SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY::PUTTUR
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Specialization: Software Engineering

I M. Tech. – I Semester (SE)

S.No.	Course Code	Subject	I	T	P	C
1	20CS5101	Object Oriented Software Engineering	3	-	-	3
2	20CS5102	Software Requirements & Estimation	3	-	-	3
		Program Elective – I				
3	20CS5112	Software Engineering and Testing	3	-	-	3
	20CS5113	Professional Aspects in Software Engineering				
	20CS5114	Protocol Software Engineering				
		Program Elective – II				
4	20CS5115	Formal Methods of Software Engineering	3	-	-	3
	20CS5116	Software Metrics & Reuse				
	20CS5117	Service Oriented Architecture				
5	20CS5103	Object Oriented Software Engineering Lab	-	-	4	2
6	20CS5104	Software Requirements & Estimation Lab	-	-	4	2
7	20HS0823	Research Methodology and IPR	2	-	-	2
		Audit Course – I				
8	20HS0818	English for Research PaperWriting	2	-	-	-
		Contact Periods / Week	16	-	08	**18**
			Total/Week 24			

I M. Tech. – II Semester (SE)

S.No.	Course Code	Subject	L	T	P	C
1.	20CS5105	Advances in Software Testing	3	-	-	3
2.	20CS5106	Software Patterns	3	-	-	3
		Program Elective – III				
3.	20CS5118	Software Quality Assurance	3	-	-	3
	20CS5119	Software Reliability				
	20CS5120	Model Driven Software Engineering				
		Program Elective – IV				
4.	20CS5121	Software Reengineering	3	-	-	3
	20CS5122	Big Data Analytics				
	20CS5123	Software Configuration Management				
5.	20CS5107	Advances in Software Testing Lab	-	-	4	2
6.	20CS5108	Software Patterns Lab	-	-	4	2
7.	20CS5109	Mini Project	-	-	4	2
		Audit Course – II				
	20HS0829	Constitution of India	2	-	-	-
		Contact Periods / Week	14	-	12	**18**
			Total/Week 26			

II M. Tech. – I Semester (SE)

S. No.	Course Code	Subject	L	T	P	C
		Program Elective-V				
1.	20CS5124	Secure Software Engineering	3	0	0	3
	20CS5125	Agile Methodologies				
	20CS5126	Fundamentals of Data Science				
		Open Elective				
2.	20HS0824	Business Analytics	3	0	0	3
	20ME3026	Industrial Safety				
	20ME3027	Advances in Operations Research				
	20CE1028	Cost Management of Engineering Projects				
	20ME3028	Composite Materials				
	20EE2128	Waste to Energy				
3.	20CS5110	Phase-I Dissertation-I /Industrial Project	0	0	20	10
Contact Periods / Week			06	00	20	**16**
			Total/Week 26			

II M. Tech. – II Semester (SE)

S.No.	Course Code	Subject	L	T	P	C
1.	20CS5111	Phase –II Dissertation II	0	0	32	16
Contact Periods / Week			0	0	32	
			Total/Week 32			

Total Number of Credits= 18 +18+16+16 = 68

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5101) OBJECT ORIENTED SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *To investigate principles of object-oriented software engineering, from analysis through testing*
2. *To learn techniques at each stage of development, including use cases and UML*
3. *To practice these principles and techniques by developing a "real world" software system prototype*
4. *To learn software development life cycle for Object-Oriented solutions for Real-World Problems.*

COURSE OUTCOMES

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

1. *Understand the fundamental principles underlying Object-Oriented software design.*
2. *Employ formal methods to produce effective software designs as solutions to specific tasks.*
3. *Develop the design structure using Object-Oriented Principles to achieve overall programming goals.*
4. *Take responsibility for your own learning through reading and the preparation of assignments, and reflect upon your learning experience.*
5. *Identify and represent domain constraints on the objects and (or) on their relationships.*
6. *Use Case Studies and tools for OOA design.*

UNIT -- I

Introduction to Software Engineering: Introduction-Software Engineering Failures, What is Software Engineering? Software Engineering Concepts, Software Engineering Development Activities, Managing Software Development. Modeling with UML: Introduction, Overview of UML, Modeling Concepts, ARENA Case Study

UNIT -- II

Analysis: Introduction –An Optical Illusion, Overview of Analysis, Analysis Concepts, Analysis Activities, Managing Analysis, ARENA Case Study

UNIT -- III

System Design: Decomposing the System-Introduction: A Floor Plan Example, Overview of System Design, System Design Concepts, System Design Activities: From Objects to Subsystems

System Design: Addressing Design Goals: Introduction- A Redundancy Example, Overview of System Design Activities, Concepts: UML Deployment Diagrams, Addressing Design Goals, Managing System Design, ARENA Case Study

UNIT -- IV

Object Design: Specifying Interfaces: Introduction - A Railroad Example, Overview of Interface Specification, Interface Specification Concepts, Interface Specification Activities, Managing Object Design, ARENA Case Study

Mapping Models to Code: Introduction – A Book Example, Overview of Mapping, Mapping Concepts, Mapping Activities, Managing Implementation, ARENA Case Study

UNIT -- V

Testing: Testing overview, concepts, activities and managing testing

Project Management: Project management overview, concepts, activities and managing project management models and activities

TEXT BOOKS

1. Bernd Bruegge & Allen H. Dutoit, "*Object-Oriented Software Engineering",* 2009.
2. Ivar Jacobson, "*Object-Oriented Software Engineering*", Pearson Education, 2009.

REFERENCES

1. Stephen R. Schach, "*Object-Oriented Classical Software Engineering*", Mc Graw Hill, 2010.
2. Yogesh Singh, "*Object-Oriented Software Engineering*", 2012

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5102) SOFTWARE REQUIREMENTS & ESTIMATION

COURSE OBJECTIVES

1. *To demonstrate knowledge of the distinction between critical and non- critical systems.*
2. *To demonstrate the ability to manage a project including planning, scheduling and risk Assessment / management.*
3. *To author a software requirements document.*
4. *To demonstrate an understanding of the proper contents of a software requirements document.*
5. *To author a formal specification for a software system.*

COURSE OUTCOMES

1. *Student can able to demonstrate an understanding of distributed system architectures and application architectures.*
2. *Student can able to demonstrate an understanding of the differences between real-time and non-real time systems.*
3. *Student can able to demonstrate proficiency in rapid software development techniques.*
4. *Student can able to demonstrate proficiency in software development cost estimation*

UNIT -- I

Software Requirements: What and Why Essential Software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management.

UNIT -- II

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 -- III

Software Requirements Modeling: Use Case Modeling, Analysis Models, Dataflow diagram, state transition diagram, and class diagrams, Object analysis, Problem Frames.

Software Requirements Management: Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain.

UNIT -- IV

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.

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

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, and SLIM (Software Life Cycle Management) Tools.

TEXT BOOKS

1. Karl E. Weigers ,*Software Requirements* ,Microsoft Press.
2. *Rajesh Naik and Swapna Kishore*, *Software Requirements and Estimation*, Tata Mc Graw Hill.

REFERENCES

1. Dean Leffingwell & Don Widrig, *Managing Software Requirements*, Pearson Education,2003.
2. Suzanne Robertson & James Robertson, *Mastering the requirements process*, second edition, Pearson Education, 2006.
3. Capers Jones, *Estimating Software Costs*, Second edition, Tata McGraw-Hill, 2007.
4. M.A. Parthasarathy, *Practical Software Estimation*, Pearson Education, 2007.
5. William A. Florac & Anita D. Carleton, *Measuring the software process*, Pearson Education,1999.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5112) SOFTWARE ENGINEERING AND TESTING

COURSE OBJECTIVES

1. To understand the software life cycle model

2. To understand the software requirements and SRS document

3. Basic software debugging methods

4. Knowing various testing methodologies

5. Testing of domains and paths

COURSE OUTCOMES

At the end of the course, students can able to

1. Define and develop a software project from requirement gathering to implementation

2. Ability to code and test the software

3. Understand the basic testing procedures

4. Generating test cases and test suites

5. Facilitate the dichotomies and taxonomy of bugs

UNIT - I

Software Process Models : The Evolving role of Software – Software – The changing Nature of Software – Legacy software – A generic view of process– A layered Technology – A Process Framework – The Capability Maturity Model Integration (CMMI) – Process Assessment – Personal and Team Process Models – Product and Process – Process Models – The Waterfall Model – Incremental Process Models – Incremental Model – The RAD Model – Evolutionary Process Models – Prototyping – The Spiral Model – The Concurrent Development Model – Specialized Process Models – the Unified Process.

UNIT - II

Requirement Engineering : Software Engineering Practice – communication Practice – Planning practice - Modeling practice– Construction Practice –Deployment - Requirements Engineering - Requirements Engineering tasks – Initiating the requirements Engineering Process- Eliciting Requirements – Developing Use cases – Building the Analysis Models – Elements of the Analysis Model – Analysis pattern – Negotiating Requirements – Validating Requirements.

UNIT – III

Analysis Modelling: Requirements Analysis – Analysis Modeling approaches – Data Modeling Concepts – Object Oriented Analysis – Scenario based Modeling – Flow Oriented Modeling – Class based Modeling

Design & Implementation : Architectural Design – Detailed Design - Design process -Design QualityDesign model - User interface Design – Implementation

UNIT -- IV

Introduction: Purpose of testing, Dichotomies, model for testing, Taxonomy of Bugs - consequences of bugs, taxonomy for 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 -- V

Transaction Flow Testing: Transaction flows, transaction flow testing techniques.

Dataflow Testing: Basics of data flow testing, strategies in dataflow testing, application of data flow testing.

Domain Testing: Domains and Paths, Nice & Ugly Domains, Domain testing, Domains and Interface testing, Domains and Testability.

TEXT BOOKS

1. Roger S. Pressman, *Software Engineering A practitioner's Approach*, Seventh Edition, 2009 McGraw Hill International Edition.
2. Boris Beizer, *Software Testing techniques* - Dreamtech Publishers, second edition.

REFERENCES

1. Ian Sommerville, *Software Engineering*, 8th Edition, Pearson Education, 2008.
2. Richard Fairley, *Software EngineeringConcepts* , McGraw Hill, 2004.
3. Stephan Schach, *Software Engineering*, Tata McGraw Hill, 2007.
4. Pfleeger and Lawrence , *Software Engineering : Theory and Practice*, Pearson Education, 2nd, 2001
5. Brian Marick, *The craft of software testing* –Pearson Education.
6. *Software Testing Techniques*–SPD(Oreille)
7. Edward Kit, *Software Testing in the Real World* –Pearson.
8. Perry, *Effective methods of Software Testing*, John Wiley.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5113) PROFESSIONAL ASPECTS IN SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *Develops a broad understanding of the discipline of the software engineering.*
2. *Seeks to complement this with the detailed knowledge of the techniques for the analysis and design of complex software systems.*
3. *To introduce ethical and professional issues and how are those relevant to software Engineers*
4. *Provides a brief account of associated professional and legal issues.*

COURSE OUTCOMES

1. *Understood the issues affecting the organization, planning and control of software based systems development.*
2. *Should able to complete the analysis and design of the intensive software systems.*
3. *Understood the Professional and technical literature on Software Engineering*

UNIT -- I

Intellectual Property rights Confidential Information, Copyright, Infringement of Copyright, Acts permitted in Relation to Copyright Works, Licensing and Assignment of Copyright, Moral Rights, Designs, Trademarks, The tort of passing off, Domain Names, Patents.

