

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

## M. Tech. SOFTWARE ENGINEERING

## EFFECTIVE FROM ACADEMIC YEAR 2019 - 20 ADMITTED BATCH

## R19 COURSE STRUCTURE AND SYLLABUS

## I YEAR I – Semester

Course Code	Course Title	L	T	P	Credits
Professional Core - I	Principles of Software Engineering	3	0	0	3
Professional Core - II	Advanced Data Structures	3	0	0	3
Professional Elective - I	1. Object Oriented Analysis & Design 2. Software Requirements & Estimation 3. Software Development Methodologies	3	0	0	3
Professional Elective - II	1. Software Project Management 2. Software Metrics 3. Mobile Application Security	3	0	0	3
Lab - I	Advanced Data Structures Lab	0	0	4	2
Lab - II	Object Oriented Analysis & Design Lab	0	0	4	2
	Research Methodology & IPR	2	0	0	2
Audit - I	Audit Course - I	2	0	0	0
	<b>Total</b>	<b>16</b>	<b>0</b>	<b>8</b>	<b>18</b>

## I YEAR II – Semester

Course Code	Course Title	L	T	P	Credits
Professional Core - III	Design Patterns	3	0	0	3
Professional Core - IV	Machine Learning	3	0	0	3
Professional Elective - III	1. Software Testing Methodologies 2. Mobile Application and API Development 3. Object Oriented Software Engineering	3	0	0	3
Professional Elective - IV	1. Software Quality Assurance and Testing 2. Secure Software Engineering 3. Internet Technologies & Services	3	0	0	3
Lab - III	Machine Learning Lab	0	0	4	2
Lab - IV	Software Testing Methodologies Lab	0	0	4	2
	Mini Project with Seminar	0	0	4	2
Audit - II	Audit Course - II	2	0	0	0
	<b>Total</b>	<b>14</b>	<b>0</b>	<b>12</b>	<b>18</b>

**II YEAR I – Semester**

Course Code	Course Title	L	T	P	Credits
Professional Elective - V	1. Agile Development Methodologies 2. Web Services Testing 3. Test Automation	3	0	0	3
Open Elective	Open Elective	3	0	0	3
Dissertation	Dissertation Work Review - II	0	0	12	6
	<b>Total</b>	<b>6</b>	<b>0</b>	<b>12</b>	<b>12</b>

**II YEAR II - SEMESTER**

Course Code	Course Title	L	T	P	Credits
Dissertation	Dissertation Work Review - III	0	0	12	6
Dissertation	Dissertation Viva-Voce	0	0	28	14
	<b>Total</b>	<b>0</b>	<b>0</b>	<b>40</b>	<b>20</b>

**\*For Dissertation Work Review - I, Please refer 7.8 in R19 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

**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.

**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, 3<sup>rd</sup> 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)**

**Pre-Requisites:** UG level course in Data Structures

**Course Objectives:**

- The student should be able to choose appropriate data structures, understand the ADT/libraries, and use it to design algorithms for a specific problem.
- Students should be able to understand the necessary mathematical abstraction to solve problems.
- To familiarize students with advanced paradigms and data structure used to solve algorithmic problems.
- Student should be able to come up with analysis of efficiency and proofs of correctness.

**Course Outcomes:** After completion of course, students would be able to:

- Understand the implementation of symbol table using hashing techniques.
- Understand the implementation of symbol table using hashing techniques.
- Develop algorithms for text processing applications.
- Identify suitable data structures and develop algorithms for computational geometry problems.

**UNIT - I**

**Dictionaries:**

Definition, Dictionary, Abstract Data Type, Implementation of Dictionaries.

**Hashing:**

Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.

**UNIT - II**

**Skip Lists:**

Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists.

**UNIT - III**

**Trees:**

Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees

**UNIT - IV**

**Text Processing:**

String Operations, Brute-Force Pattern Matching, The Boyer- Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem

**UNIT - V**

**Computational Geometry:**

One Dimensional Range Searching, Two-Dimensional Range Searching, constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quadrees, k-D Trees.

Recent Trends in Hashing, Trees, and various computational geometry methods for efficiently solving the new evolving problem

**References:**

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004.
2. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.

