NMAM INSTITUTE OF TECHNOLOGY, NITTE

SCHEME OF TEACHING AND SYLLABUS

FOR

M.TECH. SOFTWARE ENGINEERING

AUTONOMOUS SCHEME

Choice Based Credit System

AY 2017-2019

NMAM INSTITUTE OF TECHNOLOGY, NITTE

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH. SOFTWARE ENGINEERING
(AUTONOMOUS SCHEME: 2017-2019)

I Semester **(25 Credits)**

Sl. No.	Code	Subject	Credits	C.I.A.	SEE	Credits
			Lect/Tut/Prac/ Self Study			
1	17SSE101	Software Testing	4+0+0+S	50	50	5
2	17SSE102	Advanced Algorithms	4+0+2+0	50	50	5
3	17SSE103	Advances in Data Base Management Systems	4+0+2+0	50	50	5
4	17SSE11X	Elective-I	4+0+0+0	50	50	4
5	17SSE12X	Elective-II	4+0+0+0	50	50	4
6	17SSE104	Research Experience through Practice-I	0+0+4+0	50	-	2
			Total	**300**	**250**	**25**

Elective I

17SSE111 : Distributed Systems
17SSE112 : Advances in Computer Networks
17SSE113 : Decision Support Systems and ERP

Elective - II

17SSE121: Web Services
17SSE122 : Soft Computing
17SSE123 : Advances in Storage Area Networks
17SSE124 : Agile Technologies

NOTE:

At the end of ***Research Experience through Practice-I*** in the first semester, PG students should be able to identify a research problem, with clear objectives and methodologies backed by extensive literature review. All the PG students may be asked to submit a research proposal and a presentation at the end of the first semester.

NMAM INSTITUTE OF TECHNOLOGY, NITTE

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH. SOFTWARE ENGINEERING
(AUTONOMOUS SCHEME: 2017-2019)

II Semester **(25Credits)**

Sl. No.	Code	Subject	Credits Lect/Tut/Prac/Self Study	C.I.A.	S.E.E	Credits
1	17SSE201	Software Project Planning & Management	4+0+2+0	50	50	5
2	17SSE202	Enterprise Application Programming	4+0+2+0	50	50	5
3	17SSE203	Software Metrics & Quality Assurance	4+0+0+S	50	50	5
4	17SSE21X	Elective-III	4+0+0+0	50	50	4
5	17SSE22X	Elective-IV	4+0+0+0	50	50	4
6	17SSE204	Research Experience through Practice-II	0+0+4+0	50	-	2
	Total 25			**300**	**250**	**25**

Elective III

17SSE211 Data Mining & Data Warehousing
17SSE212 Information and Network Security
17SSE213 Distributed Operating Systems
17SSE214 Design Patterns

Elective IV

17SSE221 Information Retrieval
17SSE222 Internet of Things
17SSE223 General Purpose Computation on GPU

NOTE:

At the end of ***Research Experience through Practice-II*** in the second semester, students are expected to write a full research paper based on the Mathematical modelling / Design calculations / computer simulations / Preliminary experimentation / testing carried out during second semester.

Course Title: Software Testing **Course Code: 17SSE101**
Credits(L:T:P:S): 4:0:0:S **Core/Elective: Core**
Type of Course: Lecture & Practical **Total Contact Hours:52**

COURSE OBJECTIVES

- To explore the basics and goals of software testing.
- To discuss various types of software testing and its techniques
- To list out various tools which can be used for automating the testing process
- To introduce various software quality standards for establishing quality environment
- To discuss various methods and evaluation procedures for improving the quality Models

TOPICS

UNIT-I

Basics of Software Testing and Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. Examples: Generalized pseudo code, The triangle problem, The Next Date function, The commission problem, The SATM (Simple Automatic Teller Machine) problem.

10 hours

UNIT-II

Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the NextDate function, Test cases for the commission problem, Guidelines and observations. Data Flow Testing: Definition-Use testing, Slice-based testing, Guidelines and observations. Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. Integration Testing: A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations, Case study.

10 hours

UNIT-III

System Testing: Threads, Basic concepts for requirements specification, Finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example. Interaction Testing: Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing,. Issues in Object-Oriented Testing: Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism, Levels of object-oriented testing, GUI testing, Dataflow testing for object-oriented software, Examples. Class Testing: Methods as units, Classes as units.

12 hours

UNIT-IV

Object-Oriented Integration Testing: UML support for integration testing, MM-paths for object-oriented software, A framework for object-oriented dataflow integration testing. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program. Object-Oriented System Testing: Currency converter UML description, UML-based system testing, State chart-based system testing.

10 hours

UNIT-V

Exploratory Testing: The context-driven school, Exploring exploratory testing, Exploring a familiar example, Exploratory and context-driven testing observations. Model-Based Testing: Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing. Test-Driven Development: Test-then code cycles, Automated test execution, Java and JUnit example, Remaining questions, Pros, cons, and open questions of TDD, Retrospective on MDD versus TDD.

10 hours

LABORATORY WORK

1. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive at least 10 different test cases, execute these test cases and discuss the test results.
2. Design, develop, code and run the program in any suitable language to solve the NextDate problem. Analyze it from the perspective of decision table-based testing, derive at least 10 different test cases, execute these test cases and discuss the test results.
3. Design, develop, code and run the program in any suitable object-oriented language to solve the calendar problem. Analyze it from the perspective of OO testing, derive test cases to test the method that increment the date and the method that increments the month., execute these test cases and discuss the test results.
4. Design, develop, code and run the program in any suitable object-oriented language to solve the currency converter problem. Analyze it from the perspective of use case-based system testing, derive appropriate system test cases., execute these test cases and discuss the test results.

COURSE OUTCOMES:

Upon Completion of the course, students shall be able to

- Compare and pick out the right type of software testing process for any given real world problem
- Carry out the software testing process in efficient way
- Automate the testing process by using several testing tools
- Establish a quality environment as specified in standards for developing quality software
- Analyze and improve the quality procedures based on the past experience

TEXT BOOKS:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2013.

REFERENCE BOOKS:

1. Aditya P Mathur: Foundations of Software Testing, Pearson, 2008.
2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, John Wiley & Sons, 2008.

Course Title: Advanced Algorithms **Course Code: 17SSE102**
Credits(L:T:P:S): 4:0:2:0 **Core/Elective: Core**
Type of Course: Lecture & Practical **Total Contact Hours:52**

COURSE OBJECTIVES

- To learn the graph search algorithms.
- To study network flow and linear programming problems.
- To learn the hill climbing and dynamic programming design techniques.
- To develop recursive backtracking algorithms.
- To get an awareness of NP completeness and randomized algorithms.

TOPICS

UNIT-I

Review of Analysis Techniques: Growth of Functions: Asymptotic notations; Standard notations and common functions; Recurrences and Solution of Recurrence equations- The substitution method, The recurrence – tree method, The master method; Amortized Analysis: Aggregate, Accounting and Potential Methods.

10 hours

UNIT-II

Graph Algorithms: Bellman - Ford Algorithm; Single source shortest paths in a DAG; Johnson's Algorithm for sparse graphs; Flow networks and Ford-Fulkerson method; maximum bipartite matching. Polynomials and the FFT: Representation of polynomials; The DFT and FFT; Efficient implementation of FFT.

12 hours

UNIT-III

Number -Theoretic Algorithms: Elementary notions; GCD; Modular Arithmetic; Solving modular linear equations; The Chinese remainder theorem; Powers of an element; RSA cryptosystem; Primality testing; Integer factorization.

10 hours

UNIT-IV

String-Matching Algorithms: Naïve string Matching; Rabin - Karp algorithm; String matching with finite automata;
Knuth-Morris-Pratt algorithm; Boyer – Moore algorithms.

10 hours

UNIT-V

Probabilistic and Randomized Algorithms: Probabilistic algorithms; Randomizing deterministic algorithms, Monte Carlo and Las Vegas algorithms; Probabilistic numeric algorithms. **10 hours**

LABORATORY WORK

1. Design, develop, and run a program in any language to implement the Bellman-Ford algorithm and determine its performance.
2. Design, develop, and run a program in any language to implement Johnson's algorithm and determine its performance.
3. Design, develop, and run a program in any language to implement a Monte Carlo algorithm to test the primality of a given integer and determine its performance.
4. Design, develop, and run a program in any language to solve the string matching problem using naïve approach and the KMP algorithm and compare their performances.
5. Design, develop, and run a program in any language to solve modular linear equations.
6. Design, develop, and run a program in any language to implement the FFT algorithm efficiently.