UNIT -- II

Software Licenses, Copyright, Contract, Patent, Free Software and Open Source Software, MIT License, BSD, License, GNU General Public License, GNU Lesser General Public License, Q Public License, Proprietary License, Sun Community License.

UNIT -- III

Software Contracts: Basics of Software Contracts, Extent of liability, Contract for the supply of custom-built software at a fixed price, other types of software service Contract, Liability for defective software.

UNIT –- IV

Software Crime Prevention Computing and criminal Activity, Reforms of Criminal Law, Categories of Misuse, Computer Fraud, Obtaining Unauthorized Access to Computer, Unauthorized Alteration or Destruction of Information, Denying Access to an Authorized user, Unauthorized Removal of Information Stored in a Computer

UNIT -- V

Data Protection Regulations, Data Protection and Privacy, The impact of the Internet, Factors Influencing the Regulation of Data Processing, Convergence of Data Protection Practice, Defamation and the protection of Reputation.

TEXT BOOKS

1. Andrew M. St. Laurent, “*Open Source and Free Software Licensing*”, O’Reilly, Publications.
2. Frank Bott, et. al, “*Professional Issues in Software Engineering*”, Taylor

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5114) PROTOCOL SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *Able to acquire the generic software development model.*
2. *Understand the various stages of software development cycle*
3. *Maintain the software quality through software development with various protocol based environment.*

COURSE OUTCOMES

1. *Able to generate the test cases for various protocol based environment.*
2. *Understood the different protocol specification languages.*
3. *Able to perform protocol testing.*

UNIT -- I

Network Reference Model: Layered Architecture, Network Services and Interfaces, Protocol Functions, OSI Model, TCP/IP Protocol Suite, Application Protocols.

Formal Specification: Formal Specification in the Software Process, Sub-system Interface Specification, Behavioral Specification. Protocol Specification: Components of Protocol to be Specified, Communication Service Specification, Protocol Entity Specification, Interface Specifications, Interactions, Multimedia Protocol Specifications, Internet Protocol Specifications.

UNIT -- II

Architectural Design: Architectural Design Decisions, System Organization, Modular Decomposition Styles, Control Styles, Reference Architectures. Distributed Systems Architectures: Multiprocessor Architectures, Client-server Architectures, Distributed Object Architectures, Inter-organizational Distributed Computing.

UNIT -- III

Formal Description Testing for Protocol Specification, Extended State Transition Language, Language for temporal Ordering Specification, Format and Protocol Languages.

SDL: A Protocol Specification Language: SDL, Examples of SDL Based Protocol Specifications, Other Protocol Specification Languages.

Protocol Verification/Validation: Protocol Verification, Verification of a Protocol Using Finite State Machines, Protocol Validation, Protocol Design Errors, Protocol Validation Approaches, SDL Based Protocol Verification, SDL Based Protocol Validation.

UNIT -- IV

Protocol Conformance Testing: Conformance Testing, Conformance Testing Methodology and Framework, Conformance Test Architectures, Test Sequence Generation Methods, Distributed Architecture by Local Methods, Conformance Testing with TTCN, Conformance Testing in Systems with Semi controllable Interfaces, Conformance Testing of RIP, Multimedia Applications Testing, SDL Based Tools for Conformance Testing, SDL Based Conformance

Testing of MPLS.

UNIT -- V

Protocol Performance Testing: Performance Testing, SDL Based Performance Testing of TCP, SDL Based Performance Testing of OSPF, Interoperability Testing, SDL Based Interoperability Testing of CSMA/CD and CSMA/CA Protocol Using Bridge, Scalability Testing.

Protocol Synthesis: Protocol Synthesis, Interactive Synthesis Algorithm, Automatic Synthesis Algorithm, Automatic Synthesis of SDL from MSC, Protocol Resynthesize. Testing Models, PICS and PIX IT, Abstract Test Methods, Simulation Based Evaluation of Conformance Testing Methodologies. Examples include actual implementation like OSINET, based on ESTELLE tools and TTCU, PICS, PIX IT for OSINET.

TEXT BOOKS

1. Pallapa Venkataram, Sunilkumar S. Manvi, *Communication Protocol Engineering*, PHI.
2. Gregor V. Bochmann, *Protocol Specification for OSI*1* , University of Motreal, Montreal, Quebec, Canada.
3. Olivier Dubuisson, Morgan Kaufmann,*ASN.1: Communication Between Heterogeneous Systems.*

REFERENCE

1. Harry Rudin, *Tools for Protocols Driven by Formal Specifications,* Network Protocols and Tools to help produce them*, Harry Rudin, IBM Research Division, Zurich Research Laboratory, 8803 Ruschlikon, Switzerland.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5115) FORMAL METHODS OF SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *Introduce students to advanced techniques and methods in software engineering that reflect the current state of the art.*
2. *Students will be equipped with the knowledge of both formal methods and informal but precise and rigorous methods of software development.*
3. *The course places more focus on the early phases in the software development cycle.*

COURSE OUTCOMES

1. *Describe the characteristics and tradeoffs between different formal and informal methods of software development;*
2. *Specify software using a formal specification language;*
3. *Develop high quality software requirement specifications using informal or semi-formal notation.*
4. *Appreciate the use of formal and rigorous techniques in program refinement and verification.*

UNIT -- I

Introduction: Formal methods, The CICS Experience, The Z notation, The importance of Proof, Abstraction.

Propositional Logic: Proportional logic, Conjunction, Disjunction, Implication, Equivalence, Negation, Tautologies and Contradictions.

Predicate Logic: Predicate calculus, Quantifiers and declarations, Substitution, Universal Introduction and elimination, Existential introduction and elimination, Satisfaction and validity.

Equality and Definite Description: Equality, The one-point rule, Uniqueness and quantity, definite description.

UNIT -- II

Sets: Membership and extension, Set comprehension, Power sets, Cartesian products, Union, intersection, and difference, Types.

Definitions: Declarations, Abbreviations, Generic abbreviations, Axiomatic definitions, Generic definitions, Sets and predicates.

Relations: Binary relations, Domain and range, Relational inverse, Relational composition, Closures.

Functions: Partial functions, Lambda notation, Functions on relations, Overriding, Properties of functions, Finite sets.

UNIT -- III

Sequences: Sequence notation, A model for sequences, Functions on sequences, Structural induction, Bags.

Free Types: The natural numbers, Free type definitions, Proof by induction, Primitive

recursion, Consistency.
Schemas: The schema, Schemas as types, Schemas as declarations, Schemas as predicates, Renaming, Generic schemas.
Schema Operators: Conjunction, Decoration, Disjunction, Negation, Quantification and hiding, Composition.

UNIT -- IV
Promotion: Factoring operations, Promotion, Free and constrained promotion.
Preconditions: The initialization theorem, Precondition investigation, Calculation and simplification, Structure and preconditions.
A File System: A Programming interface, Operations upon files, a more complete description, a file system, Formal analysis.
Data Refinement: Refinement, Relations and non-determinism, Data types and data refinement, Simulations, Relaxing and unwinding.

UNIT -- V
Data Refinement and Schemas: Relations and schema operations, Forwards simulation, backwards simulation.
Functional Refinement: Retrieve functions, Functional refinement, Calculating data refinements, Refining promotion.
Refinement Calculus: The specification statement, Assignment, Logical constants, Sequence composition, Conditional statements, Iteration.

TEXT BOOK

1. Jim Woodcock and Jim Davies, **“***Using Z: Specification, Refinement, and Proof*”, Prentice Hall (ISBN 0-13-948472-8), 1996.

REFERENCES

1. Diller.Z, *An Introduction to Formal Methods* (2nd ed.), Wiley, 1994.
2. J. M. Spivey, “*The Z Notation: A Reference Manual”,* Second Edition, Prentice Hall, 1992.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5116) SOFTWARE METRICS & REUSE

COURSE OBJECTIVES

1. *To understand software metrics and measurement.*
2. *To emphasize the use of product and quality metrics.*
3. *To explain quality assurance and various tools used in quality management.*
4. *To learn in detail about various quality assurance models.*
5. *To understand the audit and assessment procedures to achieve quality.*
6. *To understand the knowledge of reuse and reuse problems.*
7. *To discuss COTS reuse*

COURSE OUTCOMES

1. *Able to lead and implement measurement plans for process and product assessments.*
2. *Able to analyze data for project estimation, planning and quality control in software projects.*
3. *Understood the reusable concepts that can be represented as patterns.*

UNIT - I

Basics of measurement: Measurement in everyday life, measurement in software engineering, scope of software metrics, representational theory of measurement, measurement and models, measurement scales, meaningfulness in measurement, goal-based framework for software measurement, classifying software measures, determining what to measure, software measurement validation.

UNIT - II

Empirical investigation: types of investigation, planning and conducting investigations.
Software-metrics data collection and analysis: What is good data, how to define the data, how to collect the data, how to store and extract data, analyzing software-measurement data, frequency distributions, various statistical techniques.

Measuring internal product attributes: Measuring size, aspects of software size, length, functionality and complexity, measuring structure, types of structural measures, control-flow structure, and modularity and information flow attributes, data structures.

UNIT - III

Measuring external product attributes: Modeling software quality, measuring aspects of software quality.
Metrics for object-oriented systems: The intent of object-oriented metrics, distinguishing characteristics of object-oriented metrics, various object-oriented metric suites – LK suite, CK suite and MOOD metrics.
Metrics for component-based systems: The intent of component-based metrics, distinguishing characteristics of component-based metrics, various component-based metrics.

UNIT - IV

Introduction: Software Reuse and Software Engineering, Concepts and Terms, Software Reuse products, Software Reuse processes, Software Reuse paradigms. State of the Art and the Practice: Software Reuse Management, Software Reuse Techniques, Aspects of Software Reuse, Organizational Aspects, Technical Aspects and Economic Aspects.

Programming Paradigm and Reusability: Usability Attributes, Representation and Modeling Paradigms, Abstraction and Composition in development paradigm.

UNIT - V

Object-Oriented Domain Engineering: Abstraction and Parameterization Techniques, Composition Techniques in Object Orientation.

Application Engineering: Component Storage and Retrieval, Reusable Asset Integration. **Software Reuse Technologies:** Component Based Software Engineering, COTS based development, Software Reuse Metrics, Tools for Reusability.

