing: Path Testing - DD-Paths, Test Coverage Metrics, Basis Path Testing, Guidelines and Observations. Dataflow Testing - Define/Use Testing & Examples, Slice based Testing & Examples, Guidelines and Observations.

UNIT-II: SOFTWARE TESTING CONCEPTS – II (Periods:12)

Integration Testing: Scenario Testing, Defect bash. System and acceptance testing - functional, non-functional testing. Performance testing, Regression Testing - Definition, Types, When & How to do regression testing, Best practices. Internationalization Testing - Introduction, Test Phases of Internationalization testing, and Enabling testing, Locale Testing, Language testing, Localization testing.

UNIT-III: SOFTWARE TESTING CONCEPTS – III (Periods:10)

Ad-hoc testing: Overview, Buddy, Pair, Exploratory, Iterative, Agile and Extreme Testings. Testing of Object-oriented systems - Introduction, Primer on

object oriented software, Differences in OO testing. Usability And Accessibility Testing - what is usability testing, approach to usability, when to do usability testing, how to achieve usability, quality factors for usability, accessibility testing, tools for usability.

UNIT-IV: SOFTWARE TEST MANAGEMENT AND METRICS (Periods:10)

Test planning, Test Management, Test Process and Reporting, Software Test matrices and Measurement: Type of Metrics, Project Metrics, Productivity Metrics, Progress Metrics, Release Metrics.

UNIT-V: SOFTWARE TEST AUTOMATION (Periods:13)

Test Automation: Scope of Automation, Design and Architecture of automation, Process Model for Automation. Load Runner, Selenium, QTP, RFT and RQM, Bugzilla.

Total Periods:57

TEXT BOOKS:

1. Paul C. Jorgensen, "Software Testing: A Craftman's Approach," 3rd Edition, Auerbach Publications, 2008.
2. Srinivasan Desikan and Gopalswami Ramesh "Software Testing: Principles and Practices," 1st Edition, Pearson Education, 2008.

REFERENCE BOOKS:

1. M. G. Limaye, "Software Testing: Principles and Techniques and Tools," Tata Mc Graw – Hill Education, 1st Edition, 2012.
2. Ilene Burnstein, "Practical Software Testing," Springer-Verilog International Edition, 2003.
3. Dr. K. V. K. K.Prasad, "Software Testing Tools," Dreamtech, 1st Edition, 2004.
4. The Bugzilla Guide - 4.4.2+ Release
(<http://www.bugzilla.org/docs/4.4/en/pdf/Bugzilla-Guide.pdf>)
5. Introduction to IBM Rational Functional Tester 6.1
(<http://www.ibm.com/developerworks/rational/library/04/r-3228/3228.html>)
6. Selenium-IDE — Selenium Documentation (<http://docs.seleniumhq.org/docs/>)

M.Tech. (SE) II Semester
(14MT22506) BIG DATA TECHNOLOGIES

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PREREQUISITES: A course on "Data Mining and Data Warehousing"

COURSE DESCRIPTION:

Fundamentals of Big Data; Data-parallel programming model- Hadoop, Hadoop I/O; MapReduce features, HDFS; Hive, HBase, Zookeeper; Sqoop and Case studies.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Acquire knowledge on Big Data storage, processing, querying and reporting.
2. Apply Big Data Technologies to solve real-world problems such as stock market, sentiments analysis and massively scalable architectures.
3. Gain skills on various Big Data tools: Sqoop, HBase, MapReduce and Mahout.

DETAILED SYLLABUS

UNIT-I: INTRODUCTION TO BIG DATA (Periods:12)

The Evolution of Big Data, What is Big Data, Why Big Data Matters, Big Data Sources, The Big Data Revolution, Security, Compliance, Auditing and Protection. Meet Hadoop: Data Storage and Analysis, Comparison with Other Systems, A Brief History of Hadoop, Apache Hadoop and the Hadoop Ecosystem. MapReduce: A Weather Dataset, Analyzing the Data with UNIX Tools, Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes. The Hadoop Distributed File system: The Design of HDFS, HDFS Concepts, The Command-Line Interface, Hadoop File systems, The Java Interface, Data Flow, Parallel Copying with distcp, Hadoop Archives.

