COURSE STRUCTURE

AND

DETAILED SYLLABUS

SOFTWARE ENGINEERING

For

M.TECH. TWO YEAR DEGREE COURSE

(Applicable for the batches admitted from 2018-19)

VAADGEVI COLLEGE OF ENGINEERING

VAAGDEVI COLLEGE OF ENGINEERING
AUTONOMOUS
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
M.TECH. (Software Engineering)

COURSE STRUCTURE

(R18 Regulations applicable for the batches admitted from Academic Year 2018-19 onwards)

I-SEMESTER

S.No	Course Codes	Title of the Course	L	T	P	C
1	M18CS01	Data Structures and Algorithms	3	0	0	3
2	M18SW01	Software Development Methodologies	3	0	0	3
3		**Program Elective-I**				
	M18CS05	Cloud Computing	3	0	0	3
	M18SW02	Component Based Software Engineering				
	M18SW03	Internet Technologies and services				
4		**Program Elective-II**				
	M18SW04	Software requirements and Estimation	3	0	0	3
	M18SW05	Object Oriented Software Engineering				
	M18SW06	Information Theory and Coding				
5	M18MC01	Research Methodology	2	0	0	2
6	M18AC01	**Audit Course-I** English for Research Paper Writing	2	0	0	0
7	M18SW07	Software Development Methodologies Lab	0	0	4	2
8	M18CS10	Cloud Computing Lab	0	0	4	2
		Total Credits	**16**	**0**	**8**	**18**

II-SEMESTER

S.No	Course Codes	Title of the Course	L	T	P	C
1	M18SW08	Software Quality Assurance and Testing	3	0	0	3
2	M18CS18	Software Project and Project Management	3	0	0	3
3		**Program Elective-III**				
	M18SW09	Software Architecture and Design Patterns	3	0	0	3
	M18SW10	Agile Software Development				
	M18SW11	Bigdata Analytics				
4		**Program Elective-IV**				
	M18SW12	Software Security Engineering	3	0	0	3
	M18SW13	Business Process Management				
	M18CN12	Cyber Security				
5	M18AC02	**Audit Course-II** Stress Management	2	0	0	0
6	M18SW14	Software Testing Lab	0	0	4	2
7	M18SW15	Bigdata Analytics Lab	0	0	4	2
8	M18SW16	Mini Project	0	0	2	2
		Total Credits	**14**	**0**	**10**	**18**

III-SEMESTER

S.No	Course Codes	Title of the Course	L	T	P	C
1		**Program Elective-V**				
2	M18SW17	Information Retrieval Systems	3	0	0	3
	M18SW18	Principles of Information Security				
	M18SW19	Computer Foreinsics				
		Open Elective				
3	M18MA01	Advanced Optimization Techniques	3	0	0	3
	M18SE27	Waste Management				
	M18VL07	Embedded System Design				
4	M18SW20	Dissertation Phase-I	0	0	20	10
		Total Credits	**6**	**0**	**20**	**16**

IV-SEMESTER

S.No	Course Codes	Title of the Course	L	T	P	C
1	M18SW21	Dissertation Phase-II	0	0	32	16
		Total Credits	**0**	**0**	**32**	**16**
		Grand Total				**68**

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18CS01) DATA STRUCTURES AND ALGORITHMS
(Core Course-I)

M.Tech:I-Semester **L/T/P C**
3/0/- 3

Objectives: The fundamental design, analysis, and implementation of basic data structures. Basic concepts in the specification and analysis of programs. Principles for good program design, especially the uses of data abstraction. Significance of algorithms in the computer field Various aspects of algorithm development Qualities of a good solution

UNIT I

Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big Oh, Omega and Theta notations, Complexity Analysis Examples. Data structures-Linear and non linear data structures, ADT concept, Linear List ADT, Array representation, Linked representation, Vector representation, singly linked lists -insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, circular lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

UNIT II

Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack, implementation of recursion, Circular queue-insertion and deletion, Dequeue ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap, java.util package-ArrayList, Linked List, Vector classes, Stacks and Queues in java.util, Iterators in java.util.

UNIT III

Searching–Linear and binary search methods, Hashing-Hash functions, Collision Resolution methods-Open Addressing, Chaining, Hashing in java.util-HashMap, HashSet, Hashtable. Sorting –Bubble sort, Insertion sort, Quick sort, Merge sort, Heap sort, Radix sort, comparison of sorting methods.

UNIT IV

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree ADT, representations, recursive and non recursive traversals, Java code for traversals, Threaded binary trees. Graphs- Graphs terminology, Graph ADT, representations, graph traversals/search methods DFS and DFS, Java code for graph traversals, Applications of Graphs-Minimum cost spanning tree using Kruskal’s algorithm, Dijkstra’s algorithm for Single Source Shortest Path
Problem.

UNIT V

Search trees- Binary search tree-Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees-Definition and examples only, Red Black trees –Definition and examples only, B-Trees-definition, insertion and searching operations, Trees in java.util- TreeSet, Tree Map Classes, Tries(examples only),Comparison of Search trees. Text compression-Huffman coding and decoding, Pattern matching-KMP algorithm.

Course Outcomes:

After the completion of this course the student should be able to

1. Understand the basics of Algorithms and Analyze the performance and complexity of Algorithms
2. Explain the concepts of basic data structures: Linear and Non Linear and compare how the storage and retrieval of data is done on these data structures.
3. Gain knowledge about applications of data structures including creating, inserting, deleting, searching and sorting of data for each data structure.
4. Experiment with using linear data structures like stacks, queues and linked list for real time applications.
5. Distinguish between Trees and Graphs and the areas where best applicable.
6. Be able to decide an appropriate data structure for any specific problem.

TEXT BOOKS:

1. Data structures, Algorithms and Applications in Java, S.Sahni, Universities Press.
2. Data structures and Algorithms in Java, Adam Drozdek, 3rd edition, Cengage Learning.
3. Data structures and Algorithm Analysis in Java, M.A.Weiss, 2nd edition, Addison-Wesley (Pearson Education).

REFERENCE BOOKS:

1. Java for Programmers, Deitel and Deitel, Pearson education.
2. Data structures and Algorithms in Java, R.Lafore, Pearson education.
3. Java: The Complete Reference, 8th editon, Herbert Schildt, TMH.
4. Data structures and Algorithms in Java, M.T.Goodrich, R.Tomassia, 3rd edition,Wiley India Edition.
5. Data structures and the Java Collection Frame work,W.J.Collins, Mc Graw Hill.
6. Classic Data structures in Java, T.Budd, Addison-Wesley (Pearson Education).
7. Data structures with Java, Ford and Topp, Pearson Education.
8. Data structures using Java, D.S.Malik and P.S.Nair, Cengage learning.
9. Data structures with Java, J.R.Hubbard and A.Huray, PHI Pvt. Ltd.
10. Data structures and Software Development in an Object-Oriented Domain, J.P.Tremblay and G.A.Cheston, Java edition, Pearson Education.

*****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW01) SOFTWARE DEVELOPMENT METHODOLOGIES

(Core Course-II)

M.Tech: I-Semester **L/T/P C**
3/0/- 3

Objectives:

- The course aims to provide a broad and critical understanding of all the processes for engineering high quality software and the principles, concepts and techniques associated with software development. It shall also enhance ability to analyze and evaluate problems and draw on the theoretical and technical knowledge to develop solutions and systems.
- This inculcates a range of skills focused on the analysis of requirements, design and implementation of reliable and maintainable software, with strong emphasis on engineering principles applied over the whole development lifecycle.
- Opens the horizon on an awareness of current research in software development, the analytical skills and research techniques for their critical and independent evaluation and their application to new problems.

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

Course Outcomes

CO-1: Review the basics of software engineering, processes, models and practices.

CO-2: Understand software requirement engineering and its application using various models.

CO-3: Understand design thinking at varied levels i.e architectural and component level and to also user interface

CO-4: Understand testing and its theoretical background along with metrics to test source code, applications and maintenance of application

CO-5: Develop understand on risks, risk identification, risk projection, Risk refinement, risk management and dealing with change management, survey few tools for configuration management.

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
2. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3. Fundamentals of Software Engineering, Rajib Mall,PHI, 2005
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition, 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8. Software Engineering 3: Domains, Requirements and Software Design, D.Bjorner, Springer, International Edition.
9. Software Engineering Principles and Practice, Hans Van Vliet,3rd edition, Wiley India edition.

10. Introduction to Software Engineering, R.J.Leach,CRC Press.
11. Software Engineering Fundamentals, Ali Behforooz and Frederick J.Hudson, Oxford University Press, rp2009
12. Software Engineering Handbook, Jessica Keyes, Auerbach, 2003.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18CS05) CLOUD COMPUTING
(PROGRAM ELECTIVE-I)

M.Tech: I-Semester **L/T/P C**
3/0/- 3

Objectives:

- To learn the new computing model this enables shared resources on demand over the network.
- To learn about the pay-per-use scenarios.
- To learn about the new kind of service models and deployment models.
- To learn about the virtualization technology.
- To learn the python programming or various services and models.
- To develop cloud applications in Python

UNIT-I

Principles of Parallel and Distributed Computing, Introduction to cloud computing, Cloud computing Architecture, cloud concepts and technologies, cloud services and platforms, Cloud models, cloud as a service, cloud solutions, cloud offerings, introduction to Hadoop and Map reduce.

UNIT-II

Cloud Platforms for Industry, Healthcare and education, Cloud Platforms in the Industry, cloud applications. Virtualization, cloud virtualization technology, deep dive: cloud virtualization, Migrating in to cloud computing, Virtual Machines Provisioning and Virtual Machine Migration Services, On the Management of Virtual Machines for cloud Infrastructure, Comet cloud, T-Systems,

UNIT-III

Cloud computing Applications: Industry, Health, Education, Scientific Applications, Business and Consumer Applications, Understanding Scientific Applications for Cloud Environments, Impact of Cloud computing on the role of corporate IT. Enterprise cloud computing Paradigm, Federated cloud computing Architecture, SLA Management in Cloud Computing, Developing the cloud: cloud application Design.