COURSE OUTCOMES:

Upon completion of the course, students shall be able to

- Design and apply iterative and recursive algorithms.
- Design and implement optimization algorithms in specific applications.
- Design appropriate shared objects and concurrent objects for applications.
- Implement and apply concurrent linked lists, stacks, and queues.

TEXT BOOKS:

1. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3rd Edition, Prentice-Hall of India, 2010.
2. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning, 2002.

REFERENCE BOOKS:

1. Ellis Horowitz, Sartaj Sahni, S.Rajasekharan: Fundamentals of Computer Algorithms, 2nd Edition, Universities press, 2007.

Course Title: Advances In Database Management Systems **Course Code: 17SSE103**
Credits(L:T:P:S): 4:0:2:0 **Core/Elective: Core**
Type of Course: Lecture & Practical **Total Contact Hours:52**

COURSE OBJECTIVES:

- To acquire knowledge on parallel and distributed databases and its applications.
- To study the usage and applications of Object Oriented database
- To understand the basic concepts, principles of intelligent databases.
- To understand the advanced topics of data warehousing and mining .
- To learn emerging and advanced data models
- To acquire inquisitive attitude towards research topics in databases.

TOPICS:

UNIT-I

Review of Relational Data Model and Relational Database Constraints: Relational model concepts; Relational model constraints and relational database schemas; Update operations, transactions and dealing with constraint violations.

Overview of Object-Oriented Concepts – Objects, Encapsulation, Polymorphism, Type and class hierarchies etc.

10 Hours

UNIT-II

Object and Object-Relational Databases: Object Oriented Concepts: Objects, complex objects; Object model of ODMG, Object definition Language ODL; Object Query Language OQL; Overview of C++ language binding;

Conceptual design of Object database. Overview of object relational features of SQL; Object-relational features of Oracle; Implementation and related issues for extended type systems; The nested relational model.

10 Hours

UNIT- III

Parallel and Distributed Databases: Architectures for parallel databases; Parallel query evaluation; Parallelizing individual operations; Parallel query optimizations; Introduction to distributed databases; Distributed DBMS architectures; Storing data in a Distributed DBMS; Distributed catalog management; Distributed Query processing; Updating distributed data; Distributed transactions; Distributed Concurrency control and Recovery.

10 Hours

UNIT-IV

Data Warehousing, Decision Support and Data Mining: Introduction to decision support; OLAP, multidimensional model; Window queries in SQL; Finding answers quickly; Implementation techniques for OLAP; Data Warehousing; Views and Decision support, View materialization, Maintaining materialized views. Introduction to Data Mining; Counting co-occurrences; Mining for rules; Tree-structured rules; Clustering; Similarity search over sequences; Incremental mining and data streams; Additional data mining tasks.

12 Hours

UNIT-V

Enhanced Data Models for Some Advanced Applications: Active database concepts and triggers; Temporal, Spatial, and Deductive Databases – Basic concepts. More Recent Applications: Mobile databases, Multimedia databases, Geographical Information Systems, Genome data management

10 Hours

ADBMS LABORATORY WORK

LABORATORY OBJECTIVES:

- To acquire practical knowledge on advanced databases and its applications.
- To understand and work on areas like Storage, Retrieval, Multi valued attributes, Triggers and other complex objects, Algorithms etc related to ADBMS.
- To design and implement recent applications database for better interoperability
- Note: The following experiments may be implemented on MySQL/ORACLE or any other suitable RDBMS with support for Object features

1. Develop a database application to demonstrate storing and retrieving of BLOB and CLOB objects.

Write a binary large object (BLOB) to a database as either binary or character (CLOB) data, depending on the type of the field in your data source. To write a BLOB value to the database, issue the appropriate INSERT or UPDATE statement and pass the BLOB value as an input parameter. If your BLOB is stored as text, such as a SQL Server text field, pass the BLOB as a string parameter. If the BLOB is stored in binary format, such as a SQL Server image field, pass an array of type byte as a binary parameter.

Once storing of BLOB and CLOB objects is done, retrieve them and display the results accordingly.

2. Develop a database application to demonstrate the representation of multi valued attributes, and the use of nested tables to represent complex objects. Write suitable queries to demonstrate their use.

Consider Purchase Order Example: This example is based on a typical business activity: managing customer orders. Need to demonstrate how the application might evolve from relational to object-relational, and how you could write it from scratch using a pure object-oriented approach.

Show how to implement the schema -- Implementing the Application under the Relational Model -- using only Oracle's built-in data types. Build an object-oriented application on top of this relational schema using object views.

3. Design and develop a suitable Student Database application by considering appropriate attributes. Couple of attributes to be maintained includes the Attendance of a student in each subject for which he/she has enrolled and Internal Assessment Using TRIGGERS, write active rules to do the following:

Whenever the attendance is updated, check if the attendance is less than 85%; if so, notify the Head of the Department concerned.

Whenever, the marks in an Internal Assessment Test are entered, check if the marks are less than 40%; if so, notify the Head of the Department concerned.

Use the following guidelines when designing triggers:

- Use triggers to guarantee that when a specific operation is performed, related actions are performed.
- Use database triggers only for centralized, global operations that should be fired for the triggering statement, regardless of which user or database application issues the statement.
- Do not define triggers that duplicate the functionality already built into Oracle. For example, do not define triggers to enforce data integrity rules that can be easily enforced using declarative integrity constraints.
- Limit the size of triggers (60 lines or fewer is a good guideline). If the logic for your trigger requires much more than 60 lines of PL/SQL code, it is better to include most of the code in a stored procedure, and call the procedure from the trigger.
- Be careful not to create recursive triggers. For example, creating an AFTER UPDATE statement trigger on the EMP table that itself issues an UPDATE statement on EMP causes the trigger to fire recursively until it has run out of memory.

4. Design, develop, and execute a program to implement specific Apriori algorithm for mining association rules. Run the program against any large database available in the public domain and discuss the results.

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

COURSE OUTCOMES:

Upon completion of the course, the students will be able to

- Select the appropriate high performance database like parallel and distributed database Model and represent the real world data using object oriented database
- Embed the rule set in the database to implement data warehousing of mining
- Choose and design database for recent applications database for better interoperability

TEXT BOOKS:

1. Elmasri and Navathe: Fundamentals of Database Systems, Pearson Education, 2013.
2. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw-Hill, 2013.

REFERENCE BOOKS:

1.Abraham Silberschatz, Henry F. Korth, S. Sudarshan: Database System Concepts, 6th Edition, McGraw Hill, 2010.

Course Title: Distributed Systems **Course Code: 17SSE111**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES:

- To learn Basic Concepts of DSM, Hardware DSM
- To understand File Sharing, DFS Implementation, Replication in DFS,
- To explore the concepts of Cryptography, Secure channels, Access control
- To understand real time Distributed Operating System and emerging trends in Distributed Computing

TOPICS

UNIT-I

Distributed System management: Introduction, Resource management, Task Assignment Approach, Load-Balancing Approach, Load-Sharing Approach, Process management in a Distributed Environment, Process Migration, Threads, Fault Tolerance.

.

10 hours

UNIT-II

Distributed Shared Memory: Introduction, Basic Concepts of DSM, Hardware DSM, Design Issue in DSM Systems, Issue in Implementing DSM Systems, Heterogeneous and other DSM Systems, Case Studies.

10 hours

UNIT-III

Distributed File System: Introduction to DFS, File Models, Distributed File System Design, Semantics of File Sharing, DFS Implementation, File Caching in DFS, Replication in DFS, Case studies. **Naming:** Introduction, Desirable features of a good naming system, Basic concepts, System-oriented names, Object-locating mechanisms, Issues in designing human-oriented names, Name caches, Naming and security, Case study:Domain name service.

. **10 hours**

UNIT-IV

Security in distributed systems: Introduction, Cryptography, Secure channels, Access control, Security Management, Case studies.

. **10 hours**

UNIT-V

Real-Time Distributed operating Systems: Introduction, Design issues in real-time distributed systems, Realtime communication, Real-time scheduling, Case study: Real-time communication in MARS. **Emerging Trends in distributed Computing:** Introduction to emerging trends, Grid Computing, SOA, Cloud computing, Emerging Trends. Programming to demonstrate RMI, EJB, JMS, SOAP etc.

10 hours

COURSE OUTCOMES**:**

Upon completion of the course, students shall be able to
· Explain distributed systems concepts
· Demonstrate an ability to apply theory and techniques to unseen problems.
· Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
· Explore the various resource management techniques for distributed systems

TEXT BOOK:
1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.

