DEPARTMENT OF INFORMATION TECHNOLOGY
JNTUH UCESTH

M. Tech. SOFTWARE ENGINEERING

EFFECTIVE FROM ACADEMIC YEAR 2022 - 23 ADMITTED BATCH
R22 COURSE STRUCTURE AND SYLLABUS

I YEAR I-Semester

Course Code	Course Title	L	T	P	Max Marks(100)		Credits
					Int	Ext	
Professional Core –I	Principles of Software Engineering	3	0	0	40	60	3
Professional Core-II	Advanced Data Structures	3	0	0	40	60	3
Professional Elective –I	1. Enterprise Cloud Concepts 2. Internet Technologies & Services 3. Database Programming 4. Internet of things	3	0	0	40	60	3
Professional Elective-II	1. Software Project Management 2. Software Metrics 3. Software Reliability 4. Software Agents	3	0	0	40	60	3
Lab-I	Advanced Data Structures Lab	0	0	4	40	60	2
Lab-II	Professional Elective-I Lab	0	0	4	40	60	2
	Research Methodology & IPR	2	0	0	40	60	2
Audit-I	Audit Course-I	2	0	0	100		0
	Total	**16**	**0**	**8**			**18**

Professional Elective-I and Professional Elective-I Lab must be of same course

I YEAR II -Semester

Course Code	Course Title	L	T	P	Max Marks (100)		Credits
					Int	Ext	
Professional Core –III	Design Patterns	3	0	0	40	60	3
Professional Core-IV	Machine Learning	3	0	0	40	60	3
Professional Elective –III	1. Software Testing Methodologies 2. Mobile Application and API Development 3. Full Stack Development 4. Functional Programming	3	0	0	40	60	3
Professional Elective-IV	1. Software Requirements & Estimation 2. Secure Software Engineering 3. Object Oriented Software Engineering 4. Human Computer Interaction	3	0	0	40	60	3
Lab-III	Machine Learning and Design patterns Lab	0	0	4	40	60	2
Lab-IV	Professional Elective-III Lab	0	0	4	40	60	2
	Mini Project With Seminar	0	0	4	40	60	2
Audit-II	Audit Course-II	2	0	0	100		0
	Total	**14**	**0**	**12**			**18**

Professional Elective-III and Professional Elective-III Lab must be of same course

II YEAR I -Semester

Course Code	Course Title	L	T	P	Max Marks(100)		Credits
					Int	Ext	
Professional Elective-V	1. Agile Development Methodologies 2. Web Services Testing 3. Adhoc Networks 4. Mobile Application Security	3	0	0	40	60	3
Open Elective	Open Elective	3	0	0	40	60	3
Dissertation	Dissertation Work Review-II	0	0	12	100		6
	Total	**6**	**0**	**12**			**12**

II YEAR II-Semester

Course Code	Course Title	L	T	P	Max Marks(100)		Credits
					Int	Ext	
Dissertation	Dissertation Work Review-III	0	0	12	100		6
Dissertation	Dissertation Viva-Voice	0	0	28		100	14
	Total	**0**	**0**	**40**			**20**

*For Dissertation Work Review - I, Please refer 7.8 in R22 Academic Regulations.

Audit Course I & II:
1. English for Research Paper Writing
2. Disaster Management
3. Sanskrit for Technical Knowledge
4. Value Education
5. Constitution of India
6. Pedagogy Studies
7. Stress Management by yoga
8. Personality Development Through Life Enlightenment Skills

Open Elective:

1. Fault Tolerance Systems
2. Intrusion Detection Systems
3. Optimization Techniques
4. Cyber Physical Systems
5. Graph Analytics
6. Cyber Security
7. Network Programming

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

PRINCIPLES OF SOFTWARE ENGINEERING (PC– I)

Prerequisites: Software Engineering
Course Objectives: Your studies will enable you to develop:
- a broad and critical understanding of all the processes for engineering high quality software and the principles, concepts and techniques associated with software development
- an ability to analyze and evaluate problems and draw on the theoretical and technical knowledge to develop solutions and systems
- a range of skills focused on the analysis of requirements, design and implementation of reliable and maintainable software, with strong emphasis on engineering principles applied over the whole development lifecycle
- an awareness of current research in software development, the analytical skills and research techniques for their critical and independent evaluation and their application to new problems.

UNIT - I
Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, legacy software, Software myths.
A Generic view of process: Software engineering - A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.
Process models: The waterfall model, Incremental process models, Evolutionary process models, specialized process models, The Unified process.

UNIT - II
Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. **Requirements engineering process:** Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. **System models**: Context Models, Behavioral models, Data models, Object models, structured methods.

UNIT - III
Design Engineering: Design process and Design quality, Design concepts, the design model, pattern-based software design.
Creating an architectural design: software architecture, Data design, Architectural styles and patterns, Architectural Design, assessing alternative architectural designs, mapping data flow into a software architecture.
Modeling component-level design: Designing class-based components, conducting component-level design, object constraint language, designing conventional components. Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

UNIT - IV
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.
Product metrics: Software Quality, Frame work for Product metrics, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.
Metrics for Process and Products: Software Measurement, Metrics for software quality.

UNIT - V
Risk management: Reactive vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.
Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.
Configuration Management: Configuration Management planning, Change management, Version and release management, System building, CASE tools for configuration management.

Text Books:
1. Software Engineering: A practitioner's Approach, Roger S Pressman, sixth edition.   McGraw Hill International Edition, 2005
2. Software Engineering, Ian Sommerville, seventh edition, Pearson education, 2004.

Reference Books:
1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
2. Software Engineering: A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3. Fundamentals of Software Engineering, Rajib Mall, PHI, 2005
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition, 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8. Software Engineering 3: Domains, Requirements and Software Design, D. Bjorner, Springer, International Edition.
9. Software Engineering Principles and Practice, Hans Van Vliet,3rd edition, Wiley India edition.
10. Introduction to Software Engineering, R.J. Leach, CRC Press.
11. Software Engineering Fundamentals, Ali Behforooz and Frederick J. Hudson, Oxford University Press, rp2009
12. Software Engineering Handbook, Jessica Keyes, Auerbach, 2003.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

ADVANCED DATA STRUCTURES (PC- II)

Prerequisites: A course on "Data Structures"
Course Objectives
1. Introduces the heap data structures such as leftist trees, binomial heaps, Fibonacci and min- max heaps
2. Introduces a variety of data structures such as disjoint sets, hash tables, search structures and digital search structures

Course Outcomes
1. Ability to select the data structures that efficiently model the information in a problem
2. Ability to understand how the choice of data structures impact the performance of programs
3. Design programs using a variety of data structures, including hash tables, search structures and digital search structures

UNIT – I
 Introduction:Introduction ,Stacks,queues,linked lists.
 Heap Structures: Introduction, Min-Max Heaps, Binomial Heaps, Fibonacci heaps.

UNIT - II
 Hashing and Collisions
 Introduction, Hash Tables, Hash Functions, different Hash Functions: Division Method, Multiplication Method, Mid-Square Method, Folding Method, Collisions

UNIT - III
 Search Structures: OBST, AVL trees, Red-Black trees, Splay trees,
 Multiway Search Trees: B-trees, 2-3 trees

UNIT - IV
 Digital Search Structures
 Digital Search trees, Binary tries and Patricia, Multiway Tries, Suffix trees, Standard Tries, Compressed Tries

UNIT - V
 Pattern matching
 Introduction, Brute force, the Boyer –Moore algorithm, Knuth-Morris-Pratt algorithm, Naïve String, Harspool, Rabin Karp

TEXT BOOKS:
1. Fundamentals of data structures in java Sahni, Horowitz, Mehatha, Universities Press.
2. Introduction to Algorithms, TH Cormen, PHI

REFERENCES:
1. Design methods and analysis of Algorithms, SK Basu, PHI.
2. Data Structures & Algorithm Analysis in java, Mark Allen Weiss, Pearson Education.
3. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press.

ENTERPRISE CLOUD CONCEPTS (Professional Elective - I)

M.Tech SE I Year I Sem.   **L   T   P   C**
                                                                    3   0   0   3

Course Objectives: Knowledge on significance of cloud computing and its fundamental concepts and models.

Course Outcomes:
1. Understand importance of cloud architecture
2. Illustrating the fundamental concepts of cloud security
3. Analyze various cloud computing mechanisms
4. Understanding the architecture and working of cloud computing.

UNIT - I
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.

UNIT - II
Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Center Technology, Virtualization Technology.
Cloud Computing Mechanisms:
Cloud Infrastructure Mechanisms: Logical Network Perimeter, Virtual Server, Cloud Storage Device, Cloud Usage Monitor, Resource Replication.

UNIT - III
Cloud Management Mechanisms: Remote Administration System, Resource Management System, SLA Management System, Billing Management System, Case Study Example
Cloud Computing Architecture
Fundamental Cloud Architectures: Workload Distribution Architecture, Resource Pooling Architecture, Dynamic Scalability Architecture, Elastic Resource Capacity Architecture, Service Load Balancing Architecture, Cloud Bursting Architecture, Elastic Disk Provisioning Architecture, Redundant Storage Architecture, Case Study Example

UNIT - IV
Cloud-Enabled Smart Enterprises: Introduction, Revisiting the Enterprise Journey, Service-Oriented Enterprises, Cloud Enterprises, Smart Enterprises, The Enabling Mechanisms of Smart Enterprises
Cloud-Inspired Enterprise Transformations: Introduction, The Cloud Scheme for Enterprise Success, Elucidating the Evolving Cloud Idea, Implications of the Cloud on Enterprise Strategy, Establishing a Cloud-Incorporated Business Strategy

UNIT - V
Transitioning to Cloud-Centric Enterprises: The Tuning Methodology, Contract Management in the Cloud
Cloud-Instigated IT Transformations
Introduction, Explaining Cloud Infrastructures, A Briefing on Next-Generation Services, Service Infrastructures, Cloud Infrastructures, Cloud Infrastructure Solutions, Clouds for Business Continuity, The Relevance of Private Clouds, The Emergence of Enterprise Clouds

TEXT BOOKS:
1. Erl Thomas, Puttini Ricardo, Mahmood Zaigham, Cloud Computing: Concepts, Technology &Architecture 1st Edition,
2. Pethuru Raj, Cloud Enterprise Architecture, CRC Press

REFERENCES:
1. James Bond,The Enterprise Cloud, O'Reilly Media, Inc.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

INTERNET TECHNOLOGIES AND SERVICES (Professional Elective - I)

Course Objective:
- The student who has knowledge of programming with java should be able to develop web-based solutions using multi-tier architecture.
- S/he should have good understanding of different technologies on client and server-side components as Follows:

Client Side: HTML5, CSS3, Javascript, Ajax, JQuery and JSON
Server Side: Servlets, JSP
Database: MySQL with Hibernate and Connection Pooling
Framework: Struts with validation framework, Internationalization (I18N)
SOA: Service Oriented Architecture, Web services fundamentals, Axis framework for WS

UNIT - I
Client Side Technologies: Overview of HTML - Common tags, XHTML, capabilities of HTML5 Cascading Style sheets, CSS3 enhancements, linking to HTML Pages, Classes in CSS Introduction to JavaScripts, variables, arrays, methods and string manipulation, BOM/DOM (Browser/Document Object Model), accessing elements by ID, Objects in JavaScript Dynamic HTML with JavaScript and with CSS, form validation with JavaScript, Handling Timer Events Simplifying scripting with JQuery, JASON for Information exchange.

UNIT - II
Introduction to Java Servlets: Introduction to Servlets: Lifecycle of a Servlet, Reading request and initialization parameters, Writing output to response, MIME types in response, Session Tracking: Using Cookies and Sessions, Steps involved in Deploying an application Database Access with JDBC and Connection Pooling Introduction to XML, XML Parsing with DOM and SAX Parsers in Java Ajax - Ajax programming with JSP/Servlets, creating XML Http Object for various browsers, Sending request, Processing response data and displaying it. Introduction to Hibernate

UNIT - III
Introduction to JSP: JSP Application Development: Types of JSP Constructs (Directives, Declarations, Expressions, Code Snippets), Generating Dynamic Content, Exception Handling, Implicit JSP Objects, Conditional Processing, Sharing Data Between JSP pages, Sharing Session and Application Data, Using user defined classes with jsp:useBean tag, Accessing a Database from a JSP.

UNIT - IV
Introduction to Struts Framework: Introduction to MVC architecture, Anatomy of a simple struts2 application, struts configuration file, Presentation layer with JSP, JSP bean, html and logic tag libraries, Struts Controller class, Using form data in Actions, Page Forwarding, validation frame work, Internationalization.

UNIT - V
Service Oriented Architecture and Web Services: Overview of Service Oriented Architecture – SOA concepts, Key Service Characteristics, Technical Benefits of a SOA Introduction to Web Services– The definition of web services, basic operational model of web services, basic steps of implementing web services. Core fundamentals of SOAP – SOAP Message Structure, SOAP

encoding, SOAP message exchange models, Describing Web Services –Web Services life cycle, anatomy of WSDL Introduction to Axis– Installing axis web service framework, deploying a java webservice on axis. Web Services Interoperability **–** Creating java and .Net client applications for an AxisWeb Service

(Note: The Reference Platform for the course will be open source products Apache Tomcat Application Server, MySQL database, Hibernate and Axis)

Text Books:
1. Web Programming, building internet applications, Chris Bates 3rd edition, WILEY Dreamtech .
2. The complete Reference Java 7th Edition, Herbert Schildt., TMH.
3. Java Server Pages,Hans Bergsten, SPD, O'Reilly.
4. Professional Jakarta Struts - James Goodwill, Richard Hightower, Wrox Publishers.
5. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India, rp – 2008.
6. Understanding SOA with Web Services, Eric Newcomer and Greg Lomow, Pearson Edition – 2009
7. Java Web Service Architecture, James McGovern, Sameer Tyagi et al., Elsevier – 2009

References:
1. Programming the world wide web, 4th edition, R.W. Sebesta, Pearson
2. Core SERVLETS AND JAVA SERVER PAGES VOLUME 1: CORE
3. TECHNOLOGIES, Marty Hall and Larry Brown Pearson
4. Internet and World Wide Web – How to program, Dietel and Nieto PHI/Pearson.
5. Jakarta Struts Cookbook, Bill Siggelkow, S P D O'Reilly.
6. Professional Java Server Programming, S. Allamaraju & others Apress (Dreamtech).
7. Java Server Programming, Ivan Bayross and others, The X Team, SPD
8. Web Warrior Guide to Web Programming-Bai/Ekedaw-Cengage Learning.
9. Beginning Web Programming-Jon Duckett, WROX.