UNIT-IV

Python Basics, Python for cloud, cloud application development in python, Cloud Application Development in Python. Programming Google App Engine with Python: A first real cloud Application, Managing Data in the cloud, Google app engine Services for Login Authentication, Optimizing UI and Logic, Making the UI Pretty: Templates and CSS, Getting Interactive. Map Reduce Programming Model and Implementations.

UNIT-V

Cloud management, Organizational Readiness and change management in the cloud age ,Cloud Security, Data security in the cloud, Legal Issues in the Cloud , Achieving Production Readiness for the cloud Services

Course Outcomes:

After the completion of this course the students should be able to :

1. Discuss main concepts, key strengths, and limitations for cloud computing.
2. Develop the architecture along with specific infrastructure on cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Explain the issues on cloud computing along with security, privacy, and interoperability.
4. Choose and use the appropriate technology, methods on these issues.
5. Identify problems, and explain, analyze, and evaluate various cloud computing solutions.

6. Provide the appropriate solutions on cloud computing based on the application.

TEXT BOOKS:

1. Cloud Computing: Raj Kumar Buyya , James Broberg, andrzej Goscinski, 2013 Wiley
2. Mastering Cloud Computing: Raj Kumar buyya, Christian Vecchiola,selvi-2013.
3. Cloud Computing: Arshdeep Bahga, Vijay Madisetti, 2014, University Press.
4. Cloud computing: Dr Kumar Saurab Wiley India 2011.

REFERENCES:

1. Code in the Cloud: Mark C.Chu-Carroll 2011, SPD.(Second part of IV UNIT)
2. Essentials of cloud computing : K Chandrasekharan CRC Press.
3. Cloud Computing: John W. Rittinghouse, James Ransome, CRC Press.
4. Virtualization Security: Dave shackleford 2013. SYBEX a wiley Brand.
5. Cloud computing and Software Services: Ahson , Ilyas.2011.
6. Cloud Computing Bible: Sosinsky 2012. Wiley India .
7. Cloud Computing: Dan C. Marinescu-2013, Morgan Kaufmann.
8. Distributed and Cloud Computing, Kai Hwang, Geoffery C.Fox, Jack J.Dongarra, Elsevier, 2012.
9. Fundamentals of Python Kenneth A.Lambert, B.L.Juneja

*****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW02) COMPONENT BASED SOFTWARE ENGINEERING
(PROGRAM ELECTIVE-I)

M.Tech: I-Semester	**L/T/P C** **3/0/- 3**

Objectives:

course provides knowledge on the essentials of component-based software engineering main characteristics of components and component models. This course creates awareness on software development processes for component-based systems and also to understand relations between software architecture and component models.

UNIT I

Component definition - Definition of a Software Component and its elements, The Component Industry Metaphor, Component Models and Component Services, An example specification for implementing a temperature regulator Software Component. The Case for Components- The Business Case for components, COTS Myths and Other Lessons Learned in Component-Based Software Development.

UNIT II

Planning Team Roles for CBD, Common High-Risk Mistakes, CBSE Success Factors: Integrating Architecture, Process, and Organization. Software Engineering Practices - Practices of Software Engineering, From Subroutines to Subsystems: Component-Based Software Development, Status of CBSE in Europe.

UNIT III

The Design of Software Component Infrastructures - Software Components and the UML, Component Infrastructures, Business Components, Components and Connectors, An OPEN process for CBD, Designing Models of Modularity and Integration. Software Architecture, Software Architecture Design Principles, Product-Line Architectures.

UNIT IV

The Management of Component-Based Software Systems - Measurement and Metrics for Software Components, Implementing a Practical Reuse Program for Software Components, Selecting the Right COTS Software, Building instead of Buying, Software Component Project Management, The Trouble with Testing Components, Configuration Management and Component Libraries, The Evolution, Maintenance, and Management of CBS.

UNIT V

Component Technologies - Overview of the CORBA Component Model, Overview of COM+, Overview of the EJB Component Model, Bonobo and Free Software GNOME Components, Choosing between COM+, EJB, and CCM, Software Agents as Next Generation Software Components.

COURSE OUTCOMES:

CO-1: Understand component based software development, models and approaches

CO-2: Demonstrate the role of team in building component based software development.

CO-3: identify the processes involved in Design of Software Component Infrastructures and study existing models

CO-4: Demonstrate the learnt principles in effective reuse and maintenance of software

CO-5: Survey technologies that support implementation of component based software development

TEXT BOOKS:

1. Component - Based Software Engineering, G.T. Heineman and W.T. Councill, Addison- Wesley, Pearson Education.

REFERENCES:

1. Component Software, C.Szyperski, D.Gruntz and S.Murer, Pearson Education.
2. Software Engineering, Roger S. Pressman, 6th edition, Tata McGraw-Hill.
3. Software Engineering, Ian Sommerville, seventh edition, Pearson education, 2004.
4. Software Engineering Principles and Practice, Hans Van Vliet, 3rd edition, Wiley India.

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW03) INTERNET TECHNOLOGIES AND SERVICES
(PROGRAM ELECTIVE-I)

M.Tech: I-Semester	**L/T/P C** **3/0/- 3**

Objective:

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

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

Service Oriented Architecture and Web Services: Overview of Service Oriented Architecture – SOA concepts, Key Service Characteristics, Technical Benefits of a SOA Introduction to Web Services– The

definition of web services, basic operational model of web services, basic steps of implementing web services. Core fundamentals of SOAP – SOAP Message Structure, SOAP encoding, SOAP message exchange models, Describing Web Services –Web Services life cycle, anatomy of WSDL Introduction to Axis– Installing axis web service framework, deploying a java web service on axis. Web Services Interoperability – Creating java and .Net client applications for an Axis Web Service
(Note: The Reference Platform for the course will be open source products Apache Tomcat Application Server, MySQL database, Hibernate and Axis)

COURSE OUTCOMES:

CO-1: survey client side technologies for web development.

CO-2: Understand life cycle of a java servlet and apply it to a develop software

CO-3: develop understanding on JSP and enhance the solution using JSP program.

CO-4: Create awareness on Struts framework and its application, develop complex solution using this framework.

CO-5: introduce web services and service oriented architecture to develop seamless applications that are portable and highly interoperable.

TEXT BOOKS:

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

REFERENCES:

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

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW04) SOFTWARE REQUIREMENTS AND ESTIMATION
(PROGRAM ELECTIVE-II)

M.Tech: I-Semester	**L/T/P C** **3/0/- 3**

Course Objectives:

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

UNIT I

Software Requirements: What and Why Essential Software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management **Software Requirements Engineering** Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality.

UNIT II

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

UNIT III

Software Estimation Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation

Size Estimation Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, Conversion between size measures.

UNIT IV

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

UNIT V

Tools for Requirements Management and Estimation Requirements Management Tools: Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation, **Software Estimation Tools:** Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools

COURSE OUTCOMES:

CO-1: To develop an understanding of software requirements and asses their nature.
CO-2: To analyze software requirement management.
CO-3: To be able to estimate the cost of software development by understanding various methods.
CO-4: To be able to draw conclusions on effort, schedule and cost estimation
CO-5: Survey tools for requirements management, software estimation tools

TEXT BOOK:

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

REFERENCES:

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

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW05) OBJECT ORIENTED SOFTWARE ENGINEERING
(PROGRAM ELECTIVE-II)

M.Tech: I-Semester **L/T/P C**
3/0/- 3

Course Objectives:

Introduce object-oriented life-cycle models, object-oriented analysis, object oriented design, management implications of the object-oriented paradigm, and the testing and maintenance of object-oriented software and post delivery maintenance

UNIT I
The Scope of Object-Oriented Software Engineering, Software Life-Cycle Models, Software Process
UNIT II :
Teams, Tools for the trade, testing.
UNIT III:
From Modules to Objects, Reusability and Portability
UNIT IV:
Requirement Workflow, The Analysis Workflow
UNIT V:
Design Workflow, Implementation Workflow, Post delivery Maintenance

Course Outcomes:
CO-1: To understand Scope of Object-Oriented Software Engineering, Software Life-Cycle Models, Software Process.
CO-2: To analyze role of teams, tools for the trade, testing.
CO-3: To be able to create reusable and portable applications.
CO-4: To be able to draw conclusions from requirement workflow.
CO-5: Design and implement workflow and maintain post delivery.

TEXT BOOK:

- Object-Oriented Software Engineering Stephen R. Schach Vanderbilt University,the McGraw-Hill companies References
- Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition Bernd Bruegge & Allen H. Dutoit ,Prentice Hall

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW06) INFORMATION THEORY AND CODING
(PROGRAM ELECTIVE-II)

M.Tech: I-Semester **L/T/P C**
3/0/- 3

Objectives:

- In this students will acquire knowledge about information and entropy
- In this students will acquire knowledge about Hamming weight, minimum distance decoding and different types of codes. They also learn about syndrome calculation and design of an encoder and decoder;
- In this students will gain knowledge about convolution coding. They also learn about sequential search and Viterbi algorithm
- In this Unit students gain knowledge about text compression techniques. They also learn about speech and audio coding.
- In this students know about, image compression, graphics interchange format, JPEG and MPEG standards.

UNIT-I:

INFORMATION THEORY :Information – Entropy, Information rate, classification of codes, Kraft McMillan inequality, Source coding theorem, Shannon-Fano coding, Huffman coding, Extended Huffman coding - Joint and conditional entropies, Mutual information - Discrete memoryless channels – BSC, BEC – Channel capacity, Shannon limit.