REFERENCE BOOK:
1 Andrew S. Tanenbaum: Distributed Operating Systems, Pearson Education, 2013.

Course Title: Advances in Computer Networks **Course Code: 17SSE112**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours: 50**

COURSE OBJECTIVES:

- Become familiar with the basics of Computer Networks
- Learn Network architectures
- Become thorough with design concepts of fundamental protocols

TOPICS

UNIT-I

Foundation: Building a Network, Requirements, Perspectives, Scalable Connectivity, Cost-Effective Resource sharing,
Support for Common Services, Manageability, Protocol layering, Performance, Bandwidth and Latency, Delay X Bandwidth Product, Perspectives on Connecting, Classes of Links, Reliable Transmission, Stop-and-Wait , Sliding Window, Concurrent Logical Channels.

T1: Chapter 1.1, 1.2, 1.5.1, 1.5.2., 2.1, 2.5 T2: Chapter 4

10 hours

UNIT-II

Internetworking- I: Switching and Bridging, Datagrams, Virtual Circuit Switching, Source Routing, Bridges and LAN
Switches, Basic Internetworking (IP), What is an Internetwork ?, Service Model, Global Addresses, Datagram Forwarding in IP, subnetting and classless addressing, Address Translation(ARP), Host Configuration(DHCP), Error Reporting(ICMP), Virtual Networks and Tunnels.

T1: Chapter 3.1, 3.2,

10 hours

UNIT-III

Internetworking- II: Network as a Graph, Distance Vector(RIP), Link State(OSPF), Metrics, The Global Internet, Routing Areas, Routing among Autonomous systems(BGP), IP Version 6(IPv6), Mobility and Mobile IP.

T1: Chapter 3.3, 4.1.1,4.1.3 T2: Chapter 13.1 to 13.18 , Chapter 18.

10 hours

UNIT-IV

End-to-End Protocols: Simple Demultiplexer (UDP), Reliable Byte Stream(TCP), End-to-End Issues, Segment Format, Connecting Establishment and Termination, Sliding Window Revisited, Triggering Transmission, Adaptive Retransmission, Record Boundaries, TCP Extensions, Queuing Disciplines, FIFO, Fair Queuing, TCP Congestion Control, Additive Increase/ Multiplicative Decrease, Slow Start, Fast Retransmit and Fast Recovery.

T1: Chapter 5.1, 5.2.1 to 5.2.8, 6.2, 6.3

10 hours

UNIT-V

Congestion Control and Resource Allocation: Congestion-Avoidance Mechanisms, DEC bit, Random Early Detection (RED), Source-Based Congestion Avoidance. The Domain Name System(DNS), Electronic Mail(SMTP,POP,IMAP,MIME),World Wide Web(HTTP), Network Management(SNMP)

T1: Chapter 6.4 T2: Chapter 23.1 to 23.16, Chapter 24, 25, Chapter 27.1 to 27.8

10 hours

COURSE OUTCOMES:

Upon completion of the course, students shall be able to

- List and classify network services, protocols and architectures, explain why they are layered.
- Key Internet applications and their protocols, and will apply to develop their own applications (e.g. Client Server applications, Web Services) using the sockets API.
- Explain various congestion control techniques.

TEXT BOOKS:

1.Larry Peterson and Bruce S Davis, "Computer Networks :A System Approach" 5th Edition , Elsevier 2014

2.Douglas E Comer, "Internetworking with TCP/IP, Principles, Protocols and Architecture" 6th Edition, PHI 2014

REFERENCE BOOK:

1.Uyless Black, "Computer Networks, Protocols , Standards and Interfaces", 2nd Edition - PHI, 2009

2.Behrouz A Forouzan "TCP/IP Protocol Suite" 4th Edition – Tata McGraw-Hill.

Subject : DECISION SUPPORT SYSTEM AND ERP **Subject Code: 17SSE113**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

UNIT I **10 Hours**

Decision Making and Computerized Support

Managers and Decision Making, Managerial Decision Making and Information Systems, Managers and Computerized Support, The need for Computerized Support technologies, A frame work for decision support, The concept of Decision Support systems, Group Decision Support Systems. Executive systems and Information (support) Systems, Expert Systems, Artificial Neural Networks, Hybrid Support Systems, The Evolution and Attributes of Computerized Decision aids, Introduction and Definitions, Systems, Models, The Modeling Process, Decision Making: The Intelligent Phase, Decision Making: The Design Phase, Decision Making: The Choice Phase, Evaluation, Decision Making: Implementation phase.

UNIT II **10 Hours**

Decision Support Systems-I

DSS Configuration, What is DSS?, Characteristics, Capabilities, Components of DSS, The Data Management Sub System, The Model Management Subsystem, The Knowledge Based Management System, The User Interface, The User, DSS Hardware, Distinguishing DSS from Management Science and MIS, DSS Classification.

Decision Support Systems – II

Introduction to DSS development, The Traditional System Development Life cycle, Alternate Development Methodologies, Prototyping: The DSS Development Methodology, DSS Technology Levels and Tools, DSS Development Platforms, DSS Development Tool Selection, Team-Developed DSS, End User-Developed DSS, Developing DSS: Putting the System Together, DSS Research Directions and the DSS of the future.

UNIT III **12 Hours**

Expert Systems Knowledge Based DSS, Concepts and Definitions of Artificial

Intelligence, AI versus Natural Intelligence, The Artificial Intelligence field, Types of Knowledge-Based Decision Support systems, Basic Concepts, The Human Element in Expert System, How Expert System work? Problem areas addressed by ES, Benefits, Problems and Limitations of ES, ES Success Factors, Types of Expert Systems, ES and the Internet/ Intranet/ Web, Knowledge Engineering, Scope of Knowledge, Difficulties in Knowledge Acquisition, Methods of Knowledge Acquisition, Machine Learning, Intelligent Agents, Selecting an appropriate Knowledge Acquisition Method, Knowledge Acquisition from Multiple Experts, Validation and Verification of Knowledge Base, Analyzing and Coding, Documenting and Diagramming, Numeric and Documented Knowledge Acquisition, Knowledge Acquisition and Internet/ Intranet, Induction and Table Example.

UNIT IV **10 Hours**

Enterprise Resource Planning:

An Overview, Accommodating Variety, Integrated Management Information, Seamless Integration, Supply Chain Management, Resource Management, Integrated Data Model, Scope, Technology, Benefits of ERP, Evolution, ERP revisited, ERP and its Modern Enterprise.

Business Engineering and ERP:

An overview, What is Business Engineering? Significance, Principles, BRP, ERP and IT, Business Engineering with Information Technology, ERP and Management Concerns.

UNIT V **10 Hours**

Business Modeling for ERP:

An Overview, Building Business Model.

ERP- Implementation:

An overview, Role of Consultants, Vendors and Users, Customization, Precautions, ERP-Post Implementation Options, ERP- Implementation Methodology, Guidelines for Implementation.

ERP and the Competitive advantage:

An Overview, ERP and the Competitive strategy.

Text Book:

1. Efraim Turban & Jay E. Aronson : 'Decision Support Systems and Intelligent Systems', Sixth Edition, Pearson Education Asia, 2001

1.'Enterprise Resource Planning- Concepts and Practice', Vinod Kumar Garg, N.K. Venkatakrishnan, PHI, 1999.

2.'Enterprise Resource Planning', S Sadagopan, PHI,1999.

Reference Books:

1. Giarratano & Riley: 'Expert Systems: Principles and Programming', Thomson Brooks / Cole, 2002.

2. Sprague R.H. Jr and H.J. Watson: 'Decision Support Systems' , Fourth Edition, Prentice Hall, 1996

Course Title: Web Services **Course Code: 17SSE121**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES

- To provide an in-depth knowledge of Web Services.
- To understand the fundamental concepts of Web services.
- To understand the fundamental concepts of WSDL Web Services.
- To design Web service Architecture.
- To Study Building Blocks of Web services.

TOPICS

UNIT-I

Middleware: Understanding the middle ware, RPC and Related Middle ware , TP Monitors , Object Brokers, Message-Oriented Middleware.

10 hours

UNIT-II

Web Services: Web Services Technologies, Web Services Architecture.