UNIT-II: HADOOP I/O (Periods:12)

Data Integrity, Compression, Serialization, File-Based Data Structures. Developing a MapReduce Application: The Configuration API, Configuring the Development Environment, Writing a Unit Test, Running Locally on Test Data, Running on a Cluster, Tuning a Job, MapReduce Workflows. How MapReduce Works: Anatomy of a MapReduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution. MapReduce Types and Formats: MapReduce Types, Input Formats, Output Formats.

UNIT-III: MAPREDUCE FEATURES (Periods:12)

Counters, Sorting, Joins, Side Data Distribution, MapReduce Library Classes. Setting Up a Hadoop Cluster: Cluster Specification, Cluster Setup and Installation, SSH Configuration, Hadoop Configuration, YARN Configuration, Security, Benchmarking a Hadoop Cluster, Hadoop in the Cloud. Administering Hadoop: HDFS, Monitoring, Maintenance. Pig: Installing and Running Pig,

Comparison with Databases, Pig Latin, User-Defined Functions, Data Processing Operators, Pig in Practice.

UNIT-IV: HIVE, HBase, ZOOKEEPER (Periods:11)

Installing Hive, Running Hive, Comparison with Traditional Databases, HiveQL, Tables, Querying Data, User - Defined Functions. HBase: HBasics, Concepts, Installation, Clients, HBase versus RDBMS, Praxis. ZooKeeper: Installing and Running ZooKeeper, the ZooKeeper Service, Building Applications with ZooKeeper, ZooKeeper in Production.

UNIT-V: SQOOP AND CASE STUDIES (Periods:12)

Getting Sqoop, Generated Code, Database Imports: A Deeper Look, Working with Imported Data, Importing Large Objects, Performing an Export, Exports: A Deeper Look. Case Studies: Best Practices for Big Data Analytics, Hadoop Usage at Last.fm, Hadoop and Hive at Facebook, Nutch Search Engine, Log Processing at Rackspace, Cascading, and TeraByte Sort on Apache Hadoop, Using Pig and Wukong to Explore Billion-edge Network Graphs.

Total Periods:59

TEXT BOOKS:

1. Tom White, "*Hadoop: The Definitive Guide*," O'Reilly and Yahoo Press, 3rd Edition, 2012.

REFERENCE BOOKS:

1. Frank J. Ohlhorst, "*Big Data Analytics: Turning Big Data into Big Money*," Wiley Publication, December 2012.
2. Kevin Roebuck, "*Big Data: High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*," Tebbo Publisher, 2011.
3. Alex Holmes, "*Hadoop in Practice*," Manning Publications Publisher, 2012.

M. Tech. (SE)-I Semester
(14MT22507) SOFTWARE RELIABILITY

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: Courses on "Software Engineering" and "Probability and Statistics".

COURSE DESCRIPTION:

Software reliability engineering process, developing operational profiles; Software reliability strategies, failures, faults & errors, availability; Software reliability modeling; Software metrics for reliability assessment; Best practice of software reliability engineering, and neural networks for software reliability.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge on:
 - Concepts of Software and Hardware Reliability.
 - Software Reliability Modeling.
 - Software metrics for Reliability Assessment.
 - Software Reliability Estimation.
 - Best practices of Software Reliability Engineering.
2. Analyze software system failures and develop convincing solutions.
3. Apply problem solving skills in producing failure free software intensive system.

DETAILED SYLLABUS

UNIT-I: INTRODUCTION AND THE OPERATIONAL PROFILE (Periods:12)

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.

The Operational Profile: Reliability concepts, software reliability and hardware reliability, developing operational profiles, applying operational profiles, learning operations and run concepts.

UNIT-II: SOFTWARE RELIABILITY CONCEPTS (Periods:10)

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 & failure intensities, predicting basic failure intensity.

UNIT- III: SOFTWARE RELIABILITY MODELING SURVEY (Periods:10)

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-IV: RELIABILITY ASSESSMENT AND SOFTWARE TESTING (Periods:12)

Software Metrics for Reliability Assessment: Introduction, Static Program Complexity, Dynamic Program Complexity, Software Complexity and Software Quality, Software Reliability Modeling.

Software Testing: Introduction, Overview of Software Testing, Operational Profiles, Time/Structure based Software Reliability Estimation.

UNIT-V: BEST PRACTISES AND NEURAL NETWORKS (Periods:12)

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

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

TEXT BOOKS:

1. Michael R. Lyu, "*Handbook of Software Reliability Engineering*," IEEE Computer Society Press, McGraw-Hill Book Company, 2005.
2. John D. Musa, "*Software Reliability Engineering*," 2nd Edition, Tata McGraw-Hill, 2011.

REFERENCE BOOKS:

1. Patric D. T. O Connor, "*Practical Reliability Engineering*," 4th Edition, John Wesley & Sons, 2003.
2. Anderson and PA Lee, "*Fault tolerance principles and Practice*," PHI, 1981.

M.Tech. (SE) II Semester
(14MT22508) SOFTWARE REVERSE ENGINEERING

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: Courses on "Software Engineering," and "Object Oriented Programming".

COURSE DESCRIPTION:

Fundamentals of software reverse engineering, object flow graph, low level software; reversing Tools; Locating undocumented APIs; UML diagrams; Reversing malware, and Anti reversing techniques.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain Knowledge on
 - Finding malicious code.
 - Discovering unexpected flaws and faults.
 - Finding the use of others code.
 - Learning from others products of a different domain or purpose.

2. Analyze well designed Software System and discover new opportunities for innovation.

DETAILED SYLLABUS

UNIT-I: FOUNDATIONS

(Periods:13)

Need for Reverse Engineering, Software Reverse Engineering, Reverse Applications, Low Level Software, The Reversing Process, The Tools, Is Reversing Legal and Code Samples Tools.

Object Flow Graph: Abstract Language, Object Flow Graph, Containers, Flow Propagation Algorithm, Object Sensitivity, The elib Program.

Low Level Software: High Level Perspectives, Low Level Perspectives, Assembly Language, A Primer on Compilers and Compilation, Execution Environments.

UNIT-II: REVERSING TOOLS

(Periods:07)

Different Reversing Approaches, Disassemblers, Debuggers, Decompilers, System-Monitoring Tools, Patching Tools, Miscellaneous Reversing Tools.

UNIT-III: BEYOND THE DOCUMENTATION

(Periods:08)

Reversing and Interoperability, Laying the Ground Rules, Locating Undocumented APIs, Case Study.

UNIT-IV: CLASS, OBJECT, INTERACTION AND STATE DIAGRAMS

(Periods:14)

Class Diagram: Recovery, Declared Vs Actual Types, Containers, The elib Program.

Object Diagram: Introduction, Object Sensitivity, Dynamic Analysis, The elib Program.

Interaction Diagram: Introduction, Interaction Diagram Recovery, Dynamic Analysis, The elib Program.

State Diagram: Introduction, Abstract Interpretation, State Diagram Recovery, The elib Program.

UNIT-V: PACKAGE DIAGRAM, REVERSING AND ANTI REVERSING TECHNIQUES (Periods:16)

Package Diagram: Recovery, Clustering, Concept Analysis, The elib Program, Tool Architecture, The elib Program, Perspectives.

Reversing Malware: Types of malware, Sticky software, Future malware, Uses of malware, Malware vulnerability, Polymorphism, Metamorphism, establishing a secure environment.

Anti Reversing Techniques : Introduction to Anti reversing, Basic approaches to anti reversing, Eliminating symbolic information, Code encryption, Active anti debugger techniques, Confusing Disassemblers, Code obfuscation, Control flow transformations, Data transformations.

Total Periods:58

TEXT BOOKS:

1. Paolo Tonella & Alessandra Potrich, "Reverse Engineering of Object Oriented Code," Springer-2005.
2. Eldad Eilam, "Reversing: Secrets of Reverse Engineering," Wiley, 2005.

M.Tech. (SE) II Semester
(14MT22509) SOFTWARE REUSE

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	4	-	-	4

PRE-REQUISITES: Courses on "Software Engineering," and "Object Oriented Concepts".

COURSE DESCRIPTION:

Software reuse success factors, reuse driven software engineering; Architectural styles, architectural elements; Processes- Object Oriented Business Engineering; Organizing a reuse business- Transition to a reuse business, managing the reuse business, and business engineering.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Gain knowledge on:

- Component System Engineering
- Application System Engineering
- Application Family Engineering
- Managing reuse

2. Analyze complex legacy software systems to identify reusable components.

DETAILED SYLLABUS

UNIT-I: INTRODUCING THE REUSE DRIVEN SOFTWARE ENGINEERING BUSINESS (Periods:08)

Software Reuse success factors: Software reuse a simple idea, A systematic approach makes pragmatic reuse work, Reuse requires changes in process, Reuse requires changes in organization, Adopting Reuse, Set of principles.

Reuse Driven Software Engineering is a Business: Make Reuse cost effective, Reuse business characteristics, Architect Components and Applications, Software Engineering processes, Establishing and Managing a Reuse business.

UNIT-II: ARCHITECTURAL STYLE-I (Periods:12)

Object Oriented Software Engineering, Application and Component System, Use case Components.

UNIT-III: ARCHITECTURAL STYLE-II (Periods:12)

Object Components: Object models define system architecture and design, Reusing Analysis and design components, Expressing variability in object model components, Tracing use case variability to the object model, Reusable analysis components, Reusable design and implementation components, Layered Architecture.

UNIT-IV: PROCESSES (Periods:13)

Object Oriented Business Engineering, Applying Business Engineering to Define Processes and Organization, Application Family Engineering

UNIT-V: ORGANIZING A REUSE BUSINESS**(Periods:12)**

Transition to a Reuse Business, Managing the Reuse Business, Making the Reuse Business work: Reuse improves the performance of business processes, misconceptions, Doing reuse is difficult, Reuse depends on architecture, Business engineering.

Total Periods:57**TEXT BOOK:**

1. Ivar Jacobson, Martin Gress, Patrick Johnson, "*Software Reuse*," Pearson Education, 2004.

REFERENCE BOOKS:

1. Eve-Andre Karisson, "*Software Reuse – A Holistic Approach*," John Wiley and Sons, 1996.
2. Karma McClure, "*Software Reuse Techniques – Additional reuse to the systems development process*," Prentice Hall, 1997.

M.Tech. (SE) II Semester
(14MT22521) SOFTWARE DEVELOPMENT AND TESTING LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
25	50	75	-	-	4	2

PRE-REQUISITES: A course on "Object Oriented Programming".

COURSE DESCRIPTION:

Hands-on experience in software development life cycle activities-requirements specification using RequisitePro tool, modeling using AgroUML tool; Implementation of design patterns; Refactoring using InsRefactor and SafeRefactor Eclipse Plugins.; Test plan document; Regression testing, functional testing using QTP, RFT and Selenium; Performance testing using Load Runner, RPT and Web Performance Tool.

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

1. Acquire software testing knowledge and experience by handling the design and conduct software testing processes.
2. Identify key entities and relationships in the problem domain and write succinct textual descriptions of problems in the style of a user manual.
3. Apply various testing phases and automate while testing the application and apply Software Engineering concepts and practices in
 - i. Identify customer's needs.
 - ii. Evaluate system for feasibility.
 - iii. Perform economic and technical analysis.
 - iv. Allocate functions to system elements.
 - v. Establish schedule, constraints and estimate cost.
 - vi. Create system definitions.

LIST OF EXPERIMENTS:

SOFTWARE DEVELOPMENT:

1. Prepare the Software Requirement Specification (SRS), High Level Design (HLD) and Detailed Design (DD) for the following experiments
 - (i) Employee Information System
 - (ii) Online Airline Reservation

Note: For the reference of SRS, HLD and DD templates refer department manual and use Requirement documentation tool (RequisitePro) in Rational Rose.

2. Estimation of Project Metrics for the above 1. (i) and (ii) using COCOMO-II
3. Study and prepare a report on the following tools:
 - (i) OSRMT
 - (ii) Microsoft Visio 2010
 - (iii) Enterprise Architect 7.5
4. Draw ER diagrams and generate SQL statements for the above experiments using ER Master plugin in Eclipse Helios
5. Model UML Usecase, Sequence, Collaboration and Component diagrams for the following experiments using AgroUML tool
 - (i) Students Marks Analyzing System
 - (ii) Course Registration System

6. Write the java program for the following design patterns
(i) Abstract Factory (ii) Decorator
7. Mini Project on Library Systems using Refactoring

Note: Use InsRefactor and SafeRefactor Eclipse Plugins

SOFTWARE TESTING:

1. Create a test plan document for a Desktop based application
2. Write the Functional test cases for a Desktop based application
3. Write the Regression test cases for a Desktop based application
4. Conduct Functional Testing for a Desktop based application using QTP
5. Conduct Functional Testing for a Desktop based application using RFT
6. Conduct Performance Testing for a Desktop based application using Load Runner
7. Conduct Performance Testing for a Desktop based application using RPT
8. Create a test plan document for a Web based application
9. Write the Functional test cases for a Web based application
10. Write the Regression test cases for a Web based application
11. Conduct Functional Testing using Selenium for a Web based application
12. Conduct Performance and Load testing for a Web based application using Web Performance Tool

REFERENCE BOOKS:

1. Ilene Burnstein, "*Practical Software Testing*," Springer-Verilog International Edition, 2003.
2. Dr. K. V. K. K.Prasad, "*Software Testing Tools*," Dreamtech, 1st Edition, 2004.
3. Introduction to IBM Rational Functional Tester 6.1
(<http://www.ibm.com/developerworks/rational/library/04/r-3228/3228.html>)
4. Selenium-IDE — Selenium Documentation
(<http://docs.seleniumhq.org/docs/>)
5. Roger S. Pressman, "*Software Engineering, A practitioner's Approach*," 6th Edition, Tata McGraw-Hill, Edition, 2010.
6. Sommerville, "*Software Engineering*," 8th Edition, Pearson Education, 2007.
7. James W.Cooper, "Java Design Patterns- A Tutorial," Pearson Education, 2000
8. Rajesh Naik and Swapna Kishore, "*Software Requirements and Estimation*," Tata McGraw Hill, New Delhi, 2001.

M. Tech. (SE) – II Semester
(14MT22522) SEMINAR

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
--	50	50	--	--	--	2

PRE-REQUISITES: --

COURSE DESCRIPTION:

Identification of seminar topic; Literature survey; Preparation of technical report and Presentation.

COURSE OUTCOMES:

On successful completion of the seminar work, the student will be able to:

1. Acquire in-depth knowledge in core and allied areas of interest.
2. Analyze and synthesize information related to the areas.
3. Conceptualize and construct research problems.
4. Extract information pertinent to a specific area through literature survey to conduct research.
5. Identify the applicability of modern software and tools.
6. Contribute positively to multidisciplinary groups in emerging areas with objectivity and rational analysis.
7. Plan, organize, prepare and present effective written and oral technical reports.
8. Develop independent and reflective learning.
9. Acquire awareness on professional code of conduct in the chosen area.
10. Develop independent and reflective learning.

M. Tech. (SE) – III & IV Semesters
(14MT32521& 14MT42521) PROJECT WORK

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
80	120	200	--	--	--	16

PRE-REQUISITES: --

COURSE DESCRIPTION:

Identification of topic for the project work; Literature survey; Collection of preliminary data; Identification of implementation tools and methodologies; Performing critical study and analysis of the topic identified; Time and cost analysis; Implementation of the project work; Writing of thesis and Presentation.

COURSE OUTCOMES:

On successful completion of this project work, the student will be able to:

1. Acquire in-depth knowledge in the areas of interest.
2. Analyze critically chosen project topic for conducting research.
3. Apply knowledge gained through Program, self learning and experience for solution of a given problem efficiently.
4. Undertake research confidently in the project domain.
5. Use the techniques, skills and modern engineering tools necessary for project work.
6. Perform harmonically in multi-disciplinary, multi-cultural groups, and develop a high level of interpersonal skills.
7. Manage projects in respective disciplines and multidisciplinary environments with due consideration to cost and time efficiency.
8. Develop communication skills, both oral and written for preparing and presenting reports.
9. Engage in lifelong learning to improve knowledge and competence continuously.
10. Understand professional and ethical responsibility for sustainable development of society.
11. Develop independent and reflective learning.