TEXT BOOKS

1. Norman E. Fenton and Shari Lawrence Pfleeger; *Software Metrics – A Rigorous and Practical Approach*, Thomson Asia Pte., Ltd, Singapore.
2. Stephen H. Kan; *Metrics and Models in Software Quality Engineering*, Addison Wesley, New York.
3. Hafedh Mili, Ali Mili, Sherif Yacoub and Edward Addy, *Reuse Based Software Engineering Techniques, Organization and Measurement*, John Wiley & Sons Inc
4. Carma McClure, *The Three Rs of Software Automation: Re-engineering, Repository, Reusability*, Prentice Hall New Jersey

REFERENCES

1. K. H. Möller and D. J. Paulish; *Software Metrics - A Practitioner's Guide to Improved Product Development*, Chapman and Hall, London.
2. Mark Lorenz and Jeff Kidd; *Object-Oriented Software Metrics*, Prentice Hall, New York.
3. McClure, Carma L. *Software reuse techniques : adding reuse to the system development process* , Prentice Hall
4. Poulin, Jeffrey S. , *Measuring software reuse : principles, practices, and economic models / Jeffrey S. Poulin. Reading, Mass.* : Addison-Wesley

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5117) SERVICE ORIENTED ARCHITECTURE

COURSE OBJECTIVES

1. *Understand SOA and evolution of SOA.*
2. *Understand web services and primitive, contemporary SOA.*
3. *Understand various service layers.*
4. *Understand service-oriented analysis and design based on guidelines.*

COUSE OUTCOMES

1. *Understand the concepts of Service Oriented Architecture along with the evolution of SOA.*
2. *Understand primary concepts of SOA .*
3. *Know the integration of SOA technological points with Web Services.*
4. *Implementation of SOA in development cycle of Web Services.*
5. *Integrate SOA technologies with Web Services paradigms.*
6. *Can learn the reference model of Service Oriented base line backend design for any typical environment.*

UNIT -- I

Introducing SOA: Fundamental SOA, Common Characteristics of Contemporary SOA, Common Tangible Benefits of SOA, and Common Pitfalls of Adopting SOA

The Evolution of SOA: An SOA Timeline, the Continuing Evolution of SOA, The Roots of SOA.

UNIT -- II

Web Services and Primitive SOA: The Web Services Frame Work, Services, Service Descriptions, Messaging.

Web Services and Contemporary SOA (Part I-Activity management and Composition): Message Exchange Patterns, Service Activity, Coordination, Atomic Transactions, Orchestration, and Choreography.

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

UNIT -- III

Principles of Service-Orientation: Service–Orientation and the Enterprise, Anatomy of SOA, Common Principles of Service–Orientation, Interrelation between Principles of Service-Orientation, Service Orientation and Object Orientation, Native Web Services Support for Principles of Service-Orientation.

Service Layers: Service-Orientation and Contemporary SOA, Service Layer abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration Scenarios

UNIT -- IV

SOA Delivery Strategies: SOA Delivery Lifecycle Phases, The Top-Down Strategy, The Bottom-up Strategy, The Agile Strategy.

Service Oriented Analysis (Part I-Introduction): Introduction to Service Oriented Analysis, Benefits of a Business Centric SOA, Deriving Business Services.

Service Oriented Analysis (Part-II-Service Modeling): Service Modeling, Service Modeling Guidelines, Classifying Service Model Logic, Contrasting Service Modeling Approaches.

Service Oriented Design (Part I-Introduction): Introduction to Service-Oriented Design, WSDL Related XML Schema Language Basics, WSDL Language Basics, Service Interface Design Tools.

Service Oriented Design (Part II-SOA Composition Guidelines): SOA Composing Steps, Considerations for Choosing Service Layers, Considerations for Positioning Core SOA Standards, Considerations for Choosing SOA Extensions.

UNIT -- V

Service Oriented Design (Part III- Service Design): Service Design Overview, Entity-Centric Business Service Design, Application Service Design, Task-Centric Business Service Design, Service Design Guidelines.

Service Oriented Design (Part IV-Business Process Design): WS-BPEL Language Basics, WS- Coordination Overview, Service Oriented Business Process Design.

TEXT BOOKS

1. Thomas Erl, *Service-Oriented Architecture-Concepts, Technology, and Design,* Pearson Education.
2. Eric Newcomer, Greg Lomow, *Understanding SOA with Web Services*, Pearson Education.

REFERENCES

1. Jeff Davies & others, *The Definitive guide to SOA*, Apress, Dreamtech.
2. E.Hewitt, *Java SOA Cook book,* SPD.
3. N.M.Josuttis, *SOA in Practice*, SPD.
4. M.Rosen and others, *Applied SOA*, Wiley India pvt. Ltd.
5. J.Mc Govern, and others, *Java Web Services Architecture*, Morgan Kaufmann Publishers, Elsevier.
6. Shankar.K, *SOA for Enterprise Applications,* Wiley India Edition.
7. W.Roshen, *SOA-Based Enterprise Integration*, TMH.
8. K.Rama Rao, C.Prasad, *SOA Security,* dreamtech press

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
-	-	4	2

(20CS5103) OBJECT ORIENTED SOFTWARE ENGINEERING LAB

LIST OF EXPERIMENTS:

1. Identifying Requirements from Problem Statements

   Requirements | Characteristics of Requirements | Categorization of Requirements | Functional Requirements | Identifying Functional Requirements | Preparing Software Requirements Specifications

2. Estimation of Project Metrics

   Project Estimation Techniques | COCOMO | Basic COCOMO Model | Intermediate COCOMO Model | Complete COCOMO Model | Advantages of COCOMO | Drawbacks of COCOMO | Halstead's Complexity Metrics

3. Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

   Use case diagrams | Actor | Use Case | Subject | Graphical Representation | Association between Actors and Use Cases | Use Case Relationships | Include Relationship | Extend Relationship | Generalization Relationship | Identifying Actors | Identifying Use cases | Guidelines for drawing Use Case diagrams

4. E-R Modeling from the Problem Statements

   Entity Relationship Model | Entity Set and Relationship Set | Attributes of Entity | Keys | Weak Entity | Entity Generalization and Specialization | Mapping Cardinalities | ER Diagram | Graphical Notations for ER Diagram | Importance of ER modeling

5. Identifying Domain Classes from the Problem Statements

   Domain Class | Traditional Techniques for Identification of Classes | Grammatical Approach Using Nouns | Advantages | Disadvantages | Using Generalization | Using Subclasses | Steps to Identify Domain Classes from Problem Statement | Advanced Concepts.

6. Statechart and Activity Modeling

   Statechart Diagrams | Building Blocks of a Statechart Diagram | State | Transition | Action | Guidelines for drawing Statechart Diagrams | Activity Diagrams | Components of an Activity Diagram | Activity | Flow | Decision | Merge | Fork | Join | Note | Partition | A Simple Example | Guidelines for drawing an Activity Diagram

7. Modeling UML Class Diagrams and Sequence Diagrams

   Structural and Behavioral Aspects | Class diagram | Class | Relationships | Sequence diagram | Elements in sequence diagram | Object | Life-line bar | Messages

8. Modeling Data Flow Diagrams

   Data Flow Diagram | Graphical notations for Data Flow Diagram | Symbols used in DFD | Context diagram and leveling DFD.

9. Estimation of Test Coverage Metrics and Structural Complexity

   Control Flow Graph | Terminologies | McCabe's Cyclomatic Complexity | Computing Cyclomatic Complexity | Optimum Value of Cyclomatic Complexity | Merits | Demerits

10. Designing Test Suites

    Software Testing | Standards for Software Test Documentation | Testing Frameworks | Need for Software Testing | Test Cases and Test Suite | Types of Software Testing | Unit Testing | Integration Testing | System Testing | Example | Some Remarks.

TEXT BOOKS

1. Bernd Bruegge & Allen H. Dutoit, “*Object-Oriented Software Engineering*”, 2009.
2. Ivar Jacobson, “*Object-Oriented Software Engineering*”, Pearson Education, 2009.
3 Stephen R. Schach, “*Object-Oriented Classical Software Engineering*”, Mc Graw Hill, 2010.
4 Yogesh Singh, “*Object-Oriented Software Engineering*”, 2012

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
-	-	4	2

(20CS5104) SOFTWARE REQUIREMENTS & ESTIMATION LAB

LIST OF EXPERIMENTS:

1 Draw the Work Breakdown Structure for the system to be automated

2 Schedule all the activities and sub-activities Using the PERT/CPM charts

3 Define use cases and represent them in use-case document for all the stakeholders of the system to be automated

4 Identify and analyze all the possible risks and its risk mitigation plan for the system to be automated

5 Define Complete Project plan for the system to be automated using Microsoft Project Tool

6 Define the Features, Vision, Business objectives, Business rules and stakeholders in the vision document

7 Define the functional and non-functional requirements of the system to be automated by using use cases and document them in SRS document

8 Define the following traceability matrices :

   1. Usecase vs Features
   2. Functional requirements Vs Usecases

9 Estimate the effort using the following the methods for the system to be automated:

   1. Function point metric
   2. Usecase point metric

10 Develop a tool which can be used for quantification of all the non-functional requirements

- For developing software project plan Microsoft Project or its equivalents may be used
- For developing SRS document Rational Requisite Pro Tool or its equivalents may be used

TEXT BOOKS

1. Karl E. Weigers ,*Software Requirements* ,Microsoft Press.
2. *Rajesh Naik and Swapna Kishore*, *Software Requirements and Estimation*, Tata Mc Graw Hill.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
2	-	-	2

(20HS0823) RESEARCH METHODOLOGY AND IPR

COURSE OBJECTIVES:

1. Understand some basic concepts of research and its methodologies.
2. Identify and discuss appropriate research topics, select appropriate research design, and implement a research project.
3. Understand the method of research writing and presenting research report and proposal
4. Provide an understanding on the importance of intellectual property rights
5. Understand the intricacies of grant of patent, patentability, licensing and revocation at national and international level.

COURSE OUTCOMES:

After the completion of the course, student would be able to:

1. Explain the key concepts and issues in research and basic framework of research process.
2. Formulate appropriate research problem and implement suitable research design for the research problem.
3. Identify various sources of information for literature review and data collection.
4. Develop an understanding of ethics in conducting applied research and make use of components of scholarly writing in report preparation.
5. Identify different types of Intellectual Properties (IPs), the right of ownership, scope of protection as well as the ways to create and to extract value from IP.
6. Recognize the crucial role of IP in organizations of different industrial sectors for the purposes of product and technology development.

UNIT I

Research Methodology: Meaning, Objective and importance of research - Types of research - steps involved in research - Motivation in Research, Types of Research - Significance of Research - Research Methods versus Methodology - Importance of Knowing How Research is done - Research Process - Criteria of Good Research defining research problem - Errors in selecting a research problem

UNIT II

Research Design and Data Collection: Research design - Different Research Designs - Effective literature studies - Classification of Data - Methods of Data Collection – Sampling - Sampling techniques, procedure and methods - Ethical considerations in research - Responsibility of ethics in research

UNIT III

Research Report Writing: 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" Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd , 2007.
2. Mayall, "Industrial Design", McGraw Hill, 1992. Niebel , "Product Design", McGraw Hill, 1974.
3. Asimov, "Introduction to Design", Prentice Hall, 1962.
4. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age", 2016.
5. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – I Sem.