10 hours

UNIT-III

Basic Web Services Technology: WSDL Web Services Description Language, UDDI Universal Description Discovery and Integration, Web Services at work interactions between the Specifications, Related Standards. **10 hours**

UNIT-IV

Service Coordination Protocols: Infrastructure for Coordination Protocols, WS-Coordination , WS-Transaction, RosettaNet, Other Standards Related to Coordination Protocols. **10 hours**

UNIT-V

Service Composition: Basic of Service Composition, A New Chance of Success for Composition, Services Composition Models, Dependencies between Coordination and Composition, BPEL: Business Process Execution Language for Web Services, OutLook, Applicability of the Web Services, Web services as a Problem and a Solution : An Example.

10 hours

COURSE OUTCOMES

Upon completion of the course, students shall be able to

- Bind and unbind services in UDDI.
- Develop WSDL document
- Implement web service client to call public service.
- Implement a service and exposing it as public service.

TEXT BOOK

1.Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju: Web Services(Concepts , Architectures and Applications), Springer International Edition 2009.

Course Title: Soft Computing **Course Code: 17SSE122**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES:

- To learn the key aspects of Soft computing
- To know about the components and building block hypothesis of Genetic algorithm.
- To gain insight onto Neuro Fuzzy modeling and control.
- To gain knowledge in machine learning through Support vector machines.

TOPICS

UNIT-I

Introduction to Soft computing: Neural networks, Fuzzy logic, Genetic algorithms, Hybrid systems and its applications. Fundamental concept of ANN, Evolution, basic Model of ANN, Terminologies used in ANN, MP model, Hebb model.

10 hours

UNIT-II

Perceptron Network: Adaptive linear neuron, Multiple adaptive linear neurons, Back propagation Network (Theory, Architecture, Algorithm for training, learning factors, testing and applications of all the above NN models). **10 hours**

UNIT-III

Introduction to classical sets and fuzzy sets: Classical relations and fuzzy relations, Membership functions.

10 hours

UNIT-IV

Defuzzification: Fuzzy decision making, and applications**.**

10 hours

UNIT-V

Genetic algorithms: Introduction, Basic operations, Traditional algorithms, Simple GA General genetic algorithms, The schema theorem, Genetic programming, applications.

10 hours

COURSE OUTCOMES:

Upon completion of the course, the students will be able to

- Implement machine learning through neural networks.
- Design Genetic Algorithm to solve the optimization problem
- Develop a Fuzzy expert system.
- Model Neuro Fuzzy system for clustering and classification.

TEXT BOOKS:

1. Shivanandam, Deepa S. N., Principles of Soft computing, Wiley India, ISBN 13: 9788126527410, 2011
(Chapters 1, 2, 3(Upto 3.5), 7, 8, 9, 10, 13, 15 (upto 15.6 & 15.9,15,10)

REFERENCE BOOK:

1. J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-fuzzy and soft computing, EEE edition, 2012.

Course Title: Advances in Storage Area Networks **Course Code: 17SSE123**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES

- To understand the fundamentals of storage centric and server centric systems
- To understand the metrics used for Designing storage area networks
- To understand the RAID concepts
- To enable the students to understand how data centre's maintain the data with the concepts of backup mainly remote mirroring concepts for both simple and complex systems

TOPICS

UNIT-I

Introduction: Server Centric IT Architecture and its Limitations; Storage – Centric IT Architecture and its advantages. Case study: Replacing a server with Storage Networks The Data Storage and Data Access problem; The Battle for size and access. Intelligent Disk Subsystems: Architecture of Intelligent Disk Subsystems; Hard disks and Internal I/O Channels; JBOD, Storage virtualization using RAID and different RAID levels; Caching: Acceleration of Hard Disk Access; Intelligent disk subsystems, Availability of disk subsystems.

10 hours

UNIT-II

I/O Techniques: The Physical I/O path from the CPU to the Storage System; SCSI; Fibre Channel Protocol Stack; Fibre Channel SAN; IP Storage. Network Attached Storage: The NAS Architecture, The NAS hardware Architecture, The NAS Software Architecture, Network connectivity, NAS as a storage system. File System and NAS: Local File Systems; Network file Systems and file servers; Shared Disk file systems; Comparison of Fibre Channel and NAS**.**

10 hours

UNIT-III

Storage Virtualization: Definition of Storage virtualization; Implementation Considerations; Storage virtualization on Block or file level; Storage virtualization on various levels of the storage Network; Symmetric and Asymmetric storage virtualization in the Network.

10 hours

UNIT-IV

SAN Architecture and Hardware devices: Overview, Creating a Network for storage; SAN Hardware devices; The Fibre channel switch; Host Bus Adaptors; Putting the storage in SAN; Fabric operation from a Hardware perspective. Software Components of SAN: The switch's Operating system; Device Drivers; Supporting the switch's components; Configuration options for SANs.

10 hours

UNIT-V

Management of Storage Network: System Management, Requirement of management System, Support by Management System, Management Interface, Standardized Mechanisms, Property Mechanisms, In-band Management, Use of SNMP, CIM and WBEM, Storage Management Initiative Specification (SMI-S), CMIP and DMI, Optional Aspects of the Management of Storage Networks, Summary.

10 hours

COURSE OUTCOMES:

Upon completion of the course, the students will be able to

- Identify the need for performance evaluation and the metrics used for it
- Apply the techniques used for data maintenance.
- Realize storage virtualization concept,
- Develop techniques for evaluating policies for LUN masking, file systems

TEXT BOOK:

1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, Wiley India, 2013.

REFERENCE BOOKS:

1. Robert Spalding: “Storage Networks The Complete Reference”, Tata McGraw-Hill, 2011.
2. Marc Farley: Storage Networking Fundamentals – An Introduction to Storage Devices, Subsystems, Applications, Management, and File Systems, Cisco Press, 2005.
3. Richard Barker and Paul Massiglia: “Storage Area Network Essentials A Complete Guide to understanding and Implementing SANs”, Wiley India, 2006.

Course Title: Agile Technologies **Course Code: 17SSE124**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES

- To understand how an iterative, incremental development process leads to faster delivery of more useful software
- To understand the essence of agile development methods
- To understand the principles and practices of extreme programming
- To understand the roles of prototyping in the software process •To understand the concept of Mastering Agility

TOPICS

UNIT-I

Why Agile?: Understanding Success, Beyond Deadlines, The Importance of Organizational Success, Enter Agility, How to Be Agile?:Agile Methods, Don't Make Your Own Method, The Road to Mastery, Find a Mentor.

10 hours

UNIT-II

Understanding XP: The XP Lifecycle, The XP Team, XP Concepts, Adopting XP: Is XP Right for Us?, Go!, Assess Your Agility.

10 hours

UNIT-III

Practicing XP: Thinking: Pair Programming, Energized Work, Informative Workspace, Root-Cause Analysis, Retrospectives, Collaborating: Trust, Sit Together, Real Customer Involvement, Ubiquitous Language, StandUp Meetings, Coding Standards, Iteration Demo, Reporting,Releasing:"Done Done", No Bugs, Version Control, Ten-Minute Build, Continuous Integration, Collective Code Ownership, Documentation, Planning:
Vision, Release Planning, The Planning Game, Risk Management, Iteration Planning, Slack, Stories, Estimating, Developing: Incremental Requirements, Customer Tests, Test-Driven Development, Refactoring, Simple Design, Incremental Design and Architecture, Spike Solutions, Performance Optimization, Exploratory Testing.

10 hours

UNIT-IV

Mastering Agility: Values and Principles: Commonalities, About Values, Principles, and Practices, Further Reading, Improve the Process: Understand Your Project, Tune and Adapt, Break the Rules, Rely on People: Build Effective Relationships, Let the Right People Do the Right Things, Build the Process for the People, Eliminate Waste: Work in Small, Reversible Steps, Fail Fast, Maximize Work Not Done, Pursue Throughput.

10 hours

UNIT-V

Deliver Value: Exploit Your Agility, Only Releasable Code Has Value, Deliver Business Results, Deliver Frequently, Seek Technical Excellence: Software Doesn't Exist, Design Is for Understanding, Design Tradeoffs, Quality with a Name, Great Design, Universal Design Principles, Principles in Practice, Pursue Mastery.

10 hours

COURSE OUTCOMES

Upon completion of the course, students shall be able to

- Understand the XP Lifecycle, XP Concepts, Adopting XP
- Work on Pair Programming, Root-Cause Analysis, Retrospectives, Planning, Incremental Requirements, Customer Tests
- Implement Concepts to Eliminate Waste

TEXT BOOK:

1. The Art of Agile Development (Pragmatic guide to agile software development), James shore, Chromatic, O'Reilly Media, Shroff Publishers & Distributors, 2007.