DATABASE PROGRAMMING (Professional Elective - I)

M.Tech SE I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:
1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

Course Outcomes:
1. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

UNIT - I
PL/SQL Basics: Block Structure, Behavior of Variables in Blocks, Basic Scalar and Composite Data Types, Control Structures, Exceptions, Bulk Operations, Functions, Procedures, and Packages, Transaction Scope.

UNIT - II
Language Fundamentals & Control Structures: Lexical Units, Variables and Data Types, Conditional Statements, Iterative Statements, Cursor Structures, Bulk Statements, Introduction to Collections, Object Types: Varray and Table Collections, Associative Arrays, Oracle Collection API.

UNIT - III
Functions and Procedures: Function and Procedure Architecture, Transaction Scope, Calling Subroutines, Positional Notation, Named Notation, Mixed Notation, Exclusionary Notation, SQL Call Notation, Functions, Function Model Choices, Creation Options, Pass-by-Value Functions, Pass-by-Reference Functions, Procedures, Pass-by-Value Procedures, Pass-by-Reference Procedures, Supporting Scripts.

UNIT - IV
Packages: Package Architecture, Package Specification, Prototype Features, Serially Reusable Precompiler Directive, Variables, Types, Components: Functions and Procedures, Package Body, Prototype Features, Variables, Types, Components: Functions and Procedures, Definer vs. Invoker Rights Mechanics, Managing Packages in the Database Catalog, Finding, Validating, and Describing Packages, Checking Dependencies, Comparing Validation Methods: Timestamp vs. Signature.

UNIT - V
Triggers: Introduction to Triggers, Database Trigger Architecture, Data Definition Language Triggers, Event Attribute Functions, Building DDL Triggers, Data Manipulation Language Triggers, Statement-Level Triggers, Row-Level Triggers, Compound Triggers, INSTEAD OF Triggers, System and Database Event Triggers, Trigger Restrictions, Maximum Trigger Size, SQL Statements, LONG and LONG RAW Data Types.

TEXT BOOKS:
1. Oracle Database 12c PL/SQL Programming Michael McLaughlin, McGrawHill Education

REFERENCES:
1. Benjamin Rosenzweig, Elena Silvestrova Rakhimov, Oracle PL/SQL by example Fifth Edition
2. Dr. P. S. Deshpande, SQL & PL / SQL for Oracle 11g Black Book

INTERNET OF THINGS(Professional Elective - 1)

M.Tech SE I Year I Sem

Objectives:
- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- To introduce the hardware and working principles of various sensors used for IoT
- To introduce the Python Scripting Language which is used in many IoT devices
- To introduce the Raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of web based services on IoT devices

UNIT I
Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs, IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates, Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle

UNIT II
Introduction to Python - Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling Python packages - JSON, XML, HTTPLib, URLLib, SMTPLib

UNIT III
IoT Physical Devices and Endpoints - Introduction to Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, reading input from pins.

UNIT IV
Controlling Hardware- Connecting LED, Buzzer, Switching High Power devices with transistors, Controlling AC Power devices with Relays, Controlling servo motor, speed control of DC Motor, Using unipolar and bipolar Stepper motors
Digital input- Sensing push switch, pull-up and pull-down resistors, Rotary encoder, Using keypad, Using RTC
Sensors: Light sensor, temperature sensor with thermistor, voltage sensor, ADC and ADC, Temperature and Humidity Sensor DHT11, Read Switch, Distance Measurement with ultrasound sensor

UNIT V
IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs Webserver – Web server for IoT, Cloud for IoT, Python web application framework Designing a RESTful web API

TEXT BOOK:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759
3. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

SOFTWARE PROJECT MANAGEMENT (Professional Elective - II)

Prerequisites: A course on "Software Engineering".
Course Objectives:
- To develop skills in software project management
- The topics include - software economics; software development life cycle; artifacts of the process; workflows; checkpoints; project organization and responsibilities; project control and process instrumentation;

Course Outcomes:
- Gain knowledge of software economics, phases in the life cycle of software development, project organization, project control and process instrumentation.
- Analyze the major and minor milestones, artifacts and metrics from management and technical perspective
- Design and develop software products using conventional and modern principles of software project management

UNIT - I
Conventional Software Management: The waterfall model, conventional software Management performance. Evolution of Software Economics: Software Economics, pragmatic software cost estimation.

UNIT - II
Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.
The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

UNIT - III
Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.
Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.
Model based software architectures: A Management perspective and technical perspective.
Work Flows of the process: Software process workflows, Iteration workflows**.**

UNIT - IV
Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.
Iterative Process Planning: work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.
Project Organizations and Responsibilities: Line-of-Business Organizations, Project

Organizations, evolution of Organizations. Process Automation: Automation Building blocks,

The Project Environment.

UNIT - V
Project Control and Process instrumentation: The seven core Metrics, Management
indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics

automation. **Tailoring the Process:** Process discriminates.
Future Software Project Management: modern Project Profiles, Next generation Software
economics, modern process transitions.
Case Study: The command Center Processing and Display system- Replacement (CCPDSR).

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

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

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

SOFTWARE METRICS (Professional Elective - II)

Course Objectives:
- To gain basic knowledge about metrics, measurement theory and related terminologies
- To learn measure the quality level of internal and external attributes of the software product
- To introduce the basics of software reliability and to illustrate how to perform planning, executing and testing for software reliability
- To explore various metrics and models of software reliability
- To compare various models of software reliability based on its application

Course Outcomes: Upon completion of the course, students shall be able to
- Identify and apply various software metrics, which determines the quality level of software
- Identify and evaluate the quality level of internal and external attributes of the software product
- Compare and Pick out the right reliability model for evaluating the software
- Evaluate the reliability of any given software product
- Design new metrics and reliability models for evaluating the quality level of the software based on the requirement.

UNIT - I
What Is Software Quality: Quality: Popular Views, Quality Professional Views, Software Quality, Total Quality Management, and Summary.
Fundamentals of Measurement Theory: Definition, Operational Definition, And Measurement, Level of Measurement, Some Basic Measures, Reliability and Validity, Measurement Errors, Be Careful with Correlation, Criteria for Causality, Summary.
Software Quality Metrics Overview: Product Quality Metrics, In Process Quality Metrics, Metrics for Software Maintenance, Examples for Metrics Programs, Collecting software Engineering Data.

UNIT - II
Applying the Seven Basic Quality Tools in Software Development: Ishikawa's Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts, Scatter Diagram, Control Chart, Cause, and Effect Diagram. **The Rayleigh Model:** Reliability Models, the Rayleigh Model Basic Assumptions, Implementation, Reliability and Predictive Validity.

UNIT - III
Complexity Metrics and Models: Lines of Code, Halstead's Software Science, Cyclomatic Complexity Syntactic Metrics, An Example of Module Design Metrics in Practice.
Metric and Lessons Learned for Object Oriented Projects: Object Oriented Concepts and Constructs, Design and Complexity Metrics, Productivity Metrics, Quality and Quality Management Metrics, Lessons Learned For object-oriented Projects.

UNIT - IV
Availability Metrics: Definition and Measurement of System Availability, Reliability Availability and Defect Rate, Collecting Customer Outage Data for Quality Improvement, In Process Metrics for Outage and Availability.
Conducting Software Project Assessment: Audit Ad Assessment, Software Process Maturity

Assessment and Software Project Assessment, Software Process Assessment A Proponed Software Project Assessment Method.

UNIT - V

Dos and Don'ts of Software Process Improvement: Measuring Process Maturity, Measuring Process Capability, Staged Versus Continuous Debating Religion, Measuring Levels Is Not Enough, Establishing the Alignment Principle, Take Time Getting Faster, keep it Simple or Face Decomplexification, Measuring the Value of Process Improvement, Measuring Process Compliance, Celebrate the Journey Not Just the Destination. Using Function Point Metrics to Measure Software Process Improvement: Software Process Improvement Sequences, Process Improvement Economies, Measuring Process Improvement at Activity Levels.

Text Books:
1. Norman E-Fentor and Share Lawrence Pflieger." Software Metrics". International Thomson Computer Press, 1997.
2. Stephen H Khan: Metrics and Models in Software Quality Engineering, Pearson 2nd edition 2013.

References:
1. S.A. Kelkar, "Software quality and Testing, PHI Learning, Pvt., Ltd., New Delhi 2012.
2. Watts S Humphrey, "Managing the Software Process", Pearson Education Inc, 2008.
3. Mary Beth Chrissis, Mike Konrad and Sandy Shrum, "CMMI", Pearson Education (Singapore) Pvt. Ltd., 2003
4. Philip B Crosby, " Quality is Free: The Art of Making Quality Certain ", Mass Market, 1992.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)
SOFTWARE RELIABILITY (Professional Elective - II)

COURSE OBJECTIVES:
1. To learn about the engineering techniques for developing and maintaining reliable software systems.
2. To measure the reliability of software systems.
3. To understand about fault prevention, fault removal, fault tolerance and failure forecasting in software systems.
4. To learn different time dependent and time independent software reliability models and design reliability models for software systems.

UNIT I
Basic Ideas of Software Reliability, Hardware reliability vs. Software reliability, Reliability metrics, Failure and Faults – Prevention, Removal, Tolerance, Forecast, Dependability Concept – Failure Behaviour, Characteristics, Maintenance Policy, Reliability and Availability Modeling, Reliability Evaluation Testing methods, Limits, Starvation, Coverage, Filtering, Microscopic Model of Software Risk.

UNIT   II
Computation of software reliability, Functional and Operational Profile, Operational Profiles – Difficulties, Customer Type, User Type, System Mode, Test Selection - Selecting Operations, Regression Test.

UNIT   III
Classes of software reliability Models, Time Dependent Software Reliability Models: Time between failure reliability Models, Fault Counting Reliability Models.

UNIT IV
Time Independent Software Reliability Models: Fault injection model of Software Reliability, Input Domain Reliability Model, Orthogonal defect classification, Software availability Models. Software Reliability Modeling: A general procedure for reliability modeling.

UNIT V
Short and Long Term Prediction, Model Accuracy, Analysing Predictive Accuracy – Outcomes, PLR, U and Y Plot, Errors and Inaccuracy, Recalibration – Detecting Bias, Different Techniques, Power of Recalibration, Limitations in Present Techniques, Improvements.

TEXT BOOKS:
1.   J.D. Musa, *Software Reliability Engineering*, McGraw Hill, New York , 2004
2.   H. Pham, *Software Reliability*, Springer Verlag, New York , 2000

REFERENCE BOOK:
1. Patric D. T.O Connor, *Practical Reliability Engineering, 4th Edition*, John Wesley & Sons , 2003
2. D. Reled, *Software Reliability Methods*, Springer Verlag, New York , 2001

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)
SOFTWARE AGENTS (Professional Elective - II)

OBJECTIVES:
The objective of this course is to make students to:
1. Learn the principles and fundamentals of designing agents
2. Study the architecture design of different agents.
3. Learn to do detailed design of the agents
4. Explore the role of agents in assisting the users in day to day activities

UNIT I
INTRODUCTION
Agents and Multi Agent Systems-Intelligent Agent-Concepts of Building Agent–Situated Agents–Proactive and Reactive agents-Challenging Agent Environment-Social Agents-Agent Execution Cycle-Prometheus Methodology - Guidelines for using Prometheus - Agent Oriented Methodologies - System Specification – Goal Specification – Functionalities – Scenario Development – Interface Description – Checking for Completeness and Consistency.

UNIT II
ARCHITECTURAL DESIGN
Agent Types - Grouping Functionalities - Agent Coupling - Develop Agent Descriptors – Interactions - Interaction Diagram from Scenarios - Interaction Protocol from Interaction Diagram - Develop Protocol and Message Descriptors – Architectural Design - Identifying the Boundaries of Agent System – Percepts and Action - Shared Data Objects – System Overview – Checking for Completeness and Consistency.

UNIT III
DETAILED DESIGN
Capability Diagrams – Sub Tasks - Alternative Programs – Events and Messages – Action and Percept Detailed Design – Data – Develop and Refine Descriptors – Missing or Redundant Items - Consistency between Artifacts – Important Scenarios - Implementing Agent Systems - Agent Platform – JACK

UNIT IV
AGENTS AND USER EXPERIENCE
Interact with Agents - Agents from Direct Manipulation to Delegation – Interface Agents - Designing Agents - Direct Manipulation versus Agents - Agents for Information Sharing and Coordination - Agents that Reduce Work and Information Overload - KidSim: Programming Agents without a Programming Language.

UNIT V
AGENTS FOR INTELLIGENT ASSISTANCE
Computer Characters - Software Agents for Cooperative Learning – Integrated Agents – Agent Oriented Programming - KQML as an Agent Communication Language - Agent Based Framework for Interoperability - Agents for Information Gathering – KaoS - Communicative Actions for Artificial Agents – Mobile Agents.

TEXT BOOKS:

1. Lin Padgham and Michael Winikoff. Developing Intelligent Agent Systems: A Practical Guide: John Wiley & sons Publication, 2004.
2. Jeffrey M. Bradshaw. Software Agents: MIT Press , 1997. REFERENCE BOOK: Steven F. Rai ls Back and Volker Grimm. Agent – Based and Individual Based modeling: A Practical Introduction: Princeton University Press, 2012.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

ADVANCED DATA STRUCTURES LAB (Lab - I)

Prerequisites: A course on Computer Programming & Data Structures

Course Objectives:
1. Introduces the basic concepts of Abstract Data Types.
2. Reviews basic data structures such as stacks and queues.
3. Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs, and B-trees.
4. Introduces sorting and pattern matching algorithms.

Course Outcomes:
1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and know the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and B-trees.

List of Programs
1. Write a 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.