L	T	P	C
2	-	-	0

(20HS0818) ENGLISH FOR RESEARCH PAPER WRITING

COURSE OBJECTIVES

The objectives of this course:

1. *Understand that how to improve your writing skills and level of readability.*
2. *Learn about what to write in each section.*
3. *Understand the skills needed when writing a Title.*
4. *Ensure the good quality of paper at very first-time submission.*
5. *Understand critical thinking in research writing*

COURSE OUTCOMES (COs)

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

1 .*To familiarizes students with the key concepts of linguistics and develop awareness of the latest trends in language study*

2. *To lead to a greater understanding of the human communicative action through an objective study of language*

3. *To know and appreciate the location of literature within humanities*

4. *To gain knowledge of research methods in literary studies and advanced knowledge of literature in the English language and literary theory*

5. *To carry out an independent, limited research project under supervision, in accordance with applicable norms, ideals and conditions for literary research.*

6. *To improve common and basic scholarly requirements of logical and empirical rigor.*

UNIT-I

Planning and Preparation, Word Order, Breaking up long sentences, StructuringParagraphs and Sentences, Being Concise and Removing Redundancy, AvoidingAmbiguity and Vagueness.

UNIT-II

Clarifying Who Did What, Highlighting Your Findings, Hedging and Criticising, Paraphrasing and Plagiarism, Sections of a Paper, Abstracts and Introduction.

UNIT-III

Review of the Literature, Methods, Results, Discussion, Conclusions, The Final Check.

UNIT-IV

Key skills needed when writing a Title, key skills needed when writing abstract, key skillsneeded when writing an Introduction, skills whenwriting a Review of the Literature.

UNIT-V

Skills needed when writing the Methods, skills needed when writing the Results, skills needed when writing the Discussion, skills needed when writing the Conclusions.

TEXT BOOKS

1. Goldbort R *Writing for Science,* Yale University Press. 2006
2. Day R *How to Write and Publish a Scientific Paper,* Cambridge University Press. 2006

REFERENCES

1. Highman N *Handbook of Writing for the Mathematical Sciences,* SIAM. Highman's Books,1998
2. Adrian Wallwork , *English for Writing Research Papers,* Springer New York Dordrecht. Heidelberg London, 2011.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5105) ADVANCES IN SOFTWARE TESTING

COURSE OBJECTIVES

1. *Study the significance of testing*
2. *Study the testing to be done at various levels*
3. *Understand the procedure for designing test cases*

COURSE OUTCOMES

1. *Ability to systematically test the applications*
2. *Ability to write test cases*
3. *Ability to use testing tools effectively*

UNIT -- I

Introduction – Review, Software testing strategies.
Control flow graph – basic blocks, flow graphs, paths, basic paths, path conditions and domains, Dominators and post-dominators; Program dependence graph – data dependence, control dependence, call graph,
Tests generation - Test selection Problem, equivalence partitioning, Equivalence class partitioning, boundary value analysis and category partitioning method

UNIT -- II

Finite state machines (FSM) - properties of FSM, Conformance testing, test generation, test optimization, Fault detection. **Combinatorial designs** – combinatorial test design process. **Pairwise design**: Binary factors and multi-valued factors. **Orthogonal arrays** and multi-level orthogonal arrays

UNIT -- III

Test Adequacy: Basics, measurement of test adequacy, infeasibility and test adequacy. Adequacy criteria based control – statement, block, conditions and decisions coverage techniques. Basics of Junit tool for Java.
Software Metrics: Fundamentals – **Product and Quality Metrics -** Measurement of internet product attributes- size and structure-external product attributes-measurement of quality-Software quality metrics-product quality-process quality

UNIT -- IV

Regression Testing- What is Regression Testing? Regression test process. Regression test selection techniques: Test all, Random selection, modification traversing tests, using execution trace. Regression Testing Tools

UNIT -- V

Non-functional testing
Load testing, performance testing, GUI testing, Security testing techniques and tools.
Automation: Case studies functional test automation using Selenium.

TEXT BOOKS

1. Aditya P Mathur, *Foundations of software testing*, 2nd edition, Pearson , 2013.
2. Boris Beizer, “*Software Testing Techniques*”, 2nd Edition, Dream tech press, 2003.

REFERENCES

1. M G Limaye, “*Software Testing – Principles, Techniques and Tools*”, Tata McGraw Hill, 2009.
2. Edward Kit, “*Software Testing in the Real World - Improving the Process*”, Pearson Education, 2004.
3. William E. Perry, “*Effective methods for software testing*”, 2nd Edition, John Wiley, 2000.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5106) SOFTWARE PATTERNS

COURSE OBJECTIVES

1. *Introduction to the fundamentals of software architecture.*
2. *To understand various architectural patterns of software systems.*
3. *To understand design patterns and their underlying object oriented concepts.*
4. *Software architecture and quality requirements of a software system*
5. *Identifying the appropriate patterns for design problems.*
6. *To understand design patterns and their underlying object oriented concepts.*
7. *To understand implementation of design patterns and providing solutions to real world software design problems.*
8. *To understand patterns with each other and understanding the consequences of combining patterns on the overall quality of a system.*

COURSE OUTCOMES

1. *Design and motivate software architecture for large scale software systems*
2. *Recognize major software architectural patterns, design patterns, and frameworks*
3. *Know the underlying object oriented principles of design patterns.*
4. *Understand the context in which the pattern can be applied.*
5. *Understand how the application of a pattern affects the system quality and its tradeoffs.*

UNIT -- I

Envisioning Architecture - What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views and the Architecture Business Cycle.

Creating an Architecture - Quality Attributes, Achieving qualities, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT -- II

Introduction to Patterns - What is a Pattern? What makes a Pattern? Pattern Categories, Relationships between Patterns, Pattern Description, Patterns and Software Architecture.

Architectural Patterns: Layers, Pipes and Filters, Blackboard, Broker, Microkernel, MVC, PAC, Reflection.

UNIT -- III

What is Design Pattern, Organizing catalogs, Role in solving design problems, Selection and usage.

Creational Patterns - Abstract factory, builder, factory method, prototype, and singleton.

UNIT -- IV

Structural Patterns - Adapter, bridge, composite, decorator, façade, flyweight, Proxy, Decorator, façade, flyweight, Proxy.

UNIT -- V

Behavioral Patterns - Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, and visitor.

Case Studies – Designing a Document Editor **-** Design issues of Lexi Editor in Deign Patterns, The World Wide Web - a case study in interoperability

TEXT BOOKS

1. Len Bass, Paul Clements & Rick Kazman, *Software Architecture in Practice*, second edition, Pearson Education,2003.
2. Frank Buschmann Regine Meunier, Hans Rohnert, Peter Sommerlad and Michael Stal, *Pattern-Oriented Software Architecture-A System of Patterns*, WlLEY.
3. Erich Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson Education.

REFERENCES

1. William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick , Thomas J. Mowbray, *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*,1st Edition,
2. John Thomas etc, *Java testing patterns,* wiley.
3. David M. Dikel, David Kane and James R. Wilson, *Software architecture,* Prentice Hall PTR,2001
4. Eric Freeman & Elisabeth Freeman, *Head First Design patterns*, O'REILLY, 2007.
5. Steven John Metsker & William C. Wake, *Design Patterns in Java*, Pearson education, 2006

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5118) SOFTWARE QUALITY ASSURANCE

COURSE OBJECTIVES

1. *Understand quality assurance as a fundamental component of software*
2. *Define the scope of quality assurance projects*
3. *Estimate cost of a quality assurance project and manage budgets*
4. *Prepare schedules for a quality assurance project*
5. *Develop testing & quality assurance project staffing requirements*
6. *Effectively manage a testing & quality assurance project*

COURSE OUTCOMES

1. *Critically evaluate different software development environments and contexts with respect to the application of appropriate standards and models,*
2. *Critically evaluate leading edge approaches in software development and attendant quality assurance methodologies, presenting the research using Harvard referencing.*
3. *Understand and apply key quality assurance techniques tailored for specific software development environments.*

UNIT -- I

FUNDAMENTALS OF SOFTWARE QUALITY ASSURANCE: The Role of SQA – SQA Plan – SQA considerations – SQA people – Quality Management – Software Configuration Management

UNIT -- II

MANAGING SOFTWARE QUALITY: Managing Software Organizations – Managing Software Quality – Defect Prevention – Software Quality Assurance Management

UNIT -- III

SOFTWARE QUALITY ASSURANCE METRICS: Software Quality – Total Quality Management (TQM) – Quality Metrics – Software Quality Metrics Analysis

UNIT -- IV

SOFTWARE QUALITY PROGRAM: Software Quality Program Concepts – Establishment of a Software Quality Program – Software Quality Assurance Planning – An Overview – Purpose & Scope.

UNIT -- V

SOFTWARE QUALITY MODELS AND ASSURANCE STANDARDIZATION: Models, SPICE, Malcolm Baldrige Model-P-CMM - Software Standards–ISO 9000 Quality System Standards - Capability Maturity Model and the Role of SQA in Software envelopments Maturity – SEI CMM Level 5 – Comparison of ISO 9000 Model with SEI's CMM

TEXT BOOKS

1. Mordechai Ben-Menachem / Garry S Marliss, "*Software Quality*", Vikas Publishing House, Pvt, Ltd., New Delhi.(UNIT III to V)
2. Watts S Humphrey, " *Managing the Software Process*", Pearson Education Inc.(UNIT I and II)

REFERENCES

1. Gordon G Schulmeyer, "*Handbook of Software Quality Assurance*", Third Edition, Artech House Publishers 2007
2. Nina S Godbole, "*Software Quality Assurance: Principles and Practice*", Alpha Science International, Ltd, 2004.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5119) SOFTWARE RELIABILITY

COURSE OBJECTIVES

1. *To discuss the problems of reliability specification and measurement*
2. *To introduce reliability metrics and to discuss their use in reliability specification*
3. *To show how reliability predications may be made from statistical test results.*

COURSE OUTCOMES

1. *Master attributes and assessment of quality, reliability and security of software*
2. *To describe the statistical testing process*

UNIT -- I

Introduction: The Need for Reliable Software, Software Reliability Engineering Concepts, Basic definitions, Software practitioner’s biggest problem, software reliability engineering approach, software reliability engineering process, defining the product.

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

UNIT -- II

Software Reliability Concepts: Defining failure for the product, common measure for all associated systems, setting system failure intensity objectives, determining develop software failure intensity objectives, software reliability strategies, failures, faults and errors, availability, system and component reliabilities and failure intensities, predicting basic failure intensity.

UNIT -- III

Software Reliability Modeling Survey: Introduction, Historical Perspective and Implementation, Exponential Failure Time Class of Models, Weibull and Gamma Failure Time Class of Models, Infinite Failure Category Models, Bayesian Models, Model Relationship, Software Reliability Prediction in Early Phases of the Life Cycle.

UNIT -- IV

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

Software Testing and Reliability: Introduction, Overview of Software Testing, Operational profiles, Time/Structure Based Software Reliability Estimation.

UNIT -- V

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

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

TEXT BOOKS

1. Michael R. Lyu, *Handbook of Software Reliability Engineering*, Edited published by IEEE Computer Society Press and McGraw-Hill Book Company.
2. John D. Musa, *Software Reliability Engineering*, second edition Tata McGraw-Hill.