REFERENCE BOOKS:

1. Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices, Prentice Hall; 1st edition, 2002

2. Craig Larman, "Agile and Iterative Development, A Manger's Guide", Pearson Education, First Edition, India, 2004.

Course Title: Software Project Planning and Management **Course Code: 17SSE201**
Credits(L:T:P:S): 4:0:2:0 **Core/Elective: Core**
Type of Course: Lecture & Practical **Total Contact Hours:52**

COURSE OBJECTIVES

- To define and highlight importance of software project management.
- To formulate strategy in managing projects
- To estimate the cost associated with a project
- To plan, schedule and monitor projects for the risk management
- To define the software management metrics

TOPICS

UNIT-I

Metrics: Introduction, The Metrics Roadmap, A Typical Metrics Strategy, What Should you Measure?, Set Targets and track Them, Understanding and Trying to minimize variability, Act on data, People and Organizational issues in Metrics Programs, Common Pitfalls to watch out for in Metrics Programs, Matrices implementation checklists and tools, Software configuration management: Introduction, Some Basic Definitions and terminology, the processes and activities of software configuration management, configuration status accounting, configuration audit, software configuration management in geographically distributed teams, Metrics in software configuration management, software configuration management tools and automation**.**

10 hours

UNIT-II

Risk Management: Introduction, What is risk management and why is it important?, Risk management cycle, Risk identification: common tools and techniques, Risk Quantifications, Risk Monitoring, Risk Mitigation, Risks and Mitigation in the context of global project teams, some practical techniques risk management, Metrics in risk management. Project Planning and Tracking: Components of Project Planning and Tracking, The "What " Part of a Project Plan, The "What Cost " Part of a Project Plan, The "When " Part of Project Planning, The "How " Part of a Project Planning: Tailoring of Organizational Processes For the Project, The " By Whom " Part of the Project Management Plan : Assigning Resources, Putting it all together : The Software Management Plan, Activities Specific to Project Tracking, Interfaces to the Process Database.Project Closure: When Does Project Closure Happen?. Why Should We Explicitly do a Closure?, An Effective Closure Process, Issues that Get Discussed During Closure, Metrics for Project Closure, Interfaces to the Process Database.

12 hours

UNIT-III

Software Requirements gathering: Inputs and start criteria for requirements gathering, Dimensions of requirements gathering, Steps to be followed during requirements gathering, outputs and quality records from the requirements phase, skill sets required during requirements phase, differences for a shrink-wrapped software, challenges during the requirements management phase, Metrics for requirements phase. Estimation: What is Estimation? when and why is Estimation done?, the three phases of Estimation, Estimation methodology, formal models for size Estimation, Translating size Estimate into effort Estimate, Translating effort Estimates into schedule Estimate, common challenges during Estimation , Metrics for the Estimation processes. Design and Development Phases: Some

differences in our chosen approach, salient features of design, evolving an architecture/ blueprint, design for reusability, technology choices/ constraints, design to standards, design for portability, user interface issues, design for testability, design for diagnose ability, design for maintainability, design for install ability, inter-operability design, challenges during design and development phases, skill sets for design and development, metrics for design and development phases.

10 hours

UNIT-IV

Project management in the testing phase: Introduction, What is testing?, what are the activities that makeup testing?, test scheduling and types of tests, people issues in testing, management structures for testing in global teams, metrics for testing phase. Project management in the Maintenance Phase: Introduction, Activities during Maintenance Phase, management issues during Maintenance Phase, Configuration management during Maintenance Phase, skill sets for people in the maintenance phase, estimating size, effort, and people resources for the maintenance phase, advantages of using geographically distributed teams for the maintenance phase, metrics for the maintenance phase.

10 hours

UNIT-V

Globalization issues in project management: Evolution of globalization, challenges in building global teams, Models for the execution of global projects, some effective management techniques for managing global teams. Impact of the internet on project management: Introduction, the effect of internet on project management, managing projects for the internet, Effect on the project management activities. People focused process models: Growing emphasis on people centric models, people capability maturity model(P-CMM), other people focused models in the literature, how does an organization choose the models to use? **10 hours**

LABORATORY WORK for Software Project Planning and Management
USE STAR UML TOOL or VISUAL PARADIGMN or any other equivalent tool to develop ATM and Restaurant systems
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.

1.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 ma-chine. 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 ob-tained 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 custom-er 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 cus-tomer.)

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. 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.

2. Restaurant System

The system is intended to support the day-to-day operations of a restaurant by improving the processes of making reservations and allocating tables to customers. The Restaurant system provides the facilities like

- Record Booking
- Cancel Booking
- Record Arrival
- Table Transfer

The new system can offer diners eat at the restaurant without making an advance booking, if a free table is available. This is known as Walk-in.

The new system should display the same information as the existing booking sheet and in same format, to make it easy for restaurant staff to transfer, to the new system. When new bookings are recorded or changes made to existing bookings, the display should be immediately updated, so that restaurant staff is working with the latest information available.

COURSE OUTCOMES:

Upon completion of the course, the students will be able to

- Evaluate a project to develop the scope of work, provide accurate cost estimates and to plan the various activities
- Apply risk management analysis techniques that identify the factors that put a project at risk and to quantify the likely effect of risk on project timescales
- Identify the resources required for a project and to produce a work plan and resource schedule
- Monitor the progress of a project and to assess the risk of slippage, revising targets counteract drift
- Use appropriate metrics to management the software development outcome

• Develop research methods and techniques appropriate to defining, planning and carrying out a research project within your chosen specialist area within the management of software projects.

TEXT BOOK

1. Ramesh Gopalaswamy: "Managing Global Projects ", Tata McGraw Hill, 2013.

REFERENCES:

1.Watts Humphrey, "Managing the Software Process ", Pearson Education, New Delhi, 2000
2. PankajJalote, "Software Project Management in practice", Pearson Education, New Delhi, 2002.

Course Title: Enterprise Application Programming **Course Code: 17SSE202**
Credits(L:T:P:S): 4:0: 2:0 **Core/Elective: Core**
Type of Course: Lecture & Practical **Total Contact Hours:52**

COURSE OBJECTIVES:

- To gain knowledge about metrics Web Application Development and related terminologies
- To gain knowledge about persistent framework and tools.
- To learn to build solutions using Design Patterns
- To get introduced to latest WEB frameworks

TOPICS

UNIT-I

Web application and java EE 6: Exploring the HTTP Protocol, Introducing web applications, describing web containers, exploring web architecture models, exploring the MVC architecture. Working with servlets 3.0 Exploring the features of java servlet, Exploring new features in servlet 3.0, Exploring the servlet API, explain the servlet life cycle, creating a sample servlet, creating a servlet by using annotation, working with servlet config and servlet context objects, working with the Http servlet request and Http servlet response interfaces, Exploring request delegation and request scope, implementing servlet collaboration.

10 hours

UNIT-II

Handling sessions in servlet 3.0: Describing a session, introducing session tracking, Exploring the session tracking, mechanisms, using the java servlet API for session tracking, creating login application using session tracking. Implementing event handling Introducing events, Introducing event handling, working with the servlet events, developing the online shop web application. Working with java server pages: Introducing JSP technology, Exploring new features of JSP2.1, listing advantages of JSP over java servlet, Exploring the architecture of a JSP page, Describing the life cycle of a JSP page, working with JSP basic tags and implicit objects, working with the action tags in JSP, exploring the JSP unified EL, using functions with EL.

10 hours

UNIT-III

Implementing JSP tag extensions: Exploring the elements of tag extensions, Working with classic tag handlers , Exploring the tag extensions, Working with simple tag handlers. Implementing java server pages standard tag library 1.2: Introducing JSTL, Exploring the tag libraries JSTL, working with the core tag library. Implementing filters: Exploring the need of filters, exploring the working of filters, exploring filters API, configuring a filter, creating a web application using filters, using initializing parameter in filters.

10 hours

UNIT-IV

Persistence Management and Design Patterns : Implementing java persistence using hibernate Introducing hibernate, exploring the architecture of hibernate, downloading hibernate, exploring HQL, understanding hibernate O/R mapping, working with hibernate, Implementing O/R mapping with hibernate. Java EE design patterns: Describing the java EE

application architecture, Introducing a design patterns, discussing the role of design patterns, exploring types of patterns.