2. Write a program for implementing the following sorting methods:
   a) Merge sort   b) Heap sort   c) Quick sort

3. Write a program to perform the following operations:
   a) Insert an element into a B- tree.
   b) Delete an element from a B- tree.
   c) Search for a key element in a B- tree.

4. Write a program to perform the following operations:
   a) Insert an element into a Min-Max heap
   b) Delete an element from a Min-Max heap
   c) Search for a key element in a Min-Max heap

5. Write a program to perform the following operations:
   a) Insert an element into a Lefiist tree
   b) Delete an element from a Leftist tree
   c) Search for a key element in a Leftist tree

6. Write a program to perform the following operations:
    a) Insert an element into a binomial heap
    b) Delete an element from a binomial heap.
    c) Search for a key element in a binomial heap

7. Write a program to perform the following operations:
    a) Insert an element into a AVL tree.
    b) Delete an element from a AVL search tree.
    c) Search for a key element in a AVL search tree.

8. Write a program to perform the following operations:
    a) Insert an element into a Red-Black tree.
    b) Delete an element from a Red-Black tree.
    c) Search for a key element in a Red-Black tree.

9. Write a program to implement all the functions of a dictionary using hashing.

10. Write a program for implementing Knuth-Morris-Pratt pattern matching algorithm.

11. Write a program for implementing Brute Force pattern matching algorithm.

12. Write a program for implementing Boyer pattern matching algorithm.

TEXT BOOKS:
1. Fundamentals of Data structures in C, E.Horowitz, S.Sahni and Susan Anderson Freed, 2nd Edition, Universities Press
2. Data Structures Using C – A.S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson education.
3. Introduction to Data Structures in C, Ashok Kamthane, 1st Edition, Pearson.

REFERENCES:
1. The C Programming Language, B.W. Kernighan, Dennis M.Ritchie, PHI/Pearson Education
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press
3. Data structures: A Pseudocode Approach with C, R.F.Gilberg And B.A.Forouzan, 2nd Edition, Cengage Learning.

ENTERPRISE CLOUD CONCEPTS LAB (Professional Elective - I Lab)

M.Tech SE I Year I Sem.	**L**	**T**	**P**	**C**
	0	**0**	**4**	**2**

Course Objectives: Knowledge on significance of cloud computing and its fundamental concepts and models.

Course Outcomes:
1. Understand importance of cloud architecture
2. Illustrating the fundamental concepts of cloud security
3. Analyze various cloud computing mechanisms
4. Understanding the architecture and working of cloud computing.

List of Experiments:
1. Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute SimplePrograms
3. Install Google App Engine. Create a hello world app and other simple web applications using python/java..
4. Find a procedure to transfer the files from one virtual machine to another virtual machine.
5. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
6. Install Hadoop single node cluster and run simple applications like word count.

E-Resources:
1. https://www.iitk.ac.in/nt/faq/vbox.htm
2.https://www.google.com/urlsa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjqrNG0za73Ah XZt1YBHZ21DWEQFnoECAMQAQ&url=http%3A%2F%2Fwww.cs.columbia.edu%2F~sedwards%2F classes%2F2015%2F1102-fall%2Flinuxvm.pdf&usg=AOvVaw3xZPuF5xVgk-AQnBRsTtHz
3. https://www.cloudsimtutorials.online/cloudsim/
4.https://edwardsamuel.wordpress.com/2014/10/25/tutorial-creating-openstack-instance-in-trystack/
5. https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster

INTERNET TECHNOLOGIES AND SERVICES LAB(PE-1 LAB)
M.Tech SE I Year I Sem.

Objectives:

● Write syntactically correct HTTP messages and describe the semantics of common HTTP methods and header fields
● Discuss differences between URIs, URNs, and URLs, and demonstrate a detailed understanding of http-scheme URLs, both relative and absolute
● Describe the actions, including those related to the cache, performed by a browser in the process of visiting a Web address
● Install a web server and perform basic administrative procedures, such as tuning communication parameters, denying access to certain domains, and interpreting an access log
● Write a valid standards-conformant HTML document involving a variety of element types, including hyperlinks, images, lists, tables, and forms
● Use CSS to implement a variety of presentation effects in HTML and XML documents, including explicit positioning of elements
● Demonstrate techniques for improving the accessibility of an HTML document

List of Sample Problems:

o **Internet Technologies**

1. Develop static pages (using Only HTML) of an online Book store. The pages should resemble: www.amazon.com the website should consist the following pages.
   Home page, Registration and user Login
   User Profile Page, Books catalog
   Shopping Cart, Payment By credit card
   Order Conformation
2. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
3. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
4. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
5. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.
6. Implement the "Hello World!" program using JSP Struts Framework.
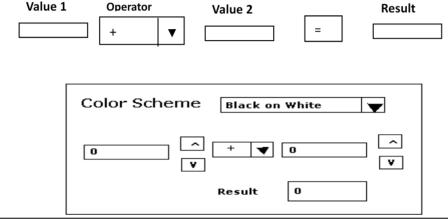
Additional Assignment Problems
1. Write an HTML page including any required Javascript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show "out of range" and if it is not a number, it should show "not a number" message in the result box.
2. Write a java swing application that takes a text file name as input and counts the characters, words and lines in the file. Words are separated with white space characters and lines are separated with new line character.
3. Write a simple calculator servlet that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operation performed on the operands. It should check in a database if the same expression is already computed and if so, just return the value from database. Use MySQL or PostgreSQL
4. Write an HTML page that contains a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
5. Write a servlet that takes name and age from an HTML page. If the age is less than 18, it should send a page with "Hello <name>, you are not authorized to visit this site" message, where <name>

should be replaced with the entered name. Otherwise it should send "Welcome <name> to this site" message.

6. Write a calculator program in HTML that performs basic arithmetic operations (+, -, /, * and %). Use CSS to change the foreground and background color of the values, buttons and result display area separately. Validate the input strings using JavaScript regular expressions. Handle any special cases like division with zero reasonably. The screen may look similar to the following:

7. Write a Java program that creates a calculator GUI, as shown in figure. Extra components may be added for convenience:

The Color Scheme may be Black on White or Blue on Yellow (selectable) and accordingly all components colors must be changed. The values can be either entered or increased or decreased by a step of 10. The operators are +, -, / and * (selectable). Once any change takes place, the result must be automatically computed by the program.

2. Write a Java Application that will read an XML file that contains personal information (Name, Mobile Number, age and place. It reads the information using SAX parser. After reading the information, it shows two input Text Fields in a window, one for tag name and the other for value. Once these two values are given, it should list all the records in the XML file that match the value of the given field in a text area (result box). For example, if the two text boxes are entered with "name" and "ABCD" then it should show all the records for which name is "ABCD"? An Illustration is given below that takes a mobile number and lists all the records that have the same mobile number.

3. Consider the following web application for implementation:
    The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
    If name and password matches, serves a welcome page with user's full name.
    If name matches and password doesn't match, then serves "password mismatch" page
    If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application in:
    1. Pure JSP
    2. Pure Servlets
    3. Struts Framework
    4. Implement a simple arithmetic calculator with +, -, /, *, % and = operations using Struts Framework The number of times the calculator is used should be displayed at the bottom(use session variable).

iii)Internet Technologies and Services Lab - Additional Problems

1. Create a web Service in Java that takes two city names from the user and returns the distance between these two from data available from a table in MySql.Write a java and a C# client which use the above service
2. Write a Java program that takes a file as input and encrypts it using DES encryption. The program should check if the file exists and its size is not zero.
3. Write a Java program that generates a key pair and encrypts a given file using RSA algorithm
4. Write a Java program that finds digest value of a given string
5. Consider the following xml file for encryption
    <?xml   version="1.0">   <transaction>   <from>12345</from>   <to>54321</to> <amount>10000</amount>
    <secretcode>abc123</secretcode> <checksum></checksum> </transaction>
    Replace <from> and <to> values with the RSA encrypted values represented with base64 encoding assuming that the public key is available in a file in local directory "pubkey.dat". Encrypt <secretcode> with AES algorithm with a password 'secret'. The checksum of all the field values concatenated with a delimiter character '+' will be inserted in the checksum and the xml file is written to encrypted.xml file.
6.   Assume that a file 'config.xml', which has the following information:
    <users>
    <user> <name>abc</name> <pwd>pwd123</pwd> <role>admin</role> <md5>xxx</md5> </user>
    <user> <name>def</name> <pwd>pwd123</pwd> <role>guest</role> <md5>xxx</md5> </user>
    </users>
    Replace name and role with DES encrypted values and pwd with RSA encrypted values (represent the values with base64 encoding). The public key is available in "public.key" file in current directory. Replace xxx with respective MD5 values of all the fields for each user. Write the resulting file back to config.xml.
7. Write an HTML page that gives 3 multiple choice (a,b,c and d) questions from a set of 5 preloaded questions randomly. After each question is answered change the color of the question to either green or blue using CSS. Finally on clicking OK button that is provided, the score should be displayed as a pop-up window. Use Java Script for dynamic content
8. Write an HTML page that has 3 countries on the left side ("USA", "UK" and "INDIA") and on the right side of each country, there is a pull-down menu that contains the following entries: ("Select Answer", "New Delhi", "Washington" and "London"). The user will match the Countries with their respective capitals by selecting an item from the menu. The user chooses all the three answers (whether right or wrong). Then colors of the countries should be changed either to green or to red depending on the answer. Use CSS for changing color
9. Write an HTML Page that can be used for registering the candidates for an entrance test. The fields are: name, age, qualifying examination (diploma or 10+2), stream in qualifying examination. If qualifying examination is "diploma", the stream can be "Electrical", "Mechanical" or "Civil". If the qualifying examination is 10+2, the stream can be "MPC" or "BPC". Validate the name to accept only characters and spaces.

10. Write an HTML page that has two selection menus. The first menu contains the states ("AP", "TN" and "KN") and depending on the selection the second menu should show the following items: "Hyderabad", "Vijayawada", "Kurnool" for AP, "Chennai", "Salem", "Madurai" for TN and "Bangalore", "Bellary", "Mysore" for KN.

11. Write an HTML page that has phone buttons 0 to 9 and a text box that shows the dialed number. If 00 is pressed at the beginning, it should be replaced with a + symbol in the text box. If the number is not a valid international number (+ followed by country code and 10 digit phone number) the color of the display should be red and it should turn to green when the number is valid. Consider only "+91, +1 and +44 as valid country codes. Use CSS for defining colors.

12. Write an HTML page that has a text box for phone number or Name. If a number is entered in the box the name should be displayed next to the number. If 00 is pressed at the beginning, it should be

replaced with a + symbol in the text box. If a name is entered in the text box, it should show the number next to the name. If the corresponding value is not found, show it in red and show it in green otherwise. Use CSS for colors. Store at least 5 names and numbers in the script for testing

13. A library consists of 10 titles and each title has a given number of books initially. A student can take or return a book by entering his/her HTNo as user ID and a given password. If there are at least two books, the book is issued and the balance is modified accordingly.
    (a) Use RDBMS and implement it with JSP.
    (b) Use XML File for data and Implement it with JSP
    (c) Use RDBMS and implement it with Servlets
    (d) Use XML File for data and Implement it with Servlets

14. A Bus Reservation System contains the details of a bus seat plan for 40 seats in 2x2 per row arrangement, where the seats are numbered from 1 to 40 from first row to last row. The customer can visit the website and can reserve a ticket of his choice if available by entering his details (Name, Address, Gender and Age). The customer can cancel the ticket by entering the seat number and his name as entered for reservation.
    (a) Use RDBMS and implement it with JSP.
    (b) Use XML File for data and Implement it with JSP
    (c) Use RDBMS and implement it with Servlets
    (d) Use XML File for data and Implement it with Servlets

15. Implement a simple messaging system with the following details:
    When a student logs in with his/her HTNO and a given password, they should get all the messages posted to him/her giving the ID of sender and the actual message. Each message may be separated with a ruler. There should be a provision for the user to send a message to any number of users by giving the IDs separated with commas in the "To" text box.
    (a) Use RDBMS and implement it with JSP.
    (b) Use XML File for data and Implement it with JSP
    (c) Use RDBMS and implement it with Servlets
    (d) Use XML File for data and Implement it with Servlets.

16. There is an image of 600x100 size which can be logically divided into 12 button areas with labels (0-9, +, =). Write a javascript calculator program that uses this image as input virtual keyboard and three text areas for two input numbers and result of sum of these numbers. Add a CSS that can be used to change the colors of text and background of text areas and the page. The input numbers can be up to 4 digits each.

17. Develop a web application that takes user name and password as input and compares them with those available in an xml user database. If they match, it should display the welcome page that contains the user's full name and last used date and time retrieved from a client cookie. On logout it stores new time to the cookie and displays a goodbye page. If authentication fails, it should store the attempt number to the client cookie and displays an error page. Add necessary CSS that takes care of the font, color of foreground and background.

18. A web application has the following specifications:
    The first page (Login page) should have a login screen where the user gives the login name and password. Both fields must be validated on client side for a minimum length of 4 characters, name should be lower case a-z characters only and password should contain at least one digit. On submitting these values, the server should validate them with a MySQL database and if failed, show the login page along with a message saying "Login Name or Password Mismatch" in Red color below the main heading and above the form. If successful, show a welcome page with the user's full name (taken from database) and and a link to Logout. On logout, a good bye page is displayed with

the total time of usage (Logout time – login time). Specify the Schema details of table and web.xml file contents.
Implement it using (a) JSP Pages (b) Servlets (c) Struts

19. Design a struts based web portal for an international conference with following specifications:
The welcome page should give the details of the conference and a link to login. If login fails, direct them back for re-login and also provide a link for registration. On successful registration/login, the user will be directed to a page where s/he can see the status (accepted/rejected) of their already submitted papers followed by a form for submitting a doc file to the conference. Provide a logout button on all pages including the home page, once the user logs in. Implement validation framework to check that the user name is in the form of CCDDCC and password is in the form of (CCSDDD) (C for character, S for special character (one of @, #, $, %, ^, & and !) and D for digit)., Database should be accessed through Connection Pool for MySql for user information. Provide scope for internationalization in future. Assume any missing information and mention it first.