REFERENCES

1. Patric D. T. O connor , *Practical Reliability Engineering*, 4th Edition, John Wesley & Sons, 2003.
2. Anderson and PA Lee, *Fault tolerance principles and Practice*, PHI, 1981.
3. Pradhan D K ,*Fault tolerant computing-Theory and Techniques*, (Ed.): Vol 1 and Vol 2, Prentice hall, 1986.
4. E. Balagurusamy, *Reliability Engineering*, Tata McGrawHill, 1994.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5120) MODEL DRIVEN SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *Develop enabling technologies for supporting model driven engineering approaches to software development*
2. *Develop improved techniques and tool support for using executable specifications and model-based testing to better capture, manage and test software against its requirements*
3. *Better integrate social networking tools and techniques into the software development process to improve the efficiency of collaborative and community development of software*
4. *Better support "early phase" decision making by providing tools and techniques to assess nonfunctional requirement adherence at early stages in the software development process.*

COURSE OUTCOMES

1. *Explain the role and importance of modeling in software development*
2. *Make and defend decisions regarding the appropriate use of modeling throughout the software development life-cycle*
3. *Demonstrate the practical application of several general purpose modeling languages*
4. *Design and demonstrate the practical application of domain specific modeling languages*
5. *Integrate a set of models to form effective software specifications*
6. *Describe concepts involved in the verification and translation of specifications*
7. *Demonstrate the translation of specifications to form executable software*

UNIT -- I

MDSD Basic Terminology: Goals of MDSD, MDSD Approach, Overview of MDA concepts, Architecture-Centric MDSD, Common MDSD concepts and terminology, Model-Driven Architecture, Generative Programming, Software factories, Model-Integrated computing, Language-Oriented Programming, Domain specific modeling.

UNIT -- II

Metamodeling: What is Metamodeling?, Metalevels vs. Level of Abstraction, MOF and UML, UML Meta models - Extending UML, Prototyping Requirement Management Meta-Model, UML profiles, Metamodeling and OCL, Examples ,Tool-supported Model validation, Metamodeling and Behavior, Pitfalls in Metamodeling, MDSD classification.

UNIT -- III

Model Transformation with QVT

History, M2M language requirements, Overall Architecture, An Example Transformation, The OMG standardization Process and Tool Availability, Assessment.

MDSD Tools: Roles, Architecture, Selection Criteria, and Pointers: Role of Tools in the Development Process, Tool Architecture and selection criteria, pointers.
The MDA Standard: Goals, Core concepts

UNIT -- IV
MDSD Process Building Blocks and Best Practices: Introduction, Separation between Application and domain Architecture Development, Two track Iterative Development, Target Architecture Development Process, Product-line Engineering.
Testing: Test Types, Tests in Model-driven Application Development, Testing the Domain Architecture
Versioning: What is Versioned? Projects and Dependencies, The structure of Application Projects, Version management and Build Process for mixed files, modeling in a team and versioning of partial models

UNIT -- V
Quality: Quality in Model Driven Engineering
Case study: Embedded Component Infrastructures - Overview, Product-Line Engineering, Modeling, Implementation of Components, Generator Adaptation, Code Generation

TEXT BOOKS

1. Thomos Stahl, Markus Volter ,*Model-Drievn Software Development-Technology, Engineering, Management* , John Wiley & Sons. ,jul 2006.
2. Jorg Rech, Christian Bunse,*Model-Driven Software Development: Integrating Quality Assurance,* Information Science Publishing.

REFERENCES

1. Sami Beydeda Matthias Book ,*Model-Driven Software Development*, Volker Gruhn, Springer.
2. Brian Nolan, Barclay Brown, Dr. Laurent Balmelli, Et Al Tim Bohn, *Model Driven Systems Development with Rational Products*, IBM,2008.
3. Dragan Milicev, *Model Driven Development with Executable UML*, Wilei India pvt Ltd. ,2009.

Kevin Lano, *Model Driven Software Development*, Ci Business Press., Apr 2009

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5121) SOFTWARE REENGINEERING

COURSE OBJECTIVES

1. *To explain why software re-engineering is a cost-effective option for system evolution*
2. *To describe the activities involved in the software re-engineering process*
3. *To distinguish between software and data re-engineering and to explain the problems of data re-engineering.*

COURSE OUTCOMES

1. *Assess which parts should be reengineered first;*
2. *Identify the risks and opportunities for a given re-engineering project;*
3. *Extract coarse-grained and fine-grained design models;*
4. *select the most appropriate migration strategy;*
5. *Solve the typical problems of an object-oriented re-engineering project.*

UNIT -- I

Software, Software Evolution and Maintenance: Software, Legacy software, well designed software, Software evolution challenges, Lehman’s laws, Software deterioration curve.

Software maintenance: Software change, Types of change encountered during the support phase, Maintenance costs, why is software maintenance expensive? Factors affecting maintenance, Maintenance process, Change and maintenance prediction.

Software Quality Factors, Quality and Maintainability Metrics: Internal and external attributes, McCall’s quality factors, ISO 9126 quality factors, Need and importance of quality and maintainability metrics, Metric for software correctness (Defects/KLOC), Metric for software integrity, Software reliability (MTBF), Metrics for maintainability (Mean-time-to-change (MTTC), Spoilage metric, Software maturity index, McCabe and Halstead metrics).

UNIT -- II

Design maintainability: Cohesion, Coupling, Understandability and Adaptability.

Legacy software structure, Software reengineering process model: Software change strategies include: Software maintenance, Architectural transformation, Software reengineering. Legacy software structure and distribution: Ideal structure, Real structure, Layered distribution model, Legacy software distribution, Architectural problems.

Business process reengineering: Business processes, A BPR Model, Software reengineering and its importance, Goals of reengineering, A software reengineering process model, Software reengineering activities.

UNIT -- III

Design Extraction: Reverse Engineering: Goals of reverse engineering, Why design extraction is needed?, Reverse engineering process, Reverse engineering to understand processing, Code duplication detection, Reverse engineering to understand data, Reverse engineering user interfaces, Design extraction with UML, Heuristics to extract the design, Tools for reverse engineering.

Restructuring (In Traditional context): Code restructuring: Characteristics of unstructured code, Characteristics of structured code, Spaghetti logic, Structured control logic, Restructuring problems, Flow graph restructuring, Warnier's logical simplification techniques, Some basic code restructuring methods: Interchange, Transposition, Combination, Resolution, Substitution.

UNIT -- IV

Data restructuring (Data reengineering): Data reengineering process, Data problems, Approaches: Data cleanup, Data extension, Data migration. Tools for restructuring.

Refactoring (Restructuring in object oriented context): What is refactoring?, Principles in refactoring: Why should you refactor?, When should you refactor?, Problems with refactoring, Refactoring and design, Refactoring and performance. Refactoring opportunities, Top ten of code bad smells, Different refactorings and their use, Refactoring tools.

UNIT -- V

Forward Engineering: What is forward engineering ? Goals of forward engineering, Forward engineering for client/server applications, Forward engineering for object oriented architectures, Forward engineering user interfaces, Tools for forward engineering.

Reengineering Metrics, Repositories, and Economics: Metrics in Reengineering: Why metrics in Reengineering?, Metrics as a reengineering tool, Which metrics to collect ?(Goal Question Metric (GQM) paradigm), Reengineering repositories: Why repositories?, Taxonomy (Functionality + Integration options), Issues, Reengineering economics.

TEXT BOOKS

1. Ed. Robert S. Arnold ,*Software Reengineering*, IEEE Computer Society, 1993.
2. Tom Mens, Serge Demeyer, *Software Evolution*, Springer publication company, 2008.

REFERENCES

1. Ian Sommerville, *Software Engineering*, Addison-Wesley, 6th Edition.
2. Roger S. Pressman, *Software Engineering, A Practitioner's Approach*, 6th Edition.
3. Martin Fowler, K.Beck, J.Brant., *Refactoring: Improving the Design of Existing Code.*

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5122) BIG DATA ANALYTICS

COURSE OBJECTIVES

The Objectives of this course

1. *To understand the competitive advantages of big data analytics*
2. *To understand the big data frameworks*
3. *To learn data analysis methods*
4. *To learn stream computing*
5. *To gain knowledge on Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics*

COURSE OUTCOMES

On successful completion of the course students will be able to

1. *Understand how to leverage the insights from big data analytics*
2. *Analyze data by utilizing various statistical and data mining approaches*
3. *Perform analytics on real-time streaming data*
4. *Develop Real Time Analytics Platform (RTAP) Applications*
5. *Understand the various NoSql alternative database models*
6. *Able to gain knowledge on Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics*

UNIT I

INTRODUCTION TO BIG DATA: Big Data – Definition, Characteristic Features – Big Data Applications - Big Data vs Traditional Data - Risks of Big Data - Structure of Big Data - Challenges of Conventional Systems - Web Data – Evolution of Analytic Scalability - Evolution of Analytic Processes, Tools and methods - analysis vs Reporting - Modern Data Analytic Tools.

UNIT II

HADOOP FRAMEWORK:

Distributed File Systems - Large-Scale FileSystem Organization – HDFS concepts - MapReduce Execution, Algorithms using MapReduce, Matrix-Vector multiplication – Hadoop YARN

UNIT III

DATA ANALYSIS

Statistical Methods: Regression modeling, Multivariate Analysis - Classification: SVM & Kernel Methods - Rule Mining - Cluster Analysis, Types of Data in Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Model Based Clustering Methods, Clustering High Dimensional Data – Predictive Analytics – Data analysis using R.

UNIT IV

MINING DATA STREAMS

Streams: Concepts – Stream Data Model and Architecture - Sampling data in a stream - Mining Data Streams and Mining Time-series data - Real Time Analytics Platform (RTAP) Applications - Case Studies – Real Time Sentiment Analysis, Stock Market Predictions.

UNIT V

BIG DATA FRAMEWORKS

Introduction to NoSQL – Aggregate Data Models – Hbase: Data Model and Implementations – Hbase Clients – Examples – .Cassandra: Data Model – Examples – Cassandra Clients – Hadoop Integration. Pig – Grunt – Pig Data Model – Pig Latin – developing and testing Pig Latin scripts. Hive – Data Types and File Formats – HiveQL Data Definition – HiveQL Data Manipulation – HiveQLQueries

TEXT BOOKS

1. Bill Franks, *Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics*, Wiley and SAS Business Series, 2012.
2. David Loshin,*Big Data Analytics: From Strategic Planning to Enterprise Integration withTools, Techniques, NoSQL, and Graph*, Morgan Kaufmann,2013.

REFERENCES

1. Michael Berthold, David J. Hand,*Intelligent Data Analysis*, Springer, Second Edition, 2007.
2. Michael Minelli, Michelle Chambers, and AmbigaDhiraj,*Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses*, Wiley, 2013.
3. P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*,Addison-Wesley Professional, 2012.
4. Richard Cotton,*Learning R – A Step-by-step Function Guide to Data Analysis*, O‘Reilly Media, 2013.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
3	-	-	3

(20CS5123) SOFTWARE CONFIGURATION MANAGEMENT

COURSE OBJECTIVES

1. *To learn the changing nature of software and need for change management.*
2. *To study the different phases involved in software configuration management.*
3. *To learn about the SCM plans, audits and reviews*
4. *To study the various SCM tools and implementation techniques*
5. *To study the SCM different scenarios and future directions*

COURSE OUTCOMES

1. *Identifying items that need to be controlled for changes.*
2. *Systematically controlling changes to them.*
3. *Establishing & maintaining integrity of these items and providing accurate status of items to relevant stakeholders (like developers, end users, and customers) throughout the Software Development Lifecycle.*

UNIT -- I
OVERVIEW THE SOFTWARE CONFIGURATION MANAGEMENT

SCM: Concepts and definitions – SCM Plan – Software development life cycle models – SDLC Phases – Need and importance of Software configuration management –SCM: Basic concepts – Baselines – Check-in and Check-out- Versions and Variants –System Building – Releases.