10 hours

UNIT-V

Web Frameworks: Working with struts 2. Introducing struts 2, understanding actions in struts 2.Working with java server faces 2.0:Introducing JSF, Explaining the features of JSF, Exploring the JSF architecture, describing JSF elements, Exploring the JSF request processing life cycle. Working with spring 3.0:Introducing features of the spring framework, exploring the spring framework architecture, exploring dependency injection & inversion of control, exploring AOP with spring, managing transactions. Securing java EE 6 applications: Introducing security in java EE 6, exploring security mechanisms, implementing security on an application server.

12 hours

LABORATORY WORK

Design Develop and Implement the following modules using any suitable language/tools.

1. Developing the profile management module

- Design, develop and implement the following theory models using any suitable language tools
- Implementing logic with servlet.
- creating the people_employee servlet.
- creating the employeeobj class.
- creating the employeedbmethods class.
- creating the generateid class, creating views.
- creating the people_insert JSP page.
- creating the people_search JSP page.
- creating the people_edit JSP page.
- creating the people_list JSP page.
- creating the people_ profile JSP page.

2. Developing the recruitment module

- Registering a new applicant.
- creating the people_applicant servlet.
- creating the applicantDBObj class.
- creating the applicantDBmethods class.
- creating the generated class.
- creating an interface for applicant registration.
- conducting rounds of test.
- creating the applicant_test_dtl servlet.
- designing JSP views.
- working of the recruitment module.

.

3. Developing the payroll module

- Updating salary statement,
- creating the people_payroll servlet.
- creating the empsal class.
- creating the employee agreement class.

- creating the payrollbean methods class.
- designing JSP views.
- creating the people_agreement JSP page.
- creating the people_agreement_edit JSP page.
- creating the salary_search.jsp file.
- creating the salary_slip JSP page.

COURSE OUTCOMES:

Upon completion of the course, students shall be able to

- Implement a WEB application.
- Manage deployment configurations are
- Implement Security mechanisms

TEXT BOOK:

1.Kogent learning solution: JAVA SERVER PROGRAMMING JAVA EE6(J2EE 1.6), Dreamtech press 2014

Course Title: Software Metrics and Quality Assurance **Course Code: 17SSE203**
Credits(L:T:P:S): 4:0:0:S **Core/Elective: Core**
Type of Course: Lecture & Self study **Total Contact Hours:52**

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

TOPICS

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.

11 hours

UNIT-II

Applying The Seven Basic Quality Tools In Software Development : Ishikawa's Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts , Scatter Diagram, Control Chart, Cause And Effect Diagram. The Rayleigh Model: Reliability Models, The Rayleigh Model Basic Assumptions, Implementation, Reliability And Predictive Validity.

10 hours

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.

10 hours

UNIT-IV

Availability Metrics: Definition And Measurement Of System Availability, Reliability Availability And Defect Rate, Collecting Customer Outage Data For Quality Improvement, In Process Metrics For Outage And Availability. Conducting Software Project Assessment, Audit Ad Assessment, Software Process Maturity Assessment And Software Project Assessment, Software Process Assessment A Proponed Software Project Assessment Method.

10 hours

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.

11 hours

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

TEXT BOOK

1.Stephen H Khan, "Metrics and Models in Software Quality Engineering", Pearson 2nd edition 2013.

REFERENCES:

1. Norman E-Fentor and Share Lawrence Pflieger. "Software Metrics". International Thomson Computer Press, 1997.
2. S.A.Kelkar, "Software quality and Testing", PHI Learing Pvt, Ltd., New Delhi 2012.
3. Watts S Humphrey, "Managing the Software Process", Pearson Education Inc, 2008.
4. Mary Beth Chrissis, Mike Konrad and Sandy Shrum, "CMMI", Pearson Education(Singapore) , 2003
5. Philip B Crosby, "Quality is Free: The Art of Making Quality Certain ", Mass Market, 1992.

Course Title: Data Mining & Data Warehousing **Course Code: 17SSE211**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES:

- To expose the students to the concepts of Data warehousing Architecture and Implementation
- To Understand Data mining principles and techniques and Introduce DM as a cutting edge business intelligence
- To learn to use association rule mining for handling large data
- To understand the concept of classification for the retrieval purposes
- To know the clustering techniques in details for better organization and retrieval of data

TOPICS

UNIT-I

Introduction: What is a Data Warehouse?, A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Data cube Technology, From Data warehousing to Data Mining, Data Mining Functionalities, Data cleaning, Data Integration and Transformation, Data Reduction.

10 hours

UNIT-II

Data Mining Primitives, Languages And System Architectures: Data Mining primitives, Presentation and Visualization of Discovered patterns, A Data Mining Query Language. MINING ASSOCIATION RULES IN LARGE DATA BASES: Association Rule Mining Single –Dimensional Boolean Association Rules From Transactional Databases, Mining Multilevel Association Rules from Transactional Databases.

10 hours

UNIT-III

Classification and Prediction: Issues regarding Classification and Prediction, classification by Decision tree induction, Bayesian classification, Classification by back propagation, Classification Based on the concepts from association rule
mining. Other classification methods, prediction.

10 hours

UNIT-IV

Cluster Analysis: What is Cluster Analysis? Types of data in cluster Analysis: a Categorization of Major Clustering Methods, Partitioning Methods, And Hierarchical methods, Density-Based Methods, Model-Based Clustering Methods:
Statistical Approach, Neural Network Approach Outliner Analysis.

10 hours

UNIT-V

Applications and Trends In Data Mining: Data mining application, Data mining system Products research Prototypes, Additional Themes on Data Mining, Data Mining and Intelligent Query Answering, Tends in Data Mining.

10 hours

COURSE OUTCOMES:

Upon completion of the course, students shall be able to

- Store voluminous data for online processing
- Preprocess the data for mining applications
- Apply the association rules for mining the data
- Design and deploy appropriate classification techniques
- Cluster the high dimensional data for better organization of the data
- Discover the knowledge imbibed in the high dimensional system

Text Books:

1. Jiawei Michelin Kamber, "Data Mining Concepts and Techniques", Morgan Kauf Mann Publishers, Reprint 2014.

Course Title: Information and Network Security **Course Code: 17SSE212**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES:

- To understand the fundamentals of Cryptography
- To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
- To understand the various key distribution and management schemes.
- To understand how to deploy encryption techniques to secure data in transit across data networks
- To design security applications in the field of Information technology

UNIT-I

Classical Encryption Techniques: Symmetric Cipher Model, Cryptography, Cryptanalysis and Brute-Force Attack, Substitution Techniques, Caesar Cipher, Mono alphabetic Cipher, Playfair Cipher, Hill Cipher, Polyalphabetic Cipher, One Time Pad. Block Ciphers and the data encryption standard: Traditional block Cipher structure, stream Ciphers and block Ciphers, Motivation for the feistel Cipher structure, the feistel Cipher, The data encryption standard, DES encryption , DES decryption , A DES example, results, the avalanche effect, the strength of DES, the use of 56-Bit Keys, the nature of the DES algorithm, timing attacks, Block cipher design principles, number of rounds, design of function F, key schedule algorithm**.** **10 hours**

UNIT-II

Public-Key Cryptography and RSA: Principles of public-key cryptosystems. Public-key cryptosystems.Applications for public-key cryptosystems, requirements for public-key cryptosystems.public-key cryptanalysis. The RSA algorithm, description of the algorithm, computational aspects, the security of RSA: OtherPublic-Key Cryptosystems: Diffiehellman key exchange, The algorithm, key exchange protocols, man in the middle attack, Elgamal Cryptographic systems, Elliptic curve arithmetic, abelian groups, elliptic curves over real numbers, elliptic curves over Zp, elliptic curves over GF(2m), Elliptic curve cryptography, Analog of Diffie-hellman key exchange, Elliptic curve encryption/ decryption, security of Elliptic curve cryptography, Pseudorandom number generation based on an asymmetric cipher, PRNG based on RSA.

10 hours

UNIT-III

Key Management and Distribution: Symmetric key distribution using Symmetric encryption, A key distribution scenario, Hierarchical key control, session key lifetime, a transparent key control scheme, Decentralized key control, controlling key usage, Symmetric key distribution using asymmetric encryption, simple secret key distribution, secret key distribution with confidentiality and authentication, A hybrid scheme, distribution of public keys, public announcement of public keys, publicly available directory, public key authority, public keys certificates, X-509 certificates**.** Certificates, X509 version 3, public key infrastructure .User Authentication: Remote user Authentication principles, Mutual Authentication, one way Authentication, remote user Authentication using Symmetric encryption, Mutual Authentication, one way Authentication, Kerberos, Motivation , Kerberos version 4, Kerberos version 5, Remote user Authentication using Asymmetric encryption,

Mutual Authentication, one way Authentication, federated identity management, identity management, identity federation, personal identity verification.