UNIT-II:

ERROR CONTROL CODING: BLOCK CODES: Definitions and Principles: Hamming weight, Hamming distance, Minimum distance decoding - Single parity codes, Hamming codes, Repetition codes - Linear block codes, Cyclic codes - Syndrome calculation, Encoder and decoder - CRC

UNIT-III:

ERROR CONTROL CODING: CONVOLUTIONAL CODES: Convolutional codes – code tree, trellis, state diagram - Encoding – Decoding: Sequential search and Viterbi algorithm – Principle of Turbo coding

UNIT-IV:

SOURCE CODING: TEXT, AUDIO AND SPEECH Text: Adaptive Huffman Coding, Arithmetic Coding, LZW algorithm – Audio: Perceptual coding, Masking techniques, Psychoacoustic model, MEG Audio layers I,II,III, Dolby AC3 - Speech: Channel Vocoder, Linear Predictive Coding

UNIT-V:

SOURCE CODING: IMAGE AND VIDEO : Image and Video Formats – GIF, TIFF, SIF, CIF, QCIF – Image compression: READ, JPEG – Video Compression: Principles-I,B,P frames, Motion estimation, Motion compensation, H.261, MPEG standard

Course Outcomes:

CO-1: Ability to learn about information and entropy

CO-2: Ability to learn about Hamming weight, minimum distance decoding and different types of codes. They also learn about syndrome calculation and design of an encoder and decoder;

CO-3 understanding the sequential search and Viterbi algorithm

CO-4 Apply knowledge on text compression techniques. They also learn about speech and audio coding.

CO-5apply knowledge on image compression, graphics interchange format, JPEG and MPEG standards.

TEXT BOOKS:

1.R Bose, “Information Theory, Coding and Cryptography”, TMH 2007

2.Fred Halsall, “Multidedia Communications: Applications, Networks, Protocols and Standards”, Perason Education Asia, 2002.

REFERENCE BOOKS:

1. K Sayood, “Introduction to Data Compression” 3/e, Elsevier 2006
2. S Gravano, “Introduction to Error Control Codes”, Oxford University Press 2007
3. Amitabha Bhattacharya, “Digital Communication”, TMH 2006

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18MC01) RESEARCH METHODOLOGY
(AUDIT COURSE)

M.Tech: I-Semester **L/T/P C**
2/0/- 2

Course Objectives:

- To develop an understanding of IPR/ research methodology in the process of creation of patents through research.
- To develop further research capabilities.
- To learn better report writing skills and Patenting.

UNIT-I:

RESEARCH METHODOLOGY: Objectives and Motivation of Research, Significance of Literature review, Types of Research, Research Approaches, and Research Methods verses Methodology, Research and Scientific Method, Importance of Research Methodology, Research Process, Criteria of Good Research.

UNIT-II:

RESEARCH DESIGN: Meaning of Research Design, Need of Research Design, Feature of a Good Design Important Concepts Related to Research Design, Different Research Designs, Basic Principles of Experimental Design, Data collection methods, Collection of primary data, Secondary data, Data organization, Methods of data grouping, Diagrammatic representation of data, Graphic representation of data.

UNIT-III:

RESEARCH REPORT WRITING: Format of the Research report, Synopsis, Dissertation, References/Bibliography/ Webliography, Research Proposal Preparation: Writing a Research Proposal and Research Report, Writing Research Grant Proposal.

UNIT-IV:

NATURE OF INTELLECTUAL PROPERTY: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development.

UNIT-V:

PATENT RIGHTS: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. New Developments in IPR: Administration of Patent System.

Course Outcomes:

After the completion of this course the student should be able to

1. Acquire knowledge on Research Design and statistical methods in research.
2. Analyze the various methods in Data Collection, Data Organization and different approaches of Data Representation.
3. Understand all the basic concepts required to prepare
    a. Research synopsis
    b. Dissertation
    c. Writing a good research proposal
4. Interpret the Scope of Patent Rights and Administration of Patent System.

TEXT BOOKS:

1. C.R Kothari, “Research Methodology, Methods & Technique”.New Age International Publishers, 2004.
2. R. Ganesan, “Research Methodology for Engineers”, MJP Publishers, 2011.
3. Robert P. Merges, Peter S. Menell, Mark A. Lemley, “Intellectual Property in New Technological Age”, Aspen Publishers, 2016.
4. T. Ramappa, “Intellectual Property Rights Under WTO”, S. Chand, 2008.
5. Satarkar, S.V,.“Intellectual property rights and copy right”. ESS Publications, 2000.

REFERENCES:

1. Ranjit Kumar, “Research Methodology: A Step by Step Guide for beginners”, SAGE Publications Ltd.
2. Halbert, “Resisting Intellectual Property”, Taylor & Francis Ltd ,2007

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
(M18AC01) ENGLISH FOR RESEARCH PAPER WRITING
(AUDIT COURSE)

M.Tech: I-Semester **L/T/P C**
2/0/- 2

Course Objectives:

- To understand the nuances of language and vocabulary in writing a Research Paper.
- To develop the content, structure and format of writing a research paper.
- To give the practice of writinga Research Paper.
- To enable the students to evolve original research papers without subjected to plagiarism.

UNIT-I:
ACADEMIC WRITING: What is Research? - Meaning & Definition of a research paper– Purpose of a research paper – Scope – Benefits – Limitations – outcomes.

UNIT-II:
RESEARCH FORMAT: Title – Abstract – Introduction – Discussion - Findings – Conclusion – Style of Indentation – Font size/Font types – Indexing – Citation of sources.

UNIT-III:
RESEARCH METHODOLOGY: Methods (Qualitative – Quantitative) – Literature Review – Who did what – Criticizing, Paraphrasing & Plagiarism.

UNIT-IV:
PROCESS OF WRITING A RESEARCH PAPER: Choosing a topic - Thesis Statement – Outline – Organizing notes - Language of Research – Word order, Paragraphs – Writing first draft – Revising/Editing - Typing the final draft

UNIT-V:
HOW TO & WHERE TO GET PUBLISHED: Reputed Journals – National/International – ISSN No, No. of volumes, Scopes Index/UGC Journals – Freepublications - Paid Journal publications – /Advantages/Benefits

Course Outcomes:
After the completion of this course the student should be able to

1. Obtain complete knowledge on Definition of a research paper, Purpose of writing any research paper , its Scope and Benefits.
2. Understand the standard English formats .for scripting the best research paper.
3. Analyze all the Qualitative and Quantitative Research Methodologies and the ethics of plagiarism.
4. Explain the detailed process of writing and publishing any research paper and perform a case study on paper writing.

TEXT BOOKS:

1. MLA Hand book for writers of Research Papers, East West Press Pvt. Ltd, New Delhi, 7th Edition.
2. C. R Kothari, Gaurav, Garg, Research Methodology Methods and Techniques, New Age International Publishers. 4th Edition.
3. Lauri Rozakis, Schaum's Quick Guide to Writing Great Research Papers, Tata McGraw Hills Pvt. Ltd, New Delhi.
4. N. Gurumani, Scientific Thesis Writing and Paper Presentation, MJP Publishers

REFERENCES:

1. NPTEL: https://onlinecourses.nptel.ac.in/noc18_mg13/preview

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW07) SOFTWARE DEVELOPMENT METHODOLOGIES Lab
(LABORATORY-I)

M.Tech: I-Semester **L/T/P/ C**
-/- / 4/ 2

Objectives:

- To understand the software engineering methodologies involved in the phases for project development.
- To gain knowledge about open source tools used for implementing software engineering methods.
- To exercise developing product-startups implementing software engineering methods. Open source Tools: StarUML / UMLGraph / Topcased .
- Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology for at least two real time scenarios or for the sample experiments.

Problem Analysis and Project Planning –

- Thorough study of the problem – Identify Project scope, Objectives and Infrastructure.
- Software Requirement Analysis – Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements.
- Data Modeling – Use work products – data dictionary.
- Software Designing - Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.
- Prototype model – Develop the prototype of the product

LIST OF SAMPLE PROBLEMS/EXPERIMENTS:

Week-1:

Student Enrolment System: A University has contracted you to develop their new student records system. The normal tasks that the system performs are as follows: **Enroll a student at the university:** A student provides his or her personal details (name, address, sex, date of birth), along with the code of the course (e.g. Bachelor of Computer Science) in which he or she wishes to enroll. A student record is created, and a unique student ID number is assigned to the student. The system automatically enrolls the student in any core first-year subjects for the course.

Enroll a student in a subject: A student provides his or her student ID number and the subject code of the subject in which he or she wish to enroll. The system checks that the subject requested by the student is allowed for the course in which the student is enrolled. If not, the enrolment request is rejected. The system checks what subjects (if any) are specified as prerequisites for the subject in which the student wishes to enroll. If the student has passed all the prerequisite subjects, he or she is enrolled in the desired subject. Otherwise, the enrolment request is rejected.

Record a mark for a student: A staff member accesses the system by giving a subject code and a password for that subject. If the password is correct, the system displays the list of students enrolled in the subject to the staff member. The staff member can then specify a mark for any student on the list.

Create a new subject: An administrator accesses the system using a password. The administrator then chooses a subject code for the new subject. The system checks that this code is not already in use in the system, and if not, creates a new subject record. The administrator then gives the subject name, the course to which it belongs, the year of the course in which it may first be taken, a flag indicating whether or not it is a core subject and the codes of any prerequisite subjects.

Print a transcript of a student's results: An administrator accesses the system using a password. The administrator then gives the student ID number of the student for whom the transcript is to be generated. The system contacts the finance system to check whether or not the student has paid all fees. If fees have been paid, the system creates a transcript showing all the subjects in which the student has been enrolled in each year, and the mark for that subject. The header of the transcript shows the student's personal details and the course in which he or she is enrolled.

Assign a staff member to a subject: An administrator accesses the system using a password. The administrator then gives the subject code for the subject to which the staff member is to be assigned, and the staff ID number of the staff member.

Week-2:

A major book retailer is planning to develop a computer system to handle their new online bookshop: Booky.com. You have been chosen to do the analysis and design.

The following requirements have been identified: Customers can search for books on the Booky.com website, either by author name, or words in the title. A list of all matching books is returned to the customer. A customer does not need to be logged-in in order to search. The system records all the customers of the Booky.com who have ever logged in. A customer may be an individual customer or a business customer. Each customer has a username and password. Business customers may have several usernames and passwords, corresponding to different divisions within the business. When a customer has selected a book to buy at the Booky.com website. The system prompts for the customer's username and password. The customer enters these details. The system verifies the customer's identity and retrieves the customer's name and address, then prompts for credit card details. The customer enters these details. The system checks the credit card details. The system shows the customer the book and delivery price. The customer confirms the transaction.