UNIT -- II
DIFFERENT PHASES OF SOFTWARE CONFIGURATION MANAGEMENT

Different Phases Of Scm – SCM System design - SCM Plan preparation – SCM Team organization – SCM Infrastructure organization – SCM Team training – Project team training – Configuration identification – Configuration Control –Configuration status accounting – Configuration audits.

UNIT -- III
CONFIGURATION AUDITS AND MANAGEMENT PLANS

When, what and who of auditing - Functional Configuration audit – Physical Configuration audit – Auditing the SCM System – Role of SCM Team in configuration audits – SCM plan and the incremental approach – SCM Plan and SCM Tools – SCM Organization.

UNIT -- IV
SOFTWARE CONFIGURATION MANAGEMENT TOOLS AND IMPLEMENTATION

Advantages of SCM tools – Reasons for the increasing popularity of SCM tools – SCM Tools and SCM Functions – SCM tool selection – Role of Technology –Selection criteria – Tool implementation – SCM implementation plan

UNIT -- V

TRENDS IN SCM: FUTURE DIRECTIONS

SCM in different scenarios – SCM and project size – SCM in integrated development environments – SCM In distributed environments – SCM and CASE Tools - Trends in SCM - Hardware and Software Management – Better integration with IDE'S and CASE environments – Customization – Better decision making capabilities – Reduction in SCM Team size – Market snapshot.

TEXT BOOKS / REFERENCES

1. Jessica Keyes, *Software Configuration Management*, Auerbach Publications, 2008.
2. Alexis Leon, *Software Configuration Management Handbook*, Artech Print on Demand; 2 edition, 2009.
3. Robert Aiello and Leslie Sachs, *Configuration Management Best Practices: Practical Methods that work in Real World*, Addison-Wesley Professional; 1 edition, 2010.
4. Stephen P. Berczuk, Brad Appleton and Kyle Brown , "*Software Configuration Management Patterns: Effective Teamwork and Practical Integration*", Addison-Wesley , 2003.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
-	-	4	2

(20CS5107) ADVANCES IN SOFTWARE TESTING LAB

COURSE OBJECTIVES

1. *To learn to use the following (or Similar) automated testing tools to automate testing:*
   - *Win Runner/QTP for functional testing.*
   - *Load Runner for Load/Stress testing.*
   - *Test Director for test management.*
   - *JUnit, HTMLUnit, CPPUnit.*
2. *To study state-of-art tools for software testing and Middleware technologies*

COURSE OUTCOMES

1. *Test the software applications using standard tools available in the market*

LIST OF EXPERIMENTS

1. Write programs in C Language to demonstrate the working of the following constructs: do...while ii) while….do iii) if…else iv) switch v) for
2. A program written in C language for Matrix Multiplication fails. Introspect the causes for its failure and write down the possible reasons for its failure.
3. Consider ATM System and Study its system specifications and report the various bugs.
4. Write the test cases for banking application.
5. Create test plan document for Library Management System.
6. Create test cases for Railway Reservation.
7. Create test plan document for Online Shopping.

Working with Tool’s:

Understand the Automation Testing Approach, Benefits, Workflow, Commands and Perform Testing on one application using the following Tool’s.

1. Win runner Tool for Testing.
2. Load runner Tool for Performance Testing.
3. Selenium Tool for Web Testing.
4. Bugzilla Tool for Bug Tracking.
5. Test Director Tool for Test Management.
6. Test Link Tool for Open Source Testing.

TEXT BOOKS / REFERENCES

1. M G Limaye, “*Software Testing – Principles, Techniques and Tools”,* Tata McGraw Hill, 2009.
2. Edward Kit, “*Software Testing in the Real World - Improving the Process*”, Pearson Education, 2004.
3. William E. Perry, “*Effective methods for software testing*”, 2nd Edition, John Wiley, 2000.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
-	-	4	2

(20CS5108) SOFTWARE PATTERNS LAB

COURSE OBJECTIVES

1. *Construct UML diagrams for static view and dynamic view of the system.*
2. *Generate creational patterns by applicable patterns for given context.*
3. *Create refined model for given Scenario using structural patterns.*
4. *Construct behavioral patterns for given applications.*
5. *Construct architectural patterns for given applications*.

COURSE OUTCOMES

1. *Understand the Case studies and design the Model..*
2. *Understand how design patterns solve design problems.*
3. *Develop design solutions using creational patterns.*
4. *Construct design solutions by using structural, behavioral and architectural patterns*

LIST OF EXPERIMENTS

1. Identify the application where you can use single pattern and implement it.
2. Using UML design one of the architectural patterns.
3. Using UML design one of the creational patterns.
4. Using UML design one of the structural patterns.
5. Using UML design one of the behavioral patterns.
6. User gives a print command from a word document. Design to represent this chain of responsibility design pattern.
7. User gives a print command from a word document. Design to represent this Singleton design pattern.
8. Identify the application where you can use multiple creational patterns and implement it.
9. Identify the application where you can use multiple structural patterns and implement it.
10. Identify the application where you can use multiple behavioral patterns and implement it.
11. Identify the application where you can use architectural patterns and implement it.

TEXT BOOKS

1.Len Bass, Paul Clements & Rick Kazman, *Software Architecture in Practice*, second edition, Pearson Education,2003.

2. Frank Buschmann Regine Meunier, Hans Rohnert, Peter Sommerlad and Michael Stal, *Pattern-Oriented Software Architecture-A System of Patterns*, WlLEY.

3.Erich Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson Education.

REFERENCES

1.William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick , Thomas J. Mowbray, *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*,1st Edition,

2.John Thomas etc, *Java testing patterns*, wiley.

3.David M. Dikel, David Kane and James R. Wilson, *Software architecture*, Prentice Hall PTR,2001

4.Eric Freeman & Elisabeth Freeman, *Head First Design patterns*, O'REILLY, 2007.

5.Steven John Metsker & William C. Wake, *Design Patterns in Java*, Pearson education, 2006

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
-	-	4	2

(20CS5109) MINI PROJECT

****

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

I M.Tech – II Sem.

L	T	P	C
2	-	-	0

(20HS0829) CONSTITUTION OF INDIA

COURSE OBJECTIVES

1. *Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.*
2. *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.*
3. *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

1. *Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.*
2. *Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.*
3. *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.*
4. *Discuss the passage of the Hindu Code Bill of 1956.*

UNIT – I
History of Making of the Indian Constitution:
History, Drafting Committee, (Composition & Working)
UNIT-- II
Philosophy of the Indian Constitution:
Preamble, Salient Features
UNIT-- III
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 -- IV
Organs of Governance: Parliament, Composition, Qualifications and Disqualifications Powers and Functions.
Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions.

UNIT -- V
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
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

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.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5124) SECURE SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *Students will demonstrate knowledge of the distinction between critical and non-critical systems.*
2. *Students will demonstrate the ability to manage a project including planning, scheduling and risk assessment/management.*
3. *Students will demonstrate an understanding of the proper contents of a software requirements document for secure software engineering.*
4. *Students will author a formal specification for secure software systems.*
5. *Students will demonstrate an understanding of distributed system architectures and application architectures.*
6. *Students will demonstrate an understanding of the differences between real-time and non-real time systems.*

COURSE OUTCOMES

1. *Ability to identify specific components of a software design that can be targeted for reuse.*
2. *Ability to learn software testing plan and metrics for secure software engineering.*

UNIT -- I

Why Is Security a Software Issue?

Introduction, The problem, Software assurance and software security, Threats to software security, Sources of software insecurity, the benefits of detecting software security defects early, managing secure software development.

What Makes Software Secure?

Defining properties of secure software, How to influence the security properties of software, How to assert and specify desired security properties.

UNIT -- II

Requirements Engineering for Secure Software

Introduction, Misuse and Abuse Cases, The SQUARE process model: SQUARE sample outputs, Requirements elicitation, Requirements Prioritization.

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.

UNIT -- III

Considerations for Secure Coding and Testing

Introduction, Code analysis, Coding practices, Software security testing, Security testing considerations throughout the SDLC.

Security and Complexity: System Assembly Challenges

Introduction, Security failures, Functional and attacker perspectives for security analysis, System complexity drivers and security, Deep technical problem complexity.

UNIT -- IV

Governance, and Managing for More Secure Software

Introduction, Governance and security, Adopting an enterprise software security framework, How much security is enough?, Security and project management, maturity of practice.

UNIT -- V

Security Metrics

Defining security metrics, Diagnosing problems and measuring technical security, Analysis Techniques, Organize, aggregate, and analyze data to bring out key insights.

TEXT BOOKS

1. Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, *Software Security Engineering: A Guide for Project Managers*, Addison-Wesley , 1st edition, 2008.
2. Andrew Jaquith, *Security Metrics: Replacing Fear, Uncertainty, and Doubt* , Addison-Wesley , 1st edition , 2007.

REFERENCES

1. Haralambos Mouratidis, Paolo Giorgini, *Integrating Security and Software Engineering: Advances and Future Vision,* IGI Global, 2006.
2. Gary McGraw , *Software Security: Building Security In* , Addison-Wesley, 2006
3. Mark Dowd, John McDonald, Justin Schuh, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*, Addison-Wesley, 1st edition, 2006
4. John Viega,Gary McGraw,*Building Secure Software: How to Avoid Security Problems the Right Way,* Addison-Wesley, 2001

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5125) AGILE METHODOLOGIES

COURSE OBJECTIVES

1. *To provide students with a theoretical as well as practical understanding of agile software development practices and how small teams can apply them to create high-quality software.*
2. *To provide a good understanding of software design and a set of software technologies and APIs.*
3. *To do a detailed examination and demonstration of Agile development and testing techniques.*
4. *To understand the benefits and pitfalls of working in an Agile team.*
5. *To understand Agile development and testing.*

COURSE OUTCOMES

1. *Realize the importance of interacting with business stakeholders in determining the requirements for a software system*
2. *Perform iterative software development processes: how to plan them, how to execute them.*
3. *Point out the impact of social aspects on software development success.*
4. *Develop techniques and tools for improving team collaboration and software quality.*
5. *Perform Software process improvement as an ongoing task for development teams.*
6. *Show how agile approaches can be scaled up to the enterprise level.*

UNIT -- I
Agile Methodology

Theories for Agile Management – Agile Software Development – Traditional Model vs. Agile Model - Classification of Agile Methods – Agile Manifesto and Principles – Agile Project Management – Agile Team Interactions – Ethics in Agile Teams - Agility in Design, Testing – Agile Documentations – Agile Drivers, Capabilities and Values

UNIT -- II
Agile Processes

Lean Production - SCRUM, Crystal, Feature Driven Development- Adaptive Software Development - Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices.