10 hours

UNIT-IV

Wireless network security: Wireless security, Wireless network threats, Wireless network measures, mobile device security, security threats, mobile device security strategy, IEEE 802.11 Wireless LAN overview, the Wi-Fi alliance, IEEE 802 protocol architecture. Security, IEEE 802.11i services, IEEE 802.11i phases of operation, discovery phase, Authentication phase, key management phase, protected data transfer phase, the IEEE 802.11i pseudorandom function, Web Security Considerations: Web Security Threats, Web Traffic Security Approaches. Secure Sockets Layer: SSL Architecture, SSL Record Protocol, Change Cipher Spec Protocol, Alert Protocol, handshake Protocol, Cryptographic Computations. Transport Layer Security: Version Number, Message Authentication Code, Pseudorandom Functions, Alert Codes, Cipher Suites, Client Certificate Types, Certificate Verify And Finished Messages, Cryptographic Computations, Padding. HTTPS Connection Initiation, Connection Closure. Secure Shell (SSH): Transport Layer Protocol, User Authentication Protocol, and Connection Protocol.

10 hours

UNIT-V

Electronic Mail Security: Pretty good privacy, notation, operational; description, S/MIME, RFC5322, Multipurpose internet mail extensions, S/MIME functionality, S/MIME messages, S/MIME certificate processing, enhanced security services, Domain keys identified mail, internet mail architecture, E-Mail threats, DKIM strategy, DKIM functional flow, .IP Security: IP Security overview, applications of IPsec, benefits of IPsec, Routing applications, IPsec documents, IPsec services, transport and tunnel modes, IP Security policy, Security associations, Security associations database, Security policy database, IP traffic processing, Encapsulating Security payload, ESP format, encryption and authentication algorithms, Padding, Anti replay service, transport and tunnel modes, combining security associations, authentication plus confidentiality, basic combinations of security associations, internet key exchange, key determinations protocol, header
and payload formats, cryptographic suits**.**

10 hours

COURSE OUTCOMES:

Upon Completion of the course, the students will be able to

- Implement basic security algorithms required by any computing system
- Analyze the vulnerabilities in any computing system and hence be able to design a security solution.
- Analyze the possible security attacks in complex real time systems and their effective counter measures
- Identify the security issues in the network and resolve it.

Text Books:

1.William Stallings: Cryptography and Network Security, Pearson 6th edition 2013.

Reference

3.V K Pachghare: Cryptography and Information Security , 2013.

Course Title: Design Patterns **Course Code: 17SSE214**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture & Practical **Total Contact Hours:52**

OBJECTIVES:

- To learn how to add functionality to designs while minimizing complexity.
- To learn What code qualities are required to maintain to keep code flexible?
- To understand the common design patterns.
- To explore the appropriate patterns for design problems.

TOPICS

UNIT-I

Introduction: what is a design pattern? Describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern. What is object-oriented development? Key concepts of object oriented design other related concepts, benefits and drawbacks of the paradigm.

10 hours

UNIT-II

Analysis a System: overview of the analysis phase, stage 1: gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain. Design and Implementation, discussions and further reading.

10 hours

UNIT-III

Design Pattern Catalog: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy**.**

10 hours

UNIT-IV

Interactive systems and the MVC architecture: Introduction , The MVC architectural pattern, analyzing a simple drawing program , designing the system, designing of the subsystems, getting into implementation , implementing undo operation , drawing incomplete items, adding a new feature , pattern based solutions**.**

11 hours

UNIT-V

Designing with Distributed Objects: Client server system, java remote method invocation, implementing an object oriented system on the web (discussions and further reading) a note on input and output, selection statements, loops arrays.

11 hours

Design Patterns Laboratory

OBJECTIVES**:**

- To understand the need of design pattern to solve problems of different context.
- To understand the common design patterns and to implement them.
- To explore the appropriate patterns for design problems.

Laboratory Work:

Note: Use appropriate tools/language to implement the following experiment:

1. Design, develop and implement an intercepting Filter Pattern which intercepts and intermediates the request received. Implement an HTML to send a request to a server, where the request is intercepted by the filter and the following details are extracted and displayed. Date, Content-Encoding, Content-Length, Content-Location ,Content-MD5
2. Design, develop and implement a front controller pattern , which forms the centralized control to handle multiple user request for effectively managing the activities of content retrieval, view management and security service invocation. Implement a java Servlet, which forms the controller and decision maker for the entire application .
3. Implement a data access object pattern to separate the data processing logic from data access logic. Implement the java class that decouples the persistence management from business logic.
4. Implement an object pooling mechanism using N-TON design pattern. Implement a Java class which creates 10 connection objects and forms a pool , and another java class to consume the connection and to persist data into the EIS.
5. Implement the session façade pattern to de-couple the business logic from accessing the third party application object in the distributed environment using EJB.

COURSE OUTCOMES**:**

Upon completion of the course, the students will be able to

- Design and implement codes with higher performance and lower complexity
- Be aware of code qualities needed to keep code flexible
- Experience core design principles and be able to assess the quality of a design with respect to these principles.
- Capable of applying these principles in the design of object oriented systems.
- Demonstrate an understanding of a range of design patterns. Be capable of comprehending a design presented using this vocabulary.
- Be able to select and apply suitable patterns in specific contexts.

TEXT BOOKS

1. Brahma dathan, Sarnathrammath, “Object-oriented analysis, design and implementation”, Universities press,2013
2. Erich gamma, Richard helan, Ralph johman , John Vlissides, “Design patterns”, ,PEARSON Publication,2013.

REFERENCE BOOKS:

1. Frank Bachmann, ReginеMeunier, Hans Rohnert, “Pattern Oriented Software Architecture”, Volume 1, 1996.
2. William J Brown et al., “Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998.

Course Title: Information Retrieval **Course Code: 17SSE221**
Credits(L:T:P:S): 4:0:0:0 **Core/Elective: Elective**
Type of Course: Lecture **Total Contact Hours:50**

COURSE OBJECTIVES:

- To understand the basics of Information Retrieval with pertinence to modeling, query operations and indexing
- To get an understanding of machine learning techniques for text classification and clustering
- To understand the various applications of Information Retrieval giving emphasis to Multimedia IR, Web Search
- To understand the concepts of queries specification judgment and search engines

TOPICS

UNIT-I

Introduction: Motivation, Basic concepts, Past, present, and future, The retrieval process. Modeling: Introduction, A taxonomy of information retrieval models, Retrieval:Adhoc and filtering, A formal characterization of IR models, Classic information retrieval, Alternative set theoretic models, Alternative algebraic models, Alternative probabilistic models, Structured text retrieval models, Models for browsing.

10 hours

UNIT-II

Retrieval Evaluation: Introduction, Retrieval performance evaluation, Reference collections. Query Languages: Introduction, keyword-based querying, Pattern matching, Structural queries, Query protocols. Query Operations: Introduction, User relevance feedback, Automatic local analysis, Automatic global analysis

10 hours

UNIT-III

Text and Multimedia Languages and Properties: Introduction, Metadata, Text, Markup languages, Multimedia. Text Operations: Introduction, Document preprocessing, Document clustering, Text compression, Comparing text compression techniques.

10 hours

UNIT-IV

Indexing and Searching: Introduction, Inverted Files, Other indices for text, Boolean queries, Sequential searching, Pattern matching, Structural queries, Compression. Parallel and Distributed IR: Introduction, Parallel IR, Distributed IR

10 hours

UNIT-V

User Interfaces and Visualization: Introduction, Human-Computer interaction, The information access process, Starting pints, Query specification, Context, Using relevance judgments, Interface support for the search process. Searching the Web: Introduction, Challenges, Characterizing the web, Search engines, Browsing, Meta searchers, Finding the needle in the haystack, Searching using hyperlinks**.**

10 hours

COURSE OUTCOMES:

Upon completion of the course, the students will be able to

- Build an Information Retrieval system using the available tools
- Identify and design the various components of an Information Retrieval system
- Apply machine learning techniques to text classification and clustering which is used for efficient Information Retrieval
- Analyze the Web content structure
- Design an efficient search engine

TEXT BOOKS:

1. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: Modern Information Retrieval, Pearson, 1999.

REFERENCE BOOKS:

1.David A. Grossman, OphirFrieder: Information Retrieval Algorithms and Heuristics, 2nd Edition, Springer, 2004.

Introduction to Internet of Things (IOT)

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - II

Subject Code	17SSE222	**Duration of SEE**	03 Hours
Number of Lecture Hours Per Week	04 Hours	**CIE Marks**	50 Marks
Total Number of Lecture Hours	50 Hours	**SEE Marks**	50 Marks
L–T–P–S	4-0–0–0	**CREDITS**	04

Course Objectives: This course will enable students to

1. Explore the IoT concept and Applications
2. Describe Security and Privacy Framework issues in IoT
3. Explain the IoT Architectures and requirements, smart office use case
4. Discuss the market aspects of IOT
5. Explain the cloud services to IOT

UNITS	Teaching Hours
UNIT – I	
.**Introduction**, **Putting the Internet of Things Forward to the Next Level** The Internet of Things Today, The Internet of Things Tomorrow, Potential Success Factors, **Internet of Things Strategic Research and Innovation Agenda -** Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Smart-X Applications, Internet of Things and Related Future Internet Technologies, Networks and Communication, Processes, Data Management, Security, Privacy & Trust, Device Level Energy Issues, IoT Protocols Convergence	13 Hours
UNIT – II	
Internet of Things Global Standardization - State of Play – Introduction, IoT Vision, IoT Standardization Landscape, **Dynamic Context-Aware Scalable and Trust-based IoT Security, Privacy Framework –** Introduction, Main Concepts and Motivation of the Framework, A Policy-based Framework for Security and Privacy in Internet of Things	13 Hours

UNIT – III

Scalable Integration Framework for Heterogeneous Smart Objects, Applications and Services – Introduction, IPv6 Potential, IoT6, IPv6 for IoT, Adapting IPv6 to IoT Requirements, IoT6 Architecture, DigCo Integration with the Cloud and EPICS, Enabling Heterogeneous Integration, IoT6 Smart Office Use-case, Scalability Perspective, Conclusions. 13 Hours

UNIT-IV

Internet of Things Applications - From Research and Innovation to Market Deployment – Introduction, OpenIoT, Icore, Compose, SmartSantander, Fitman, OSMOSE 5

UNIT V

Insights on Federated Cloud Service Management and the Internet of Things, Introduction, Federated Cloud Services Management , Federated Management Service Life Cycle, Self-management Lifecycle 6

Course Outcomes: After Studying this course, students will be able to

1. Demonstrate understanding of IOT Concept and Applications
2. Analyze Security and Privacy Framework issues in IoT
3. Apply IoT Architecture and requirements in understanding use cases
4. Analyze the market aspects of IOT
5. Apply cloud services to IOT

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Design/Development of Solutions
3. Problem Analysis
4.The Engineer and Society

Text Books:

Internet of Things From Research Innovation to Market Development , Dr, Ovidiu Vermesan

SINTEF Norway, Dr. Peter Friess, EU Belgium, River Publishers, Aalborg

Reference Books:

The Definitive Guide to the Internet of Things for Business, 2nd Edition, By Syed Zaeem Hosain, CTO, Aeris

E-Books / Online Resources:

http://www.internet-of-things-research.eu/pdf/IoT-From%20Research%20and%20Innovation%20to%20Market%20Deployment_IERC_Cluster_eBook_978-87-93102-95-8_P.pdf, http://www.internet-of-things-research.eu/pdf/Digitising_the_Industry_IoT_IERC_2016_Cluster_eBook_978-87-93379-82-4_P_Web.pdf

MOOC:

1. Stanford:https://www.class-central.com/mooc/6748/coursera-introduction-to-architecting-smart-iot-devices
2. https://www.class-central.com/mooc/4338/coursera-introduction-to-the-internet-of-things-and-embedded-systems

GENERAL PURPOSE COMPUTATION ON GPU

Subject Code: 17SSE223 **L-T-P-S: 4+0+0+0**
Teaching Hours: 52 **Credits: 4**

UNIT I **10 Hrs**

Heterogeneous Architecture and Parallel Computing: Introduction to parallel programming, Introduction to heterogeneous architecture- GPU in particular. Introduction to GPU computing, Why GPU, evolution of GPU pipeline and general purpose computation on GPU, GPU architecture case studies: NVIDIA G80, GT200, Fermi, AMD Radeon, AMD Fusion APU etc.

UNIT II **10 Hrs**

Execution Model: Features of CUDA and OpenCL, Comparision of CUDA and OpenCL, Thread organization, Kernel, error handling, and execution in CUDA and OpenCL.

UNIT III **12 Hrs**

Programming Model: CUDA Introduction, basics of CUDA C, Complete CUDA structure, basic details of API and libraries, OpenCL overview, OpenCL basic specification, OpenCL C language, Vectorization.

UNIT IV **10 Hrs**

Memory Model: Introduction to memory model and GPU interaction with CPU, Memory model of CUDA and OpenCL, Memory hierarchy (local/register, shared and global) and optimizations, memory optimized programming, coding tips.

UNIT V **10 Hrs**

Tools and programming: Introduction to installation and compilation process, usage of tools, profilers and debuggers. CUDA by Examples and OpenCL by Examples, Future Directions.

TEXT BOOKS:

1. Programming Masiively Parallel Processors: A Hands-on Approach by David B Kirk and Wen-Mei W. Hwu
2. CUDA by Example: An Introduction to General Purpose GPU Programming by Jason Sanders and Edward Kandrot
3. http://developer.amd.com/zones/OpenCLZone/courses/pages/Introduction-OpenCL-Programming-May2010

NMAM INSTITUTE OF TECHNOLOGY, NITTE

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH.SOFTWARE ENGINEERING
(AUTONOMOUS SCHEME: 2017-2019)

III Semester **(20 Credits)**

Sl. No.	Subject Code	Name of the Subject	Duration	C.I.A.	EXAM	Credits
			Practical/ Field work / Assignment			
1	17SSE301	Industrial Training/ Mini Project /Certification Exam	Full Time 8 weeks	50(Report) 50(Presentation)	--	8
2	17SSE302	Seminar	--	100	--	2
3	17SSE303	Project Part-I (Problem Stmt, Literature, Initial Design)	Remaining weeks of 3rd sem.	100 (Report) 100 (Presentation)	--	10
		Total		400		20

Seminar : Topics should be chosen from IEEE/ACM/Elsevier/Springer/any Refereed - Journals /Transactions. Encourage students to convert these seminar topics into a good survey paper or Technical paper.

NMAM INSTITUTE OF TECHNOLOGY, NITTE

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH.SOFTWARE ENGINEERING
(AUTONOMOUS SCHEME: 2017-2019)

IV Semester **(30 Credits)**

Subject Code	Name of the Subject	Duration	C.I.E.	SEE	Credits
		Practical/ Field work			
17SSE401	Project part II (Project Implementation, Testing and Report submission)	Full time	200 [PPE-I – 100 PPE-II – 100]	200	30
TOTAL			**200**	**200**	**30**
GRAND TOTAL from 1st to 4th semester : 100 credits					

Note:
***Lab Classes for these Core Subjects are Compulsory (Practical will be Evaluated for 20 marks and Internal assessment for 30 marks). Lab journals should be maintained.**
 # Seminar : Topics should be chosen from IEEE/ACM/Elsevier/Springer/any Refereed - Journals /Transactions. Encourage students to convert these seminar topics into a good survey paper or Technical paper.

1) Project Phase – I : 6 weeks duration shall be carried out between II and III Semester. Candidates in consultation with guide shall carryout literature survey / visit to Industries to finalize the topic of dissertation.

2) Internship: 24 weeks Duration in 3rd Semester, Evaluation of Marks - Presentation : 25 marks, Report writing and Submission :75 marks and at the end of Internship Viva-Voce Exams shall be conducted for 50 marks.

3) Project Work : 20 weeks duration in IV Semester carries total marks of 250.

4) Project Phase II: 4 days for project work in a week during IV Semester. Evaluation shall be taken during the 8th week of the IV Semester. Total Marks shall be 25.

5) Project Phase – III : Evaluation shall be taken up once in midterm, and at the end of the IV Semester. Midterm evaluation is for 50+50 marks. After the Project report is submitted, Project Work Evaluation and Viva-Voce Examination shall be conducted. Total Marks Shall be 200+200=400 for final viva-voce.

Marks of Evaluation of Project:

I) The I.A. Marks of Project Phase – II & III shall be sent to the University along with Project Work report at the end of the Semester.
II) The Project Valuation and Viva-Voce will be conducted by a committee consisting of the following:

a) Head of the Department (Chairman)
b) Guide
c) Two Examiners appointed by the university. (out of two external examiners at least one should be present).