The system records all books available at Booky.com. For each book, the author, title and ISBN number are recorded. The number of each book in stock is also stored, along with the number on order by customers and the number on order from publishers. Books may be temporarily unavailable. All books are stored in the Booky.com warehouse. The warehouse can be contacted via a secure internet connection. For each customer, a permanent record of books bought by that customer is maintained. Likewise, for each book, a record of customers who have bought that book is kept. A customer order consists of one or more order lines, each corresponding to a particular book. A customer may choose to defer the shipment of an order until all the order lines have been filled. When the warehouse fills all or part of customer order, an email is sent to the customer informing them of what has been shipped.

If a book ordered by a customer turns out to be unavailable, the corresponding order line is flagged and an email is sent to the customer informing them of the problem. At this stage the customer can cancel this order line. When a book corresponding to a previously-unavailable order line becomes available, an email is sent to the customer and a copy of the book is held for seven days, after which it is returned to normal stock if the customer has not confirmed the order.

The shop keeps track of which publishers produce particular book titles. Some books may be available from more than one publisher. Although Booky.com will initially sell only books, it is envisaged that in future it will offer further products, such as CDs. The list of possible future products has not yet been finalized.

Week-3:

A **course management system** (CMS) is a collection of software tools providing an online environment for course interactions. A CMS typically includes a variety of online tools and environments, such as:

- An area for faculty posting of class materials such as course syllabus and handouts
- An area for student posting of papers and other assignments
- A gradebook where faculty can record grades and each student can view his or her grades
- An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class
- A chat tool allowing synchronous communication among class participants
- A threaded discussion board allowing asynchronous communication among participants

In addition, a CMS is typically integrated with other databases in the university so that students enrolled in a particular course are automatically registered in the CMS as participants in that course. The Course Management System (CMS) is a web application for department personnel, Academic Senate, and Registrar staff to view, enter, and manage course information formerly submitted via paper. Departments can use CMS to create new course proposals, submit changes for existing courses, and track the progress of proposals as they move through the stages of online approval.

Week-4:

This project is aimed at developing a web based Leave Management Tool, which is of importance to either an organization or a college.

The Easy Leave is an Intranet based application that can be accessed throughout the organization or a specified group/Dept. This system can be used to automate the workflow of leave applications and their approvals. The periodic crediting of leave is also automated. There are features like notifications, cancellation of leave, automatic approval of leave, report generators etc in this Tool.

Functional components of the project:

There are registered people in the system. Some are approvers. An approver can also be a requestor. In an organization, the hierarchy could be Engineers/Managers/Business Managers/Managing Director etc. In a college, it could be Lecturer/Professor/Head of the Department/Dean/Principal etc.

Following is a list of functionalities of the system:

A person should be able to o login to the system through the first page of the application

a. change the password after logging into the system
b. see his/her eligibility details (like how many days of leave he/she is eligible for etc)
c. query the leave balance
d. see his/her leave history since the time he/she joined the company/college
e. apply for leave, specifying the from and to dates, reason for taking leave, address for communication while on leave and his/her superior's email id

f. see his/her current leave applications and the leave applications that are submitted to him/her for approval or cancellation
g. approve/reject the leave applications that are submitted to him/her
h. withdraw his/her leave application (which has not been approved yet)
i. Cancel his/her leave (which has been already approved). This will need to be approved by his/her Superior
j. get help about the leave system on how to use the different features of the system

2. As soon as a leave application /cancellation request /withdrawal /approval /rejection /password-change is made by the person, an automatic email should be sent to the person and his superior giving details about the action
3. The number of days of leave (as per the assumed leave policy) should be automatically credited to everybody and a notification regarding the same be sent to them automatically

An automatic leave-approval facility for leave applications which are older than 2 weeks should be there. Notification about the automatic leave approval should be sent to the person as well as his superior

Week-5:

Auctions are among the latest economic institutions in place. They have been used since antiquity to sell a wide variety of goods, and their basic form has remained unchanged. In this dissertation, we explore the efficiency of common auctions when values are interdependent- the value to a particular bidder may depend on information available only to others-and asymmetric. In this setting, it is well known that sealed-bid auctions do not achieve efficient allocations in general since they do not allow the information held by different bidders to be shared. Typically, in an auction, say of the kind used to sell art, the auctioneer sets a relatively low initial price. This price is then increased until only one bidder is willing to buy the object, and the exact manner in which this is done varies. In my model a bidder who drops out at some price can "reenter" at a higher price.

With the invention of E-commerce technologies over the Internet the opportunity to bid from the comfort of ones own home has seen a change like never seen before. Within the span of a few short years, what may have began as an experimental idea has grown to an immensely popular hobby, and in some cases, a means of livelihood, the Auction Patrol gathers tremendous response every day, all day. With the point and click of the mouse, one may bid on an item they may need or just want, and in moments they find that either they are the top bidder or someone else wants it more, and you're outbid! The excitement of an auction all from the comfort of home is a completely different experience.

Society cannot seem to escape the criminal element in the physical world, and so it is the same with Auction Patrols. This is one area wherein a question can be raised as to how safe Auction Patrols.

Proposed system

1. To generate the quick reports
2. To make accuracy and efficient calculations
3. To provide proper information briefly
4. To provide data security
5. To provide huge maintenance of records
6. Flexibility of transactions can be completed in time

Week-6:

This project is mainly developed for the Account Division of a Banking sector to provide better interface of the entire banking transactions. This system is aimed to give a better out look to the user interfaces and to implement all the banking transactions like:

Supply of Account Information, New Account Creations, Deposits, Withdraws, Cheque book issues, Stop payments, Transfer of accounts, Report Generations.

Proposed System:

- The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.
- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- Readily upload the latest updates, allows user to download the alerts by clicking the URL. There is no risk of data mismanagement at any level while the project development is under process. It provides high level of security with different level of authentication.

Week-7:

Verification and Validation is a part of S/W Quality Assurance. Verification refers to the set of activities that ensure correctly implements a specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

Verification:" Are we building the product right", Validation: "Are we building the right product"

The project entitled Independent Project Metrics is an effort, to develop a tool to manage the Verification and Validation process.

The specific purpose of the Independent Verification and Validation Process o f Project Metrics Tool is to bring out the various Verification and validation tasks to be performed. The scope of the Project Metrics is to cover the developed for system.

The goals of the V&V effort is to ensure that the software and the documents are developed are of high quality as expected from any mission critical software. This project generates the plan for Verification and validation process. This project maintain the document names, source code module names, version number, released date, receiving date size of document and source code modules of receiving projects for Verification and validation.

Using this application we assign the tasks/activities to different persons and also calculate the expected efforts and actual efforts. The V&V co-coordinator does this work.

Proposed System:

The general description gives an "executive overview" and is very client-oriented. It expounds on the functional and data requirements of the application. It also lists the limitations, assumptions and dependencies of the application. It also touches on the performance and quality requirements of the application and provides a solid definition of the interface. The computerization of this system would avoid the wrong interpretation and bad calculation of data .The system help the user to see any documents, source code, tasks, activities, team information with details at the click of a button. The record data is maintained and backed up such a way that data is not loss. The speed of the system could also increased

Week-8:

ERP is a powerful human resource tool for maintaining employee and company information. More than a data storage program, ERP helps you manage your employees. ERP offers a wide variety of reports that give you exactly the information you need. View payroll information by department, or find everyone who is receiving company

Module Description: *Payroll, Employee, Employee pay slip, Selection process, Reports, Mailing System Training, Add Company Information*

PROPOSED SYSTEM

The proposed system is designed to eliminate all the drawbacks of the existing system. The system is part of a large HRMS Application and shall be responsible for maintaining information about employees, positions, company benefits, departments, new recruit checklists, employee achievements, warnings, evaluation reports, education & training, administration, work changes and several ad hoc reports.

The major advantage of the proposed system is,

- It's online, so that information is available anytime.
- High integrity and security.
- Ability to incorporate newly available data.
- It is user friendly
- Speed and accuracy is increased
- Fully automated.
- Security is associated with user authentication
- Duplication of information is curbed

Course Outcomes:

CO-1: Review the basics of software engineering, processes, models and practices.

CO-2: understand software requirement engineering and its application using various models.

CO-3: understand design thinking at varied levels i.e architectural and component level and to also user interface

CO-4: understand testing and its theoretical background along with metrics to test source code, applications and maintenance of application

CO-5: develop an understand on risks, risk identification, risk projection, Risk refinement, risk management and dealing with change management, survey few tools for configuration management

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18CS10) Cloud Computing Lab
(LABORATORY-II)

M.Tech: I-Semester **L/T/P/ C**
-/- / 4/ 2

Experiments:

Week 1: Introduction to basic cloud computing concepts.

Week 2: Draw a neat diagram of cloud computing Architecture.

Week 3:
Find procedure to run the virtual machine of different configuration. Check how many virtual machines can be utilized at particular time.

Week 4: Find procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine.

Week 5: Install a C compiler in the virtual machine and execute a sample program.

Week 6: Show the virtual machine migration based on the certain condition from one node to the other.
Week 7: Find procedure to install storage controller and interact with it.

Week 8: Installation and Configuration of Hadoop.

Week 9: Find procedure to set up the one node Hadoop cluster.

Week 10: Mount the one node Hadoop cluster using FUSE.

Week 11: Write a program to use the API's of Hadoop to interact with it.

Week 12: Using Hadoop for counting word frequency with map reduce.

Week 13: Write a word count program to demonstrate the use of Map and Reduce tasks

Week 14: Installation & Configuration of Oracle Virtual box for windows xp & android.

Week 15: Installing open Solaris as a guest OS to Sun xVM Virtual Box using the 7-Zip archive tool

Week 16: Evaluation of performance of services over cloud: Google App &Amazon web services.
Course Outcomes:
After the completion of this course the students should be able to :

1. Develop the architecture along with specific infrastructure on cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
2. Explain the issues on cloud computing along with security, privacy, and interoperability.
3. Identify problems, and explain, analyze, and evaluate various cloud computing solutions.
4. Provide the appropriate solutions on cloud computing based on the application.