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

**OBJECT ORIENTED ANALYSIS & DESIGN (Professional Elective - I)**

**Prerequisites:** Software Engineering

**Course Objectives:**

- To train students on object modeling
- To apply unified process phases
- To apply unified modeling language for software design of any applications
- To study case studies for OOAD

**Course Outcomes:**

- Will be able to use UML notations
- Can apply unified process in software development
- Will be able to perform analysis and design using object modeling

**UNIT - I**

Introduction to UML: Importance of modeling, principles of modeling, object-oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

**UNIT - II**

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.

Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

**UNIT - III**

Basic Behavioral Modeling-I: Interactions, Interaction diagrams.

Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

**UNIT - IV**

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

**UNIT - V**

Patterns and Frameworks, Artifact Diagrams. **Case Study:** The Unified Library application

**Text Books:**

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education 2nd Edition
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

**Reference Books:**

1. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY-Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TMH.

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

**SOFTWARE REQUIREMENTS & ESTIMATION (Professional Elective - I)**

**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 – I Semester (Software Engineering)**

**SOFTWARE DEVELOPMENT METHODOLOGIES (Professional Elective-I)**

**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, Behavioural 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 software architecture.

**Software Design Approaches, Structured Analysis, Structured Design.**

**UNIT - IV**

**Object Oriented Concepts and Principles,** Object Oriented Analysis, Object Oriented Design,

**Modelling component-level design:** Designing class-based components, conducting component-level design, object constraint language, designing conventional components.

**UNIT - V**

**User Interface Design, Performing User interface design:** Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation. Coding and Documentation.

**Text Books:**

1. Software Engineering: A practitioner's Approach, Roger S Pressman, sixth edition. McGraw

- Hill International Edition, 2005 (Unit 1, 2, 3, 5)
2. Software Engineering by Jibitesh Mishra, Ashok Mohanty. Pearson. (Unit 4, 5)

**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, 3<sup>rd</sup> 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, RP 2009
12. Software Engineering Handbook, Jessica Keyes, Auerbach, 2003.

**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, Pareto 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 Proposed 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)**

**MOBILE APPLICATION SECURITY (Professional Elective - II)**

**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. – 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 Leftist 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, 2<sup>nd</sup> 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, 1<sup>st</sup> 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, 2<sup>nd</sup> Edition, Cengage Learning.

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

**OBJECT ORIENTED ANALYSIS & DESIGN LAB (Lab – II)**

**Course Objectives:**

- Main objective of this lab is to enable the student to practice the object-oriented analysis and design through UML on a particular application (project).
  - Understand how UML supports the entire OOAD process.
  - Become familiar with all phases of OOAD
1. The student should take up the case study of *Unified Library Application* which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
  2. Draw the following diagrams using UML for an ATM system whose description is given below.

UML diagrams to be developed are:

1. Use Case Diagram.
2. Class Diagram.
3. Sequence Diagram.
4. Collaboration Diagram.
5. State Diagram
6. Activity Diagram.
7. Component Diagram
8. Deployment Diagram.
9. Test Design.

**Description for an ATM System**

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) – both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

*The ATM must be able to provide the following services to the customer:*

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM,

subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.

3. A customer must be able to make a transfer of money between any two accounts linked to the card.

4. A customer must be able to make a balance inquiry of any account linked to the card.

5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine. The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.) If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back. If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction. The ATM will provide the customer with a printed receipt for each successful transaction. The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. 143 COMPUTER SCIENCE AND ENGINEERING 2013-14 When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

3. Study of any testing tool (e.g. Win runner)

4. Study of any web testing tool (e.g. Selenium)

5. Study of any bug tracking tool (e.g. Bugzilla, bugbit)

6. Study of any test management tool (e.g. Test Director)

7. Study of any open source-testing tool (e.g. Test Link)

**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.
4. Head First Design Patterns, Eric Freeman, O'reily publications



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

**MACHINE LEARNING (PC – IV)**

**Course Objectives:**

- To learn the concept of how to learn patterns and concepts from data without being explicitly programmed in various IOT nodes.
- To design and analyse various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
- Explore supervised and unsupervised learning paradigms of machine learning.
- To explore Deep learning technique and various feature extraction strategies.

**Course Outcomes:** After completion of course, students would be able to:

- Extract features that can be used for a particular machine learning approach in various IOT applications.
- To compare and contrast pros and cons of various machine learning techniques and to get an insight of when to apply a particular machine learning approach.
- To mathematically analyse various machine learning approaches and paradigms.

**UNIT - I**

**Supervised Learning (Regression/Classification)**

Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes.

Linear models: Linear Regression, Logistic Regression, Generalized Linear Models.