UNIT -- III
Agility And Knowledge Management

Agile Information Systems – Agile Decision Making - Earl_S Schools of KM – Institutional Knowledge Evolution Cycle – Development, Acquisition, Refinement, Distribution, Deployment , Leveraging – KM in Software Engineering – Managing Software Knowledge – Challenges of Migrating to Agile Methodologies – Agile Knowledge Sharing – Role of Story-Cards – Story-Card Maturity Model (SMM).

UNIT -- IV

Agility And Requirements Engineering

Impact of Agile Processes in RE–Current Agile Practices – Variance – Overview of RE Using Agile – Managing Unstable Requirements – Requirements Elicitation – Agile Requirements Abstraction Model – Requirements Management in Agile Environment, Agile Requirements Prioritization – Agile Requirements Modeling and Generation – Concurrency in Agile Requirements Generation.

UNIT -- V

Agility And Quality Assurance

Agile Product Development – Agile Metrics – Feature Driven Development (FDD) – Financial and Production Metrics in FDD – Agile Approach to Quality Assurance - Test Driven Development – Agile Approach in Global Software Development.

TEXT BOOKS

1. David J. Anderson and Eli Schragenheim, *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*‖, Prentice Hall, 2003.
2. Hazza and Dubinsky, *Agile Software Engineering, Series: Undergraduate Topics in Computer Science*‖*,* Springer, 2009.

REFERENCES

1. Craig Larman, *Agile and Iterative Development: A Manager_s Guide*, Addison-Wesley, 2004.
2. Kevin C. Desouza, *Agile Information Systems: Conceptualization, Construction, and Management*, Butterworth-Heinemann, 2007.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CS5126) FUNDAMENTALS OF DATA SCIENCE

COURSE OBJECTIVES

1. *Understand the machine learning*
2. *To get familiar with regression and its techniques*
3. *To Perform effective data analysis using "R"*
4. *To solve computing issues which includes "Reading R Data", "accessing R Packages" etc?*
5. *To work practically in data warehouse modeling i.e OLAP, data cube etc*

COURSE OUTCOMES

1. *Have a good understanding of the fundamental issues and challenges of* machine learning*: data, model selection, model complexity, etc.*
2. *Understood the strengths and weaknesses of many popular* machine learning a*ble to solve the complexity issues using "R" Packages*

UNIT - I

Introduction, What Is Statistical Learning?, Why Estimate f?, How Do We Estimate f?, The Trade-Off Between Prediction Accuracy and Model Interpretability, Supervised Versus Unsupervised Learning, Regression Versus Classification Problems, Assessing Model Accuracy, Measuring the Quality of Fit, The Bias-Variance Trade-of, The Classification Setting, Introduction to R, Basic Commands, Graphics, Indexing Data, Loading Data, Additional Graphical and Numerical Summaries.

UNIT – II

Linear Regression, Simple Linear Regression, Multiple Linear Regression, Other Considerations in the Regression Model, Comparison of Linear Regression with K-Nearest Neighbours, Linear Regression.

UNIT-III

Classification, Logistic Regression, Linear Discriminant Analysis, A Comparison of Classification Methods, Logistic Regression, LDA, QDA, and KNN.

UNIT- IV

Programming for basic computational methods such as Eigen values and Eigen vectors

Data Wrangling: Data Acquisition, Data Formats, Imputation, The split-apply-combine paradigm. Data wrangling with r.

UNIT-V

Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

Data Warehouse: Basic Concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation.

TEXT BOOKS

1. Gareth James Daniela Witten Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, February 11, 2013, web link: www.statlearning.com.
2. Mark Gardener, *Beginning R The statistical Programming Language*, Wiley, 2015.
3. Han , Kamber, and J Pei, *Data Mining Concepts and Techniques*, 3rd edition, Morgan Kaufman, 2012.

REFERENCES

1. Sinan Ozdemir, *Principles of Data Science*, Packt Publishing Ltd Dec 2016.
2. Joel Grus, *Data Science from Scratch*, Oreilly media, 2015.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20HS0824) BUSINESS ANALYTICS

COURSE OBJECTIVES:

1. Understand the concepts and methods of business analytics.
2. To gain an understanding of how managers use business analytics to formulate and solve business problems and to support managerial decision making.
3. Identify the management related issues and processes to resolve
4. Understand the significance of forecasting models helpful in decision making
5. To become familiar with processes needed to develop, report, and analyze business data.

COURSE OUTCOMES:

After the completion of the course, student would be able to:

1. Analyze data using statistical and data mining techniques and understand relationships between the underlying business processes of an organization.
2. Design alternatives to solve business problems utilizing quantitative analysis, critical thinking and sound ethical decision making.
3. Summarize, process and transform data for obtaining meaningful conclusions
4. Interpret data using latest data analytics tools to address organisational problems
5. Organize and critically apply the concepts and methods of business analytics
6. Assess decision problems and build models for creating solutions using business analytical tools.

UNIT I

Business analytics: Overview of Business analytics - Scope of Business analytics - Business Analytics Process - Relationship of Business Analytics Process and organisation - competitive advantages of Business Analytics - Statistical Tools: Statistical Notation, Descriptive Statistical methods, Review of probability distribution and data modelling, sampling and estimation methods overview.

UNIT II

Trendiness and Regression Analysis: Modelling Relationships and Trends in Data - simple Linear Regression - Important Resources - Business Analytics Personnel - Data and models for Business analytics - problem solving - Visualizing and Exploring Data, Business Analytics Technology

UNIT III

Organization Structures of Business analytics: Team management - Management Issues - Designing Information Policy – Outsourcing - Ensuring Data Quality - Measuring contribution of Business analytics - Managing Changes - Descriptive Analytics - predictive analytics - predicative Modelling - Predictive analytics analysis - Data Mining - Data Mining Methodologies - Prescriptive analytics and its step in the business analytics Process - Prescriptive Modelling - nonlinear Optimization.

UNIT IV

Forecasting Techniques: Qualitative and Judgmental Forecasting - Statistical Forecasting Models - Forecasting Models for Stationary Time Series - Forecasting Models for Time Series with a Linear Trend - Forecasting Time Series with Seasonality - Regression Forecasting with Casual Variables - Selecting Appropriate Forecasting Models - Monte Carlo Simulation and Risk Analysis: Monte Carle Simulation Using Analytic Solver Platform - New-Product Development Model - Newsvendor Model - Overbooking Model - Cash Budget Model.

UNIT V

Decision Analysis: Formulating Decision Problems - Decision Strategies with the Outcome Probabilities - Decision Trees - The Value of Information - Utility and Decision Making - Recent Trends in Embedded and collaborative business intelligence - Visual data recovery - Data Storytelling and Data journalism.

TEXT BOOKS

1. S. Christian Albright, *Business Analytics: Data analysis & Decision making*, Cengage Learning
2. Jeffery Camm, & others, *Essentials of Business Analytics,* Cengage Learning

REFERENCES

1. Marc J. Schniederjans, Dara G. Schniederjans, Christopher M. Starkey,*Business analytics Principles, Concepts, and Applications,* Pearson FT Press
2. James Evans, *Business Analytics,* Pearsons Education.
3. Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, Kenneth C. Lichtendahl Jr., *Data mining for business analytics: Concepts, Techniques and Applications,* WILEY.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20ME3026) INDUSTRIAL SAFETY

COURSE OBJECTIVES

1. *To learn about mechanical and electrical hazards.*
2. *To learn about Fundamentals of Maintenance Engineering.*
3. *To learn about Wear and Corrosion and their prevention.*
4. *To know about Fault Tracking*
5. *To learn about Periodic and preventive maintenance.*

COURSE OUTCOMES

Students undergoing this course are able to

1. *Understand the points of factories act 1948 for health and safety.*
2. *Understand the cost & its relation with replacement economy.*
3. *Understand the concepts of Wear and Corrosion Prevention*
4. *Understand the concepts of sequence of fault finding activities*
5. *Understand the Program and schedule of preventive maintenance of mechanical and electrical equipment.*
6. *Understand the Periodic Maintenance of Equipments*

UNIT-I

Industrial Safety: Accident, causes, types, results and control, mechanical and electrical hazards, types, causes and preventive steps/procedure, describe salient points of factories act 1948 for health and safety, wash rooms, drinking water layouts, light, cleanliness, fire, guarding, pressure vessels, etc, Safety color codes. Fire prevention and fire fighting, equipment and methods.

UNIT-II

Fundamentals of Maintenance Engineering: Definition and aim of maintenance engineering, Primary and secondary functions and responsibility of maintenance department, Types of maintenance, Types and applications of tools used for maintenance, Maintenance cost & its relation with replacement economy, Service life of equipment.

UNIT-III

Wear and Corrosion and their Prevention: Wear- types, causes, effects, wear reduction methods, lubricants-types and applications, Lubrication methods, general sketch, working and applications, i. Screw down grease cup, ii. Pressure grease gun, iii. Splash lubrication, iv. Gravity lubrication, v. Wick feed lubrication vi. Side feed lubrication, vii. Ring lubrication, Definition, principle and factors affecting the corrosion. Types of corrosion, corrosion prevention methods.

UNIT-IV

Fault Tracing: Fault tracing-concept and importance, decision tree concept, need and applications, sequence of fault finding activities, show as decision tree, draw decision tree for problems in machine tools, hydraulic, pneumatic, automotive, thermal and electrical equipment's like, I. Any one machine tool, ii. Pump iii. Air compressor, iv. Internal combustion engine, v. Boiler, vi. Electrical motors, Types of faults in machine tools and their general causes.

UNIT-V

Periodic and Preventive Maintenance: Periodic inspection-concept and need, degreasing, cleaning and repairing schemes, overhauling of mechanical components, overhauling of electrical motor, common troubles and remedies of electric motor, repair complexities and its use, definition, need, steps and advantages of preventive maintenance. Steps/procedure for periodic and preventive maintenance of: I. Machine tools, ii. Pumps, iii. Air compressors, iv. Diesel generating (DG) sets, Program and schedule of preventive maintenance of mechanical and electrical equipment, advantages of preventive maintenance. Repair cycle concept and importance

TEXT BOOKS

1. Higgins & Morrow, *Maintenance Engineering Handbook*, Da Information Services.
2. H. P. Garg, *Maintenance Engineering*, S. Chand and Company.

REFERENCE BOOKS

1. Audels, *Pump-hydraulic Compressors*, Mcgrew Hill Publication.
2. Winterkorn, *Foundation Engineering Handbook*,Chapman & Hall London.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20ME3027) ADVANCES IN OPERATIONS RESARCH

COURSE OBJECTIVES:

1. *Enumerate the fundamentals of Linear Programming*
2. *Learn classical optimization techniques*
3. *Develop the best strategy of Game and identifying the Queuing theory.*
4. *Understand about sequence and optimum Duration of the Project*
5. *Develop the importance of Replacement models and Inventory control*

COURSE OUTCOMES

On successful Completion of this course the student will be able to

1. *Create mathematical models of the real time situations.*
2. *Implement Transportation and Assignment problems to solve in real time industry*
3 *Choose the best strategy of Game and capable of identifying the suitable queuing theory*
4. *Enumerate fundamental techniques and apply it to solve various optimization areas*
5. *Investigate, study, Apply knowledge in Replacement models and*
6. *Understand the Inventory control Models*

.