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW08) SOFTWARE QUALITY ASSURANCE AND TESTING

(CORE COURSE-IV)

M.Tech: II Semester **L/T/P C**
3/0/- 3

Objectives:
The student should be able to:

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

UNIT-I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

Course Outcomes:

1. Apply modern software testing processes in relation to software development and project management.
2. Create test strategies and plans, design test cases, prioritize and execute them.
3. Ability to learn and manage incidents using software testing tools.
4. Contribute to efficient delivery of software solutions and implement improvements in the software development processes.
5. To gain expertise in designing, implementation and development of computer based systems and IT processes.

TEXT BOOKS:

1. Effective Methods for Software Testing, Third edition, *William E. Perry*, Wiley India, 2009
2. Software Testing – Principles and Practices, *Naresh Chauhan,* Oxford University Press, 2010.
3. Software Quality Assurance – From Theory to Implementation, *Daniel Galin,* Pearson Education, 2009.

REFERENCES:

1. Testing Computer Software, Cem Kaner, Jack Falk, Hung Quoc Nguyen, Wiley India, rp2012.
2. Software Testing – Principles, Techniques and Tools, *M.G.Limaye*, Tata McGraw-Hill, 2009.
3. Software Testing - A Craftsman's approach, *Paul C. Jorgensen*, Third edition, Auerbach Publications, 2010.
4. Foundations of Software Testing, *Aditya P. Mathur*, Pearson Education, 2008.
5. Software Testing and Quality Assurance – Theory and Practice, *Kshirasagar Naik, Priyadashi Tripathy*, Wiley India, 2010.

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18CS18) SOFTWARE PROCESS AND PROJECT MANAGEMENT

(CORE COURSE – V)

M.Tech:II-Semester **L/T/P C**
3/0/- 3

Objectives:

- Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
- Compare and differentiate organization structures and project structures.
- Implement a project to manage project schedule, expenses and resources with the application of suitable project management tools.

UNIT-I

Software Process Maturity Software maturity Framework**,** Principles of Software Process Change, Software Process Assessment, The Initial Process, The Repeatable Process, The Defined Process, The Managed Process, The Optimizing Process.

Process Reference Models Capability Maturity Model (CMM)**,** CMMI, PCMM, PSP, TSP.

UNIT-II

Software Project Management Renaissance Conventional Software Management, Evolution of Software Economics, Improving Software Economics, The old way and the new way.

Life-Cycle Phases and Process artifacts Engineering and Production stages, inception phase, elaboration phase, construction phase, transition phase, artifact sets, management artifacts, engineering artifacts and pragmatic artifacts, model based software architectures.

UNIT-III

Workflows and Checkpoints of process Software process workflows, Iteration workflows, Major milestones, Minor milestones, Periodic status assessments.

Process Planning Work breakdown structures, Planning guidelines, cost and schedule estimating process, iteration planning process, Pragmatic planning.

UNIT-IV

Project Organizations Line-of- business organizations, project organizations, evolution of organizations, process automation.

Project Control and process instrumentation The seven core metrics, management indicators, quality indicators, life-cycle expectations, Pragmatic software metrics, and metrics automation.

UNIT-V

CCPDS-R Case Study and Future Software Project Management Practices Modern Project Profiles, Next-Generation software Economics, Modern Process Transitions.

Course Outcomes:

After the completion of this course the students should be able to :

1. Discuss and plan to execute projects based on required standards.
2. Understand the range of tools used on project management**.**
3. Analyze the concepts related on project governance and methodologies.
4. Apply critical analysis on solving problems and planning process.
5. Describe planning, Risk and issues management.
6. Plan process, pragmatic planning service delivery and quality assurance

TEXT BOOKS:

1. Managing the Software Process, *Watts S. Humphrey*, Pearson Education.
2. Software Project Management, *Walker Royce,* Pearson Education.

REFERENCE BOOKS:

1. Effective Project Management: Traditional, Agile, Extreme, Robert Wysocki, Sixth edition, Wiley India, rp2011.
2. An Introduction to the Team Software Process, Watts S. Humphrey, Pearson Education, 2000
3. Process Improvement essentials, James R. Persse, O'Reilly, 2006
4. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, TMH, 2006
5. Applied Software Project Management, Andrew Stellman & Jennifer Greene, O'Reilly, 2006.
6. Head First PMP, Jennifer Greene & Andrew Stellman, O'Reilly, 2007
7. Software Engineering Project Managent, Richard H. Thayer & Edward Yourdon, 2nd edition, Wiley India, 2004.
8. The Art of Project Management, Scott Berkun, SPD, O'Reilly, 2011.
9. Applied Software Project Management, Andrew Stellman & Jennifer Greene, SPD, O'Reilly, rp2011.
10. Agile Project Management, Jim Highsmith, Pearson education, 2004.

*****

VAADGEVI COLLEGE OF ENGINEERING (AUTONOMOUS)

(M18SW09) SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

(PROGRAM ELECTIVE-III)

M.Tech: III-Semester (CNIS) **L/T/P C**
3/0/- 3

Course Objectives:

- To learn how to add functionality to designs while minimizing complexity
- To learn what design patterns really are, and are not
- To know about specific design patterns.
- To learn how to use design patterns to keep code quality high without over design.

UNIT I

Envisioning Architecture: The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views. **Creating an Architecture:** Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT II

Analyzing Architectures: Architecture Evaluation, Architecture design decision making, ATAM, CBAM. **Moving from one system to many:** Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT III

Patterns: Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage.**Creational and Structural patterns:** Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight.

UNIT IV

Behavioural patterns: Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy. template method, visitor.

UNIT V

Case Studies: A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development,

COURSE OUTCOMES

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

- To understand the concept of patterns and the Catalog.
- To discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
- To understand the variety of implemented bad practices related to the Business and Integration tiers.

- To highlight the evolution of patterns.
- To learn how to add functionality to designs while minimizing complexity

TEXT BOOKS:

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick Kazman, Pearson Education, 2003.
2. Design Patterns, Erich Gamma, Pearson Education.

REFERENCES:

1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O’REILLY, 2007.
5. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
6. J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003.
7. Design Patterns in C#, Steven John metsker, Pearson education, 2004.
8. Pattern Oriented Software Architecture, F.Buschmann &others, John Wiley & Sons.

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW10) AGILE SOFTWARE DEVELOPMENT
(PROGRAM ELECTIVE-III)

M.Tech: II-Semester **L/T/P C**
3/0/- 3

Course Objectives:
To introduce agile software development, and who can a team use agile software development. it also provides knowledge on the agile tools such as Scrum meetings or paired programming, different approaches to creating software using agile software development process.

UNIT I: The History and Value of Agile Software Development , Organizational Culture Considerations with Agile

UNIT II: Understanding the Different Types of Agile, Describing the Different Roles,

UNIT III: The New Way to Collect and Document Requirements

UNIT IV: Grooming and Planning, Testing, Quality, and Integration

Unit V: Tracking and Reporting , Agile beyond IT

Course Outcomes:
After the completion of this course the student should be able to

1. Understand the architecture, creating it and moving from one to any, different structural patterns.
2. Analyze the architecture and build the system from the components.
3. Design creational and structural patterns.
4. Learn about behavioural patterns.
5. Do a case study in utilizing architectural structures

TEXT BOOKS:
Introduction to Agile Methods Sondra Ashmore, Ph.D. Kristin Runyan Addison-Wesley.2015

REFERENCES:

1. Agile Software Development Methods: Review and Analysis Authors: Pekka Abrahamsson, Outi Salo, Jussi Ronkainen and Juhani Warsta
2. http://agilemanifesto.org/

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW11) BIGDATA ANALYTICS

(PROGRAM ELECTIVE-III)

M.Tech: II-Semester **L/T/P C**
3/0/- 3

Course Objective:

- This course brings together several key big data technologies used for storage, analysis and manipulation of data.
- To recognize the key concepts of Hadoop framework, MapReduce, Pig, Hive, and No-SQL.
- To prepare a sample project in Hadoop API.

UNIT I
INTRODUCTION TO BIG DATA

Big Data and its Importance – Four V's of Big Data – Drivers for Big Data –Introduction to Big Data Analytics – Big Data Analytics applications.

BIG DATA TECHNOLOGIES Hadoop's Parallel World – Data discovery – Open source technology for Big Data Analytics – cloud and Big Data –Predictive Analytics – Mobile Business Intelligence and Big Data – Crowd Sourcing Analytics – Inter- and Trans-Firewall Analytics - Information Management.

UNIT II

PROCESSING BIG DATA Integrating disparate data stores - Mapping data to the programming framework- Connecting and extracting data from storage - Transforming data for processing - Subdividing data in preparation for Hadoop Map Reduce.

UNIT III

HADOOP MAPREDUCE Employing Hadoop Map Reduce - Creating the components of Hadoop Map Reduce jobs - Distributing data processing across server farms –Executing Hadoop Map Reduce jobs - Monitoring the progress of job flows - The Building Blocks of Hadoop Map Reduce - Distinguishing Hadoop daemons - Investigating the Hadoop Distributed File System Selecting appropriate execution modes: local, pseudo-distributed, fully distributed.

UNIT IV

ADVANCED ANALYTICS PLATFORM Real-Time Architecture – Orchestration and Synthesis Using Analytics Engines– Discovery using Data at Rest – Implementation of Big Data Analytics – Big Data Convergence – Analytics Business Maturity Model.

UNIT V BIG DATA TOOLS AND TECHNIQUES

Installing and Running Pig – Comparison with Databases – Pig Latin – User- Define Functions – Data Processing Operators – Installing and Running Hive – Hive QL – Tables – Querying Data – User-Defined Functions – Oracle Big Data.

Course Outcomes:

- Understand what Big Data is and why classical data analysis techniques are no longer adequate

- Understand the benefits that Big Data can offer to businesses and organizations
- Understand conceptually how Big Data is stored
- Understand how Big Data can be analysed to extract knowledge
- Communicate with data scientists

REFERENCES:

1. Michael Minelli, Michehe Chambers, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Business", 1st Edition, Ambiga Dhiraj, Wiely CIO Series, 2013.
2. Arvind Sathi, "Big Data Analytics: Disruptive Technologies for Changing the Game", 1st Edition, IBM Corporation, 2012.
3. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", 1st Edition, Wiley and SAS Business Series, 2012.
4. Tom White, "Hadoop: The Definitive Guide", 3rd Edition, O'reilly, 2012.