Support Vector Machines, Nonlinearity and Kernel Methods.

Beyond Binary Classification: Multi-class/Structured Outputs, Ranking.

**UNIT – II**

**Unsupervised Learning:**

Clustering: K-means/Kernel K-means.

Dimensionality Reduction: PCA and kernel PCA.

Matrix Factorization and Matrix Completion.

Generative Models (mixture models and latent factor models).

**UNIT - III**

Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning

Theory, Ensemble Methods (Boosting, Bagging, Random Forests)

**UNIT - IV**

Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning

**UNIT - V**

Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference.

Recent trends in various learning techniques of machine learning and classification methods for IOT applications. Various models for IOT applications.

**References:**

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012

2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online)
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007

**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
- 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:**

1. Best for Programmers With Java Experience: Android Programming: The Big Nerd Ranch Guide
2. Best for Quick Answers: Android Cookbook: Problems and Solutions for Android Developers
3. 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)**

**OBJECT ORIENTED SOFTWARE ENGINEERING (Professional Elective - III)**

**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)**

**SOFTWARE QUALITY ASSURANCE AND TESTING (Professional Elective - IV)**

**Course Objectives:** The student should be able to:

- To understand software testing and quality assurance as a fundamental component of software life cycle
- To define the scope of SW T&QA projects
- To efficiently perform T&QA activities using modern software tools
- To estimate cost of a T&QA project and manage budgets
- To prepare test plans and schedules for a T&QA project
- To develop T&QA project staffing requirements
- To effectively manage a T&QA project

**UNIT - I**

**Software Quality Assurance and Standards:** The Software Quality challenge, What is Software Quality, Software Quality factors, The components of Software Quality Assurance system, Software Quality Metrics, Costs of Software Quality, Quality Management Standards, Management and its role in Software Quality Assurance, SQA unit and other actors in SQA system. - **(Chapters: 1-4, 21-23, 25, 26) of T3 Quality Standards:** ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma and other latest quality standards **(Refer Internet and R11, R12, R13).**

**UNIT - II**

**Software Testing Strategy and Environment:** Minimizing Risks, Writing a Policy for Software Testing, Economics of Testing, Testing-an organizational issue, Management Support for Software Testing, Building a Structured Approach to Software Testing, Developing a Test Strategy **Building Software Testing Process:** Software Testing Guidelines, workbench concept, Customizing the Software Testing Process, Process Preparation checklist - **(Chapters: 2,3) of T1 Software Testing Techniques:** Dynamic Testing – Black Box testing techniques, White Box testing techniques, Static testing, Validation Activities, Regression testing -**(Chapters: 4, 5, 6, 7, 8) of T2**

**UNIT - III**

**Software Testing Tools:** Selecting and Installing Software Testing tools – **(Chapter 4) of T1.** Automation and Testing Tools - **(Chapter 15) of T2** Load Runner, Win runner and Rational Testing Tools, Silk test, Java Testing Tools, JMetra, JUNIT and Cactus. **(Refer Internet and R9, R10)**

**UNIT - IV**

**Testing Process Seven Step Testing Process – I:** Overview of the Software Testing Process, Organizing of Testing, Developing the Test Plan, Verification Testing, Validation Testing. **(Chapters 6, 7, 8, 9, 10) of T1**

**UNIT - V**

**Seven Step Testing Process – II:** Analyzing and Reporting Test results, Acceptance and Operational Testing, Post-Implementation Analysis **Specialized Testing Responsibilities:** Software Development Methodologies, Testing Client/Server Systems **(Chapters 12, 13, 14, 15) of T1.**

**TEXT BOOKS:**

1. Effective Methods for Software Testing, Third edition, *William E. Perry*, Wiley India, 2009
2. Software Testing – Principles and Practices, *Naresh Chauhan*, Oxford University Press, 2010.