UNIT-I

Introduction to OR and Linear Programming-OR definition–Types of Operations Research models; Linear Programming- Problem Formulation, Graphical Method, Simplex Method, Big-M Method, Degeneracy - Problems

UNIT-II

Transportation Problem – Formulation; Initial Basic Feasible Solution-North-West Corner Rule, Least Cost Method, Vogel‘s Approximation Method, Modified Distribution (MODI) Method, Unbalanced Transportation - Problems

Assignment Problem – Formulation, Optimal Solution -Traveling Salesman problem.

UNIT-III

Game Theory - Introduction – Minimax (Maxi mini) Criterion and Optimal Strategy, Saddle Point, Solution of Games with Pure Strategy and Mixed Strategies – 2 X 2 Games – Dominance Principle.

Queuing Theory- Introduction to queuing system–Service Channel, Arrival Pattern, Size of Population, Service Pattern, Queue Discipline, Customer Behavior, Probability Distribution-Birth & Death Process, Simple Problems on Single Service channel only.

UNIT-IV

Sequencing –Terminology - Johnson’s Algorithm for n-jobs x 2 Machines and n-jobs x 3 machines models - Problems

PERT & CPM: Introduction, Difference between PERT and CPM, Terminology- Activities, Events, Predecessor, Early Start, Early Finish, Late Start & Late Finish Times, Earliest Occurrence and Latest Occurrence of the Event, Total Float, Free Float, Independent Float; CPM- Deterministic Model; PERT- Probabilistic Model, Critical Path, Optimal Project

Duration, Least Possible Project Duration- Problems.

UNIT-V

Replacement – Failure Mechanism of Items**,** Types of Replacements- Individual Replacement policy, Group Replacement policy, Replacement of items fail suddenly – problems

Inventory - Necessity for maintaining inventory, inventory costs, classification of fixed order quantity inventory models, selective inventory management techniques.

.

TEXT BOOKS

1. S D. SHARMA *Operations Research* KNRN Publications. 17th edition 2015
2. Hamdy A Taha , *Operations Research* Pearson Publications, 9 th edition 2015

REFERENCES

1. Manohar Mahajan *Operations Research,* Dhanpat Rai &Co 2016
2. Er. Prem kumar Guptha & Dr.D.S.Hira *Operations Research*, Schand publications 2012.
3. R Panneerselvam *Operations Research* PHI, 2nd edition, 2012.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20CE31028) COST MANAGEMENT OF ENGINEERING PROJECTS

COURSE OBJECTIVES

The Objectives of this course

1. *To Implement CPM and PERT concepts inconstruction*
2. *To provide techniques to develop personal skills of practical use in the Management and implementation of Civil Engineeringprojects*
3. *To know the Management techniques, the development of personal, interpersonal and Project Managementskills*
4. *To provide a fundamental of understanding of the social, economic, resource management within which the Construction Project takesplace.*

COURSE OUTCOMES

On successful completion of the course students will be able to

1. *Implement generic and special Construction Project Management skills to a higher level*
2. *Understand the special management skills required in multidisciplinary and global ConstructionIndustry*
3. *Integrate and apply theoretical concepts, ideas, tools and techniques to Construction practice.*
4. *Can plan, execute, monitor and control construction projects using Construction Project Management Tools such as CPM &PERT*
5. *Analyse Cost Behavior and Profit Planning according to the projects*
6. *Efficiently perform Budgetary Control.*

UNIT-I

Introduction: Introduction to Mobile Computing, Introduction to Android Development Environment, Factors in Developing Mobile Applications, Mobile Software Engineering, Frameworks and Tools, Generic UI Development Android User.

UNIT -II

Introduction and Overview of the Strategic Cost Management Process Cost concepts in decision-making; Relevant cost, Differential cost, Incremental cost and Opportunity cost. Objectives of a Costing System; Inventory valuation; Creation of a Database for operational control; Provision of data for Decision-Making.

UNIT-III

Project: meaning, Different types, why to manage, cost overruns centres, various stages of project execution: conception to commissioning. Project execution as conglomeration of technical and nontechnical activities.Detailed Engineering activities. Pre project execution main clearances and documents Project team: Role of each member. Importance Project site: Data required with significance. Project contracts.Types and contents. Project execution Project cost control. Bar charts and Network diagram. Project commissioning: mechanical and process

UNIT-IV

Cost Behavior and Profit Planning Marginal Costing; Distinction between Marginal Costing and Absorption Costing; Break-even Analysis, Cost-Volume-Profit Analysis. Various decision-making problems. Standard Costing and Variance Analysis. Pricing strategies: Pareto Analysis. Target costing, Life Cycle Costing. Costing of service sector. Just-in-time approach, Material Requirement Planning, Enterprise Resource Planning, Total Quality Management and Theory of constraints.

Activity-Based Cost Management, Bench Marking; Balanced Score Card and Value-Chain Analysis.

UNIT-V

Budgetary Control; Flexible Budgets; Performance budgets; Zero-based budgets. Measurement of Divisional profitability pricing decisions including transfer pricing.

Quantitative techniques for cost management, Linear Programming, PERT/CPM, Transportation problems, Assignment problems, Simulation, Learning Curve Theory.

TEXT BOOKS

1. *Cost Accounting A Managerial Emphasis*, Prentice Hall of India, New Delhi
2. Charles T. Horngren and George Foster, *Advanced Management Accounting*
3. Robert S Kaplan Anthony A. Alkinson, *Management & Cost Accounting*,
4. Ashish K. Bhattacharya,*Principles & Practices of CostAccounting*, A. H. Wheeler publisher
5. N.D. Vohra, *Quantitative Techniques in Management*, Tata McGraw Hill Book Co. Ltd.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20ME3028) COMPOSITE MATERIALS

COURSE OBJECTIVES

1. *To understand the mechanical behavior of composite materials*
2. *To get an overview of the methods of manufacturing composite materials.*
3. *To know the fundamentals of composite materials.*
4. *To understand the fabrication and process of composites.*
5. *To recognize the applications of composite materials.*

COURSE OUTCOMES

Upon completion of this course, the students will have an overview of

1. *Fundamental concept of composite materials.*
2. *Different types of composite materials.*
3. *Fabrication and processing of composite materials.*
4. *MMC & CMC*
5. *Mechanical behavior of composite materials.*
6. *Application of composite materials .*

UNIT-I

Introduction To Composites: Fundamentals of composites – need– enhancement of properties – classifications —Introduction to Reinforcement composites–types. Applications. Fiber production techniques for glass, carbon and ceramic fibers –Resin materials-Types.

UNIT-II

Polymer Matrix Composites: Fabrication of PMC’s ,Fabrication of Fibers, Plastic Fiber Forms, Pre-pregs, Molding Compounds-Processes, Lay-Ups, Filament Winding, Pultrusion, and Recycling. Matrix – Reinforcement Interface, Wettability.

UNIT-III

MMC&CMC : Fabrication of MMC’S, Liquid Infiltration- Casting, Solid State Processes-Diffusion Bonding &In Situ Technique. Fabrication of CMC’s, Hot-Pressing, Infiltration, In Situ Chemical reaction Techniques. CVD& CVI, Sol-gel.

UNIT-IV

Mechanics of Composites: Basic assumptions of laminated anisotropic plates, symmetric laminates, angle ply laminates, cross ply laminates, laminate structural moduli, evaluation of lamina properties, determination of lamina stresses, maximum stress and strain criteria, Von - Mises Yield criterion for isotropic materials, generalized Hill’s criterion for anisotropic materials, Tsai-Hill’s criterion for composites, prediction of laminate failure, thermal analysis of composite laminates

UNIT-V

Applications Of Composites: Applications of advanced composite materials. Environmental effects in Composites, Green composites, Synthesis and Properties of Nano composites. Surface Composites & Surface metal matrix composites: Need, Synthesis, Properties and applications.

TEXT BOOKS

1. Mathews F. L. and Rawlings R. D., "*Composite Materials: Engineering and Science*", 1st Edition, Chapman and Hall, London, England, 1994.
2. Chawla K. K., "*Composite materials*", Second Edition, Springer – Verlag, 1998.

REFERENCES

1. Clyne, T. W. and Withers, P. J., "*Introduction to Metal Matrix Composites*", Cambridge University Press, 1993.
2. Strong, A.B., "*Fundamentals of Composite Manufacturing*", SME, 1989.
3. Sharma, S.C., "*Composite materials*", Narosa Publications, 2000.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR (AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
3	-	-	3

(20EE2128) WASTE TO ENERGY

COURSE OBJECTIVES

The Objectives of this course

1. *To understand the importance of gaining energy from the waste*
2. *To Understand and analyze the pattern of renewable energy resources Suggest methodologies / technologies for its utilization Economics of the utilization and environmental aspects.*
3. *To undusted the need and production of for biogas.*

COURSE OUTCOMES

On successful completion of the course students will be able to

1. *Identify the new methodologies / technologies for effective utilization of renewable energy sources.*
2. *Analyse over different types of waste for energy conception.*
3. *Understand different types of Bio mass utilizations.*

UNIT-I

INTRODUCTION TO ENERGY FROM WASTE:

Classification of waste as fuel – Agro based, Forest residue, Industrial waste - MSW – Conversion devices – Incinerators, gasifiers, digestors.

UNIT-II

BIOMASS PYROLYSIS:

Pyrolysis – Types, slow fast – Manufacture of charcoal – Methods -Yields and application – Manufacture of pyrolytic oils and gases, yields and applications.

UNIT-III

BIOMASS GASIFICATION:

Gasifiers – Fixed bed system – Downdraft and updraft gasifiers –Fluidized bed gasifiers – Design, construction and operation – Gasifier burner arrangement for thermal heating – Gasifier engine arrangement and electrical power – Equilibrium and kinetic consideration in gasifier operation.

UNIT-IV

BIOMASS COMBUTION:

Biomass stoves – Improved chullahs, types, some exotic designs, fixed bed combustors, Types, inclined grate combustors, Fluidized bed combustors, Design, construction and operation - Operation of all the above biomass combustors.

UNIT-V

PROPERTIES OF BIOGAS (CALORIFIC VALUE AND COMPOSITION)

Biogas plant technologyand status - Bio energy system - Design and constructional features - Biomass resources and their classification - Biomass conversion processes - Thermo chemical conversion - Direct combustion - biomass gasification - pyrolysis and liquefaction - biochemical conversion - anaerobic digestion - Types of biogas Plants – Applications- Alcohol production from biomass - Bio diesel production - Urban waste to energy conversion - Biomass energy programme in India.

TEXT BOOKS

1. Desai, Ashok V., *Non-Conventional Energy*, Wiley Eastern Ltd.,1990.
2. Khandelwal, K. C. and Mahdi, S. S.,*Biogas Technology - A Practical HandBook* - Vol. I & II, Tata McGraw Hill Publishing Co. Ltd.,1983.
3. Challal, D. S., *Food, Feed and Fuel from Biomass,* IBH Publishing Co. Pvt. Ltd.,1991.

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

II M.Tech – I Sem.

L	T	P	C
-	-	20	10

(20CS5110) PHASE-I DISSERTATION-I /INDUSTRIAL PROJECT

****

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

II M.Tech – II Sem.

L	T	P	C
-	-	32	16

(20CS5111) PHASE –II DISSERTATION II

****