DATABASE PROGRAMMING LAB (Lab – PE1)

M.Tech SE I Year I Sem.   **L   T   P C**
                                                                               0   0   4 2

Course Objectives:
1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

Course Outcomes:
1. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

List of Experiments:
1. Write a Pl/SQL program using FOR loop to insert ten rows into a database table.
2. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID), write a cursor to select the five highest paid employees from the table.
3. Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction might be done.
4. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.
5. Write a PL/SQL program to demonstrate Exceptions.
6. Write a PL/SQL program to demonstrate Cursors.
7. Write a PL/SQL program to demonstrate Functions.
8. Write a PL/SQL program to demonstrate Packages.
9. Write PL/SQL queries to create Procedures.
10. Write PL/SQL queries to create Triggers.

INTERNET OF THINGS LAB(PE-1 LAB)

M.Tech SE I Year I Sem.

Python Basic exercises

1. Write a Python program that reads 10 integers from keyboard and prints the average of even numbers and odd numbers separately

2. Write a Python program that prints the grade of a student when internal and external marks are given. A candidate is declared Failed (Grade = F), if Total marks < 50 or External marks < 25.

If a candidate is passed, then Grade is given as follows:

Condition	Grade
50 <= total marks < 60	E
60 <= total marks < 70	D
70 <= total marks < 80	C
80 <= total marks < 90	B
total marks >= 70	A

3. Create a table in MySQL that stores the status of devices in a house with the following data (Device ID, Device Name and Device State, last altered date and time). Now write a Python program that reads and alters the state of a given device. The date format is "YYYY-MM-DD:HH-mm-ss" where mm is minutes and ss is seconds.

4. Write a Python program that loads all the states of the devices into a dictionary from the table mentioned above.

5. Write a Python program that sorts the device states based on the last altered time

6. Write a Python program that reads a string from keyboard and prints the count of each alphabet in the string.

7. Write a Python program that reads a page from internet and prints it on the screen.

8. Write a Python program that reads and modifies an XML file

9. Write a Python program that reads and alters JSON data from a database table

10. Write a client-server Python program that uses socket connection to implement a time server. The client will connect to the server and the server sends the current time as "YYYY-MM-DD:HH-mm-ss" format. This value should be printed on the client side.

11. Write a Python program that generates 10 random numbers and stores them in a text file one per line. Now write another Python program that reads this data into a list and shows them

12. Write a program that reads key-value pair data from a file and stores them in a database table

13. Write a Python program that reads a time string in the format of "YYYY-MM-DD:HH-mm-ss" and prints its components separately.

14. Write a Python program that reads data from a table and writes it to a text file using tab as field separator and new line as record separator and vice versa.

Raspberry Pi Experiments: *Use Raspberry Pi for all the experiments*

1. Connect an LED to GPIO pin 25 and control it through command line

2. Connect an LED to GPIO pin 24 and a Switch to GPIO 25 and control the LED with the switch. The state of LED should toggle with every press of the switch

3. Use DHT11 temperature sensor and print the temperature and humidity of the room with an interval of 15 seconds

4. Use joystick and display the direction on the screen

5. Use Light Dependent Resistor (LDR) and control an LED that should switch-on/off depending on the light.

6. Create a traffic light signal with three colored lights (Red, Orange and Green) with a duty cycle of 5-2-10 seconds.

7. User rotary encoder and print the position of the shaft on the console

8. Control a servo motor angle that is taken from the keyboard

9. Switch on and switch of a DC motor based on the position of a switch

10. Convert an analog voltage to digital value and show it on the screen.

11. Create a door lock application using a reed switch and magnet and give a beep when the door is opened.

12. Control a 230V device (Bulb) with Raspberry Pi using a relay

13. Control a 230V device using a threshold temperature, using temperature sensor.

14. Simulate an earthquake alarm using vibration sensor and give an alarm when vibration is detected.

15. Create an application that has three LEDs (Red, Green and white). The LEDs should follow the cycle (All Off, Red On, Green On, White On) for each clap (use sound sensor).

16. Create a web application for the above applications wherever possible with suitable modifications to get input and to send output.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – I Semester (Software Engineering)

RESEARCH METHODOLOGY & IPR

Prerequisite: None

Course Objectives:
- To understand the research problem
- To know the literature studies, plagiarism and ethics
- To get the knowledge about technical writing
- To analyze the nature of intellectual property rights and new developments
- To know the patent rights

Course Outcomes: At the end of this course, students will be able to
- Understand research problem formulation.
- Analyze research related information
- Follow research ethics
- Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
- Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular.
- Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

UNIT-I:
Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations

UNIT-II:
Effective literature studies approaches, analysis, Plagiarism, Research ethics

UNIT-III:
Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee

UNIT-IV:
Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

UNIT-V:
Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications. New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional

knowledge Case Studies, IPR and IITs.

TEXT BOOKS:
1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'"
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"

REFERENCES:
1. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners"
2. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd ,2007.
3. Mayall, "Industrial Design", McGraw Hill, 1992.
4. Niebel, "Product Design", McGraw Hill, 1974.
5. Asimov, "Introduction to Design", Prentice Hall, 1962.
6. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age", 2016.
7. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)
Design Patterns(PC– III)

Prerequisites:
1. A Course on Software Engineering"
2. A Course on "Object Oriented Programming Through Java"

Course Objectives:
- The aim of the course is to appreciate the idea behind Design Patterns in handling common problems faced during building an application
- This course covers all pattern types from creational to structural, behavioral to concurrency and highlights the scenarios when one pattern must be chosen over others.

Course Outcomes:
- Create software designs that are scalable and easily maintainable
- Understand the best use of Object-Oriented concepts for creating truly OOP programs
- Use creational design patterns in software design for class instantiation
- Use structural design patterns for better class and object composition
- Use behavioral patterns for better organization and communication between the objects
- Use refactoring to compose the methods for proper code packaging
- Use refactoring to better organize the class responsibilities of current code

UNIT - I:
Introduction: What is a design pattern? design patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

UNIT - II:
Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary

UNIT - III:
Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

UNIT - IV:
Structural Pattern: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy

UNIT - V:
Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.

Text Book:

1. Design Patterns, Erich Gamma, Pearson Education

Reference Books:

1. Pattern's in Java, Vol –I, Mark Grand, Wiley DreamTech.
2. Patterns in Java, Vol-II, Mark Grand, Wiley DreamTech.
3. Java Enterprise Design Patterns Vol-III, Mark Grand, Wiley DreamTech.
   Head First Design Patterns, Eric Freeman, O'reilypublications

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
MACHINE LEARNING (PC – IV)

M.Tech SE I Year II Sem.   L   T   P   C
                                                                      3   0   0   3

Course Objectives:
1. To introduce students to the basic concepts and techniques of Machine Learning.
2. To have a thorough understanding of the Supervised and Unsupervised learning techniques
3. To study the various probability based learning techniques
4. To understand graphical models of machine learning algorithms

Course Outcomes: Upon completion of the course, the students will be able to:
1. Distinguish between, supervised, unsupervised and semi-supervised learning
2. Apply the apt machine learning strategy for any given problem
3. Suggest supervised, unsupervised or semi-supervised learning algorithms for any given problem
4. Design systems that use the appropriate graph models of machine learning
5. Modify existing machine learning algorithms to improve classification efficiency

UNIT - I:
Introduction: Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Design a Learning System – Perspectives and Issues in Machine Learning – Concept Learning Task – Concept Learning as Search – Finding a Maximally Specific Hypothesis – Version Spaces and the Candidate Elimination Algorithm – Linear Discriminants: – Perceptron – Linear Separability – Linear Regression.

UNIT - II:
Linear Models: Multi-layer Perceptron– Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perceptron in Practice – Examples of using the MLP – Overview – Deriving Back-Propagation – Radial Basis Functions and Splines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

UNIT - III:
Tree and Probabilistic Models: Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Basic Statistics – Gaussian Mixture Models – Nearest Neighbor Methods – Unsupervised Learning – K means Algorithms

UNIT - IV:
Dimensionality Reduction and Evolutionary Models: Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis – Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap – Least Squares Optimization – Evolutionary Learning – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms – Reinforcement Learning – Overview – Getting Lost Example

UNIT - V:
Graphical Models: Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain Monte Carlo – Graphical Models – Bayesian Networks – Markov Random Fields – Hidden Markov Models – Tracking Methods

TEXT BOOKS:
1. Stephen Marsland, ―Machine Learning – An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
2. Tom M Mitchell, ―Machine Learning, First Edition, McGraw Hill Education, 2013.

REFERENCES:
1. Peter Flach, ―Machine Learning: The Art and Science of Algorithms that Make Sense of Dataǁ, First Edition, Cambridge University Press, 2012.
2. Jason Bell, ―Machine learning – Hands on for Developers and Technical Professionalsǁ, First Edition, Wiley, 2014
3. Ethem Alpaydin, ―Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series), Third Edition, MIT Press, 2014

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

SOFTWARE TESTING METHODOLOGIES (Professional Elective – III)

Prerequisites: A course on "Software Engineering"

Course Objectives:
- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- To develop skills in software test automation and management using latest tools.

Course Outcomes:
- Ability to design and develop the best test strategies in accordance to the development models
- Acquire skills to perform dataflow testing, domain testing, logic testing.

UNIT - I
Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: - Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II:
Transaction Flow Testing: transaction flows, transaction flow testing techniques.
Dataflow testing: Basics of data flow testing, strategies in data flow testing, application of dataflow testing.
Domain Testing: domains and paths, nice & ugly domains, domain testing, domains and interfaces' testing, domain and interface testing, domains and testability.

UNIT - III:
Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV:
State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V:
Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

Text Books:
1. Software Testing techniques - Boris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K.V.K.K. Prasad, Dreamtech.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

MOBILE APPLICATION AND API DEVELOPMENT (Professional Elective – III)

Course Objectives: Your studies will enable you to develop:
- Structure your app, design flexible and interactive interfaces, run services in the background, make your app work on various smartphones and tablets
- Build efficient and secure RESTful web APIs in Java
- Design solutions to produce, consume and visualize RESTful web services using WADL, RAML, and Swagger
- Familiarize the role of RESTful APIs usage in emerging technology trends like Cloud, IoT, Social Media.
- Introduce yourself to the RESTful software architectural style and the REST API design principles
- Make use of the JSR 353 API, JSR 374 API, JSR 367 API and Jackson API for JSON processing q
- Build portable RESTful web APIs, making use of the JAX-RS 2.1 API
- Simplify API development using the Jersey and RESTEasy extension APIs
- Secure your RESTful web services with various authentication and authorization mechanisms
- Get to grips with the various metadata solutions to describe, produce, and consume RESTful web services
- Understand the design and coding guidelines to build well-performing RESTful APIs
- See how the role of RESTful web services changes with emerging technologies and trends

UNIT - I
Introduction to RESTful API Architecture: Introduction to REST Architecture style, architectural elements, methods. Java API for JSON processing

UNIT - II
JAX-RS API: JAX-RS API Annotations, Building API, advance features, filters and interceptors, Jersey and HATEOAS models, REST frameworks and extensions.

UNIT - III
Design and Securing REST API: Securing API using OAuth, RESTful modelling language, Swagger, REST API design guidelines, package and deploy JAX-RS applications.

UNIT - IV
Android Development Basics: Understanding Android Platform, Setting up the environment, pre-requisites for building application

UNIT - V
Build Mobile Apps using API: Setting up the Android environment, building basic app, activities and layout, run the app, watch the progress and go-live.

Text Books:
1. Best for Visual Learners: Head First Android Development: A Brain-Friendly Guide by Dawn Griffitha and David Griffitha
2. RESTful Java Web Services: A pragmatic guide to designing and building RESTful APIs using Java, 3rd Edition by Bogunuva Mohanram Balachandar

Reference Books:
Best for Programmers With Java Experience: Android Programming: The Big Nerd Ranch Guide
1. Best for Quick Answers: Android Cookbook: Problems and Solutions for Android Developers
2. Building Progressive Web Apps- Bringing the Power of Native to the Browser

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

FULL STACK DEVELOPMENT

(Professional Elective – III)

Course Objectives:
- To learn the core concepts of both the frontend and backend programming course. 2. To get familiar with the latest web development technologies.
- To learn all about databases.
- To learn complete web development process
- To provide an in-depth study of the various web development tools

Course Outcomes:
- Develop a fully functioning website and deploy on a web server.
- Gain Knowledge about the front end and back end Tools
- Find and use code packages based on their documentation to produce working results in a project.
- Create web pages that function using external data.

UNIT-I:
Web Development Basics: Web development Basics - HTML & Web servers Shell - UNIX CLI Version control - Git & Github HTML, CSS

UNIT-II:
Frontend Development: Javascript basics OOPS Aspects of JavaScript Memory usage and Functions inJS AJAX for data exchange with server jQuery Framework jQuery events, UI components etc. JSON data format.

UNIT-III:
REACT JS: Introduction to React React Router and Single Page Applications React Forms, Flow Architecture and Introduction to Redux More Redux and Client-Server Communication

UNIT-IV:
Java Web Development: JAVA PROGRAMMING BASICS, Model View Controller (MVC) Pattern MVC Architecture using Spring RESTful API using Spring Framework Building an application using Maven

UNIT-V:
Databases & Deployment: Relational schemas and normalization Structured Query Language (SQL) Data persistence using Spring JDBC Agile development principles and deploying application in Cloud

TEXT BOOKS:
1. Web Design with HTML, CSS, JavaScript and JQuery Set Book by Jon Duckett Professional JavaScript for Web Developers Book by Nicholas C. Zakas
2. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon
3. Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB. Copyright © 2015 BY Azat Mardan

REFERENCES:
1. Full-Stack JavaScript Development by Eric Bush.
2. Mastering Full Stack React Web Development Paperback – April 28, 2017 by Tomasz Dyl, Kamil Przeorski, Maciej Czarnecki

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

FUNCTIONAL PROGRAMMING (Professional Elective - III)

M.Tech SE I Year II Sem.   **L   T   P   C**
                                                                               3   0   0   3

Course Objectives
1. Understand the concepts and terms used to describe languages that support the imperative, functional, object-oriented, and logic programming paradigms.
2. Solve problems using the functional paradigm.
3. Solve problems using the object-oriented paradigm.
4. Solve problems using the logic programming paradigm.