3. Software Quality Assurance – From Theory to Implementation, *Daniel Galin*, Pearson Education, 2009.

**References:**

1. Testing Computer Software, Cem Kaner, Jack Falk, Hung Quoc Nguyen, Wiley India, rp2012.
2. Software Testing – Principles, Techniques and Tools, *M.G.Limaye*, Tata McGraw-Hill, 2009.
3. Software Testing - A Craftsman's approach, *Paul C. Jorgensen*, Third edition, Auerbach Publications, 2010.
4. Foundations of Software Testing, *Aditya P. Mathur*, Pearson Education, 2008.
5. Software Testing and Quality Assurance – Theory and Practice, *Kshirasagar Naik, Priyadashi Tripathy*, Wiley India, 2010.
6. Software Testing, *Ron Patton*, Second edition, Pearson Education, 2006.
7. Software Testing and Analysis – Process, Principles and Techniques, *Mauro Pezze, Michal Young*, Wiley India, 2008.
7. Software Testing Techniques, Boris Beizer, Second edition, Wiley India, 2006
8. Foundations of Software Testing, Dorothy Graham, et al., Cengage learning, 2007, rp 2010.
9. Software Testing - Effective Methods, Tools and Techniques, *Renu Rajani, Pradeep Oak*, Tata McGraw-Hill, rp2011.
10. Software Automation Testing Tools for Beginners, *Rahul Shende*, Shroff Publishers and Distributors, 2012.
11. Software Testing Tools, *K.V.K.K. Prasad*, Dream Tech Press, 2008.
12. Software Testing Concepts and Tools, *Nageswara Rao Pusuluri*, Dream Tech press, 2007.
13. Software Quality Assurance, *Milind Limaye*, Tata McGraw-Hill, 2011.
14. Software Quality – Theory and Management, *Alan C. Gillies*, Second edition, Cengage Learning, 2009.
15. Software Quality – A Practitioner's approach, *Kamna Malik, Praveen Choudhary*, Tata McGraw-Hill, 2008.
16. Software Quality Models and Project Management in a Nutshell, *Shailesh Mehta*, Shroff Publishers and Distributors, 2010.
17. Software Quality Engineering – Testing, Quality Assurance and Quantifiable Improvement, *Jeff Tian*, Wiley India, 2006.
18. Software Quality, *Mordechai Ben-Menachem/Garry S. Marliss*, Cengage Learning, 2010.



**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 throughout 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)**

**INTERNET TECHNOLOGIES AND SERVICES (Professional Elective - IV)**

**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 web

service on axis. Web Services Interoperability – Creating java and .Net client applications for an Axis Web 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 3<sup>rd</sup> edition, WILEY Dreamtech .
2. The complete Reference Java 7<sup>th</sup> 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, 4<sup>th</sup> 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.

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

**MACHINE LEARNING 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 the probability 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 design single 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

**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)**

**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, 2<sup>nd</sup> 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. – 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 together, Real customer involvement, 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, 11<sup>th</sup> Indian Reprint, O'Reilly, 2018

**References:**

1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4<sup>th</sup> Indian Reprint, 2018
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5<sup>th</sup> 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 Molyneux

**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)**

**TEST AUTOMATION (Professional Elective - V)**

**Course Objectives:** Your studies will enable you to develop:

- Know the real value to be expected from test automation
- Discover the key traits that will make your test automation project succeed
- Be aware of the different considerations to consider when planning automated tests vs. manual tests
- Determine who should implement the tests and the implications of this decision
- Architect the test project and fit it to the architecture of the tested application
- Design and implement highly reliable automated tests
- Begin gaining value from test automation earlier
- Integrate test automation into the business processes of the development team
- Leverage test automation to improve your organization's performance and quality, even without formal authority
- Understand how different types of automated tests will fit into your testing strategy, including unit testing, load and performance testing, visual testing, and more

**UNIT - I**

**Why and What is Test Automation:** Value of test automation, From Manual to Automated Testing, Tools, Code coverage

**UNIT - II**

**Test Automation and Architecture:** Business Processes, Test Architecture, Test Environments

**UNIT - III**

**How to perform automation testing:** Testing approaches, setting up the environment, designing test case

**UNIT - IV**

**Test case development:** Designing and coding first test case, completing the test case, investigating failures, adding more tests

**UNIT - V**

**CI, CD and CT:** Continuous Integration, Deployment and Testing, Acceptance test driven development, units tests and TDD, other types of automated testing, performance testing.

**Text Books:**

1. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Software Projects by Arnan Axelrod
2. Agile Automation and Unified Functional Testing by Rajeev Gupta

**Reference Books:**

1. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series)
2. Test Automation Using Selenium Webdriver with Java: Step by Step Guide by Navneesh Garg
3. Experiences of Test Automation: Case Studies of Software Test Automation by Graham

**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. Akyeamong K (2003) Teacher training in Ghana - does it count? Multi-site teacher education research project (MUSTER) country report 1. London: DFID.

4. Akyeamong 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](http://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 Training-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.