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW12) SOFTWARE SECURITY ENGINEERING

(PROGRAM ELECTIVE-IV

M.Tech: II-Semester	**L/T/P C** **3/0/- 3**

Objectives:

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

UNIT – I Security a software Issue: introduction, the problem, Software Assurance and Software Security, Threats to software security, Sources of software insecurity, Benefits of Detecting Software Security What Makes Software Secure: Properties of Secure Software, Influencing the security properties of software, Asserting and specifying the desired security properties?

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

UNIT – III Secure Software Architecture and Design: Introduction, software security practices for architecture and design: architectural risk analysis, software security knowledge for architecture and design: security principles, security guidelines and attack patterns Secure coding and Testing: Code analysis, Software Security testing, Security testing considerations throughput the SDLC

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

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

Course Outcomes

1. An ability to analyze security and privacy and properties of systems.
2. An ability to conduct user-cantered design for security engineering.
3. An ability to understand programming constraints with systems security.
4. An understanding of limitations and advantages of security protocols, functional and attacker perspectives, password authentication and various alternative systems.
5. Discussing the Security adopting considerations and limitations

TEXT BOOK:

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

REFERNCES:

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

*****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(M18SW13) BUSINESS PROCESS MANAGEMENT
(PROGRAM ELECTIVE-IV)

M.Tech: II Semester **L/T/P C**
3/0/- 3

Course Objectives:

- To recognize the role of business processes within an Infinity based application
- To understand the importance of parameter sets to a business process
- To learn common patterns and best practices for formatting and restricting the output from a business process
- To understand the difference between a business process and a business process instance
- To learn how data processing occurs within a business process
- To list the Infinity SDK software developer responsibilities for building and supporting the functionality required for a business process

UNIT I

UNDERSTANDING BPM - I: How can we demystify business process management? What is business process management? Why is it important to improve business process before automating them? When should you do BPM – what are the main drivers and triggers? Who should be involved in BPM?

UNIT II

UNDERSTANDING BPM - II: Why are organizational strategy and process architecture important in BPM implementation? How do you sell BPM technology to the organization? What are the critical success factors in a BPM project? What are the critical implementation aspects for a BPM solution?

UNIT III

FRAMEWORK - I: Framework overview, Guidelines on how to use the framework, Organization strategy phase, Process architecture phase, Launch pad phase, Understand phase, Innovate phase.

UNIT IV

FRAMEWORK – II: People phase, Develop phase, Implement phase, Realize value phase, Sustainable performance phase, Essentials introduction, Project management, People change management, Leadership.

UNIT V

BPM AND THE ORGANIZATION:BPM maturity, Embedding BPM within the organization.

Course Outcomes:

- Develop new or improved innovative business processes from gap analysis through process design in support of a company's strategic objectives in a socially responsible manner.
- Develop business models that support a company's strategic objectives.
- Articulate the interdependence between financial and operational metrics used in value chain analysis to key decision makers.

- Appraise the impact on financial and operational performance of specific business processes/models.
- Evaluate the opportunities for business process and supply chain improvement based on current best practices across industries, as well as new breakthrough thinking.
- Analyze the key business processes that drive the value chain of an organization throughout the entire product life cycle.

TEXT BOOKS:

1. Business Process Management, Practical guidelines to successful implementations, John Jeston and Johan Nelis, Second edition, Elsevier, 2009.
2. Management by Process, A roadmap to sustainable Business Process Management, John Jeston and Johan Nelis, Elsevier, 2009.

REFERENCES:

1. Business Process Management Systems, Strategy and Implementation, James F. Chang, Auerbach Publications, Taylor and Francis group, 2005

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18CN12) CYBER SECURITY
(PROGRAM ELECTIVE-IV)

M.Tech: II-Semester **L/T/P C**
3/-/- 3

Course Objective:

- To learn about cyber crimes and how they are planned
- To learn the vulnerabilities of mobile and wireless devices
- To learn about the crimes in mobile and wireless devices

UNIT-I

Introduction to Cybercrime: Introduction, Cybercrime and Information security, who are cybercriminals, Classifications of Cybercrimes, Cybercrime: The legal Perspectives and Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes.

UNIT-II

Cyber offenses: How criminals Plan Them Introduction, How Criminals plan the Attacks, Social Engineering, Cyber stalking, Cyber cafe and Cybercrimes, Botnets: The Fuel for Cybercrime, Attack Vector, Cloud Computing.

UNIT III

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

UNIT IV

Tools and Methods Used in Cybercrime: Introduction, Proxy Servers and Anonymizers, Phishing, Passwaord Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks

UNIT V

Understanding Computer Forensics Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Chain of Custody concept, Network Forensics, Approaching a computer, Forensics Investigation, Challenges in Computer Forensics, Special Tools and Techniques,Forensics Auditing

Course Outcomes:
After the completion of this course the student should be able to

1. Understand the different kinds of security attacks, services and mechanisms.
2. Define a internetwork security model and identify the TCP, UDP session hijacking.

3. Identify and classify the different types of attacks and suggest appropriate conventional encryption algorithms to be applied.
4. Gain complete knowledge in number system and areas of applications in public key cryptography algorithms.
5. Interpret the importance of digital signatures, digital Certificates, Certificate Authority for electronic document transfer on internet.
6. Demonstrate IP security architecture and explain how Pretty Good Privacy (PGP) and S/MIME provides Email privacy.

TEXT BOOKS:

1. **Cyber Security**: *Understanding Cyber Crimes, Computer Forensics and Legal Perspectives*, Nina Godbole and Sunil Belapure, Wiley INDIA.
2. **Introduction to Cyber Security** , Chwan-Hwa(john) Wu,J.David Irwin.CRC Press T&F Group

REFERENCES

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18AC02) STRESS MANAGEMENT
(AUDIT COURSE)

M.Tech: II-Semester	**L/T/P C**
	2/0/0 0

UNDERSTANDING STRESS Meaning – Symptoms – Work Related Stress – Individual Stress – Reducing Stress -sources of stress –consequence of stress-burnout-symptoms of Burnout- stress verses Burnout-model of stress-strategies for coping stress (individual and organizational strategies) –case study

TIME MANAGEMENT Techniques – Importance of Planning the day –developing concentration – Prioritizing Beginning at the start – Techniques for conquering procrastination – Sensible delegation – Taking the right breaks – Learning to say "No"

CAREER PLATEAU Career plateau – Identifying Career plateaus – Structural and Content - Plateauing – Making a fresh start – Importance of Sabbaticals – Counseling out – Executive leasing – Sustaining a marketable Career.

CRISIS MANAGEMENT Implications – People issues – Structure issues – Environmental issues – Learning to keep calm - Preventing interruptions – Controlling crisis – Pushing new ideas – Empowerment – Work place Humor, Developing a sense of Humor – Learning to laugh – role of group cohesion and team spirit.

SELF DEVELOPMENT Improving personality – Leading with Integrity – Enhancing Creativity – Effective decision making – Sensible Communication – The Listening Game – Managing Self – Mediation for peace – Yoga for Life.

Course Outcomes:
After the completion of this course, the students should be able to

1. Maintain a stress awareness log. Include identification of causes, symptoms, and analysis of effects.
2. Gather information on current stress management techniques and evaluate personal relevance.
3. Practice specific techniques, track effectiveness, and revise to meet personal preferences.
4. Create an adaptable stress management plan for academic success incorporating selected techniques.

TEXT BOOKS

1. Bhatia R.L., The Executive Track: An Action Plan for Self Development Wheeler Publishing.
2. Charavathy.S.K, "Human Values for Manager", McGraw Hill/HenelyManagement Series

REFERENCES

1. Jeffr Davison, Managing Stress, Prentice Hall of India, New Delhi
2. Jerrold S Greenberg, Comprehensive Stress Management, Jain Books, 2009

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18SW14) SOFTWARE TESTING LAB
(LABORATORY-III)

M.Tech: II-Semester **L/T/P C**
0/-/4 2

Course Objectives:

The student should be able to:

- To understand software testing and quality assurance as a fundamental component of software life cycle
- To define the scope of SW T&QA projects
- To efficiently perform T&QA activities using modern software tools
- To estimate cost of a T&QA project and manage budgets.

Software Testing Objectives:

To learn to use the following (or similar) automated testing tools to automate testing:

a) Win Runner/QTP for functional testing.
b) LoadRunner for Load/Stress testing.
c) Test Director for test management.
d) JUnit,HTMLUnit,CPPUnit.

Week-1:

Write programs in ‘C’ Language to demonstrate the working of the following constructs:
i) do...while ii) while….do iii) if…else iv) switch v) for

Week-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.

Week-3:

Take any system (e.g. ATM system) and study its system specifications and report the various bugs.

Week-4:

Write the test cases for any known application (e.g. Banking application)

Week-5:

Create a test plan document for any application (e.g. Library Management System)

Week- 6:

Refer Page no 115 in Text book 2(Foundations of software testing by Rex Black,Erik Van Veenendaal,Dorthy Graham) for the described scenario and observe the given
i. Equivalence Partioning /Boundary Value Analysis ii. Decision Tables
ii. State transition iv. Statement and decision testing.consider any other scenario of your choice and do the same.

Week-7:

Refer Page no 158 in Text book 2(Foundations of software testing by Rex Black,Erik Van Veenendaal,Dorthy Graham) for the described scenario and observe the given **Incident Report** and consider any other scenario of your choice and do the same.

Week -8:

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

Week- 9:

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

Week-10:

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

Course Outcomes

1. Understanding Selenium tool to perform testing
2. Writing test suits for applications
3. Construct and test simple programs.
4. Understanding the use of bug tracking and testing tool
5. Ability to learn any open source Testing tool

***

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18SW15) BIGDATA ANALTYICS LAB
(LABORATORY-IV)

M.Tech: II-Semester **L/T/P C**
0/4/- 2

List of Lab Programs
Course Objectives:

The objective of course is to provide an insight to Bigdata techniques, techniques along with attacks on data hiding and integrity of data.

The following programs should be implemented using Hadoop.

Week – 1: Installation of JDk, Hadoop 3.0.3.

Week – 2: Write a program Word Count. Write a very simple Hadoop program that counts the number of occurrences of each word in a text file. In Hadoop, this program, known as Word Count is the equivalent of the standard Hello, world! program you typically write when you learn a new programming language..

Week – 3: MapReduce for Parallelizing Computations: We will now estimate the value of Euler's constant (e) using a Monte Carlo method. Let X1, X2, ..., Xn be an infinite sequence of independent random variables drawn from the uniform distribution on [0,1]. Let V be the least number n such that the sum of the first n samples exceeds 1: V = min{n | X1 +X2 +...+Xn > 1} The expected value of V is e: E(V) = e Each Map task will generate random points using a uniform distribution on [0,1] in order to find a fixed number of values of n. It will output the number of time each value of n has been produced. The Reduce task will sum the results, and using them, the program will calculate the expected value of V and print the result

Week – 4: NCDC Weather Data: The NCDC produces CSV (Comma-Separated Values) files with worldwide weather data for each year. Each line of one of these files contains: • The weather station's code. • The date, in the ISO-8601 format. • The type of value stored in that line. All values are integers. TMIN (resp. TMAX) stands for minimum (resp. maximum) temperature. Temperatures are expressed in tenth of degrees Celsius. AWND stands for average wind speed, and PRCP stands for precipitation (rainfall), etc. Several other types of records are used (TOBS, SNOW, ...). • The next field contains the corresponding value (temperature, wind speed, rainfall, etc.) • All lines contain five more fields that we won't use in this exercise.

Week – 5: Back to Counting: Create a new class in your project CMPT732A1-WordCount named WordCountByLength that counts the number of words of each length: it will return the number of 1-letter words, of 2-letter words, and so on. Run your program on one of the gutenberg-.txt files, and plot the results (you can use gnuplot). Look up the distribution of word lengths in English. You can check the following paper on Arxiv, for instance: http://arxiv.org/pdf/1207.2334.pdf Are your results close?.

Week – 6: Create a new class in the same project named LetterCount that calculates the frequency of each letter in a file. Plot the results obtained using one of the gutenberg-.txt files. You can find the relative frequencies of each letter in the English language at the following URL: http://en.wikipedia.org/wiki/Letter_frequency.

Week -7 Write a MapReduce program to find degree of each vertex in a given graph.

Week-8: Write a MapReduce program to find lowest degree vertex in a given graph.

Week-9: Write a MapReduce program to neighbours of each vertexin a given graph.

Week-10: Write a MapReduce program for stop word elimination problem.

Input: A large tetual file containing one sentence per line. A small file containing a set of stop words (one stop word per line)

Output: A textual file containing the same sentences of the large input file without the words

appearing in the small file.

Course Outcomes:

- Understand what Big Data is and why classical data analysis techniques are no longer adequate
- Understand the benefits that Big Data can offer to businesses and organizations
- Understand conceptually how Big Data is stored
- Understand how Big Data can be analysed to extract knowledge
- Communicate with data scientists

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18SW17) INFORMATION RETRIEVAL SYSTEMS
(Program Elective-V)

M.Tech: III-Semester **L/T/P C**
3/-/- 3

Course Objectives:

To enable students to make more effective use of data stored in databases. To create a clean and consistent repository of data within a data warehouse. To utilize various levels and types of summarization of data to support management decision making. To discover patterns and knowledge that is embedded in the huge quantities of data records using different data mining techniques.

UNIT I

Introduction to Information Retrieval Systems : Definition of Information Retrieval System, Objectives of Information Retrieval System, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses; Boolean retrieval. The term vocabulary and postings lists. Dictionaries and tolerant retrieval. Index construction. Index compression.

UNIT II

Scoring, term weighting and the vector space model. Computing scores in a complete search system. Evaluation in information retrieval. Relevance feedback and query expansion.

UNIT III

XML retrieval. Probabilistic information retrieval. Language models for information retrieval. Text classification. Vector space classification.

UNIT IV

Support vector machines and machine learning on documents. Flat clustering. Hierarchical clustering. Matrix decompositions and latent semantic indexing.

UNIT V

Web search basics. Web crawling and indexes. Link analysis.

TEXT BOOKS:

1. Introduction to Information Retrieval , Christopher D. Manning and Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2008.

2. Information Storage and Retrieval Systems: Theory and Implementation, Kowalski, Gerald, Mark T Maybury,Springer.

REFERENCES :

1. Modern Information Retrival , Ricardo Baeza-Yates, Pearson Education, 2007.
2. Information Retrieval: Algorithms and Heuristics, David A Grossman and Ophir Frieder, 2nd Edition, Springer, 2004.
3. Information Retrieval Data Structures and Algorithms, William B Frakes, Ricardo Baeza-Yates, Pearson Education, 1992.
4. Information Storage & Retieval , Robert Korfhage , John Wiley & Sons.

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18SW18) PRINCIPLES OF INFORMATION SECURITY
(PROGRAM ELECTIVE- V)

M.Tech: III-Semester	**L/T/P C**
	3/-/- 3

Course Objectives:

- To introduce the concepts of Information Security
- To understand the concepts of cryptography
- To understand the various encryption algorithms
- To understand various authentication algorithms

.

UNIT – I

Information Security: Introduction, History of Information security, What is Security, CNSS Security Model, Components of Information System, Balancing Information Security and Access, Approaches to Information Security Implementation, The Security Systems Development Life Cycle.

UNIT – II

Symmetric Key Algorithms and AES: Introduction, Algorithm Types and Modes, Symmetric Key Cryptography, Data Encrption Standard(DES), International Data Encryption Algorithm(IDEA), RC4, RC5, Blowfish, Advanced Encrption Standard(AES), Key Management, **Asymmetric key Algorithms: Overview of A**symmetric Key Cryptography, RSA Algorithm, Diffie-Hellman Key Exchange, Key Management

UNIT – III

Message Authentication and Hash Functions: Authentication requirements and functions, MAC and Hash Funtions, **MAC Algorithms**: Secure Hash Algorithm, Whirlpool, HMAC, Digital signatures, X.509, Kerberos

UNIT – IV

Security at layers(Network, Transport, Application): IPSec, Secure Socket Layer(SSL), Transport Layer Security(TLS), Secure Electronic Transaction(SET), Pretty Good Privacy(PGP), S/MIME

UNIT – V

Inruders, Virus: Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, **Case Studies on Cryptography and security:** Single Sign On(SSO), Secure Inter-branch Payment Transactions, Secret Splitting, Secure Multiparty Calculation.

TEXT BOOKS:

1. Principles of Information Security: Michael E. Whitman, Herbert J. Mattord, CENGAGE Learning, 5th Edition.
2. Cryptography and Network Security : Atul Kahate, Mc Graw Hill, 2nd Edition
3. Cryptography and Network Security : William Stallings, Pearson Education,4th Edition

REFERENCES:

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.

2. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning
3. Cryptography and Network Security : Forouzan Mukhopadhyay, Mc Graw Hill, 2nd Edition
4. Principles of Computer Security: WM.Arthur Conklin, Greg White, TMH.

****

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONMOUS)

(M18SW19) COMPUTER FORENSICS
(PROGRAM ELECTIVE- V)

M.Tech: III-Semester **L/T/P C**
3/-/- 3

Course Objectives:
To understand the cyberspace, forensics fundamentals, evidence capturing process and preservation of digital evidence

.

UNIT I : Computer Forensics Fundamentals: Introduction to Computer Forensics, Use of Computer Forensics in Law Enforcement, Computer Forensics Assistance to Human Resources/Employment Proceedings, Computer Forensics Services, Benefits of Professional Forensics Methodology, Steps Taken by Computer Forensics Specialists, Who Can Use Computer Forensic Evidence?.**Types of Computer Forensics Technology :** Types of Military Computer Forensic Technology, Types of Law Enforcement Computer Forensic Technology, Types of Business Computer Forensics Technology.

UNIT II : Computer Forensics Evidence and Capture: Data Recovery: Data Recovery Defined, Data Backup and Recovery, The Role of Backup in Data Recovery, The Data-Recovery Solution, Case Histories.**Evidence Collection and Data Seizure:** Why Collect Evidence?, Collection Options, Obstacles, Types of Evidence, The Rules of Evidence, Volatile Evidence, General Procedure, Collecting and Archiving, Methods of Collection, Artifacts, Collection Steps, Controlling Contamination: The Chain of Custody.

UNIT III: Duplication and Preservation of Digital Evidence: Preserving the Digital Crime Scene, Computer Evidence Processing Steps, Legal Aspects of Collecting And Preserving Computer Forensic Evidence. **Computer Image Verification and Authentication :** Special Needs of Evidential Authentication, Practical Considerations, Practical Implementation.

UNIT IV: Computer Forensics Analysis: Discovery of Electronic Evidence: Electronic Document Discovery: A Powerful New Litigation Tool, **Identification of Data:** Timekeeping, Time Matters,Forensic Identification and Analysis of Technical Surveillance Devices. **Reconstructing Past Events**: How to Become a Digital Detective, Useable File Formats, Unusable File Formats, Converting Files. **Networks:** Network Forensics Scenario, A Technical Approach, Destruction of Email, Damaging Computer Evidence, International Principles Against Damaging of Computer Evidence, Tools Needed for Intrusion Response to the Destruction of Data, Incident Reporting and Contact Forms.

UNIT V: Current Computer Forensics Tools: Evaluating Computer Forensics Tool Needs, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software.

TEXT BOOKS:

1. “Computer Forensics : Computer Crime Scene Investigation”, JOHN R. VACCA, Firewall Media.
2. “Guide to Computer Forensics and Investigations”4e, Nelson, Phillips Enfinger, Steuart, Cengage Learning.

REFERENCES:

1. “Computer Forensics and Cyber Crime”, Marjie T Britz, Pearson Education.

2. “Computer Forensics”, David Cowen, Mc Graw Hill.
3. Brian Carrier , "File System Forensic Analysis" , Addison Wesley, 2005
4. Dan Farmer & Wietse Venema ,"Forensic Discovery", Addison Wesley, 2005
5. Eoghan Casey , ―Digital Evidence and Computer Crime ―, Edition 3, Academic Press, 2011
6. Chris Pogue, Cory Altheide, Todd Haverkos ,Unix and Linux Forensic Analysis DVD ToolKit, Syngress Inc. , 2008
7. Harlan Carvey ,Windows Forensic Analysis DVD Toolkit, Edition 2, Syngress Inc. , 2009
8. Harlan Carvey ,Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry , Syngress Inc, Feb 2011
9. Eoghan Casey, Handbook of Digital Forensics and Investigation, Academic Press, 2009
10. Gonzales/ Woods/ Eddins, Digital Image Processing using MATLAB, 2nd edition, Gatesmark Publishing, ISBN 9780982085400
11. N.Efford, Digital Image Processing, Addison Wesley 2000, ISBN 0-201-59623-7
12. M Sonka, V Hlavac and R Boyle, Image Processing, Analysis and Machine Vision, PWS
13. 1999, ISBN 0-534-95393-
14. Pratt.W.K., Digital Image Processing, John Wiley and Sons, New York, 1978

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
(M18MA01) ADVANCED OPTIMIZATION TECHNIQUES
(Open Elective)

M.Tech:III-Semester **L/T/P C**
3/0/- 3

UNIT-I:

Single Variable Non-Linear Unconstrained Optimization: One dimensional Optimization methods:- Uni-modal function, elimination methods, ,, Fibonacci method, golden section method, interpolation methods – quadratic & cubic interpolation methods.

UNIT-II:

Multi variable non-linear unconstrained optimization: Direct search method – Univariant method - pattern search methods – Powell's- Hook -Jeeves, Rosenbrock search methods- gradient methods, gradient of function, steepest decent method, Fletcher Reeves method, variable metric method.

UNIT-III:

Linear Programming: Formulation – Sensitivity analysis. Change in the constraints, cost coefficients, coefficients of the constraints, addition and deletion of variable, constraints. Simulation – Introduction – Types- steps – application – inventory – queuing systems

UNIT –IV"

Integer Programming: Introduction – formulation – Gomory cutting plane algorithm – Zero or one algorithm, branch and bound method Stochastic programming: Basic concepts of probability theory, random variables- distributions-mean, variance, correlation, co variance, joint probability distribution-stochastic linear, dynamic programming.

UNIT-V:

Geometric Programming: Polynomials – arithmetic - geometric inequality – unconstrained G.Pconstrained G.P (<= TYPE ONLY) Non-traditional optimization Techniques: Genetic Algorithms-Steps-Solving simple problemsComparitions of similarities and dissimilarities between traditional and non-traditional techniquesParticle Swarm Optimization (PSO)- Steps(Just understanding)-Simulated Annealing-Steps-Simple problems.

Course Outcomes:

After the completion of this course the student should be able to:

1. Describe problem clearly, identify and analyze the individual functions.
2. Analyze study on solving optimization problem.
3. Translate verbal formula on optimization problem.
4. Design algorithms, reliably to find an approximate solution.
5. Evaluate and compare the performance of an algorithm.
6. Discovery, study, understand and solve optimization techniques using algorithms.

REFERENCES:

1. Optimization theory & Applications / S.S. Rao / New Age International.
2. Engineering Optimization-Kalyan Deb/ PHI
3. Introductory to operation Research / Kasan & Kumar / Springar
4. Optimization Techniques theory and practice / M.C.Joshi, K.M. Moudgalya/ Narosa
5. Publications
6. Operation Research / H.A. Taha /TMH

7. Optimization in operations research / R.L Rardin 8. Optimization Techniques /Benugundu & Chandraputla / Pearson Asia

***

VAAGDEVI COLLEGE OF ENGINEERING (AUTONOMOUS)

(M18SE27) WASTE MANAGEMENT (Open Elective)

M.Tech: III-Semester **L/T/P C**
3/0/- 3

Week This course covers various aspects of hazardous waste, biomedical waste and E-waste such as collection, segregation, recovery, labeling requirements, storage areas, treatment and disposal facilities.

UNIT-I:

Sources, Composition and characteristic of hazardous waste, Hazardous Waste (Management and Handling) Rules, 1989 and amendments, Federal Hazardous Waste Regulations under RCRA, Superfund, CERCLA and SARA. Toxicology, public health impact, Protocols, issues and challenges in transportation of hazardous waste.

UNIT-II:

Municipal Solid Waste Management – Fundamentals Sources; composition; generation rates; collection of waste; separation, transfer and transport of waste; treatment and disposal options Radioactive Waste Management – Fundamentals Sources, measures and health effects; nuclear power plants and fuel production; waste generation from nuclear power plants; disposal options

UNIT-III:

Characterization of medical waste- Bio-medical wastes (Management and Handling) Rules, 1998, Amendments and guidelines, segregation, packaging, storage, transport of infectious waste. Techniques of Biomedical waste management. Health and safety rules. Protocols, issues and challenges in transportation of Biomedical waste.

UNIT-IV:

Treatment method- Autoclave, Hydroclave, Microwave, Chemical Disinfection, Solidification and stabilization, Bioremediation, Thermal Conversion Technologies, accumulation and storage of hazardous waste, land disposal of hazardous waste, other treatment and disposal method. Common Hazardous Waste Treatment facilities (TSDF).

UNIT- V:

E-waste: Introduction, toxicity due to hazardous substances in e-waste and their impacts, domestic e-waste disposal, e-waste management, technologies for recovery of resource from electronic waste, guidelines for environmentally sound management of e-waste, occupational and environmental health perspectives of recycling e-waste in India.

Course Outcomes:

After the completion of this course the student should be able to:

1. Evaluate the subject from the technical, legal and economical points .
2. Learn solid waste management.
3. Describe environment for sound management.
4. Understand a municipal solid waste management system.
5. Plan a solid waste management system for decision makers.
6. Design an incineration facility.

Reference Books:

- Tchobanoglous G., Theisen H., Viquel S.A., “Integrated Solid Waste Management: Engineering,
- Principles and Management issues”, Tata McGraw Hill Publishing Company Ltd., New Delhi.

- CPHEEO Manual on Municipal Solid Waste Management.
- Peavy H.S., Rowe D.R., Tchobanoglous G., “Environmental Engineering”, Tata McGraw Hill Publishing Company Ltd., New Delhi.
- Cunningham W.P., Cunningham M.A., “Principles of Environmental Science”, Tata McGraw Hill Publishing Company Ltd., New Delhi.

VAAGDEVI COLLEGE OF ENGINEERING (AUTONMOUS)

(M18VL07) EMBEDDED SYSTEM DESIGN (Open Elective)

M.Tech: III-Semester **L/T/P C**
3/-/- 3

Course Objectives:

To explain various embedded system applications and design requirements.

- To construct embedded system hardware.
- To develop software programs to control embedded system.
- To generate product specification for embedded system.

UNIT-I:

Introduction to Embedded Systems: Embedded Systems, Processor Embedded into a System, Embedded Hardware Units and Devices in a System, Embedded Software, Complex System Design, Design Process in Embedded System, Formalization of System Design, Classification of Embedded Systems

UNIT-II:

8051 and Advanced Processor Architecture: 8051 Architecture, 8051 Micro controller Hardware, Input/output Ports and Circuits, External Memory, Counter and Timers, Serial data Input/output, Interrupts, Introduction to Advanced Architectures, Real World Interfacing, Processor and Memory organization - Devices and Communication Buses for Devices Network: Serial and parallel Devices & ports, Wireless Devices, Timer and Counting Devices, Watchdog Timer, Real Time Clock, Networked Embedded Systems, Internet Enabled Systems, Wireless and Mobile System protocols

UNIT-III:

Embedded Programming Concepts: Software programming in Assembly language and High Level Language, Data types, Structures, Modifiers, Loops and Pointers, Macros and Functions, object oriented Programming, Embedded Programming in C++ & JAVA

UNIT-IV:

Real – Time Operating Systems: OS Services, Process and Memory Management, Real – Time Operating Systems, Basic Design Using an RTOS, Task Scheduling Models, Interrupt Latency, Response of Task as Performance Metrics - RTOS Programming: Basic functions and Types of RTOSES, RTOS VxWorks, Windows CE

UNIT-V:

Embedded Software Development Process and Tools: Introduction to Embedded Software Development Process and Tools, Host and Target Machines, Linking and Locating Software, Getting Embedded Software into the Target System, Issues in Hardware-Software Design and Co-Design - Testing, Simulation and Debugging Techniques and Tools: Testing on Host Machine, Simulators, Laboratory Tools.

Course Outcomes:

After the completion of this course the student should be able to

1. Explain the different embedded system design techniques and the metrics or challenges in designing them.
2. Understand the complete architecture of 8051 and Advanced Processor.
3. Demonstrate Software programming in Assembly language and High Level Language.

4. Develop code for object oriented Programming, Embedded Programming using Macros and Functions in c++ and java.
5. Classify the different Real Time Operating System (RTOS), RTOS Vx Works, Windows CE.
6. Understand the Embedded Software Development Process and Tools.

TEXT BOOK:

1. Embedded Systems, Raj Kamal, Second Edition TMH.
2. Introduction to Embedded Systems by K.V.Shibu.

REFERENCE BOOKS:

1. Embedded/Real-Time Systems, Dr. K.V.K.K. Prasad, dream Tech press
2. The 8051 Microcontroller and Embedded Systems, Muhammad Ali Mazidi, Pearson.
3. The 8051 Microcontroller, Third Edition, Kenneth J. Ayala, Thomson.
4. An Embedded Software Primer, David E. Simon, Pearson Education.
5. Micro Controllers, Ajay V Deshmukhi, TMH.
6. Microcontrollers, Raj Kamal, Pearson Education.
7. Introduction to Embedded Systems, Shibu K. V, TMH.

****