Course Outcomes: At the end of the course the student will be able to:
1. Write programs in a functional style.
2. Reason formally about functional programs.
3. Use polymorphism and higher-order functions.
4. Reason about the time and space complexity of programs.

UNIT - I
Functional Programming: Introduction, Differences between Functional Programming and Object-Oriented Programming, concepts of functional programming, Functional Programming in Python: Introduction to Python, Built-in Functions, Dictionary Methods, String Methods, LIST/ARRAY Methods,Tuple Methods, Set Methods

UNIT - II
Python Exceptions, File Handling, Tuple Methods, Defining Iteration, Conditional Iterations, Random Module, Math Module, CMath Module, Python File I/O

UNIT - III
Python Sending Mail, Python CSV, Python OOP Concepts, Python Iterators, Python Generators, Python Decorators, Python Database Connections

UNIT - IV
Introduction to Haskell and Laziness, Structure, Modularity, Maintainability, Polymorphism, higher order functions, strings & characters, lazy evaluation, Data Types using Patterns

UNIT - V
LISP Programming: Basic LISP Programming, Data Types, Functions, Editing, Loading, Compiling LISP Programs, Control Structures: Recursions and Conditionals, LISTS, SETS, Structural Recursion with LISTS, Symbols

TEXT BOOKS:
1. The Haskell School of Expression: Learning Functional Programming through Multimedia, Paul Hudak.
2. Functional Programming in Python, David Mertz, O'Reilly Media.
3. LISP, Patrick Henry Winston, Bertbold Klaus Paul Horn, Pearson Education

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

Software Requirements & Estimation (Professional Elective - IV)

Course Objectives:
- Students will demonstrate knowledge of the distinction between critical and non- critical systems.
- Students will demonstrate the ability to manage a project including planning, scheduling and risk assessment/management.
- Students will author a software requirements document.
- Students will demonstrate an understanding of the proper contents of a software requirements document.
- Students will author a formal specification for a software system.
- Students will demonstrate an understanding of distributed system architectures and application architectures.
- Students will demonstrate an understanding of the differences between real-time and non-real time systems.
- Students will demonstrate proficiency in rapid software development techniques.
- Students will demonstrate proficiency in software development cost estimation
- Students will author a software testing plan.

UNIT - I
Software Requirements: What and Why 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 Requirements Management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain **Software Requirements Modeling** Use Case Modeling, Analysis Models, Dataflow diagram, state transition diagram, class diagrams, Object analysis, Problem Frames

UNIT - III
Software Estimation 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, Conversion between size measures.

UNIT - IV

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

UNIT - V

Tools for Requirements Management and Estimation Requirements Management Tools: 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

Text Book:
1. Software Requirements and Estimation by *Rajesh Naik and Swapna Kishore*, Tata Mc Graw Hill.

Reference Books:
1. Software Requirements by Karl E. Weigers, Microsoft Press.
2. Managing Software Requirements, Dean Leffingwell & Don Widrig, Pearson Education, 2003.
3. Mastering the requirements process, second edition, Suzanne Robertson & James Robertson, Pearson Education, 2006.
4. Estimating Software Costs, Second edition, Capers Jones, TMH, 2007.
5. Practical Software Estimation, M.A. Parthasarathy, Pearson Education, 2007.
6. Measuring the software process, William A. Florac & Anita D. Carleton, Pearson Education, 1999.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

SECURE SOFTWARE ENGINEERING (Professional Elective - IV)

Course Objectives:
- Students will demonstrate knowledge of the distinction between critical and non-critical systems.
- Students will demonstrate the ability to manage a project including planning, scheduling and risk assessment/management.
- Students will author a software requirements document.
- Students will demonstrate an understanding of the proper contents of a software requirements document.
- Students will author a formal specification for a software system.
- Students will demonstrate an understanding of distributed system architectures and application architectures.
- Students will demonstrate an understanding of the differences between real-time and non-real time systems.
- Students will demonstrate proficiency in rapid software development techniques.
- Students will be able to identify specific components of a software design that can be targeted for reuse.
- Students will demonstrate proficiency in software development cost estimation.
- Students will author a software testing plan.

UNIT – I
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
What Makes Software Secure: Properties of Secure Software, Influencing the security properties of software, Asserting and specifying the desired security properties?

UNIT – II
Requirements Engineering for secure software: Introduction, the SQUARE process Model, Requirements elicitation and prioritization

UNIT – III
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: Code analysis, Software Security testing, Security testing considerations throughput the SDLC

UNIT – IV
Security and Complexity: System Assembly Challenges: introduction, security failures, functional and attacker perspectives for security analysis, system complexity drivers and security

UNIT – V

Governance and Managing for More Secure Software: Governance and security, Adopting an enterprise software security framework, How much security is enough?, Security and project management, Maturity of Practice

Text book:
1. Software Security Engineering: Julia H. Allen, Pearson Education

References:
1. Developing Secure Software: Jason Grembi, Cengage Learning
2. Software Security: Richard Sinn, Cengage Learning

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

OBJECT ORIENTED SOFTWARE ENGINEERING (Professional Elective - IV)

UNIT – I
Software life cycle models: Waterfall, RAD, Spiral, Open-source, Agile process
Understanding software process: Process metric, CMM levels

UNIT – II
Planning & Estimation: Product metrics, Estimation- LOC, FP, COCOMO models.
Project Management: Planning, Scheduling, racking.

UNIT - III
Workflow of Software life cycle,
Requirement Workflow: Functional and Nonfunctional, Characteristics of Requirements
Requirement Elicitation Techniques, Requirement Documentation –Use case specification, Activity
Diagram
Analysis workflow: Static Analysis, Identifying Object – Methods of identifying objects and types -
Boundary, Control, Entity, **Dynamic Analysis-Identifying Interaction –** Sequence and Collaboration
diagrams, State chart diagram
Design Workflow: System Design Concept – Coupling and Cohesion, Architectural Stylesm
Identifying Subsystems and Interfaces, Design Patterns

UNIT-IV
Implementation Workflow: Mapping models to Code, Mapping Object Model to Database Schema
Testing: FTR – Walkthrough and Inspection, Unit Testing, Integration, System and Regression
Testing, User Acceptance Testing
Software Quality: Quality Standards, Quality Matrices Testing & SQA: FTR, unit testing, integration
testing, product testing, and acceptance testing

UNIT - V
Software Configuration Management: Managing and controlling Changes, Managing and
controlling versions
Maintenance: Types of maintenance, Maintenance Log and defect reports. Reverse and re-
engineering

References:
1. Bernd Bruegge, "Object oriented software engineering", Second Edition, Pearson Education.
2. Stephan R. Schach, "Object oriented software engineering", Tata McGraw Hill.
3. Roger Pressman, "Software Engineering", sixth edition, Tata McGraw Hill.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)
HUMAN COMPUTER INTERACTION (Professional Elective - IV)

UNIT I
Introduction: Importance of user Interface – definition, importance of good design. Benefits of good design. A brief history of Screen design,
The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

UNIT II
Design process – Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, understanding business junctions.
Screen Designing:- Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

UNIT III
Windows – New and Navigation schemes selection of window, selection of devices based and screen based controls.
Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

UNIT IV
Software tools – Specification methods, interface – Building Tools.
UNIT V
Interaction Devices – Keyboard and function keys – pointing devices – speech recognition digitization and generation – image and video displays – drivers.

TEXT BOOKS:
1. The essential guide to user interface design, Wilbert O Galitz, Wiley DreamTech.
2. Designing the user interface, 3rd Edition Ben Shneidermann , Pearson Education Asia.

REFERENCES:
1. Human – Computer Interaction. Alan Dix, Janet Fincay, Gre Goryd, Abowd, Russell Bealg, Pearson Education
2. Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech.
3. User Interface Design, Soren Lauesen , Pearson Education.
4. Human –Computer Interaction,D.R.Olsen,Cengage Learning.
5. Human –Computer Interaction,Smith - Atakan,Cengage Learning.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

MACHINE LEARNING & Design patterns LAB (Lab - III)

Course Objective:
1. The objective of this lab is to get an overview of the various machine learning techniques and can able to demonstrate them using python.

Course Outcomes: After the completion of the "**Machine Learning**" lab, the student can able to:
1. Understand complexity of Machine Learning algorithms and their limitations;
2. Understand modern notions in data analysis-oriented computing;
3. Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
4. Be capable of performing experiments in Machine Learning using real-world data.

List of Experiments
2. The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is theprobability that a student is absent given that today is Friday? Apply Baye's rule in python to get the result. (Ans: 15%)
3. Extract the data from database using python
4. Implement k-nearest neighbours classification using python
5. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)

VAR1	VAR2	CLASS
1.713	1.586	0
0.180	1.786	1
0.353	1.240	1
0.940	1.566	0
1.486	0.759	1
1.266	1.106	0
1.540	0.419	1
0.459	1.799	1
0.773	0.186	1

6. The following training examples map descriptions of individuals onto high, medium and low credit-worthiness.

    medium skiing designsingle twenties no -> highRisk
    high   golf   trading   married forties yes -> lowRisk
    low   speedway transport married thirties yes -> medRisk
    medium football banking   single thirties yes -> lowRisk
    high   flying media   married fifties yes -> highRisk
    low   football security   single twenties no -> medRisk
    medium golf   media   single thirties yes -> medRisk
    medium golf   transport married forties yes -> lowRisk

high   skiing banking   single thirties yes -> highRisk
low   golf   unemployed married forties yes -> highRisk

Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `medRisk' in the dataset?

7. Implement linear regression using python.

8. Implement Naïve Bayes theorem to classify the English text

9. Implement an algorithm to demonstrate the significance of genetic algorithm

10. Implement the finite words classification system using Back-propagation algorithm

Learning Objective:

After completing this course, the student should be able to:

● Understand the concept of patterns and the Catalog.
● Discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
● Understand the variety of implemented bad practices related to the Business and Integration tiers.
● Highlight the evolution of patterns.
● How to add functionality to designs while minimizing complexity
● What design patterns really are, and are not
● About specific design patterns.
● What code qualities you need to maintain to keep code flexible.
● How to use design patterns to keep code quality high without overdesign.

List of Sample Problems:

1. Design and execute an application, where "abstract factory" pattern is applied.
2. Design and execute an application, where "adapter" pattern is applied.
3. Design and execute an application, where "bridge" pattern is applied.
4. Design and execute an application, where "command" pattern is applied.
5. Design and execute an application, where "composite" pattern is applied.
6. Design and execute an application, where "facade" pattern is applied.
7. Design and execute an application, where "observer" pattern is applied.
8. Design and execute an application, where "proxy" pattern is applied.
9. Design and execute an application, where "strategy" pattern is applied.
10. Specify and implement the most convenient pattern to apply to the following scenarios:
    a) A component on the web tier requires access to business components.
    b) There a need to provide several buttons on a web form which executes different actions.
    c) Messages need to be sent to citizens each time a typhoon approaches.

Text Books:
1. Machine Learning – Tom M. Mitchell, MGH
2. Fundamentals of Speech Recognition By Lawrence Rabiner and Biing – Hwang Juang.

Reference Book:
1. Machine Learning: An Algorithmic Perspective, Stephen Marsland, Taylor & Francis

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

SOFTWARE TESTING METHODOLOGIES LAB (Lab – IV)

(PE -3 LAB)

Prerequisites: A basic knowledge of programming.

Course Objectives:
1. To provide knowledge of Software Testing Methods.
2. To develop skills in software test automation and management using latest tools.

Course Outcome:
1. Design and develop the best test strategies in accordance to the development model.

List of Experiments
1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects
5. a) Bitmap checkpoint for object/window
   a) Bitmap checkpoint for screen area
6. Database checkpoint for Default check
7. Database checkpoint for custom check
8. Database checkpoint for runtime record check
9. a) Data driven test for dynamic test data submission
   b) Data driven test through flat files
   c) Data driven test through front grids
   d) Data driven test through excel test
10. a) Batch testing without parameter passing
    b) Batch testing with parameter passing
11. Data driven batch
12. Silent mode test execution without any interruption
13. Test case for calculator in windows application

Text Books:
1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.
2. Software Testing Tools, Dr.K.V.K.K.Prasad, Dreamtech.

References:
1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World, Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing, Meyers, John Wiley.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

MOBILE APPLICATION DEVELOPMENT LAB (Lab – IV)
(PE -3 LAB)

Objectives:To learn how to develop Applications in android environment.

To learn how to develop user interface applications.
To learn how to develop URL related applications.
The student is expected to be able to do the following problems, though not limited.

1. a. Create an Android application that shows Hello + name of the user and run it on an emulator.
    (b) Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.

2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.

3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.

4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.

5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.

6. Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message.

7. Create a user registration application that stores the user details in a database table.

8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.

10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.

11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.

12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

13. Create an application that shows the given URL (from a text field) in a browser.

14. Develop an application that shows the current location's latitude and longitude continuously as the device is moving (tracking).

15. Create an application that shows the current location on Google maps.

Note:
Android Application Development with MIT App Inventor:For the first one week, the student is advised to go through the App Inventor from MIT which gives insight into the various properties of each component.
The student should pay attention to the properties of each components, which are used later in Android programming. Following are useful links:
1. http://ai2.appinventor.mit.edu
2. https://drive.google.com/file/d/0B8rTtW_91YclTWF4czdBMEpZcWs/view

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – I Year – II Semester (Software Engineering)

(PE -3 LAB)

FULL STACK DEVELOPMENT LAB

Course Objectives: The students should be able:
- To implement Forms, inputs and Services using AngularJS
- To develop a simple web application using Nodejs; Angular JS and Express
- To implement data models using MongoDB

Course Outcomes:
- Develop a fully functioning website and deploy on a web server.
- Gain Knowledge about the front end and back end Tools
- Find and use code packages based on their documentation to produce working results in a project.
- Create web pages that function using external data.

