MAHATMA GANDHI UNIVERSITY

SCHEME AND SYLLABI

FOR

M. Tech. DEGREE PROGRAMME

IN

COMPUTER SCIENCE AND ENGINEERING

WITH SPECIALIZATION IN

COMPUTER SCIENCE AND SYSTEMS ENGINEERING

(2013 ADMISSION ONWARDS)

SCHEME AND SYLLABI
FOR
M.Tech DEGREE PROGRAMME
IN
COMPUTER SCIENCE AND ENGINEERING
WITH SPECIALIZATION IN
COMPUTER SCIENCE AND SYSTEMS ENGINEERING

SEMESTER - II

Sl. No	Course No.	Subjects	Hrs/week			Evaluation Scheme (Marks)					Credits (C)
						Sessional			ESE Theory / Practical	Total	
			L	T	P	TA	CT	Sub Total			
1	MCSSE 201	Object Oriented Systems Engineering	3	1	0	25	25	50	100	150	4
2	MCSSE 202	Computer System Design and Architecture	3	1	0	25	25	50	100	150	4
3	MCSSE 203	Distributed Computing Systems	3	1	0	25	25	50	100	150	4
4	MCSSE 204	Automated Verification	3	1	0	25	25	50	100	150	4
5	MCSSE 205	Elective III	3	0	0	25	25	50	100	150	3
6	MCSSE 206	Elective IV	3	0	0	25	25	50	100	150	3
7	MCSSE 207	CASE Tool Lab	0	0	3	25	25	50	100	150	2
8	MCSSE 208	Seminar II	0	0	2	50	0	50	0	50	1
Total			18	4	5	225	175	400	700	1100	25

L – Lecture, **T** – Tutorial, **P** – Practical

Elective – III (MCSSE 205)		**Elective – IV (MCSSE 206)**	
MCSSE 205 -1	Program Analysis & Verification	MCSSE 206 -1	Programming Languages and Type System
MCSSE 205 -2	Engineering System Architectures	MCSSE 206 -2	Systems Modeling and Simulation
MCSSE 205 -3	Information Systems Security	MCSSE 206 -3	Advanced Software Testing
MCSSE 205 -4	Semantic Web	MCSSE 206 -4	Data Compression Techniques

TA –	Teachers' Assessment (Quizzes, attendance, group discussion, tutorials, seminar, field visit etc)
CT –	Class Test (Minimum of two tests to be conducted by the Institute)
ESE –	University End Semester Exam will have to be conducted by the institute through concerned affiliating University.

MCSSE 201 OBJECT ORIENTED SYSTEMS ENGINEERING

L	T	P	C
3	1	0	4

SUBJECT DESCRIPTION AND OBJECTIVES

Object Oriented Systems Engineering paves a way for more effectively and efficiently interface with Software Engineering throughout full system lifecycles. To develop the students' ability to analyze and model requirements, as well as to develop software using object-oriented methodology. To be familiar with UML models and finally to experiment using CASE tools for software analysis and design.

Module 1

System Concepts – Project Organization – Communication – Project Management. Life cycle models – Unified Process – Iterative and Incremental – Workflow – Agile Processes

Module 2

Requirements Elicitation – Use Cases – Unified Modeling language: Introduction, UML Diagrams – Class diagrams, Sequence diagrams, Object diagrams, Deployment diagrams, Use case diagrams, State diagrams, Activity diagram, Component diagrams – Case Study.

Module 3

Analysis Object Model (Domain Model) – Analysis Dynamic Models – Non-functional requirements – Analysis Patterns. System Design, Architecture – Design Principles - Design Patterns – Dynamic Object Modeling – Static Object Modeling – Interface Specification – Object Constraint Language.

Module 4

Mapping Design (Models) to Code – Model Transformation - Refactoring - Mapping Associations - Mapping Activities – Testing - Usability – Deployment – Configuration Management – Maintenance process- System documentation- program evolution dynamics.

References:

1. Bernd Bruegge, Alan H Dutoit, "Object-Oriented Software Engineering," 2nd ed, Pearson Education, 2004.
2. Craig Larman, "Applying UML and Patterns" 3rd ed, Pearson Education,2005.
3. Stephen Schach, " Software Engineering" 7th ed, McGraw-Hill, 2007.
4. Ivar Jacobson, Grady Booch, James Rumbaugh, "The Unified Software Development Process", Pearson Education, 1999.
5. Alistair Cockburn, "Agile Software Development" 2nd ed, Pearson Education, 2007.

MCSSE 202 COMPUTER SYSTEM DESIGN AND ARCHITECTURE

L	T	P	C
3	1	0	4

SUBJECT DESCRIPTION AND OBJECTIVES

This subject deals with internal structure of the Digital Computer including structure of instruction, memory architecture, multicore architecture and techniques for implementing parallel processing. Understanding the concepts of these topics is essential for exploiting the recent development in the hardware architecture of Digital Computer. Sound knowledge in this subject is indispensible for other important subjects like Compilers.

Module 1

Fundamentals: Technology trend -performance measurement –Comparing and summarizing performance- quantitative principles of computer design –Amdahl's law- instruction set architectures –memory addressing- –type and size operand - encoding an instruction set - role of compilers – Review of pipelining - MIPS architecture

Module 2

Instruction level parallelism and its limits: Dynamic scheduling —dynamic hardware prediction - multiple issue processor – multiple issue with dynamic scheduling-hardware based speculation- limitation of ILP- Introduction to multicore processors

Module 3

Static scheduling and Memory hierarchy design: Static scheduling- loop unrolling-static branch prediction- software pipelining-hardware support for exploring more parallelism at compile time Memory hierarchy design - reducing cache misses and miss penalty, reducing hit time - main memory organization - virtual memory and its protection -. Memory issues in multicore processor based systems

Module 4

Multiprocessor and thread level parallelism: Classification of parallel architecture-models of communication and memory architecture-Symmetric shared memory architecture-cache coherence protocols-distributed shared memory architecture-directory based cache coherence protocol- Memory consistency-relaxed consistency models-multi threading- exploiting thread level parallelism in multicore architecture

References:

1. Hennesy J. L. & Pattersen D. A., Andrea C. Arpaci-Dusseau, "Computer Architecture: A Quantitative approach", 4/e, Morgan Kaufman, 2007
2. Pattersen D. A. & Hennesy J. L., "Computer Organisation and Design: The Hardware/ Software Interface", 3/e, Harcourt Asia Pte Ltd (Morgan Kaufman), Singapore
3. V. P. Heuring and H. F. Jordan, "Computer System Design and Architecture", Prentice Hall, 2003.

MCSSE 203 DISTRIBUTED COMPUTING SYSTEMS

L	T	P	C
3	1	0	4

SUBJECT DESCRIPTION AND OBJECTIVES

This subject is designed to study software components of distributed computing systems. The communication and interconnection architecture of multiple computer systems is introduced. The design issues of distributed computing systems are discussed. The subject emphasizes developing applications on various distributed computing platforms or environments. The main goals of this subject are to gain an understanding of the principles and techniques behind the design of distributed systems, such as locking, concurrency, scheduling, and communication across the networks and to gain practical experience in designing, implementing, and debugging real distributed systems.

Module -1

Distributed System – Goals, Hardware Concepts, Software concepts, Client Server Model. Communication – Layered Protocols, Remote Procedure Call, Remote Object Invocation, Message Oriented Communication, Stream Oriented Communication

Module -2

Processes – Threads, Clients, Servers, Code Migration, Software Agents Naming – Naming Entities, Locating Mobile Entities, Removing UnReferenced Entities, Synchronization – Clock Synchronization, Logical Clocks, Global State, Election Algorithms, Mutual Exclusion, Distributed Transactions.

Module -3

Consistency and Replication – Data Centric Consistency Models, Client Centric Consistency Models, Distributed Protocols, Consistency Protocols, Orca 347, Casually-Consistent Lazy Replication. Fault Tolerance – Failure Models, Failure Masking by redundancy, Process Resilience, Reliable Client Server Communication, Reliable Group Communication. Distributed Commit, Recovery.

Module -4

Security – Security Threats, Policies and Mechanisms, Secure Channels, Access Control, Security Management, KERBEROS, SESAME, Electronic Payment System Example. Distributed Object Based Systems – CORBA, Distributed COM, GLOBE, Comparison of CORBA, DCOM, and GLOBE

References:

1. Andrew S Tanenbaum, Maarten van Steen : “Distributed Systems Principles and Paradigms” Prentice Hall, 2002, ISBN: 0-13-088893-1
2. Andrew S. Tanenbaum: “Modern Operating Systems”, 2nd Edition, Prentice Hall, 2001, ISBN: 0-13-031358-0
3. Andrew S. Tanenbaum,”Computer Networks” 4/e, Prentice Hall, 2003, ISBN: 0-13-066102-3

MCSSE 204 AUTOMATED VERIFICATION

L	T	P	C
3	1	0	4

SUBJECT DESCRIPTION AND OBJECTIVES

Formal verification is used in proving the correctness of systems such as: software expressed as source code, cryptographic protocols, combinational circuits, and digital circuits with internal memory. The verification of these systems is done by providing a formal proof on an abstract mathematical model of the system. There are various tools (Alloi, Rodin, Spin, NuSMV, Pex, FindBug etc) developed by R&D organizations to help the different stages in Software development. The content of this course introduces the formal methods which serve as the foundation for all such kind of tools. Hence studying this subject is essential for understanding, improving and developing such tools. In spite of these, this subject is an indispensible area in Computer Science and Systems Engineering.

Module 1

Introduction, Transition System Models of Systems: Model Checking, Characteristics of Model Checking- Model Checking Process, Strength and Weaknesses; Transition Systems – Executions, Modelling Hardware and Software Systems, State-Space Exploration Problem.

Module 2

Linear Time Properties: Deadlock, Linear-Time Behaviour, Safety Properties and Invariants, Liveness Properties, Fairness.

Module 3

Regular Properties : Automata on Finite Words, Model Checking Regular Safety Properties – Regular Safety Properties, Verifying Regular Safety Properties, Automata On Infinite Words, w-Regular Languages and Properties, Nondeterministic, Deterministic and Generalized Buchi Automata, Model Checking w-Regular Properties.

Module 4

Model Checking, System, Tools, Properties : Linear Temporal Logic (LTL) – Syntax, Semantics, Specifying Properties, Automata Based LTL Model Checking – complexity, LTL Satisfiability and Validity Checking, Mutual Exclusion Problem, NuSMV Model Checker, Running NuSMV, Example Problems – Mutual Exclusion, Ferryman, Alternating bit protocol.

References:

1. Christel Baier, Joost-Pieter Katoen: “Principles of Model Checking”, MIT Press, 2008.
2. Michael Huth, Mark Ryan: “Logic in Computer Science: Modelling and Reasoning about Systems”, Cambridge University Press, 2004.

MCSSE 205-1 PROGRAM ANALYSIS AND VERIFICATION

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

The major hurdle in the software development process is ensuring the promised functionality. Most real software have lot of defects. They are bane of software industry and often they cause catastrophes. The basic causes of such defects are improper specification of requirements and implementation errors. Defect detection is hard because of the difficulty in exhaustive testing and limitation of static analysis tools. This subject introduces formal and rigorous way of specifying the requirements and methods for analysing/verifying the behaviour of the developed system with respect to the formal specification. This is an active research area with research group in various Universities and R&D groups in the industry. This subject is relevant for a PG course in Computer Science and Systems Engineering.

Module 1

Introduction : Nature of Program Analysis, Data Flow Analysis, Constrained Based Analysis, Abstract Interpretation, Partially Ordered Sets – Basic Definitions, Complete Lattices, Chain, Fixed Points.

Module 2

Dataflow analysis : Intraprocedural Analysis – Available Expressions, Reaching Definitions, Live Variables; Monotone and Distributive Frameworks – Basic Definition; Kildall's algorithm, Interprocedural analysis using Call String Approach.

Module 3

Abstract Interpretation : Abstract join-over-all-paths analysis of a program, Correctness of abstract information: Galois connections, abstract interpretation as an over-approximation of concrete semantics

Module 4

Pointer Analysis and Hoare Logic: Points-to Analysis, Alias Analysis, Static Program Analysis, Correctness and Precision, Andersen's Analysis, Steensgaards Analysis, Correctness and Precision, Hoare Logic – Hoare Triples as assertions, Programs as State Transformers, Hoare Logic Rules, Weakest Preconditions,

References:

Nielson, and Hankin, "Principles of Program Analysis", Springer- Verlag.

MCSSE 205-2 ENGINEERING SYSTEM ARCHITECTURES

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

This subject deals with methods for architecting successful systems. The first module describes the foundation of architecture and systems engineering. Different modeling languages are explained in the second module. The third module discusses functional, physical and operational architecture developments. The last module considers the software and informational technology systems.

Module 1

Foundations of Architecture and Systems Engineering : Classical Architecting Paradigm.Logical and Scientific Approach. Structured Analysis and Design. Waterfall Model of Systems Acquisition.

Module 2

Modeling Languages : Architecture Modeling language. System Modeling Language(SysML). Graphical Tools (DFD,IDEF0). Systems Engineering use of IDEF0 Models.

Module 3

Functional Architecture Development - Functional Decomposition - Relating Requirements of Functions. **Physical Architecture Development-** Definition of Elements of a System- Relating Functions and Requirements to Physical Elements. **Operational Architecture Development-** Input/Output Requirements- System/Subsystem Qualification Requirements.

Module 4

Software and Information Technology Systems: Status of Software Architecting. Software as a System Component. Systems, Software, and Process Models. Architecture in Software-Centered Systems. **Case Study :** Global Positioning System.

References:

1. Charles Dickerson, Dimitri N. Mavris.” Architecture and Principles of Systems Engineering”, Auerbach Publications,2009.
2. Mark W. Maier,” The Art of Systems Architecting, Third Edition” 2009 by CRC Press.
3. Buede, Dennis M. "The Engineering Design of Systems : Models and Methods", John Wiley &Sons, Inc., 2000
4. Alexander Kossiakoff, William N. Sweet, Sam Seymour, Steven M. Biemer “ Systems Engineering Principles and Practice”, (Wiley Series in Systems Engineering and Management) ,2nd Edition,2011.

MCSSE 205-3 INFORMATION SYSTEMS SECURITY

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

The goal of Information Systems Security is to familiarize students with the security issues and technologies involved in modern information systems, including computer systems and networks. The objective of the subject is to provide students with the necessary foundations to apply cryptography techniques in the emerging fields.

Module 1

Overview of Information Security: Basic Concepts, Cryptosystems, Cryptoanalysis, Ciphers & Cipher modes. **Symmetric Key Cryptography**: DES, AES. **Asymmetric Key Cryptography:** RSA algorithm, Key management protocols, Diffie Hellman Algorithm. **Digital Signature**: Digital Signatures, Public Key Infrastructure.

Module 2

SYSTEM SECURITY: Program Security - Security problems in Coding, Malicious Logic, Protection. **Database Security** –Access Controls, Security & Integrity Threats, Defence Mechanisms. **OS Security** - Protection of System Resources, Models for OS security.

Module 3

NETWORK & INTERNET SECURITY: LAN Security - Threats, Authentication & access control, Secured communication Mechanisms (IPSec, Kerberos, Biometric, PKI), Secured Design for LAN. **Firewall & IDS -** Firewall Techniques, Firewall Architecture, Types of IDS, IDS Tools. **Email & Transaction Security Mechanisms -** Privacy Enhanced Mail(PEM), S/MIME, SET protocol, Client-Server Security on web.

Module 4

WIRELESS SECURITY: Wi-Fi & IEEE 802.11 Security - Protocol architecture, WEP, Access controls. **Wireless Transport Layer Security -** Transport Layer Security, SSL, IPSEC, WAP security. **Bluetooth Security -** Protocol architecture, Attacks, Security architecture.

References:

1. Charles P. Pfleeger, “ Security in Computing (Second Edition)”, Prentic- Hall International, Inc., 1996.
2. Rolf Oppliger, “ Security Technologies for World Wide Web”, Artech House: Inc.
3. Behrouz Forouzan “Cryptography and Network Security” TMGH, 2007
4. Charles P. Pfleeger, Shari Lawrence Pfleeger “Security in computing” prentice hall ,2002
5. Bruce Schneier, “ Applied Cryptography Protocols, Algorithms, and Source Code in C (Second Edition)”, John Wiley & Sons, Inc., 1995.

MCSSE 205-4 SEMANTIC WEB

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. This subject introduce the basic concept of semantic web, ontology-based systems, Information retrieval from natural language based documents and Tools for Ontology construction. The objective is to equip students with the latest developments in the Semantic Web scenario.

Module 1

Introduction to semantic web, architecture, examples, semantic web technologies- Vocabularies, Taxonomies and Ontologies - Overview of Ontology Elements Requirements of ontology languages, logic agents, semantic web versus artificial intelligence

Module 2

Ontology-based Systems: Ontology based knowledge management, ontology construction, generating, storing, aligning and maintaining ontologies, languages for semantic web and ontologies: Web Documents in XML – RDF - Schema – Web Resource Description using RDF-RDF Properties – Topic Maps and RDF – Overview – Syntax Structure – Semantics – Pragmatics - Traditional Ontology Languages – LOOM- OKBC – OCML – Flogic Ontology Markup Languages – SHOE – OIL - DAML + OIL- OWL

Module 3

Information retrieval from natural language based documents; ontology evolution; ontological indexing and searching techniques for Searching web, Taxonomy for Ontology Learning Phases of Ontology Learning – Importing and Processing Ontologies and Documents – Ontology Learning Algorithms -Evaluation

Module 4

Ontology Management and Tools : Overview – need for management – development process – target ontology – ontology mapping – skills management system – ontological class – constraints – issues. Evolution– Development of Tools and Tool Suites – Ontology Merge Tools – Ontology based Annotation Tools. Programming ontology, Creation and manipulation of ontology using OWL API, Case Study.

References:

1. John Davies, Rudi Studer, and Paul Warren. “Semantic Web Technologies: Trends and Research in Ontology-based Systems,” Wiley.

2. John Davies, Dieter Fensel, Frank van Harmelen, and Frank van Harmelen. “Towards the Semantic Web: Ontology-Driven Knowledge Management” Wiley.
3. Grigoris Antoniou, Frank van Harmelen, “A Semantic Web Primer (Cooperative Information Systems)”, The MIT Press, 2004

4. Dieter Fensel (Editor), Wolfgang Wahlster, Henry Lieberman, James Hendler, "Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential", The MIT Press, 2002

5. .Asuncion Gomez-Perez, Oscar Corcho, Mariano Fernandez-Lopez "Ontological Engineering: with examples from the areas of Knowledge Management, e- Commerce and the Semantic Web" Springer, 2004

6. Alexander Maedche, "Ontology Learning for the Semantic Web", Springer; 1 edition, 2002

7. Soumen Chakrabarti, Morgan Kaufmann publishers, Mining the Web, Discovering Knowledge from Hypertext Data, Elsevier, B&N, Amazon, Pages 352, ISBN 1-55860-754-4, ISBN 81-8147-886-X

L	T	P	C
3	0	0	3

MCSSE 206-1 PROGRAMMING LANGUAGES AND TYPE SYSTEM

SUBJECT DESCRIPTION AND OBJECTIVES

Programming Languages have an important role in the quality and reliability of programs developed. Language with a strong Type System can help the programmer in detecting and removing various forms of errors/defects in the development process. Defect detection and removal is identified as one of the greatest difficulty in software development. This subject introduces a rigours Type System and techniques for detecting and removing defects in programs. This subject is essential for a PG course in Computer Science and Systems Engineering.

Module 1

Introduction : Types in Computer Science, Use of Type Systems, Type Systems and Language Design; Untyped Arithmetic Expressions – Introduction, Syntax, Induction on Terms, Semantics, Evaluation.

Module 2

Untyped Lambda Calculus and Typed Arithmetic Expressions : Basics, Abstract and Concrete Syntax, Scope, Operational Semantics. Programming in Lambda Calculus, Church Numerals, Enriching the calculus, Recursion, Substitution. Types, Typing Relation, Type Safety.

Module 3

Simply Typed Lambda Calculus, extention and Type Inference : Function Types, Typing Relation, Properties of Typing, Base Types, Unit type, Sequencing abd Wildcards, Ascription, Let bindings, Pairs, Tuples, Records, Sums, Variants.

Module 4

Type Reconstruction: Type Variables and Substitutions, Two views of type variables, Constraint-Based Typing, Completeness, Unification Algorithm, Implicit Type Annotations, Let Polymorphism.

References:

1. Benjamin C. Pierce, “Types and Programming Languages” , MIT Press, 2002
2. David A. Schmidt. “Programming Language Semantics”. In Allen B. Tucker,
3. Handbook of Computer Science and Engineering. CRC Press, 1996.
4. Michael L. Scott. “Programming Language Pragmatics” Elsevier, 2004

MCSSE 206-2 SYSTEMS MODELING AND SIMULATION

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

This subject covers modeling and simulation principles with techniques useful in systems engineering. Topics include: basic concept of modelling and simulation, concepts in discrete-event simulation, modeling complex systems and simulation of computer systems. Students will learn to develop and execute their own simulation models.

Module 1

Systems, Models, and Simulation. Systems and System Environment. Components of a System. Model of a System. Types of Models. Types of Simulation. Parallel and Distributed Simulation. Internet and Web-Based Simulation. Steps in a Simulation Study.

Module 2

Concepts in Discrete-Event Simulation. Event Scheduling/Time Advance Algorithm. Manual Simulation Using Event Scheduling. List processing in Simulation. Simulation Software. Classification of Simulation Software. Object-Oriented Simulation.

Module 3

Modeling Complex systems: A Simple Simulation Language - simlib. Single -Server Queuing Simulation with simlib. Time - Shared Computer Model. Job - Shop Model. Efficient Event - List Manipulation.

Module 4

Simulation of Computer Systems: Simulation Tools. Model Input. High-Level Computer-System Simulation.CPU Simulation. Memory Simulation. Simulation of Computer Networks: Traffic Modeling. Media Access Control. Data Link Layer. TCP Model Construction.

References:

1. Jerry Banks, John S Carson, Barry L Nelson, and David M Nicol, "Discrete- Event System Simulation", Fifth Edition, Prentice-Hall, 2005
2. Law and Kelton, "Simulation Modeling and Analysis", Third Edition, McGraw Hill, 2000.
3. J.B. Sinclair, "Simulation of Computer Systems and Computer Networks: A Process-Oriented Approach",2004.
4. Banks, J. Handbook of simulation: Principles, methodology, advances, applications and practice. Wiley,1998.

MCSSE 206-3 ADVANCED SOFTWARE TESTING

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

This subject provides the knowledge of a variety of ways to test software and understanding of some of the tradeoffs between testing techniques. Students will become acquainted with both the strengths and limitations of various functional and structural testing methods. This subject also discuss the testing applications on the web, web security testing, Testing Tools and Automation with case studies of JUnit and Selenium.

Module 1

Overview of the Software Testing Process : Organizing for testing, Developing the test plan, Verification Testing, Validation Testing, Analyzing and Reporting test result, Acceptance and Operational Testing, Post-Implementation Analysis. Testing in the Software Life Cycle. Test Planning and Control. Test Analysis and Design. Test Implementation and Execution.

Module 2

Test Techniques: Specification-Based Techniques. Structure-Based Techniques Defect-Based Techniques. Experience-Based Testing Techniques. Static Analysis Dynamic Analysis Choosing Testing Techniques. **Testing of Software Characteristics**: Quality Attributes for Test Analysts -Functional Testing, Usability Testing. Quality Attributes for Technical Test Analysts - Technical Testing in General, Technical Security Testing, Reliability Testing, Efficiency Testing, Maintainability Testing, Portability Testing.

Module 3

Testing Applications on the Web: Web Testing versus traditional testing, web systems, bug inheritance, back-end data accessing, thin-client versus thick- client processing. Mobile web application platform test. **Web security testing-** anatomy of an attack, attacking intents, testing goals and responsibilities, testing for security.

Module 4

Process Maturity Models- CMM *, CMMI*. **Testing Improvement Models -** TMM (Testing Maturity Model) , TPI (Test Process Improvement Model) , CTPs (Critical Testing Processes) , (Systematic Test and Evaluation Process). **Testing Tools and Automation:** Testing Tool Acquisition, Testing Tool Introduction and Deployment, Testing Tool Categories. **Case studies :** JUnit and Selenium Testing Tools.

References:

1. Perry, William E., “*Effective Methods for Software Testing*”, Third Edition Publisher: John Wiley & Sons.
2. Rex Black; Jamie L Mitchell “*Advanced Software Testing—Vol. 3*” Publisher: Rocky Nook
3. Hung Q. Nguyen; Bob Johnson; Michael Hackett “*Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems*”, Second Edition Publisher: John Wiley & Sons
4. Petar Tahchiev; Felipe Leme; Vincent Massol; Gary Gregory “*JUnit in Action*” Second Edition Publisher: Manning Publications
5. David Burns; “*Selenium 1.0 Testing Tools*” Publisher: Packt Publishing

MCSSE 206-4 DATA COMPRESSION TECHNIQUES

L	T	P	C
3	0	0	3

SUBJECT DESCRIPTION AND OBJECTIVES

Data compression is grounded in information theory, and there are many fundamental algorithms that must deal with our information transmission and storage tasks. In this subject discuss the theoretical underpinnings of data compression and cover many fundamental algorithms.

Module 1

Compression Techniques: Loss less compression, Lossy Compression, Measures of prefonnance, Modeling and coding, Mathematical Preliminaries for Lossless. compression: A brief introduction to information theory, Models: Physical models, Probability models, Markov models, composite source model, Coding: uniquely decodable codes, Prefix codes.

Module 2

The Huffman coding algorithm: Minimum variance Huffman codes, Adaptive Huffman coding: Update procedure, Encoding procedure, Decoding procedure. Golomb codes, Rice codes, Tunstall codes, Applications of Hoffman coding: Loss less image compression, Text compression, Audio Compression.

Module 3

Coding a sequence, Generating a binary code, Comparison of Binary and Huffman coding, Applications: Bi-level image compression-The JBIG standard, JBIG2, Image compression. Dictionary Techniques: Introduction, Static Dictionary: Diagram Coding, Adaptive Dictionary. The LZ77 Approach, The LZ78 Approach, Applications: File Compression-UNIX compress, Image Compression: The Graphics Interchange Format (GIF).

Module 4

Mathematical Preliminaries for Lossy Distortion criteria, Models, Scalar Ouantization: The Quantization problem, Uniform Quantizer, Adaptive Quantization, Non uniform Quantization. Vector Quantization Advantages of Vector Quantization over Scalar Quantization, The Linde-Buzo-Gray Algorithm, Tree structured Vector Quantizers. Structured Vector Quantizers.

References:

1. Khalid Sayood, "*Introduction to Data Compression,*" