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-19

The Courses (with course outcomes) under SVEC-19 Regulations which focus on ***employability/ entrepreneurship/ skill development*** are highlighted with the following colours.

Skill

Employability

Entrepreneurship

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12501) ADVANCED SOFTWARE ENGINEERING

(Software Engineering)

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

PREREQUISITES: Nil

COURSE DESCRIPTION: Software Life Cycle Models; Software Project Management; Software Design, Software Reliability And Software Quality Management; Software Maintenance, Software Reuse And Emerging Trends In Software; Devops.

COURSE OUTCOMES:

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

CO1: Use various Software Development Lifecycle Models in software development.

CO2: Identify project management approaches as well as cost and schedule estimation strategies for quality software development.

CO3: Apply software design approaches for development of software system.

CO4: Understand software reliability, quality, reuse, and maintenance concepts.

CO5: Choose DevOps Tools to accelerate software development and reduce defects.

DETAILED SYLLABUS:

UNIT-I: SOFTWARE LIFE CYCLE MODELS

Software Development Projects, Exploratory Style of Software Development, Emergence of Software Engineering, Notable Changes in Software Development Practices, Waterfall Model and its Extensions, Rapid Application Development, Agile Development Models, Spiral Model.

UNIT II: SOFTWARE PROJECT MANAGEMENT

Software Project Management Complexities, Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO, Halstead's Software Science—An Analytical Technique, Staffing Level Estimation, Scheduling, Organisation and Team Structures, Staffing, Software Configuration Management, Miscellaneous Plans.

UNIT III: SOFTWARE DESIGN, RELIABILITY AND QUALITY MANAGEMENT

Overview of the Design Process, Good Software Design, Cohesion and Coupling, Layered Arrangement of Modules, Approaches to Software Design, Software Reliability, Statistical Testing, Software Quality, Software Quality Management System, ISO 9000, SEI Capability Maturity Model, Six Sigma.

UNIT IV: SOFTWARE MAINTAINANCE, REUSE AND EMERGING TRENDS

Maintenance: Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Model, Estimation of Maintenance Cost.

Reuse: Need of Reuse, Basic Issues in any Reuse Program, A Reuse Approach, Reuse at Organization Level.

Emerging Trends: Client-Server Software, Client-server Architectures, CORBA, COM/DCOM, Service-Oriented Architecture, Software as a Service.

UNIT V: DevOps

DevOPs: Cloud as a Platform, Deployment Pipeline-Architecture, Building and Testing, Deployment, Case study- Supporting Multiple Datacenters.

TEXT BOOKS:

1. Rajib Mall, "*Fundamentals of Software Engineering",* 4th Edition, PHI, 2014*.*
2. Len Bass, Ingo Weber, Liming Zhu, "*DevOps: A Software Architect's Perspective*",1st Edition, Pearson Education. 2016.

REFERENCE BOOKS:

1. Roger S. Pressman, "*Software Engineering*", 6th Edition, McGraw Hill, 2005.
2. Carlo Ghezzi, Mehdi Jazayeri, and Dino Madrioli, "Fundamentals of Software Engineering", 2nd Edition, PHI Learning, 2010.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12502) DATA STRUCTURES AND ALGORITHMS

(Software Engineering)

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

PREREQUISITES: A course on "Computer Programming".

COURSE DESCRIPTION: Introduction to Data Structures and Algorithms; Searching and Sorting; Trees and Graphs; Divide and Conquer; Greedy method; Dynamic Programming, Back Tracking; Branch and Bound.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate knowledge on:

- Linear data structures including Stack, Queue and Linked Lists and Non-linear data structures like Trees and Graphs.
- Divide and Conquer, Greedy Method, Dynamic Programming, Backtracking and Branch & Bound algorithms.

CO2: Analyze the efficiency of algorithms using space and time complexities.

CO3: Apply Algorithm design techniques for searching and sorting applications.

CO4: Choose data structures for optimum way of data organization and retrieval mechanism.

CO5: Use Dynamic programming techniques to optimize shortest path finding.

DETAILED SYLLABUS

UNIT-I: Introduction to Data Structures and Algorithms

Data Structures: Review of Data Structures - Stack, Queue, Circular Queue, Linked Lists, Applications, **Algorithm Analysis:** Efficiency of algorithms, Apriori Analysis, Asymptotic Notations, Polynomial vs Exponential Algorithms, Average, Best and Worst Case Complexities, Analyzing Recursive Algorithms.

UNIT-II: Searching, Sorting and Trees & Graphs

Searching and Sorting: Linear Search, Fibonacci Search, Counting Sort, Bucket Sort, Radix Sort, **Trees and Graphs:** Introduction to trees, representation of trees, binary trees, binary tree traversal techniques, Introduction to graphs, representation of graphs, graph traversal techniques.

UNIT-III: Binary Search Trees, AVL Trees, B- Trees and Hash Tables

Binary Search Trees: Definition, Operations, Applications, AVL Trees: Definition, Operations, Applications, **Heaps:** Definition, Heap Implementation, Applications, **Hash Tables:** Definition, Hash Functions, Applications.

UNITI-IV: Divide and Conquer & Greedy Methods

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

Greedy Method: General Method, Job sequencing with deadlines, Minimum Cost Spanning Tree and Single Source Shortest Path.

UNIT-V: Dynamic Programming, Back Tracking & Branch and Bound

Dynamic Programming: General Method, All Pairs Shortest Path, 0/1 Knapsack problem, Traveling Salesperson Problem, **Back Tracking:** General Method, 8 – Queen's Problem, Graph Coloring, **Branch and Bound:** General Method, LC Search, LIFO and FIFO branch and bound solutions of 0/1 Knapsack Problem.

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. Richard Gileberg, Behrouz A. Forouzan*, "Data Structures: A Pseudocode Approach with C,"* Second Edition, 2007.
2. Mark Allen Weiss, *"Data Structures and Algorithm Analysis in C++,"* 3rd Edition, Pearson Education, 2007.

   Sartaj Sahni, *"Data structures, Algorithms and Applications in C++,"* 2nd Edition, Universities press (India) Pvt. Ltd., 2005.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12503) MOBILE APPLICATION DEVELOPMENT

(Software Engineering)

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

PRE-REQUISITES: Courses on "Java Programming".

COURSE DESCRIPTION: Introduction to Android Platform; Activities; Basic Views; Understanding the Components of a Mobile Screen; Display Orientations; Menus; File Storage; Database Storage; SMS; e-mail; Location-Based Services; Web Services using HTTP; Android Services; Communication between a Service and an Activity; Threading; Introduction to iOS.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1. Understand the programming concepts of Android and iOS for the development of mobile applications.

CO2. Analyze the states of Android activities, activity navigations, and screen orientations.

CO3. Design user interface for mobile applications using views and develop Android code to process the user data, exchange the data between activities and provide navigation among activities.

CO4. Apply Android programming principles to create files and databases for storing and retrieving of the user data.

CO5. Apply Android programming principles of messaging, location-based services (LBS), and networking to develop mobile applications like SMS, e-Mail, LBS, and web services.

DETAILED SYLLABUS:

UNIT - I: INTRODUCTION TO ANDROID PROGRAMMING

Introduction, Android versions, features of Android, architecture of Android, Required tools, launching of simple Android application, understanding activities, linking activities using intents, fragments.

UNIT - II: ANDROID USER INTERFACE

Understanding the components of a screen, adapting to display orientation, managing changes to screen orientation, utilizing the action bar, creating the user interface programmatically, listening for UI notifications.

UNIT - III: USER INTERFACE DESIGN WITH VIEWS

Basic views, picker views, list views, image views, menus with views, web view, saving and loading user preferences, persisting data to files, creating and using databases.

UNIT - IV: MESSAGING, LOCATION-BASED SERVICES, AND NETWORKING

SMS messaging, sending e-mail, displaying maps, getting location data, monitoring a Location, consuming web services using HTTP.

UNIT - V: ANDROID SERVICES AND iOS

Creation of Android services, communication between a service and an activity, binding activities to services, understanding threading.
iOS Tools, iOS project, debugging iOS apps, objective-C basics, Hello World app, building the derby app in iOS.

TEXT BOOKS:

1. J. F. DiMarzio, "*Beginning Android Programming with Android Studio*", Wiley India, Fourth Edition, 2017.
2. Jeff McWherter, Scott Gowell, *"Professional Mobile Application Development",* Wiley India, First Edition, 2012.

REFERENCE BOOKS:

1. Neils Smyth "*Android Stduio Development Essentials*", Creative Space Independent publishing platform, Seventh edition 2016.
2. Paul Deital and Harvey Deital, "*Android How to Program*",Detial associates publishers, First Edition, 2013.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M. Tech. I Semester

(19MT16303) CLOUD COMPUTING

(Program Elective – 1)

(Common to CN&IS, CS and SE)

Internal Marks	External Marks	Total Marks	L	T	P	C
40	60	100	3	-	-	3

PRE-REQUISITES: Courses on Computer Networks and Operating Systems.

COURSE DESCRIPTION:

Characteristics and taxonomy of virtualization techniques, Cloud services, Cloud architecture - NIST and other models, communication protocols, and applications, Cloud programming concepts - concurrent programming, task programming, data intensive computing,; Trends and industrial platforms.

COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1: Create virtual environments to deploy cloud services by using the concepts of virtualization and cloud computing.

CO2: Analyze and design cloud architectures for providing cloud services to cater needs of diverse applications.

CO3: Develop skills to research and develop cloud applications that ae resilient, elastic and cost-efficient.

DETAILED SYLLABUS:

UNIT I - INTRODUCTION TO VIRTUALIZATION

Characteristics of virtualized environments, Taxonomy of virtualization techniques, Virtualization and cloud computing, Pros and cons of virtualization, Technology examples - XEN, VMware, Microsoft Hyper-V.

UNIT II - FUNDAMENTAL CLOUD COMPUTING

Understanding Cloud Computing - Origins and influences, Basic concepts and terminology, Goals and benefits, Risks and challenges; Fundamental concepts and models - Roles and boundaries, Cloud characteristics, Cloud delivery models, Cloud deployment models; Cloud enabling technology - Broadband networks and Internet architecture, Data center technology; Fundamental cloud security - Basic terms and concepts, Threat agents, Cloud security threats, Additional considerations.

UNIT III - DEFINING CLOUD SERVICES

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) - Introduction to identity, Networked identity service classes, Identity system codes of conduct, IDaaS interoperability; Defining Compliance as a Service (CaaS).

UNIT IV - CLOUD PROGRAMMING CONCEPTS

Concurrent programming - Introduction to parallelism for single machine computation, Programming applications with threads; High throughput computing - Task programming, Task based application models; Data intensive computing - Introduction to data intensive computing and technologies for data intensive computing.

UNIT V – INDUSTRIAL PLATFORMS AND TRENDING DEVELOPMENTS

Case Studies on Cloud Platforms: Software as a Service (SaaS) - Salesforce.com; Platform as a Service (PaaS) - Google App Engine, MS-Azure and IBM Bluemix; Infrastructure as a Service (IaaS) - Amazon EC2, Amazon S3 and Netflix; Enhancements in cloud - Energy efficiency in clouds, Market based management of clouds, Federated clouds / InterCloud, Third party cloud services.

TEXT BOOKS:

1. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, Morgan Kaufmann, 1st Edition, 2013.
2. Thomas Erl, Zaigham Mahmood, Ricardo Puttini, *Cloud Computing: Concepts, Technology & Architecture*, PHI, 1st Edition, 2013.

REFERENCE BOOKS:

1. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, *Cloud Computing: A Practical Approach*, McGraw Hill, 1st Edition, 2010.
2. George Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*, O'Reilly, 1st Edition, 2009.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12504) INFORMATION RETRIEVAL SYSTEMS

Program Elective-I

(Software Engineering)

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

PREREQUISITES: A course on Database Management Systems

COURSE DESCRIPTION: Architecture of Information Retrieval Systems; Functional Capabilities; Data Structures; Mathematical Algorithms; Indexing; Similarity and Clustering; Human Perception and Presentation; Text Search Techniques and Evaluation Measures.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate knowledge on different information retrieval techniques.

CO2: Use different indexing and data presentation methods.

CO3: Analyze clustering algorithms to group similar data items and text search techniques for efficient search.

CO4: Demonstrate Human Perception and Presentation Techniques in information retrieval.

CO5: Choose different search and evaluation methods for information retrieval mechanism.

DETAILED SYLLABUS:

UNIT - I: INTRODUCTION

Primary Information Retrieval Problems, Objectives of Information Retrieval System, Functional Overview, Understanding the Search Functions, Relationship to DBMS, Digital libraries and Data Warehouses, Data structures and Mathematical Algorithms.

UNIT - II: INGEST AND INDEXING

Introduction, Item Receipt, Duplicate Detection, Item Normalization, Zoning and Creation of Processing Tokens, Stemming, Entity Processing, Categorization, Citational Metadata, Manual Indexing Process, Automatic Indexing of Text and Multimedia.

UNIT - III: SEARCH AND CLUSTERING

Similarity measures and Ranking, Hidden Markov Models Techniques, Ranking Algorithms, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches for Boolean Systems, Multimedia Searching, Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters.

UNIT - IV: INFORMATION PRESENTATION

Introduction, Presentation of the Hits, Display of the Item, Collaborative Filtering, Multimedia Presentation, Human Perception and Presentation.

UNIT - V: SEARCH ARCHITECTURE AND EVALUATION

Index Search Optimization, Text Search Optimization, GOOGLE Scalable multiprocessor architecture, Information System Evaluation, Measures used in system evaluation

TEXT BOOK:

1. Gerald Kowalski, "Information Retrieval Architecture and Algorithms," Springer, First Edition, 2013.

REFERENCE BOOKS:

2. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "*An Introduction to Information Retrieval,*" Cambridge University Press, First Edition, 2012.
3. Ricardo Baeza-Yates and Berthier Ribiero-Neto, "*Modern Information Retrieval the concepts and technology behind search,*" Addison Wesley, Second Edition, 2010

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester
(19MT12505) PYTHON PROGRAMMING
Program Elective-I
(Software Engineering)

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

PRE-REQUISITES: --Nil--

COURSE DESCRIPTION: Visual Programming through Scratch; Introduction to Python Programming and Control Structures; Python Data Structures, Strings, Functions and Files; Introduction to Object Oriented Programming and Python modules.

COURSE OUTCOMES:

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

CO1: Understand the basic concepts of visual programming through Scratch tool and Python variables, operators, control structures, data structures, strings, functions and files.

CO2: Develop basic scripts for performing Input, Output and Computations using Scratch tool and Python programming.

CO3: Identify the strengths and weaknesses of different Scratch Palettes and Python data structures.

CO4: Design and deploy appropriate data structures, string functions and Python modules for solving computing problems.

CO5: Develop Python Programs for solving real time applications using object oriented concepts and Python modules.

DETAILED SYLLABUS:

UNIT I – VISUAL PROGRAMMING THROUGH SCRATCH

Scratch: Scratch Programming Environment, Paint Editor, First Scratch Game, Scratch Blocks, Arithmetic Operators and Functions; Motion and Drawing- Motion Commands, Pen Commands, The Power of Repeat; Looks and Sound- The Looks Palette, The Sound Palette; Procedures- Message Broadcasting and Receiving, Creating Large Programs in Small Steps, Working with Procedures; Variables- Data Types in Scratch, Introduction to Variables, Variable Monitors in Applications, Getting Input from Users; Making Decisions- Comparison Operators, Decision Structures, Logical Operators; Repetition- Loop Blocks, Stop Commands, Counters.

UNIT II – INTRODUCTION TO PYTHON PROGRAMMING AND CONTROL STRUCTURES

Introduction: The Python programming language, First program in Python, Literals, Variables and Identifiers, Operators, Expressions, Case study on Restaurant Tab Calculation.

Control Structures: Control structures, Boolean expressions, Selection control and Iterative control, importing modules.

UNIT III – PYTHON DATA STRUCTURES

Lists: Numbers, List structures, Lists in Python, Iterations over lists, Assigning and copying lists, List comprehensions, Case studies-Chinese Zodia and Password Encryption/Decryption,

Dictionaries, Tuples and Sets: Dictionary types in Python, Implementation of Dictionary, Tuples, Set data type - the Set data type in Python.

UNIT IV – STRINGS, FUNCTIONS AND FILES

Strings: String Processing- String Traversal, String-Applicable Sequence Operations and String Methods.

Functions: Program Routines, More on Functions, Case Studies-Temperature Conversion and Credit Card Calculation.

Files: Text File, Opening Text Files, Reading and Writing Text Files.

UNIT V –OBJECT ORIENTED PROGRAMMING AND PYTHON MODULES

Introduction to Object Oriented Programming: Object-oriented programming, User-defined compound data types, Attributes, Methods, Instances as arguments, parameters and return values, PyGame.

Python Modules: Numpy, Pandas, Openpyxl, PyPDF2 and CSV.

TEXT BOOKS:

1. Focus Charles Dierbach, *"Introduction to Computer Science using Python: A Computational Problem Solving"*, Wiley India Edition, 2016.
2. Games, Art, Science, and Math, Majed Marji, *"Learn to Program with Scratch: A Visual Introduction to Programming"* No Starch Press, Inc., 2014.

REFERENCE BOOKS:

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, "How to Think Like a Computer Scientist: Learning with Python-3 Documentation", 3rd Edition, 2012, "Numpy and Pandas Tutorial",.
2. Automate the Boring Stuff with Python, Al Sweigart, No Starch Press, Inc., 2014.
3. "Learn to Program with Scratch", https://nostarch.com/learnscratch/#content.
4. Fundamentals of Python, Kenneth Lambert and B.L. Juneja, , Cengage Learning, 3rd Edition, 2012.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester
(19MT12506) USER INTERFACE DESIGN
(Software Engineering)

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

PREREQUISITES: A course on "Advanced 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:

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

CO1: Gain the knowledge on concepts of user interfaces and related business functions.

CO2: Analyze user requirements necessary for UI development.

CO3: Design interfaces using appropriate menus, windows, interfaces.

CO4: Usage and customize of advanced tools for various window layouts in project management and development of UI computing systems.

UNIT I: FOUNDATIONS OF HCI

The Human: I/O channels – Memory – Reasoning and problem solving; The Computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms. – Case Studies

UNIT II: DESIGN & SOFTWARE PROCESS

Interactive Design: Basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process: Software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules: principles, standards, guidelines, rules. Evaluation Techniques – Universal Design

UNIT III: MODELS AND THEORIES

HCI Models: Cognitive models: Socio-Organizational issues and stakeholder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.

UNIT IV: MOBILE HCI

Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools. – Case Studies.

UNIT V: WEB INTERFACE DESIGN

Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow – Case Studies

TEXT BOOKS:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "*Human Computer Interaction*", 3rd Edition, Pearson Education, 2004 (UNIT I , II & III).
2. Brian Fling, "*Mobile Design and Development*", First Edition, O'Reilly Media Inc., 2009 (UNIT –IV).
3. Bill Scott and Theresa Neil, "*Designing Web Interfaces*", First Edition, O'Reilly, 2009. (UNIT-V).

REFERENCE BOOK:

1. Alan Cooper, "*The Essential of User Interface Design*", Wiley – Dream Tech Ltd.,2002.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12507) BIG DATA TECHNOLOGIES

(Software Engineering)

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

PRE-REQUISITES: Courses on Database Management Systems and Data Warehousing and Data Mining.

COURSE DESCRIPTION: Introduction to Big Data, Hadoop; Hadoop Distributed File Systems; Hadoop I/O; MapReduce; Hive; Pig; HBase; Zookeeper; Sqoop and Case studies.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate knowledge on Structure of Big Data, Big Data characteristics, storage, processing, querying and reporting.

CO2: Analyze large dataset issues with Hadoop Distributed File System and MapReduce techniques.

CO3: Use MapReduce features and file types and formats in Hadoop environment.

CO4: Apply MapReduce, Hive, Pig, Sqoop, HBase, and Zookeeper tools for data analytics.

CO5: Design and Develop classification and clustering models for health and facebook dataset applications.

DETAILED SYLLABUS:

UNIT I - INTRODUCTION TO BIG DATA AND HADOOP

Introduction to Big Data: Evolution and Definition of Big Data, Structure of Big Data, Characteristics, Advantages, Applications and Tools.

Hadoop: Data storage and analysis, Comparison with other systems, History of Hadoop, Apache Hadoop and the Hadoop Ecosystem, Hadoop Releases.

UNIT II – HADOOP DISTRIBUTED FILE SYSTEM AND HADOOP I/O

Hadoop Distributed File system: HDFS concepts, Command-Line Interface, Hadoop file systems, Java Interface, Data Flow, Hadoop Archives.

Hadoop I/O: Data Integrity, Compression, Serialization, File-Based Data Structures.

UNIT III – MAPREDUCE, TYPES & FORMATS AND FEATURES

MapReduce: Analyzing the data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes.

Types and Formats: MapReduce Types, Input Formats, Output Formats.

Features: Counters, Sorting, Joins, Side Data Distribution and MapReduce Library Classes.

UNIT IV – HIVE, PIG AND HBASE

Hive: Comparison with Traditional Databases, HiveQL, Tables, Querying Data, and User-Defined Functions.
Pig: Comparison with Databases, Pig Latin, User-Defined Functions, Data Processing Operators.
HBase: Basics, Concepts, Clients, HBase vs. RDBMS, Praxis.

UNIT V – ZOOKEEPER, SQOOP AND CASE STUDIES

Zookeeper: Zookeeper Service, Building applications with Zookeeper, Zookeeper in production.

Sqoop: Database Imports, working with Imported Data, Importing Large Objects, Performing an Export.

Case Studies: Mahout, WhatsApp, Healthcare, Facebook and Twitter.

TEXT BOOKS:

1. Tom White, *"Hadoop: The Definitive Guide,"* Third Edition, O'REILLY Publications, 2012.
2. Anil Maheswari, **"***Big Data***,"** Tata McGraw Hill, First Edition, 2017.

REFERENCE BOOKS:

1. Bart Baesens, **"***Analytics in a Big Data World: The Essential Guide to Data Science and its Applications,"* Wiley Publications, 2014.
2. Paul Zikopoulos, IBM, Chris Eaton and Paul Zikopoulos *"Understanding Big Data: Analytics for Enterprise Class Hadoop and streaming Data,"* The McGraw-Hill Companies, 2012.
3. Chuck Lam *"Hadoop in action,"* Manning Publications, 2011.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester
(19MT12508) Information Security
(Software Engineering)

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

PREREQUISITES: -

COURSE DESCRIPTION:

Introduction to Security; Need for Security; Risk Management; Planning for Security; Security Technology

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Understand the basics of Information Security

CO2: Demonstrate the reasons for security issues.

CO3: Understand and manage risks.

CO4: Gain knowledge on security standards.

CO5: Understand the role of access control systems and security systems.

DETAILED SYLLABUS

UNIT-I: *INTRODUCTION*

History, Security Concepts, Critical Characteristics of Information, CNSS Security Model, Components of an Information System, Balancing Security and Access, The SDLC, The Security SDLC.

UNIT II: *NEED FOR SECURITY*

Business Needs, Threats, Attacks - Malicious Code, Hoaxes, Doors, Password Crack, Brute Force, Dictionary, Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS), Spoofing, Man-in-the-Middle, Spam, Mail Bombing, Sniffers, Social Engineering, Pharming, Timing Attack; Secure Software Development

UNIT III: *RISK MANAGEMENT*

Risk Identification, Risk Assessment, Risk Control Strategies, Selecting a Risk Control Strategy, Quantitative Vs Qualitative risk Control Practices.

UNIT IV: PLANNING FOR SECURITY

Information Security Planning and Governance; Information Security Policy, Standards and Practices; Information security Blueprint; Security Education, Training and Awareness Program; continuity Strategies.

UNIT V: SECURITY TECHNOLOGY

Access Control, Firewalls, Intrusion Detection and Prevention Systems; Honeypots, Honeynets, and Padded Cell Systems; Scanning and Analysis Tools; Biometric Access Controls.

TEXT BOOK:

1. Michael E Whitman and Herbert J Mattord, "*Principles of Information Security*", 6th Edition, Course Technology Press, Boston, US 2018

REFERENCES

1. Micki Krause, Harold F. Tipton, "*Handbook of Information Security Management*", 6th Edition, Auerbach Publications, 2013.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M. Tech. I Semester

(19MT10509) MACHINE LEARNING

(Program Elective – 2)

(Common to CS, CN&IS and SE)

Internal Marks	External Marks	Total Marks	L	T	P	C
40	60	100	3	-	-	3

PRE-REQUISITES: A course on statistics.

COURSE DESCRIPTION:

Concepts of supervised, unsupervised and reinforcement learning, Bayesian decision theory, Learning rules from data, Performance evaluation of classification algorithms, Ensemble learners, Elements of Reinforcement Learning.

COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1: Understand, select and apply appropriate machine learning strategies of supervised, unsupervised and reinforcement learning for solving a given problem.

CO2: Develop applications by using appropriate ensemble models to enhance the accuracy of data analysis.

CO3: Evaluate the performance of machine learning algorithms and select optimal models to suit needs of a given problem.

CO4: Redesign existing machine learning algorithms to improve efficiency of classification models.

DETAILED SYLLABUS:

UNIT I – INTRODUCTION

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.

UNIT II – BAYESIAN DECISION THEORY AND LINEAR DISCRIMINATION

Bayesian Decision Theory: Classification, Losses and risks, Discriminant functions, Utility theory, Value of information, Bayesian networks, Influence diagrams, Association rules.

Linear Discrimination: Generalizing linear model, Geometry of the linear discriminant, Pairwise separation, Parametric discrimination, Gradient descent, Support vector machines.

UNIT III – DECISION TREES AND CLUSTERING

Decision trees: Univariate trees, Pruning, Rule extraction from trees, Learning rules from data, Multivariate trees.

Clustering: Mixture densities, K-Means clustering, EM algorithm, Supervised learning after clustering, Hierarchical clustering, Choosing the number of clusters.

UNIT IV – PERFORMANCE EVALUATION OF CLASSIFICATION ALGORITHMS

Cross validation and resampling methods, Measuring error, Interval estimation, Hypothesis testing, Assessing a classification algorithm's performance, Comparing two classification algorithms.

Combining Multiple Learners: Rationale, Voting, Bagging, Boosting, The mixture of experts revisited, Stacked generalization, Cascading.

UNIT V – REINFORCEMENT LEARNING

Single state case - *K*-Armed Bandit; Elements of reinforcement learning, Model based learning, Temporal difference learning, Generalization, Partially observable states.

TEXT BOOK:

1. Ethem Alpaydin, *Introduction to Machine Learning,* MIT Press (Adaptive Computation and Machine Learning Series), 3rd Edition, 2014.

REFERENCE BOOKS:

1. Stephen Marsland, *Machine Learning: An Algorithmic Perspective*, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2nd Edition, 2014.
2. Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, Wiley, 2nd Edition, 2012.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12509) SERVICE ORIENTED ARCHITECTURE

(Software Engineering)

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

PRE-REQUISITES:A Course on "Software Engineering" and "Web Technologies"

COURSE DESCRIPTION: Introduction to SOA, Web services & Primitive SOA, Contemporary SOA, 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), SOA support in .NET and J2EE.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate knowledge on:

- Fundamental of web services
- Development strategies, Principles, Services, Layers and building of SOA.

CO2: Analyze complex business process critically in identifying appropriate service model logic like Activity Management, Messaging and Data Security. Understanding key policies behind SOA.

CO3: Understand Service policies and web service standardswith web services technology elements in SOA.

CO4: Build service models for Entity and Task centric business with modeling guidelines and approaches.

CO5: Gain skills on XML Schema, WSDL, SOAP, BPEL and apply the modern tools and techniques of .NET and J2EE to modeling the web services.

DETAILED SYLLABUS

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.

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

UNIT-II: SOA AND WS-* EXTENSIONS

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

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

Service Layers: Service-Orientation 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 & DESIGN)

Analysis 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.

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

UNIT-V: BUILDING SOA (DESIGN & TECHNOLOGY)

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, Service Oriented Business process Design.

SOA Platforms: SOA platform basics, SOA Support in J2EE, SOA Support in .NET

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.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12531) ADVANCED SOFTWARE ENGINEERING LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	4	2

PREREQUISITES: Courses on "Advanced Software Engineering", and "Data Structures and Algorithms".

COURSE DESCRIPTION: Software development life cycle activities- requirements specification using open source Requirement documentation tool, modeling using AgroUML tool; Raptor-Flowchart based programming tool; Marathon IDE; Jenkins and Apache Jmeter tools; Project Management Web application using Redmine; JIRA and Scala Oriented Build tools; Implementation of various linear and non-linear data structures using C++; MIT app Inventor and Thingspeak Cloud.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate hands-on experience on:

- Requirements Engineering and Management
- Estimation of software-size, effort, schedule and cost.

CO2: Design and Develop basic flowcharts and programs for performing Input, Output and Computations using Agro UML, Raptor tools, Marathon IDE and C++ concepts.

CO3: Identify and apply designing, estimating tools and methodologies for complex engineering problems.

CO4: Develop mobile applications for solving real time applications using MIT app Inventor and Thingspeak Cloud.

CO5: Work individually and in teams collaboratively in implementing mini projects.

CO6: Demonstrate communication skills both oral and written for preparing and presenting reports.

CO7: Engage in life-long learning and enthusiasm to upgrade knowledge and skills in latest technologies and tools.

LIST OF EXERCISES:

1. Prepare the Software Requirement Specification (SRS), High Level Design (HLD) and Detailed Design (DD) for the following experiments

   (i) Online Job Portal (ii) E-Voting System

   Note: For the reference of SRS, HLD and DD templates refer department manual and use any open source Requirement documentation tool.

2. Estimate project parameters such as size, effort and time for development for a Library Information system using Basic COCOMO model.

3. Model UML Use case, Sequence, Collaboration and Component diagrams for the following experiments using Argo UML tool (i) Online Job Portal (ii) E-Voting System.
4. Study and prepare a report on the following tools:

   (i) Raptor-Flowchart based programming tool

   (ii) Marathon IDE.
5. Study and prepare a report on

   (i) Jenkins tool

   (ii) Apache Jmeter
6. Create any project management web application using Redmine Open source tool.
7. Consider any one mobile application and track the issues and bugs related to it using JIRA tool.
8. Develop a Scala application using Scala oriented Build Tool (SBT).
9. Write C++ program to implement the operations of doubly linked list.
10. Write a C++ program to perform the following operations of BST:

    a) Node Insertion b) Node Deletion c) Key Search
11. Write C++ program for implementing the following search and sorting techniques.

    a) Binary search b) Fibonacci Search c) Quick Sort d) Shell Sort
12. Write C++ program to construct the Minimum Cost Spanning Tree using Kruskal's algorithm.
13. Write a C++ program to implement 0/1 Knapsack problem.
14. Mini Projects (MPs):

    Form a group of maximum 3 members as a team and assign mini projects related to mobile and IoT based applications using MIT app Inventor and Thingspeak Cloud.

REFERENCE BOOKS:

1. Roger S. Pressman, "*Software Engineering, A practitioner's Approach*," 6th Edition, Tata McGraw-Hill, Edition, 2010.
2. Sommerville, "*Software Engineering*," 8th Edition, Pearson Education, 2007.
3. Rajesh Naik and Swapna Kishore, "*Software Requirements and Estimation*," Tata McGraw Hill, New Delhi, 2001.
4. "Estimation of Project Estimation and Metrics," http://vlssit.iitkgp.ernet.in/isad/isad/2/.
5. "Marathon IDE", https://marathontesting.com/
6. "Jenkins tool", https://jenkins.io/doc/book/managing/tools/
7. "Apache Jmeter", https://jmeter.apache.org/
8. "Redmine", https://www.redmine.org/
9. Refer online repositories for more information about Sikuli, JIRA, SBT tools, MIT app Inventor and Thingspeak Cloud.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. I Semester

(19MT12532) MOBILE APPLICATION DEVELOPMENT LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	4	2

PRE-REQUISITES: Course on Mobile Application Development

COURSE DESCRIPTION: Hands-on experience on development of Android Mobile applications using Views; Menus; Layouts; Buttons; Date Picker, and database creation and access with Android SQLite.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate programming skills for the development of mobile applications using Android platform.

CO2: Analyze the requirements of the user to develop Android applications.

CO3: Design user interface for mobile applications using views and develop Android applications based on user requirements.

CO4: Apply Android programming principles to create files and databases for storing and retrieving of the user data.

CO5: Apply Android programming principles of messaging, location-based services (LBS), and networking to develop mobile applications like SMS, e-Mail, LBS, and web services.

CO6: Work effectively as an individual and as a member in team for mini-project implementation.

CO7: Demonstrate communication skills, both oral and written for preparing and presenting reports.

List of Experiments:

1. Test an Android development environment by performing the following:
    a. Developing a simple application using Android studio.
    b. Creating an Android Virtual Device (AVD).
    c. Running a sample application using Android Emulator.
2. Design an Android application that demonstrates the use of Relative Layout with different attributes.
3. Design an Android application that demonstrates the use of Linear Layout with different attributes.
4. Develop an Android application that displays the custom Button and to handle the onClick() event.

5. Design an Android application that shows the Table layout to display child View elements in rows and columns.
6. Design an Android application that displays the List view.
7. Develop an Android application that shows Sub menus.
8. Develop an Android application that shows Context menu (Floating List of Menu Items).
9. Develop an Android application that demonstrates the use of Time and Date pickers.
10. Develop an Android application to insert, delete, display, and update the employee details using SQLite database.
11. Design and develop an Android application for user sign-up and sign-in. Utilize SQLite database for storing user details.
12. Mini project.

REFERENCE BOOKS:

R1. J. F. DiMarzio, "*Beginning Android Programming with Android Studio*", Wiley India, Fourth Edition, 2017.

R2. Paul Deital and Harvey Deital, *"Android How to Program",*Detial associates publishers, First Edition, 2013.

R3. Neils Smyth "*Android Stduio Development Essentials*", Creative Space Independent publishing platform, Seventh edition 2016.

R4. Jeff McWherter, Scott Gowell, **"***Professional Mobile Application Development",* Wiley India, First Edition, 2012.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester
(19MT22501) Full Stack Technologies
(Software Engineering)

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

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

COURSE DESCRIPTION:

HTML5, CSS3; JQuery, Bootstrap; PHP, PHP with MySQL database; Node.js; AngularJS.

COURSE OUTCOMES:

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

CO1. Demonstrate knowledge on HTML5 and CSS3.

CO2. Develop and validate interactive web pages.

CO3. Design web pages compatible for various devices.

CO4. Demonstrate programming skills to develop web applications.

CO5. Apply Node.js and AngularJS to make web pages more interactive, scalable and user friendly.

DETAILED SYLLABUS:

UNIT I – HTML5 AND CSS3
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, Offline Web Applications, Web Communications.
CSS3: Introduction, Features of CSS3, Syntax of CSS, Exploring CSS selectors, Inserting CSS in HTML Document, State of CSS3.

UNIT II – JQUERY AND BOOTSTRAP
JQUERY: Fundamentals of JQuery, JQuery Selectors, JQuery Methods to Access HTML Attributes and Traversing, JQuery Manipulators, Events and Effects.
BOOTSTRAP: Getting Started with Bootstrap, Creating Responsive Layouts Using Bootstrap CSS, Basic HTML structure for Bootstrap, Responsive classes, Rendering images, the grid system, Constructing data entry forms.

UNIT III – INTRODUCTION TO PHP AND DATABASE CONNECTION
Introduction, Data Types, Variables, Constants, Expressions, String Interpolation, Control Structures, Functions, Arrays, Embedding PHP Code in Web Pages, PHP and Web Forms, PHP with MySQL, Interacting with the Database.

UNIT IV – NODE.JS

Installing Node.js, Selecting a Node.js IDE, Working with Node Packages, Creating a Node.js Application, Creating a Node.js Package Module, Writing Data to Console, Node.js Event Model, Adding Work to the Event Queue, Implementing Timers, Implementing Event Emitters and Listeners, Implementing Callbacks, Handling Data I/O in Node.js, Accessing the File System from Node.js**.**

UNIT V – *ANGULARJS*

Introduction: Introduction to Angular, Modules, Directives, Data Binding, Services, Using the Angular CLI, Creating a Basic Angular Application. **Angular Components:** Component Configuration, Defining a selector, Building a Template, Using Inline CSS and HTML in Angular Applications. Using Constructors, Using External Templates. Expressions, Data Binding, Built-in Directives.

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. Brad Dayley, Brendan Dayley, Caleb Dayley, *"Node.js, MongoDB and Angular Web Development,"* 2nd Edition, Pearson Education, 2014.

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.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22502) Software Architecture and Design Patterns

(Software Engineering)

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

PREREQUISITES: Course on "Advanced Software Enginering," and "Object Oriented Programming".

COURSE DESCRIPTION:

This course is used to capture the persistent parts of the program and to derive the transient versions using architecture refinement. Patterns contribute to capturing the persistent parts of a program by describing often used solutions to problems in a context. This course covers the fundamental design principles and strategies for software architecture. Architectural styles, Quality attributes, Notations and Documents, Evaluating the architecture, MVC, Creational Patterns, Structural Patterns and Behavioral Patterns.

COURSE OUTCOMES:

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

CO1: Demonstrate knowledge on:

- Software architecture styles and business life cycle.
- Various design issues and patterns.

CO2: Understand the requirements of a software systems and design, document and reconstruct suitable software architectures.

CO3: Identify suitable software architectures to design quality software system.

CO4: Analyze and identify suitable design patterns like creational, structural and behavioral patterns.

DETAILED SYLLABUS

UNIT I: ENVISIONING ARCHITECTURE

Introduction to Software Architecture, Software Processes and the ABC, Architectural Patterns, Architectural structures and views, Reference Model and Reference Architecture, Documenting software architectures.

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

UNIT II: CREATING ARCHITECTURE

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

Evaluating the Architecture: The ATAM, The CBAM. The World Wide Web-A Case Study in Interoperability

Moving From one System to Many: Software Product Lines, Celsius Tech- A Case Study in Product Line Development, Building Systems from off the shelf components, Software Architecture in the future.

UNIT IV: INTRODUCTION TO DESIGN PATTERN AND CREATIONAL PATTERNS

Introduction to Design Patterns: 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, Singleton.

UNIT V: Structural and Behavioral Patterns

Structural Patterns: Adapter, Composite, Decorator, Flyweight.

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

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. Eric Freeman and Elisabeth Freeman, *"Head First Design patterns,"* O'REILLY, 2004.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22503) ADVANCED DATABASES

(Software Engineering)

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

PRE-REQUISITES: Courses on Database Management Systems and Computer Networks.

COURSE DESCRIPTION: Parallel Databases; Object Based Databases; Distributed Databases; Distributed Transaction Management; Emerging Database Technologies and Applications.

COURSE OUTCOMES:

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

CO1: Demonstrate knowledge on:

- Parallel databases.
- Object based and Object Relational databases.
- Distributed databases, horizontal and vertical data fragmentations.
- Mobile databases, Geographic Information Systems, Genome Data Management, Multimedia Database and NoSQL.

CO2: Demonstrate skills in Query optimization, Data Fragmentation, Transaction Management and Concurrency Control for Distributed Transactions.

CO3: Design Parallel, Object-Oriented, Object-Relational and NoSQL databases.

CO4: Solve Concurrency Problems in Distributed Transactions.

CO5: Use database techniques for Mobile, Geographic Information Systems, Genome Data Management, and Multimedia Data.

CO6: Create databases as per societal needs such as airline reservation, banking systems etc.

DETAILED SYLLABUS:

UNIT-I: PARALLEL DATABASES

Introduction, I/O Parallelism, Inter query Parallelism, Intra query Parallelism, Intra operation Parallelism, Interoperation Parallelism, Query Optimization, Design of Parallel Systems, Parallelism on Multicore Processors.

UNIT-II: OBJECT-BASED DATABASES

Overview, Complex Data Types, Structured Types and Inheritance in SQL, Table Inheritance, Array and Multi set Types in SQL, Object-Identity and Reference Types in SQL, Implementing O-R Features, Persistent Programming Languages, Object-Relational Mapping, Object-Oriented versus Object-Relational.

UNIT-III: DISTRIBUTED DATABASES

Features of Distributed versus Centralized Databases, Reference Architecture for Distributed Databases, Types of Data Fragmentation, Integrity Constraints in Distributed databases, Distributed Database Design

UNIT-IV: DISTRIBUTED TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL

Distributed Transaction Management: Framework for Transaction Management, Supporting Atomicity of Distributed transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions; **Concurrency Control:** Foundation of Distributed Concurrency Control, Distributed Deadlocks, Concurrency Control Based on Timestamps.

UNIT-V: EMERGING DATABASE TECHNOLOGIES AND APPLICATION

Mobile Database, Geographic Information Systems, Genome Data Management, Multimedia Database; NoSQL-An Overview of NoSQL, Characteristics of NoSQL, NoSQL Storage Types

TEXT BOOKS:

1. A. Silberschatz, H. F. Korth and S. Sudarshan, "*Database System Concepts,"* Tata McGraw hill , Fifth Edition, 2005.
2. Stefand Ceri and Giuseppe Pelagatti, **"***Distributed Databases Principles and Systems,"* McGraw hill, First Edition, 2008.

REFERENCE BOOKS:

1. Ramea Elmasri and Shamkant B.Navathe, "*Fundamentals of database Systems*," Pearson Education, Fifth Edition,2007.
2. Gaurav Vaish **"***Getting Started with NoSQL***,"** Packt Publishing, First Edition, 2013.(e-book)

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT10504) ARTIFICIAL INTELLIGENCE

(Common to M.Tech. (SE and CS))

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

PREREQUISITES: *Courses on "Object Oriented Programming through C++" and "Computer Graphics"*

COURSE DESCRIPTION: --Nil--

COURSE OUTCOMES: Artificial intelligence concepts, Intelligent agents, Problem solving agents, Logical agents, Knowledge representation and processing, Probabilistic learning, Natural language processing.

COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1: Perceive different artificial intelligence related concepts including intelligent agents, searching strategies, knowledge representation and learning, probabilistic reasoning, and Markov decision process.

CO2: Analyze different optimal searching techniques for a given problem and select optimal search technique.

CO3: Design and develop solutions for natural language processing applications.

CO4: Apply statistical models, and algorithms for solving real-world artificial intelligence problems.

DETAILED SYLLABUS:

UNIT I – ARTIFICIAL INTELLIGENCE & PROBLEM SOLVING

The Underlying assumption of AI; AI Technique: simple Tic-Tac-Toe program; Problem solving: State space search; Production systems: control strategies, search space control: depth-first, breadth-first search; Heuristic search: Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis.

UNIT II – KNOWLEDGE REPRESENTATION & LEARNING

Knowledge Representation; Predicate Logic: Simple facts in logic, resolution, Natural deduction; Procedural versus Declarative Knowledge; Forward reasoning versus Backward reasoning; Semantic Nets; Frames; slots; conceptual dependency; scripts; NonMonotonic Reasoning, Probabilistic reasoning, use of certainty factors, fuzzy logic systems & Neural nets: Basic concepts; Concept of learning.

UNIT III – *EXPERT SYSTEM AND APPLICATIONS*

Introduction, Phases in Building Expert Systems, Expert System Architecture, Expert Systems Vs Traditional Systems, Truth Maintenance Systems, Application of Expert Systems, List of Shells and Tools. Uncertainty Measure – Probability Theory: Introduction, Probability Theory, Bayesian Belief Networks, Certainty Factor Theory, Dempster-Shafer Theory.

UNIT IV – MACHINE-LEARNING PARADIGMS

Introduction. Machine Learning Systems, Supervised and Unsupervised Learning. Inductive Learning. Learning Decision Trees (Text Book 2), Deductive Learning. Clustering, Support Vector Machines. Artificial Neural Networks: Introduction, Artificial Neural Networks, Single- Layer Feed-Forward Networks, Multi-Layer Feed-Forward Networks, Radial- Basis Function Networks, Design Issues of Artificial Neural Networks, Recurrent Networks.

UNIT V – ADVANCED KNOWLEDGE REPRESENTATION TECHNIQUES

Case Grammars, Semantic Web Natural Language Processing: Introduction, Sentence Analysis Phases, Grammars and Parsers, Types of Parsers, Semantic Analysis, Universal Networking Knowledge.

TEXT BOOKS:

1. Rich, Knight, Nair, "*Artificial intelligence*", Tata McGraw Hill, Third Edition 2009.
2. Saroj Kaushik, "Artificial Intelligence", Cengage Learning, 2011.

REFERENCE BOOKS:

1. D.W. Patterson, "*Introduction to AI and Expert Systems*", PHI, 1992.
2. Russell, Norvig, "*Artificial intelligence, A Modern Approach*", Pearson Education, Second Edition, 2004.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22504) .NET TECHNOLOGIES

(Software Engineering)

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

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

COURSE DESCRIPTION: Knowledge on .NET Framework; The element of VB.NET; Writing Software with Visual Basic .NET; C# Programming; Object-oriented concepts with C#; Exception handling mechanism; Interfaces; Generics; Delegates and Events in C#; Database access with ADO.NET; Web application development using Web forms and Web controls.

COURSE OUTCOMES:

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

CO1: Demonstrate the knowledge on basics of .NET Platform.

CO2: Building simple VB .NET applications and Demonstrate problem solving skills for creating VB .NET Applications.

CO3: Demonstrate basic knowledge on C# and problem solving skills for creating C# applications.

CO4: Design and Develop C# application to access the database.

CO5: Design and develop Graphical User Interface and Web applications using ASP .NET technologies.

DETAILED SYLLABUS:

UNIT - I: INTRODUCING .NET PLATFORM and VB .NET

Benefits of the .NET platform, Building blocks of the .NET platform, Overview of .NET assemblies, Common type system, Common language specification, Common language runtime, Platform-independent nature of .NET. The role of the .NET framework 4.5 SDK, Building .NET application using visual studio,

Visual Basic fundamentals: The Visual Basic .NET Development Environment, The element of VB.NET, VB.NET operators, Conditional structure and control flow, Methods.

UNIT – II: CORE VB .NET PROGRAMMING

Classes and Objects: Types, Structure and Enumeration, Classes, Methods, Interfaces, Exception handling and Classes, Collections, Arrays and other Data Structure, Delegates and Events Data Processing and I/O, Writing Software with Visual Basic .NET, Interfacing with the End User.

UNIT - III: CORE C# PROGRAMMING

Anatomy of a simple C# program, System data types and corresponding C# keywords. C# iteration constructs, Decision constructs and the Relational/equality operators. Understanding C# arrays, Introducing the C# class type, Constructors, Pillars of OOP, C# access modifiers, C# encapsulation services, The basic mechanics of inheritance, The Details of Inheritance, C#'s polymorphic support.

The role of .NET Exception Handling, The simplest possible example, System-level exceptions, Application level exceptions, Processing multiple exceptions.

UNIT - IV: INTERFACES, GENERICS, DELEGATES AND EVENTS, ADO.NET

Understanding interface types, Implementing an Interface, Role of generic type parameters , Creating custom generic methods, Creating custom generic structures and classes, Understanding the .NET delegate type, Delegate example, Generic Delegate, and C# Events, Understanding operator overloading.

High level definition of ADO.NET, ADO.NET data provider, ADO.NET namespaces, Data Readers , Database transactions, Role of the dataset, Working with Data Columns, Data Rows, Data Table, Data Adapters, Binding Data Table objects to windows forms GUIs.

UNIT - V: ASP.NET WEB FORMS AND WEB CONTROLS, STATE MANAGEMENT TECHNIQUES

The role of http, Web applications and web servers, Role of client side scripting, Posting back to the web server, Overview of ASP.NET API, Building a single file ASP.NET web page, Building an ASP.NET webpage using Code Files, ASP.NET web sites vs. ASP.NET Web applications, The life cycle of an ASP.NET web page, Understanding the nature of web controls, Major categories of ASP.NET web control, Maintaining session data, Cookies.

TEXT BOOKS:

1. Jeffrey R. Shapiro "*The Complete Reference Visual Basic .NET*" Tata Mcgraw Hill (2002 Edition).
2. Andrew Troelsen "*Pro C# 5.0 and the .NET 4.5 Framework*," Apress, Sixth Edition, 2012.

REFERENCE BOOKS:

1. Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner, "*Professional C# 4 and .NET 4,*" WROX Publications, First Edition, 2010.
2. Mathew Mac Donald, "*The Complete Reference ASP.NET,*" TATA McGraw Hill, First Edition, 2010.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22505) SOFTWARE MEASUREMENT AND METRICS

(Software Engineering)

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

PRE-REQUISITES: A course on "Advanced 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:

After successful completion of this course, the students will be able to:

CO1: Identify and apply various software metrics for determining the quality level of software.

CO2: Analyze goal based framework and prepare fault and failure reports for External products.

CO3: Identify and evaluate the quality level of internal and external attributes of a given software product.

CO4: Understand the process for calculating Complexity metrics.

CO5: Apply quality metrics for software development and evaluate the quality level of the software based on the requirement.

DETAILED SYLLABUS

UNIT-I: FUNDAMENTALS OF MEASUREMENT

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

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

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

Measuring External Product Attributes **-** Modeling Software Quality, Measuring Aspects of Quality. Planning a measurement Program.

Object-Oriented Metrics - Object-Oriented Concepts and Constructs, Design and Complexity metrics, Productivity Metrics, Quality and Quality Management Metrics.

UNIT-V: QUALITY METRICS

Software Quality Metrics Overview - Product Quality Metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metrics Programs-Motorola, HP, IBM, Collecting Software Engineering Data, Applying the Seven Basic Quality Tools in Software Development.

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.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M. Tech. (SE) – II Semester

(19MT20501) DATA SCIENCE

(Program Elective - 4)

(Common to CS and SE)

Internal Marks	External Marks	Total Marks	L	T	P	C
40	60	100	3	-	-	3

PRE-REQUISITES: Courses on Data Warehousing and Data Mining, Probability and Statistics.

COURSE DESCRIPTION:

Introduction to Data science, Using python for data science, Exploratory data analysis - Probability and distributions; Predictive modeling - Time series analysis; Data extraction - Feature selection, Single value decomposition, Principal component analysis; Data visualization - Using visualization for data science, Visualization tools.

COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1: Analyze data by recognizing data science process and develop simple applications using relevant python libraries.

CO2: Design and develop predictive models for a given problem to support forecasting.

CO3: Select and apply appropriate data visualization models to identify trends, patterns and outliers in given datasets.

CO4: Investigate and evaluate various predictive models to contribute tools/techniques to applications of diverse domains.

DETAILED SYLLABUS:

UNIT I – INTRODUCTION TO DATA SCIENCE

Data science history, Data science project life cycle, Managing a data science project, Using python for data science - Understanding the basic concepts of python, Essential python libraries; Numpy, Pandas, Matplotlib, IPython, Scipy, Scikitlearn, StatsModels, Using python to analyze data.

UNIT II – STATISTICAL THINKING FOR PROGRAMMERS

Exploratory Data analysis: Distributions - Representing and plotting histograms, Outliers, Summarizing distributions, Variance, Reporting results; Probability mass function - Plotting PMFs, Other visualizations, The class size paradox Data frame indexing; Cumulative distribution functions - Limits of PMFs, Representing CDFs, Percentile based statistics, Random numbers, Comparing percentile ranks; Modeling distributions - Exponential distribution, Normal distribution, Lognormal distribution.

UNIT III – PREDICTIVE MODELING

Time series Analysis - Importing and cleaning, Plotting, Moving averages, Missing values, Serial correlation, Autocorrelation; Predictive modeling - Overview, Evaluating predictive models, Building predictive model solutions, Sentiment analysis.

UNIT IV – DATA EXTRACTION

Extracting meaning from data - Feature selection, User retention, Filters, Wrappers, Entropy, Decision tree algorithm; Random forests, The dimensionality problem, Single value decomposition, Principal component analysis.

UNIT V - DATA VISUALIZATION

Need for data visualization, Creating visualizations - Comparison charts, Composition charts, Distribution charts, Relationship charts; Using visualization for data science – Popular visualization tools.

TEXT BOOKS:

1. Ofer Mendelevitch, Casey Stella, Douglas Eadline, *Practical Data science with Hadoop and Spark: Designing and Building Effective Analytics at Scale*, Addison Wesley (Data & Analytics Series), 2017.
2. Alen B. Downey, *Think Stats: Exploratory Data Analysis*, O'Reilly Publications, 2nd Edition, 2014.

REFERENCE BOOKS:

1. Jacob T. VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data,* O'Reilly Publications, 2016.
2. Cathy O'Neil, Rachell Schutt, *Doing Data Science: Straight Talk from the Frontline*, O'Reilly Publications, 2013.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M. Tech. (SE) – II Semester

(19MT26305) INTERNET OF THINGS

(Program Elective - 4)

(Common to CN&IS, CS, SE, DECS and CMS)

Internal Marks	External Marks	Total Marks	L	T	P	C
40	60	100	3	-	-	3

PRE-REQUISITES: Courses on Computer Networks, Python Programming.

COURSE DESCRIPTION:

Concepts of Domain Specific IoTs, M2M and system management with Netconf-Yang, IoT privacy and security, IoT physical devices, Amazon Web Services for IoT and case studies illustrating IoT design.

COURSE OUTCOMES:

CO1: Understand the concepts of IoT, IoT protocols, privacy and security issues in IoT applications to analyze domain specific IoT's.

CO2: Design solutions through implementing IoT applications on raspberry pi, AWS and develop security solutions to strengthen IoT environment.

CO3: Carry out research to provide IoT-based solutions to solve problems in different domains.

CO4: Acquire professional integrity through understanding the need for data privacy, security and authenticity in IoT-based applications.

DETAILED SYLLABUS:

UNIT I – CONCEPTS OF IoT

Definition and characteristics of IoT, Physical design of IoT – IoT protocols, Logical design of IoT, IoT enabling technologies, IoT levels and deployment templates.

UNIT II – DOMAIN SPECIFIC IoTs & IoT AND M2M

Domain Specific IoTs: Home automation, Cities, Environment, Energy, Logistics, Agriculture, Industry.

IoT and M2M: Introduction, M2M, Difference between IoT and M2M, SDN and NFV for IoT.

UNIT III – IoT SYSTEM MANAGEMENT WITH NETCONF-YANG AND DEVELOPING INTERNET OF THINGS

Need for IoT systems management, Simple Network Management Protocol (SNMP), Network operator requirements, NETCONF-YANG, IoT systems management with NETCONF-YANG.

Developing Internet of Things: Introduction, IoT design methodology.

UNIT IV – IoT PRIVACY, SECURITY AND VULNERABILITIES SOLUTIONS AND IoT PHYSICAL DEVICES

Introduction, Vulnerabilities, Security requirements and treat analysis, Use cases and misuse cases, IoT security tomography and layered attacker model, Identity management and establishment, Access control and secure message communication, Security models, Profiles and protocols for IoT.

IoT Physical Devices & Endpoints: What is an IoT device, Exemplary device, About the board, Linux on Raspberry Pi, Raspberry Pi interfaces, Programming Raspberry Pi with Python and other IoT devices.

UNIT V –AMAZON WEB SERVICES FOR IoT AND CASE STUDIES ILLUSTRATING IoT DESIGN

Amazon Web Services for IoT: Amazon EC2, Amazon AutoScaling, Amazon S3, Amazon RDS, Amazon DynamoDB.

Case Studies Illustrating IoT Design: Home automation, Cities, Environment and Agriculture.

TEXT BOOKS:

1. Arshdeep Bahga, Vijay Madisetti, *Internet of Things: A Hands-on Approach*, Universities Press, 2015.
2. Raj Kamal, *Internet of Things: Architecture and Design Principles*", McGraw Hill, 1st Edition, 2017.

REFERENCE BOOKS:

1. Adrian McEwen, Hakim Cassimally, *Designing the Internet of Things*, Wiley, 2013.

Jeeva Jose, *Internet of Things*, Khanna Publishing, 1st Edition, 2018.

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22506) SOFTWARE QUALITY ASSURANCE

(Software Engineering)

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

PRE-REQUISITES: --Nil--

COURSE DESCRIPTION: Introduction to SQA, Software quality factors, components; Contract review, Development and quality plans; Reviews, Software testing strategies; Procedures and work instructions, Corrective and preventive actions, Documentation control; Software quality metrics, cost and quality standards.

COURSE OUTCOMES:

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

CO1: Understand the basic components of Software Quality Assurance, factors, reviews, testing and standards.

CO2: Develop basic test suite design for improving software quality through Debugging.

CO3: Identify the strengths and weaknesses of different quality metrics and quality management standards.

CO4: Design and deploy novel software quality components through appropriate Procedures, work instructions and prepare effective documentation.

CO5: Identify suitable metrics and estimate software cost for delivering quality software products.

DETAILED SYLLABUS:

UNIT I: INTRODUCTION TO SOFTWARE QUALITY ASSURANCE

Introduction: Software quality, Software quality assurance – definition and objectives.

Software quality factors: Classifications of software requirements into software quality factors, Product operation, Product revision and product transition software quality factors.

The components of the software quality assurance system: The SQA system – an SQA architecture, Pre-project components, Software project life cycle components, Management SQA components, SQA standards, system certification, and assessment components

UNIT II: PRE-PROJECT SOFTWARE QUALITY COMPONENTS

Contract review: Process and its stages, objectives, Implementation, Subjects, Internal projects

Development and quality plans: Objectives, Elements of the development plan and quality plan, Development and quality plans for small projects and for internal projects

UNIT III: SQA COMPONENTS IN THE PROJECT LIFE CYCLE

Integrating quality activities in the project life cycle: Factors affecting intensity of quality assurance activities in the development process, Verification, validation and qualification, A model for SQA defect removal effectiveness and cost.

Reviews: objectives, Formal design reviews (DRs), Peer reviews, A comparison of the team review methods

Software testing strategies: Definition and objectives, strategies, White box and Black box testing

UNIT IV: SOFTWARE QUALITY INFRASTRUCTURE COMPONENTS

Procedures and work instructions: The need, Manuals, Preparation, Implementation and updating

Corrective and preventive actions: Definitions, Actions process, Analysis of collected information, Development of solutions and their implementation, Follow-up of activities.

Documentation control: Introduction, Controlled documents and quality records, The controlled documents list, Controlled document preparation, Issues of controlled document approval, Issues of controlled document storage and retrieval

UNIT V: MANAGEMENT COMPONENTS OF SOFTWARE QUALITY, STANDARDS

Software quality metrics: Objectives of quality measurement, Classification of software quality metrics, Process metrics, Product metrics, Implementation and Limitations.

Costs of software quality: Objectives, The classic model and an extended model, Application, problems

Quality management standards: ISO 9001 and ISO 9000-3, CMM and CMMI, The Bootstrap methodology, The SPICE project and the ISO/IEC 15504, Structure and content of IEEE software engineering standards.

Total Periods: 45

TEXT BOOKS:

1. Daniel Galin, "*Software Quality Assurance: From theory to implementation*", Pearson Education, 2004.
2. G. Gordon Schulmeyer, "*Handbook of Software Quality Assurance*", 4th Edition, Artech House, 2008.

REFERENCE BOOKS:

1. Stephen H. Kan, *"Metrics and Models in Software Quality Engineering",* 2nd Edition, Pearson Publication, 2008.
2. M.G.Limaye, "*Software Testing: Principles and Techniques and Tools*", 2010.
3. Dr.K.V.K.K.Prasad,"*Software Testing Tools",*Dreamtech, 2012

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22531) Full Stack Technologies Lab

(Software Engineering)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	4	2

PRE-REQUISITES: --Nil--

COURSE DESCRIPTION: Hands on practice in designing, developing and executing HTML5, CSS3, JQuery, Bootstrap, PHP, Node.js and AngularJS.

COURSE OUTCOMES:

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

CO1: Demonstrate knowledge on web page design elements, dynamic content and database Interaction.

CO2: Analyze user requirements to develop web applications.

CO3: Design client-server applications using web technologies.

CO4: Use HTML, CSS, JavaScript, JQuery, Bootstrap, PHP,Node.js, AngularJS technologies for device independent web application development.

CO5: Apply web technologies to develop interactive, dynamic and scalable web applications for societal needs.

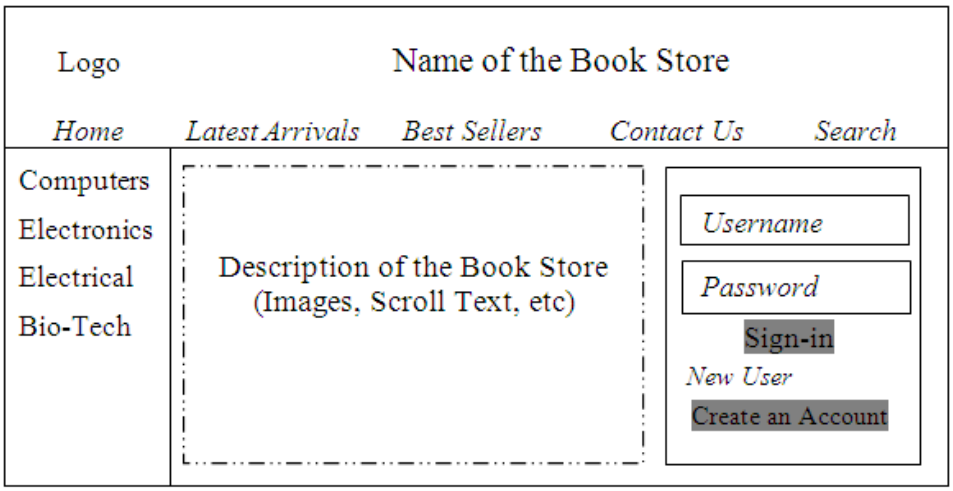
CO6: Work effectively as an individual and as a member in team for mini-project implementation.

CO7: Demonstrate communication skills, both oral and written for preparing and presenting reports.

List of Experiments:

1. Design the following static web pages required for an online book store web site.

a. Home Page:



The Home page must have the following three frames:

Top frame: Logo and the book store name and links to Home page, Latest arrivals, Best sellers, Contact us and Search.

Left frame: At least four links for navigation, which will display the books catalogue relevant to engineering disciplines. For e.g. when the link "Computers" is clicked, the catalogue relevant to computer science books will be displayed in the right frame.

Right frame: The pages of navigated links in the left and top frame must be loaded in the right frame. Initially it will load the Home page that can include the description of the book store, sign-in and create account information.

b. Catalogue Page:

The catalogue page should display the following details of available books.

i. Snap shot of cover page	ii. Title of the text book	iii. Author name
iv. Publisher	v. Price	vi. More details link.

c. Registration Page:

Design the Registration page with the following fields and navigate it with create an account link.

i. First Name	ii. Last Name	iii. Gender
iv. Date of Birth	v. Username	vi. Password
vii. Confirm Password	viii. Address	ix. Postal Code
x. Mobile No.	xi. Email-Id	

2. a. Design a web page to store username and password information using the local storage concept.
   b. Design a web page to store employee information including Name, Emp. Id, Department, Salary and Address on a client's machine using a real SQL database.

3. Apply the following styles to all web pages of online book store web application.
   a. Fonts and Styles: font-family, font-style, font-weight and font-size.
   b. Backgrounds and colors: color, background-color, background-image and background-repeat.
   c. Text: text-decoration, text-transformation, text-align and text-indentation, text-align
   d. Borders: border, border-width, border-color and borderstyle
   e. Styles for links: A: link, A: visited, A:active, A:hover f. Selectors, Classes, Layers and Positioning elements.

4. Write a JQuery code to validate the following fields of the Registration web page.
   a. First Name/Last Name - should contain only alphabets and the length should not be less than 8 characters.
   b. Username - It should contain combination of alphabets, numbers and underscore. It should not allow spaces and special symbols.
   c. Password - It should not less than 8 characters in length and it contains one uppercase letter and one special symbol.
   d. Date of Birth - It should allow only valid date; otherwise display a message stating that entered date is invalid. Ex. 29 Feb. 2009 is an invalid date.
   e. Postal Code: It must allow only 6 digit valid number.
   f. Mobile No. - It should allow only numbers and total number of digits should be equal to 10.

g. e-mail id - It should allow the mail id with the following format: Ex. mailid@domainname.com

5. Design a web page with the following features using HTML5, JavaScript and JQuery

a. Displaying of images with Custom animated effects
b. Playing of selected video from the list of videos
c. Showing the animated text in increasing and decreasing font size
d. Changing the size of the area in a web page using DIV tag
e. Hiding and Showing elements in a web page.

6. Design a web page with the following features using Bootstrap and Media Query.

a. Components
b. Responsive tables
c. Responsive images and videos

7. a. Deploy and navigate web pages of online book store using WAMP/XAMPP web server.

b. Write a PHP program to read user name and favorite color from the HTML form.

Display the name of the user in green color and sets user favorite color as a background for the web page.

8. Write a PHP code to read user details entered through the registration web page and store the same into MySQL database.

9. Write a PHP code for storing books details like Name of the book, author, publisher, edition, price, etc into MySQL database. Embed a PHP code in catalogue page of the online book store to extract books details from the database.

10. Develop an interactive web page using Node.JS to insert registration details and display those details dynamically.

11. Design a web page with the following features of Node.js

a) Timers

b) Event Emitters and Listeners

c) Callbacks

12. Develop a Simple AngularJS web application that includes different components and also apply Inline CSS.

13. **Mini Project**

REFERENCES:

1. Kogent Learning Solutions Inc, HTML 5 Black Book: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP and JQuery, Dreamtech Press, Second Edition, 2016.

2. W. Jason Gilmore, Beginning PHP and MySQL, APress, Fourth Edition, 2011.

3. Snig Bahumik, Bootstrap Essentials, PACKT Publishing, 2015. (e-book).

4. www.w3schools.com

5. www.tutorialspoint.com

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M.Tech. II Semester

(19MT22532) SOFTWARE ARCHITECTURE AND DESIGN PATTERNS LAB

(Software Engineering)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	4	2

PREREQUISITES: Courses on "Advanced Software Engineering" and "Object Oriented Programming".

COURSE DESCRIPTION: Software development life cycle activities- Implementation of design models using enterprise architect; Creation of web service client; Implementation of Orchestration with BPEL; Test plan document; Regression testing, functional testing using UFT, and Selenium; Performance testing using Load Runner, and Web Performance Tool.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

CO1: Demonstrate hands-on experience on:

- Web services
- Design issues and patterns

CO2: Design and develop basic design patterns, simulations for different software applications.

CO3: Identify and apply test case design methodologies for complex software engineering problems.

CO4: Develop and test web applications for debugging real time applications using Selenium and UFT tools.

CO5: Work individually and in teams collaboratively in implementing mini-projects.

CO6: Demonstrate communication skills both oral and written for preparing and presenting reports.

CO7: Engage in life-long learning and enthusiasm to upgrade knowledge and skills in latest technologies and tools.

LIST OF EXERCISES:

1. Implement the following diagrams by using Enterprise Architect 14.1
   i. Traceability ii. Roundtrip iii. visualization
   iv. Debugging v. Database modeling vi. Project Management
2. Write a Java program for 'Abstract Factory' and 'Decorator' design patterns.
3. Creation of web services and client applications to consume the following services:
   (i) Arithmetic Operations (ii) Finding prime numbers upto a given number 'N'.
4. Implementation of orchestration with BPEL for authenticating user credentials.
5. Create a test plan document for a web based application.
6. Write the Functional test cases for a Desktop based application. (Eg: Calculator)
7. Conduct the following functional testing for the flight reservation application using UFT 14.51
   i. The width of the GUI object "Name" in Flight Reservation window must be 150.
   ii. The flight schedule frame of Flight Reservation window should have the following properties; Height-283, Width-138, X-11,Y-84.
8. Conduct the following functional testing using UFT 14.51
   i. Test the EMPDB Application with CRUD operations.
   ii. Implement the following Checkpoints on the flight reservation application
      a. Standard Checkpoint
      b. Text Checkpoint
      c. Text Area Checkpoint
      d. Bitmap Checkpoint
      e. Database Checkpoint
9. Conduct Performance testing for a Desktop based application using Load Runner.
10. Write a test suite of functional and regression testing for Web based application using
    Selenium.
11. Write and test a program to provide total number of objects present on the page using
    Selenium.
12. Write and test a program to login a specific web page using Selenium.
13. Develop a mini project which includes all SDLC activities for any web based application.

REFERENCE BOOKS:

1. James W.Cooper, "Java Design Patterns- A Tutorial," Pearson Education, 2000
2. Eric Newcomer and Greg Lomow, "Understanding SOA with Web Services," Pearson Education, 2009.
3. Ilene Burnstein, "Practical Software Testing," Springer, International Edition, 2003.
4. Dr. K. V. K. K.Prasad, "Software Testing Tools," Dreamtech, 1st Edition, 2004.
5. "UFT Documentation", https://www.learnqtp.com/uft-14-features/
6. "Selenium Documentation," http://docs.seleniumhq.org/docs/

SREE VIDYANIKETHAN ENGINEERING COLLEGE (Autonomous)

Department of Information Technology

M. Tech. II Semester

(19MT2AC01) STATISTICS WITH R PROGRAMMING

(Audit Course)

(Common to All M. Tech. Programs)

Internal Marks	External Marks	Total Marks	L	T	P	C
--	--	--	2	-	-	-

PRE-REQUISITES: A course on Statistics.

COURSE DESCRIPTION:

Concepts of R programming basics, Bivariate and multivariate data, Confidence intervals, Goodness of fit, Analysis of variance.

COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1: Import, manage, manipulate, and structure data files using R programming.

CO2: Implement models for statistical analysis of a given dataset.

CO3: Build data visualization using charts, plots and histograms to identify trends, patterns and outliers in data.

CO4: Summarize and graph data, test hypotheses, analyze variance, and assess goodness-of-fit using advanced packages.

DETAILED SYLLABUS:

UNIT I - INTRODUCTION

Data, R's command line, Variables, Functions, The workspace, External packages, Data sets, Data vectors, Functions, Numeric summaries, Categorical data.

Unit II - BIVARIATE AND MULTIVARIATE DATA

(7 periods)

Lists, Data frames, Paired data, Correlation, Trends, Transformations, Bivariate categorical data, Measures of association, Two-way tables, Marginal distributions, Conditional distributions, Graphical summaries, Multivariate data - Data frames, Applying a function over a collection, Using external data, Lattice graphics, Grouping, Statistical transformations.

UNIT III - POPULATIONS

Populations, Discrete random variables, Random values generation, Sampling, Families of distributions, Central limit theorem, Statistical Inference - Significance tests, Estimation, Confidence intervals, Bayesian analysis.

UNIT IV - CONFIDENCE INTERVALS

Confidence intervals for a population proportion, *p* - population mean, Other confidence intervals, Confidence intervals for differences, Confidence intervals for the median, Significance test - Significance test for a population proportion, Significance test for the mean (t-tests), Significance tests and confidence intervals, Significance tests for the median.

UNIT V - GOODNESS OF FIT

The chi-squared goodness-of-fit test, The multinomial distribution, Pearson's χ^2-statistic, chi-squared test of independence and homogeneity, Goodness-of-fit tests for continuous distributions, ANOVA - One-way ANOVA, Using *lm* for ANOVA.

TEXT BOOKS:

1. John Verzani, *Using R for Introductory Statistics*, CRC Press, 2nd Edition, 2014.
2. Sudha G Purohit, Sharad D Gore, Shailaja R Deshmukh, *Statistics Using R*, Narosa Publishing house, 2nd Edition, 2015.

REFERENCE BOOKS:

1. Francisco Juretig, *R Statistics Cookbook*, Packt Publishing, 1st Edition, 2019.
2. Prabhanjan N. Tattar, Suresh Ramaiah, B. G. Manjunath, *A Course in Statistics with R*, Wiley, 2018.

M. Tech. - III Semester
(19MT32532) PROJECT WORK PHASE-I

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	-	10

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 problem identified; submitting a Report.

COURSE OUTCOMES:

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

CO1. Apply contextual knowledge to identify specific domain in Software Engineering and allied areas of discipline.

CO2. Conduct literature review, analyze, cognize and comprehend the extracted information to recognize the current status of research pertinent to the chosen domain.

CO3. Select appropriate tools, techniques and resources for implementation of project work.

CO4. Function effectively as an individual to recognize the opportunities in the chosen domain of interest

CO5. Write and present a technical report/document to present the findings on the chosen problem.

CO6. Engage lifelong learning for development of technical competence in the field of Software Engineering.

M. Tech. - IV Semester
(19MT42531) PROJECT WORK PHASE-II

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
150	150	300	-	-	-	16

PRE-REQUISITES: --

COURSE DESCRIPTION:

Time and cost analysis; undertaking practical investigations of project work; implementation; analysis of results; validation and report writing.

COURSE OUTCOMES:

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

CO1. Solve engineering problems pertinent to the chosen topic for feasible solutions.

CO2. Utilize appropriate tools, techniques and resources for implementation of project work.

CO3. Function effectively as an individual to recognize the opportunities in the chosen domain of interest

CO4. Write and present a technical report/document to present the findings on the chosen problem.

CO5. Do time and cost analysis on the project.

CO6: Engage lifelong learning for development of technical competence in the field of Software Engineering.