List of Experiments:
1. Develop a Form and validate using AngularJS
2. Create and implement modules and controllers in AngularJS
3. Implement Error Handling in AngularJS
4. Create and implement Custom directives
5. Create a simple web application using Express, Node JS and Angular JS
6. Implement CRUD operations on MongoDB
7. Create a react application for the student management system having registration, login, contact,about pages and implement routing to navigate through these pages.
8. Create a service in react that fetches the weather information from openweathermap.org and thedisplay the current and historical weather information using graphical representation using chart.js
9. Create a TODO application in react with necessary components and deploy it into github.
10. A. Develop an express web application that can interact with REST API to perform CRUDoperations on student data. (Use Postman)
    B. For the above application create authorized end points using JWT (JSON Web Token).

FUNCTIONAL PROGRAMMING LAB (Professional Elective - III Lab)

M.Tech SE I Year II Sem.   **L   T   P   C**

                                                                    0   0   4   2

Course Objectives:
1. To be able to introduce core programming basics and program design with functions using functional programming languages.
2. To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.

Course Outcomes:
1. Students should be able to understand the basic concepts of scripting and the contributions of Functional Programming Languages.
2. Ability to explore python especially the object-oriented concepts, and the built in objects of Python.

List of Programs:
1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017"
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula : c/5 = f-32/9]
10. Write a recursive function, (defun nth (pos list) ???), that returns the n'th item from a list. Assume the list has at least n items. (nth 1 aList) is to return the first item in aList.
11. Write simple lisp functions such as the following. Take into account lists which are too short.
    • (remove-first '(a b c ...)) -> (b c ...) --- remove the first item from the list.
    • (remove-second '(a b c ...)) -> (a c ...) -- remove the second item from the list.
    • (insert-as-second 'b '(a c ...)) -> (a b c ...) --- insert as the second element.
12. Write a Lisp macro mycase that translates the following macro call. Assume the input will be error free. The input lists can be any length. You must document your solution.
    (mycase (C1 C2 ... Cn) (P1 P2 ... Pn))
    translates to the following
    (mycond (C1 P1) (C2 P2) ... (Cn Pn))
13. Write a Lisp macro mycase that translates the following macro call as shown. Assume the input will be error free. The input lists can be any length. Use standard Lisp functionals. If you need support functions, your answer should have only non-recursive support functions.
    (mycase (C1 C2 ... Cn) (P1 P2 ... Pn))
    translates to the following
    (mycond (C1 (P1 P2 ... Pn)) (C2 (P2 ... Pn)) ... (Cn (Pn)))
14. Assume the following forms have been typed into the Lisp interpreter and evaluated.

    (defun a (x) (values (list x) x))
    (setq a '(a b))
    (defun b (x) `(x ,x))
    (setq b (cdr a))

```
( setq c ( car a ) )
( setq d c )
( setq e ( (lambda ( x ) ( list x ) ) d ) )
```

15. What will the following forms evaluate
    to?(cons c (car a))
    ```
    ( cons e b )
    ( eval a )
    ( let (( a b ) ( y a )) ( append a y ) )
     ( multiple-value-list ( a a ) )
     ( b c )
     ( set ( car a ) (cdr a) )
    ( setf ( car a ) (cdr a) )
    ```

TEXT BOOKS:
1. The Haskell School of Expression: Learning Functional Programming through Multimedia, Paul Hudak.
2. Functional Programming in Python, David Mertz, O'Reilly Media.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – II Year – I Semester (Software Engineering)

AGILE DEVELOPMENT METHODOLOGIES (Professional Elective - V)

UNIT - I
Introduction Extreme Programming (XP) - Agile Development
Why Agile - Understanding Success, Beyond Deadlines, Importance of Organizational Success, Introduction to Agility
How to Be Agile - Agile methods, Don't make your own method, Road to mastery
Understanding XP (Extreme Programming) - XP life cycle, XP team, XP Concepts
Adopting XP - Knowing whether XP is suitable, Implementing XP, assessing Agility
Practicing XP - Thinking - Pair Programming, Energized work, Informative Workspace, Root cause Analysis, Retrospectives

UNIT - II
Collaborating
Trust, Sit toghther, Real customer involvemnet, Ubiquitous language, meetings,
coding standards, Iteration demo, Reporting

UNIT - III
Releasing
Bugfree Release, Version Control, fast build, continuous integration, Collective ownership,
Documentation

UNIT - IV
Planing
Version, Release Plan, Risk Management, Iteration Planning, Slack, Stories, Estimating

UNIT - V
Developing:
Incremental requirements, Customer tests, Test driven development, Refactoring,
Incremental design and architecture, spike solutions, Performance optimization, Exploratory testing

Text Book:
1. The art of Agile Development, James Shore and Shane Warden, 11th Indian Reprint, O'Reilly, 2018

References:
1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4th Indian Reprint, 2018
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5th Indian Reprint, 2015
3. Agile Project Management - Jim Highsmith, Pearson Low price Edition 2004

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – II Year – I Semester (Software Engineering)

WEB SERVICES TESTING (**Professional Elective - V**)

Course Objectives: Your studies will enable you to develop:
- Become more proficient in testing web services included in your service-oriented solutions
- Find, analyze, reproduce bugs effectively by adhering to best web service testing approaches
- Learn with clear step-by-step instructions and hands-on examples on various topics related to web services testing using soapUI

UNIT - I
Introduction to Web Services Testing and soapUI: SOA and web services, SOAP, SOAPUI basics, project pre-requisites. Designing and implementing web services.

UNIT - II
First hands-on experience: Understanding web services definition, creation of soapUI projects, generate SOAP faults. Creating test suites, test cases, assertions, properties in test cases and suites.

UNIT - III
Load and Performance Testing: Non-functional testing, planning for web services performance testing using soapUI, load assertions. Web services simulation with soapUI, mock services in action.

UNIT - IV
Advance testing scenarios: Advanced functional testing, WS-Addressing, WS-Security, Apache Axis2, security with binding and authentication. REST testing, REST parameters, REST functional testing using soapUI, testing databases using soapUI and assertions, JMS testing with soapUI and assertions.

UNIT - V
Automated testing using scripts: Introduction to Groovy script, Groovy scripting in soapUI, model items, request and response handling using scripts, test automation, Junit integration, soapUI command line execution, maven soapUI plugin, WS-I validation, sending attachments with SOAP messages using soapUI.

Text Books:
1. Web Services Testing with soapUI by Charitha Kankanamge
2. The Art of Application Performance Testing by Ian Molyneaux

Reference Books:
1. Testing Applications on the Web by Hung Nguyen, Bob Johnsonm and Michael Hack

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – II Year – I Semester (Software Engineering)

Adhoc Networks (Professional Elective - V)

Course objectives:
This course will enable students to
- Explain fundamental principles of Adhoc Networks
- MAC Protocols for Adhoc Wireless Networks
- Routing Protocols for Adhoc Wireless Networks
- Transport Layer Protocols for Ad-hoc Networks
- Security Protocols for Adhoc Networks
- Wireless Sensor Networks

UNIT I
Ad-hoc Wireless Networks: Introduction, Issues in Ad-hoc Wireless Networks, Ad-hoc Wireless Internet; **MAC Protocols for Ad-hoc Wireless Networks:** Introduction, Issues in Designing a MAC Protocol, Design Goals of MAC Protocols, Classification of MAC protocols, Contention-Based Protocols, Contention-Based Protocols with Reservation Mechanisms.

UNIT II
Routing Protocols for Ad-hoc Wireless Networks: Introduction, Issues in Designing a Routing Protocol for Ad-hoc Wireless Networks, Classification of Routing Protocols, Table Driven Routing Protocols; On-Demand Routing Protocols, Hybrid Routing Protocols.

UNIT III
Transport Layer Protocols for Adhoc Networks: Introduction, Issues in Designing a Transport Layer Protocol, Design Goals of a Transport Layer Protocol, Classification of Transport Layer Solutions, TCP over Adhoc Wireless Networks

UNIT IV
Security Protocols for Adhoc Networks: Security in Adhoc Wireless Networks, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management, Secure Routing in Adhoc Wireless Networks.

UNIT V
Wireless Sensor Networks: Introduction, Sensor Network Architecture, Data Dissemination, Data Gathering, MAC Protocols for Sensor Networks

TEXT BOOKS:

1. C. Siva Ram Murthy & B. S. Manoj: Adhoc Wireless Networks: Architectures and Protocols, 2nd Edition, Pearson Education, 2011

REFERENCE BOOKS:

1. Ozan K. Tonguz and Gianguigi Ferrari: Ad-hoc Wireless Networks, John Wiley, 2007.
2. Xiuzhen Cheng, Xiao Hung, Ding-Zhu Du: Ad-hoc Wireless Networking, Kluwer Academic Publishers, 2004.
3. C.K. Toh: Ad-hoc Mobile Wireless Networks- Protocols and Systems, Pearson

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. – II Year – I Semester (Software Engineering)

MOBILE APPLICATION SECURITY (Professional Elective - V)

Course Objectives:
- To understand the mobile issues and development strategies
- To understand the WAP and mobile security issues
- To understand the Bluetooth security issues.

UNIT - I
Top Mobile Issues and Development Strategies: Top Issues Facing Mobile Devices, Physical Security , Secure Data Storage (on Disk), Strong Authentication with Poor Keyboards , Multiple-User Support with Security, Safe Browsing Environment , Secure Operating Systems, Application Isolation, Information Disclosure, Virus, Worms, Trojans, Spyware, and Malware , Difficult Patching/Update Process, Strict Use and Enforcement of SSL, Phishing , Cross-Site Request Forgery (CSRF), Location Privacy/Security, Insecure Device Drivers, Multifactor Authentication, Tips for Secure Mobile Application Development .

UNIT - II
WAP and Mobile HTML Security: WAP and Mobile HTML Basics, Authentication on WAP/Mobile HTML Sites, Encryption, Application Attacks on Mobile HTML Sites, Cross-Site Scripting, SQL Injection, Cross-Site Request Forgery, HTTP Redirects, Phishing, Session Fixation, Non-SSL Login, WAP and Mobile Browser Weaknesses, Lack of HTTPOnly Flag Support, Lack of SECURE Flag Support, Handling Browser Cache, WAP Limitations.

UNIT - III
Bluetooth Security: Overview of the Technology , History and Standards , Common Uses , Alternatives , Future , Bluetooth Technical Architecture , Radio Operation and Frequency, Bluetooth Network Topology , Device Identification , Modes of Operation , Bluetooth Stack ,Bluetooth Profiles , Bluetooth Security Features , Pairing , Traditional Security Services in Bluetooth, Security "Non-Features" , Threats to Bluetooth Devices and Networks, Bluetooth Vulnerabilities , Bluetooth Versions Prior to v1.2, Bluetooth Versions Prior to v2.1.

UNIT - IV
SMS Security: Overview of Short Message Service, Overview of Multimedia Messaging Service, Wireless Application Protocol (WAP), Protocol Attacks, Abusing Legitimate Functionality, Attacking Protocol Implementations, Application Attacks, iPhone Safari, Windows Mobile MMS, Motorola RAZR JPG Overflow, Walkthroughs, Sending PDUs, Converting XML to WBXML.

UNIT - V
Enterprise Security on the Mobile OS: Device Security Options, PIN, Remote, 346 Secure Local Storage, Apple iPhone and Keychain, Security Policy Enforcement, Encryption, Full Disk Encryption, E-mail Encryption, File Encryption, Application Sandboxing, Signing, and Permissions, Application Sandboxing, Application Signing, Permissions, Buffer Overflow Protection, Windows Mobile, iPhone, Android, BlackBerry, Security Feature Summary.

Text Book:

1. "Mobile Application Security", Himanshu Dwivedi, Chris Clark, David Thiel, TATA McGraw-Hill.

References:

1. "Mobile and Wireless Network Security and Privacy", Kami S.Makki,et al, Springer.
2. "Android Security Attacks Defenses", Abhishek Dubey, CRC Press.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

ENGLISH FOR RESEARCH PAPER WRITING (Audit Course - I & II)

Prerequisite: None

Course objectives: Students will be able to:
- Understand that how to improve your writing skills and level of readability
- Learn about what to write in each section
- Understand the skills needed when writing a Title Ensure the good quality of paper at very first-time submission

UNIT-I:
Planning and Preparation, Word Order, Breaking up long sentences, Structuring Paragraphs and Sentences, Being Concise and Removing Redundancy, Avoiding Ambiguity and Vagueness

UNIT-II:
Clarifying Who Did What, Highlighting Your Findings, Hedging and Criticizing, Paraphrasing and Plagiarism, Sections of a Paper, Abstracts. Introduction

UNIT-III:
Review of the Literature, Methods, Results, Discussion, Conclusions, The Final Check.

UNIT-IV:
key skills are needed when writing a Title, key skills are needed when writing an Abstract, key skills are needed when writing an Introduction, skills needed when writing a Review of the Literature,

UNIT-V:
skills are needed when writing the Methods, skills needed when writing the Results, skills are needed when writing the Discussion, skills are needed when writing the Conclusions. useful phrases, how to ensure paper is as good as it could possibly be the first- time submission

TEXT BOOKS/ REFERENCES:
1. Goldbort R (2006) Writing for Science, Yale University Press (available on Google Books)
2. Day R (2006) How to Write and Publish a Scientific Paper, Cambridge University Press
3. Highman N (1998), Handbook of Writing for the Mathematical Sciences, SIAM. Highman's book.
4. Adrian Wallwork, English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

DISASTER MANAGEMENT (Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to
- learn to demonstrate a critical understanding of key concepts in disaster risk reduction and humanitarian response.
- critically evaluate disaster risk reduction and humanitarian response policy and practice from multiple perspectives.
- develop an understanding of standards of humanitarian response and practical relevance in specific types of disasters and conflict situations.
- critically understand the strengths and weaknesses of disaster management approaches,
- planning and programming in different countries, particularly their home country or the countries they work in

UNIT-I:
Introduction:
Disaster: Definition, Factors and Significance; Difference Between Hazard and Disaster; Natural and Manmade Disasters: Difference, Nature, Types and Magnitude.
Disaster Prone Areas in India:
Study of Seismic Zones; Areas Prone to Floods and Droughts, Landslides and Avalanches; Areas Prone to Cyclonic and Coastal Hazards with Special Reference to Tsunami; Post-Disaster Diseases and Epidemics

UNIT-II:
Repercussions of Disasters and Hazards:
Economic Damage, Loss of Human and Animal Life, Destruction of Ecosystem. Natural Disasters: Earthquakes, Volcanisms, Cyclones, Tsunamis, Floods, Droughts and Famines, Landslides and Avalanches, Man-made disaster: Nuclear Reactor Meltdown, Industrial Accidents, Oil Slicks and Spills, Outbreaks of Disease and Epidemics, War and Conflicts.

UNIT-III:
Disaster Preparedness and Management:
Preparedness: Monitoring of Phenomena Triggering A Disaster or Hazard; Evaluation of Risk: Application of Remote Sensing, Data from Meteorological and Other Agencies, Media Reports: Governmental and Community Preparedness.

UNIT-IV
Risk Assessment Disaster Risk:
Concept and Elements, Disaster Risk Reduction, Global and National Disaster Risk Situation. Techniques of Risk Assessment, Global Co-Operation in Risk Assessment and Warning, People's Participation in Risk Assessment. Strategies for Survival.

UNIT-V:
Disaster Mitigation:

Meaning, Concept and Strategies of Disaster Mitigation, Emerging Trends In Mitigation. Structural Mitigation and Non-Structural Mitigation, Programs of Disaster Mitigation in India.

TEXT BOOKS/ REFERENCES:
1. R. Nishith, Singh AK, "Disaster Management in India: Perspectives, issues and strategies "'New Royal book Company.
2. Sahni, Pardeep Et. Al. (Eds.)," Disaster Mitigation Experiences and Reflections", Prentice Hall of India, New Delhi.
3. Goel S. L., Disaster Administration and Management Text and Case Studies", Deep &Deep Publication Pvt. Ltd., New Delhi.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

SANSKRIT FOR TECHNICAL KNOWLEDGE (Audit Course - I & II)

Prerequisite: None

Course Objectives:
- To get a working knowledge in illustrious Sanskrit, the scientific language in the world
- Learning of Sanskrit to improve brain functioning
- Learning of Sanskrit to develop the logic in mathematics, science & other subjects enhancing the memory power
- The engineering scholars equipped with Sanskrit will be able to explore the huge knowledge from ancient literature

Course Outcomes: Students will be able to
- Understanding basic Sanskrit language
- Ancient Sanskrit literature about science & technology can be understood
- Being a logical language will help to develop logic in students

UNIT-I:
Alphabets in Sanskrit,

UNIT-II:
Past/Present/Future Tense, Simple Sentences

UNIT-III:
Order, Introduction of roots,

UNIT-IV:
Technical information about Sanskrit Literature

UNIT-V:
Technical concepts of Engineering-Electrical, Mechanical, Architecture, Mathematics

TEXT BOOKS/ REFERENCES:
1. "Abhyaspustakam" – Dr. Vishwas, Samskrita-Bharti Publication, New Delhi
2. "Teach Yourself Sanskrit" Prathama Deeksha-Vempati Kutumbshastri, Rashtriya Sanskrit Sansthanam, New Delhi Publication
3. "India's Glorious Scientific Tradition" Suresh Soni, Ocean books (P) Ltd., New Delhi.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

VALUE EDUCATION (Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to
- Understand value of education and self- development
- Imbibe good values in students
- Let the should know about the importance of character

Course outcomes: Students will be able to
- Knowledge of self-development
- Learn the importance of Human values
- Developing the overall personality

UNIT-I:
Values and self-development –Social values and individual attitudes. Work ethics, Indian vision of humanism. Moral and non- moral valuation. Standards and principles. Value judgements

UNIT-II:
Importance of cultivation of values. Sense of duty. Devotion, Self-reliance. Confidence, Concentration. Truthfulness, Cleanliness. Honesty, Humanity. Power of faith, National Unity. Patriotism. Love for nature, Discipline

UNIT-III:
Personality and Behavior Development - Soul and Scientific attitude. Positive Thinking. Integrity and discipline, Punctuality, Love and Kindness.

UNIT-IV:
Avoid fault Thinking. Free from anger, Dignity of labour. Universal brotherhood and religious tolerance. True friendship. Happiness Vs suffering, love for truth. Aware of self-destructive habits. Association and Cooperation. Doing best for saving nature

UNIT-V:
Character and Competence –Holy books vs Blind faith. Self-management and Good health. Science of reincarnation, Equality, Nonviolence, Humility, Role of Women. All religions and same message. Mind your Mind, Self-control. Honesty, Studying effectively

TEXT BOOKS/ REFERENCES:
1. Chakroborty, S.K. "Values and Ethics for organizations Theory and practice", Oxford University Press, New Delhi

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

CONSTITUTION OF INDIA (Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to:
- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in the early years of Indian nationalism.
- To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.

Course Outcomes: Students will be able to:
- Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.
- Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.
- Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution.
- Discuss the passage of the Hindu Code Bill of 1956.

UNIT-I:
History of Making of the Indian Constitution: History Drafting Committee, (Composition & Working), **Philosophy of the Indian Constitution:** Preamble**,** Salient Features.

UNIT-II:
Contours of Constitutional Rights & Duties: Fundamental Rights Right to Equality**,** Right to Freedom**,** Right against Exploitation**,** Right to Freedom of Religion**,** Cultural and Educational Rights**,** Right to Constitutional Remedies, Directive Principles of State Policy**,** Fundamental Duties.

UNIT-III:
Organs of Governance: Parliament**,** Composition**,** Qualifications and Disqualifications**,** Powers and Functions**,** Executive**,** President**,** Governor**,** Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualification, Powers and Functions.

UNIT-IV:
Local Administration: District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation. Pachayati raj: Introduction, PRI: Zila Pachayat. Elected officials and their roles, CEO Zila Pachayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy.

UNIT-V:

Election Commission: Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners. State Election Commission: Role and Functioning. Institute and Bodies for the welfare of SC/ST/OBC and women.

TEXT BOOKS/ REFERENCES:
1. The Constitution of India, 1950 (Bare Act), Government Publication.
2. Dr. S. N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015.
3. M. P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

PEDAGOGY STUDIES (Audit Course - I & II)

Prerequisite: None

Course Objectives: Students will be able to:
- Review existing evidence on the review topic to inform programme design and policy making undertaken by the DfID, other agencies and researchers.
- Identify critical evidence gaps to guide the development.

Course Outcomes: Students will be able to understand:
- What pedagogical practices are being used by teachers in formal and informal classrooms in developing countries?
- What is the evidence on the effectiveness of these pedagogical practices, in what conditions, and with what population of learners?
- How can teacher education (curriculum and practicum) and the school curriculum and guidance materials best support effective pedagogy?

UNIT-I:
Introduction and Methodology: Aims and rationale, Policy background, Conceptual framework and terminology Theories of learning, Curriculum, Teacher education. Conceptual framework, Research questions**.** Overview of methodology and Searching.

UNIT-II:
Thematic overview: Pedagogical practices are being used by teachers in formal and informal classrooms in developing countries. Curriculum, Teacher education.

UNIT-III:
Evidence on the effectiveness of pedagogical practices, Methodology for the indepth stage: quality assessment of included studies. How can teacher education (curriculum and practicum) and the scho curriculum and guidance materials best support effective pedagogy? Theory of change. Strength and nature of the body of evidence for effective pedagogical practices. Pedagogic theory and pedagogical approaches. Teachers' attitudes and beliefs and Pedagogic strategies.

UNIT-IV:
Professional development: alignment with classroom practices and follow-up support, Peer support, Support from the head teacher and the community. Curriculum and assessment, Barriers to learning: limited resources and large class sizes

UNIT-V:
Research gaps and future directions: Research design**,** Contexts**,** Pedagogy**,** Teacher education**,** Curriculum and assessment**,** Dissemination and research impact.

TEXT BOOKS/ REFERENCES:

1. Ackers J, Hardman F (2001) Classroom interaction in Kenyan primary schools, Compare, 31 (2): 245-261.
2. Agrawal M (2004) Curricular reform in schools: The importance of evaluation, Journal of Curriculum Studies, 36 (3): 361-379.
3. Akyeampong K (2003) Teacher training in Ghana - does it count? Multi-site teacher education research project (MUSTER) country report 1. London: DFID.
4. Akyeampong K, Lussier K, Pryor J, Westbrook J (2013) Improving teaching and learning of basic maths and reading in Africa: Does teacher preparation count? International Journal Educational Development, 33 (3): 272–282.
5. Alexander RJ (2001) Culture and pedagogy: International comparisons in primary education. Oxford and Boston: Blackwell.
6. Chavan M (2003) Read India: A mass scale, rapid, 'learning to read' campaign.
7. www.pratham.org/images/resource%20working%20paper%202.pdf.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

STRESS MANAGEMENT BY YOGA (Audit Course - I & II)

Prerequisite: None

Course Objectives:
- To achieve overall health of body and mind
- To overcome stress

Course Outcomes: Students will be able to:
- Develop healthy mind in a healthy body thus improving social health also
- Improve efficiency

UNIT-I:
Definitions of Eight parts of yog. (Ashtanga)

UNIT-II:
Yam and Niyam.

UNIT-III:
Do`s and Don't's in life.
i) Ahinsa, satya, astheya, bramhacharya and aparigraha
ii) Shaucha, santosh, tapa, swadhyay, ishwarpranidhan

UNIT-IV:
Asan and Pranayam

UNIT-V:
i) Various yog poses and their benefits for mind & body
ii) Regularization of breathing techniques and its effects-Types of pranayam

TEXT BOOKS/ REFERENCES:
1. 'Yogic Asanas for Group Tarining-Part-I": Janardan Swami Yogabhyasi Mandal, Nagpur
2. "Rajayoga or conquering the Internal Nature" by Swami Vivekananda, Advaita Ashrama (Publication Department), Kolkata

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
M. Tech. (Software Engineering)

PERSONALITY DEVELOPMENT THROUGH LIFE ENLIGHTENMENT SKILLS
(Audit Course - I & II)

Prerequisite: None
Course Objectives:
- To learn to achieve the highest goal happily
- To become a person with stable mind, pleasing personality and determination
- To awaken wisdom in students

Course Outcomes: Students will be able to
- Study of Shrimad-Bhagwad-Geeta will help the student in developing his personality and achieve the highest goal in life
- The person who has studied Geeta will lead the nation and mankind to peace and prosperity
- Study of Neetishatakam will help in developing versatile personality of students

UNIT-I:
Neetisatakam-Holistic development of personality
- Verses- 19,20,21,22 (wisdom)
- Verses- 29,31,32 (pride & heroism)
- Verses- 26,28,63,65 (virtue)

UNIT-II:
Neetisatakam-Holistic development of personality
- Verses- 52,53,59 (dont's)
- Verses- 71,73,75,78 (do's)

UNIT-III:
Approach to day to day work and duties.
- Shrimad Bhagwad Geeta: Chapter 2-Verses 41, 47,48,
- Chapter 3-Verses 13, 21, 27, 35, Chapter 6-Verses 5,13,17, 23, 35,
- Chapter 18-Verses 45, 46, 48.

UNIT-IV:
Statements of basic knowledge.
- Shrimad Bhagwad Geeta: Chapter2-Verses 56, 62, 68
- Chapter 12 -Verses 13, 14, 15, 16,17, 18
- Personality of Role model. Shrimad Bhagwad Geeta:

UNIT-V:
- Chapter2-Verses 17, Chapter 3-Verses 36,37,42,
- Chapter 4-Verses 18, 38,39
- Chapter18 – Verses 37,38,63

TEXT BOOKS/ REFERENCES:
1. "Srimad Bhagavad Gita" by Swami Swarupananda Advaita Ashram (Publication Department), Kolkata.
2. Bhartrihari's Three Satakam (Niti-sringar-vairagya) by P.Gopinath, Rashtriya Sanskrit Sansthanam, New Delhi.

FAULT TOLERANCE SYSTEMS (Open Elective)

M.Tech SE II Year I Sem.   **L   T   P   C**
                                                                        3   0   0   3

Course Objectives:
1. To know the different advantages and limits of fault avoidance and fault tolerance techniques.
2. To impart the knowledge about different types of redundancy and its application for the design of computer system being able to function correctly even under presence of faults and data errors.
3. To understand the relevant factors in evaluating alternative system designs for a specific set of requirements.
4. To understand the subtle failure modes of "fault-tolerant" distributed systems.

Course Outcomes: Upon the Successful Completion of the Course, the Students would be able to:
1. Become familiar with general and state of the art techniques used in design and analysis of fault tolerant digital systems.
2. Be familiar with making system fault tolerant, modeling and testing, and benchmarking to evaluate and compare systems.

UNIT - I
Introduction to Fault Tolerant Computing: Basic concepts and overview of the course; Faults and their manifestations, Fault/error modeling, Reliability, availability and maintainability analysis, System evaluation, performance reliability tradeoffs.

UNIT - II
System level fault diagnosis: Hardware and software redundancy techniques. Fault tolerant system design methods, Mobile computing and Mobile communication environment, Fault injection methods.

UNIT - III
Software fault tolerance: Design and test of defect free integrated circuits, fault modeling, built in self-test, data compression, error correcting codes, simulation software/hardware, fault tolerant system design, CAD tools for design for testability.

UNIT - IV
Information Redundancy and Error Correcting Codes: Software Problem. Software Reliability Models and Robust Coding Techniques, Reliability in Computer Networks Time redundancy. Re execution in SMT, CMP Architectures, Fault Tolerant Distributed Systems, Data replication.

UNIT - V
Case Studies in FTC: ROC, HP Non-Stop Server. Case studies of fault tolerant systems and current research issues.

TEXT BOOK:
1. Fault Tolerant Computer System Design by D. K. Pradhan, Prentice Hall.

REFERENCES:
1. Fault Tolerant Systems by I. Koren, Morgan Kauffman.
2. Software Fault Tolerance Techniques and Implementation by L. L. Pullum, Artech House Computer Security Series.
3. Reliability of Computer Systems and Networks: Fault Tolerance Analysis and Design by M. L. Shooman, Wiley.

INTRUSION DETECTION SYSTEMS (Open Elective)

M.Tech SE II Year I Sem.   **L   T   P   C**
                                                               3   0   0   3

Prerequisites: Computer Networks, Computer Programming

Course Objectives:
1. Compare alternative tools and approaches for Intrusion Detection through quantitative analysis to determine the best tool or approach to reduce risk from intrusion.
2. Identify and describe the parts of all intrusion detection systems and characterize new and emerging IDS technologies according to the basic capabilities all intrusion detection systems share.

Course Outcomes: After completion of the course, students will be able to:
1. Possess a fundamental knowledge of Cyber Security.
2. Understand what vulnerability is and how to address most common vulnerabilities.
3. Know basic and fundamental risk management principles as it relates to Cyber Security and Mobile Computing.
4. Have the knowledge needed to practice safer computing and safeguard your information using Digital Forensics.
5. Understand basic technical controls in use today, such as firewalls and Intrusion Detection systems.
6. Understand legal perspectives of Cyber Crimes and Cyber Security.

UNIT - I
The state of threats against computers, and networked systems-Overview of computer security solutions and why they fail-Vulnerability assessment, firewalls, VPN's -Overview of Intrusion Detection and Intrusion Prevention, Network and Host-based IDS

UNIT - II
Classes of attacks - Network layer: scans, denial of service, penetration Application layer: software exploits, code injection-Human layer: identity theft, root access-Classes of attackers-Kids/hackers/sop Hesitated groups-Automated: Drones, Worms, Viruses

UNIT - III
A General IDS model and taxonomy, Signature-based Solutions, Snort, Snort rules, Evaluation of IDS, Cost sensitive IDS

UNIT - IV
Anomaly Detection Systems and Algorithms-Network Behavior Based Anomaly Detectors (rate based)-Host-based Anomaly Detectors-Software Vulnerabilities-State transition, Immunology, Payload Anomaly Detection

UNIT - V
Attack trees and Correlation of alerts- Autopsy of Worms and Botnets-Malware detection -Obfuscation, polymorphism- Document vectors.
Email/IM security issues-Viruses/Spam-From signatures to thumbprints to zero-day detection-Insider Threat issues-Taxonomy-Masquerade and Impersonation Traitors, Decoys and Deception-Future: Collaborative Security

TEXT BOOKS:
1. Peter Szor, The Art of Computer Virus Research and Defense, Symantec Press ISBN 0-321-30545-3.
2. Markus Jakobsson and Zulfikar Ramzan, Crimeware, Understanding New Attacks and Defenses.

REFERENCE BOOKS:
1. Saiful Hasan, Intrusion Detection System, Kindle Edition.
2. Ankit Fadia, Intrusion Alert: An Ethical Hacking Guide to Intrusion Detection.

Online Websites/Materials:
1. https://www.intechopen.com/books/intrusion-detection-systems/

Online Courses:
1. https://www.sans.org/course/intrusion-detection-in-depth
2. https://www.cybrary.it/skill-certification-course/ids-ips-certification-training-course

OPTIMIZATION TECHNIQUES (Open Elective)

M.Tech SE II Year I Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Prerequisite: Mathematics –I, Mathematics –II

Course Objectives:
1. To introduce various optimization techniques i.e classical, linear programming, transportation problem, simplex algorithm, dynamic programming
2. Constrained and unconstrained optimization techniques for solving and optimizing electrical and electronic engineering circuits design problems in real world situations.
3. To explain the concept of Dynamic programming and its applications to project implementation.

Course Outcomes: After completion of this course, the student will be able to:
1. explain the need of optimization of engineering systems.
2. understand optimization of electrical and electronics engineering problems.
3. apply classical optimization techniques, linear programming, simplex algorithm, transportation problem.
4. apply unconstrained optimization and constrained non-linear programming and dynamic programming.
5. Formulate optimization problems.

UNIT - I
Introduction and Classical Optimization Techniques: Statement of an Optimization problem – design vector – design constraints – constraint surface – objective function – objective function surface - classification of Optimization problems.

Linear Programming: Standard form of a linear programming problem – geometry of linear programming problems – definitions and theorems – solution of a system of linear simultaneous equations – pivotal reduction of a general system of equations – motivation to the simplex method – simplex algorithm.

UNIT - II
Transportation Problem: Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel's approximation method – testing for optimality of balanced transportation problems. Degeneracy.

Assignment problem – Formulation – Optimal solution - Variants of Assignment Problem; Traveling Salesman problem.

UNIT - III
Classical Optimization Techniques: Single variable Optimization – multi variable Optimization without constraints – necessary and sufficient conditions for minimum/maximum – multivariable Optimization with equality constraints: Solution by method of Lagrange multipliers – Multivariable Optimization with inequality constraints: Kuhn – Tucker conditions.

Single Variable Nonlinear Unconstrained Optimization: Elimination methods: Uni Model function-its importance, Fibonacci method & Golden section method.

UNIT - IV
Multi variable nonlinear unconstrained optimization: Direct search methods – Univariant method, Pattern search methods – Powell's, Hooke - Jeeves, Rosenbrock's search methods. Gradient methods: Gradient of function & its importance, Steepest descent method, Conjugate direction methods: Fletcher-Reeves method & variable metric method.

UNIT - V

Dynamic Programming: Dynamic programming multistage decision processes – types – concept of sub optimization and the principle of optimality – computational procedure in dynamic programming – examples illustrating the calculus method of solution - examples illustrating the tabular method of solution.

TEXT BOOKS:
1. Optimization Techniques & Applications by S.S.Rao, New Age International.
2. Optimization for Engineering Design by Kalyanmoy Deb, PHI

REFERENCES:
1. George Bernard Dantzig, Mukund Narain Thapa, "Linear programming", Springer series in Operations Research 3rd edition, 2003.
2. H. A. Taha, "Operations Research: An Introduction", 8th Edition, Pearson/Prentice Hall, 2007.
3. Optimization Techniques by Belegundu & Chandrupatla, Pearson Asia.
4. Optimization Techniques Theory and Practice by M.C. Joshi, K.M. Moudgalya, Narosa Publications

CYBER PHYSICAL SYSTEMS (Open Elective)

M.Tech SE II Year I Sem.   **L   T   P   C**
                                                                                      3   0   0   3

Course Objective: To learn about design of cyber-physical systems

Course Outcomes: Upon the Successful Completion of the Course, the Students would be able to:
1. Understand the core principles behind CPS
2. Identify Security mechanisms of Cyber physical systems
3. Understand Synchronization in Distributed Cyber-Physical Systems

UNIT - I
Symbolic Synthesis for Cyber-Physical Systems
Introduction and Motivation, Basic Techniques - Preliminaries, Problem Definition, Solving the Synthesis Problem, Construction of Symbolic Models, Advanced Techniques: Construction of Symbolic Models, Continuous-Time Controllers, Software Tools

UNIT - II
Security of Cyber-Physical Systems
Introduction and Motivation, Basic Techniques - Cyber Security Requirements, Attack Model, Countermeasures, Advanced Techniques: System Theoretic Approaches

UNIT - III
Synchronization in Distributed Cyber-Physical Systems: Challenges in Cyber-Physical Systems, A Complexity-Reducing Technique for Synchronization, Formal Software Engineering, Distributed Consensus Algorithms, Synchronous Lockstep Executions, Time-Triggered Architecture, Related Technology, Advanced Techniques

UNIT - IV
Real-Time Scheduling for Cyber-Physical Systems
Introduction and Motivation, Basic Techniques - Scheduling with Fixed Timing Parameters, Memory Effects, Multiprocessor/Multicore Scheduling, Accommodating Variability and Uncertainty

UNIT - V
Model Integration in Cyber-Physical Systems
Introduction and Motivation, Causality, Semantic Domains for Time, Interaction Models for Computational Processes, Semantics of CPS DSMLs, Advanced Techniques, ForSpec, The Syntax of CyPhyML, Formalization of Semantics, Formalization of Language Integration.

TEXT BOOKS:
1. Raj Rajkumar, Dionisio De Niz, and Mark Klein, Cyber-Physical Systems, Addison-Wesley Professional.
2. Rajeev Alur, Principles of Cyber-Physical Systems, MIT Press, 2015

GRAPH ANALYTICS (Open Elective)

M.Tech SE II Year I Sem.   L   T   P C
                                                                                                3   0   0 3

Course Objectives:
1. To explore the concept of Graphs and related algorithms.
2. To learn new ways to model, store, retrieve and analyze graph-structured data.
3. To be aware of advanced concepts in graph analytic techniques and its applications.

Course Outcomes: Upon the Successful Completion of the Course, the Students would be able to:
1. Understand Large-scale Graph and its Characteristics
2. Analyze Breadth-First Search Algorithm
3. Illustrate Recent Advances in Scalable Network Generation

UNIT - I
Introduction and Application of Large-scale Graph: Characteristics, Complex Data Sources - Social Networks, Simulations, Bioinformatics; Categories- Social, Endorsement, Location, Co-occurrence graphs; Graph Data structures, Parallel, Multicore and Graph Algorithms

UNIT - II Algorithms: Search and Paths
A Work-Efficient Parallel Breadth-First Search Algorithm (or How To Cope With the Nondeterminism of Reducers), Multi-Objective Shortest Paths

UNIT - III Algorithms: Structure
Multicore Algorithms for Graph Connectivity Problems, Distributed Memory Parallel Algorithms for Massive Graphs, Massive-Scale Distributed Triangle Computation and Applications

UNIT - IV Models
Recent Advances in Scalable Network Generation, Computational Models for Cascades in Massive Graphs, Executing Dynamic Data-Graph Computations Deterministically Using Chromatic Scheduling.

UNIT - V Frameworks and Software
Graph Data Science Using Neo4j, A Cloud-Based Approach to Big Graphs, Interactive Graph Analytics at Scale in Arkouda

TEXT BOOKS:
1. David A. Bader, Massive Graph Analytics, CRC Press

REFERENCES:
1. Stanley Wasserman, Katherine Faust, "Social Network Analysis: Methods and Applications", (Structural Analysis in the Social Sciences), Cambridge University Press, 1995.
2. Matthew O. Jackson, "Social and Economic Networks", Princeton University Press, 2010.
3. Tanja Falkowski, "Community Analysis in Dynamic Social Networks", (Dissertation), University Magdeburg, 2009.

CYBER SECURITY

M.Tech SE II Year I Sem.   **L   T   P C**
                                                                      3   0   0 3

(OPEN ELECTIVE)

1. To understand various types of cyber-attacks andcyber-crimes
2. To learn threats and risks within context of the cybersecurity
3. To have an overview of the cyber laws & concepts of cyberforensics
4. To study the defensive techniques against theseattacks

Course Outcomes:

1. Analyze and evaluate the cyber security needs of anorganization.
2. Understand Cyber Security Regulations and Roles of InternationalLaw
3. Design and develop a security architecture for anorganization.
4. Understand fundamental concepts of data privacyattacks

UNIT - I

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT - II

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy.
Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics

UNIT - III

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.

UNIT- IV

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations

UNIT - V

Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Datalinking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc
Cybercrime: Examples and Mini-Cases
Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances. **Mini-Cases:** The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

TEXT BOOKS:
1. Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives,Wiley
2. B. B. Gupta, D. P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN9780815371335,2018.

REFERENCES:
1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&FGroup.

Network Programming

M.Tech SE II Year I Sem.

L	T	P	C
3	0	0	3

(OPEN ELECTIVE)

Objectives:
- To understand Linux utilities
- To understand file handling, signals
- To understand IPC, network programming in Java
- To understand processes to communicate with each other across a Computer Network.

UNIT I
Linux Utilities- File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking utilities, Filters, Text processing utilities and Backup utilities. Bourne again shell(bash) - Introduction, pipes and redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples.Review of C programming concepts-arrays, strings (library functions), pointers, function pointers, structures, unions, libraries in C.

UNIT II
Files- File Concept, File types File System Structure, Inodes, File Attributes, file I/O in C using system calls, kernel support for files, file status information-stat family, file and record locking-lockf and fcntl functions, file permissions- chmod, fchmod, file ownership-chown, lchown , fchown, links-soft links and hard links – symlink, link, unlink. File and Directory management – Directory contents, Scanning Directories- Directory file APIs. Process- Process concept, Kernel support for process, process attributes, process control – process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process.

UNIT III
Signals- Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.Interprocess Communication - Introduction to IPC mechanisms, Pipes- creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs(Named pipes), differences between unnamed and named pipes, popen and pclose library functions, Introduction to message queues, semaphores and shared memory.Message Queues- Kernel support for messages, UNIX system V APIs for messages, client/server example.Semaphores-Kernel support for semaphores, UNIX system V APIs for semaphores.

UNIT IV
Shared Memory- Kernel support for shared memory, UNIX system V APIs for shared memory, client/server example. Network IPC - Introduction to Unix Sockets, IPC over a network, Client-Server model ,Address formats(Unix domain and Internet domain), Socket system calls for Connection Oriented - Communication, Socket system calls for Connectionless-Communication, Example-Client/Server Programs- Single Server-Client connection, Multiple simultaneous clients, Socket options – setsockopt, getsockopt, fcntl.

UNIT V

Network Programming in Java-Network basics, TCP sockets, UDP sockets (datagram sockets), Server programs that can handle one connection at a time and multiple connections (using multithreaded server), Remote Method Invocation (Java RMI)-Basic RMI Process, Implementation details-Client-Server Application.

TEXT BOOKS:
1. Unix System Programming using C++, T.Chan, PHI.(Units II,III,IV)
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.(Unit I)
3. An Introduction to Network Programming with Java, Jan Graba, Springer, 2010.(Unit V)
4. Unix Network Programming ,W.R. Stevens, PHI.(Units II,III,IV)
5. Java Network Programming,3rd edition, E.R. Harold, SPD, O'Reilly.(Unit V)

REFERENCES:
1. Linux System Programming, Robert Love, O'Reilly, SPD.
2. Advanced Programming in the UNIX environment, 2nd Edition, W.R.Stevens, Pearson Education.
3. UNIX for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education.
4. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones, Wrox, Wiley India Edition.
5. Unix Network Programming The Sockets Networking API, Vol.-I,W.R.Stevens, Bill Fenner, A.M.Rudoff, Pearson Education.
6. Unix Internals, U.Vahalia, Pearson Education.
7. Unix shell Programming, S.G.Kochan and P.Wood, 3rd edition, Pearson Education.
8. C Programming Language, Kernighan and Ritchie, PHI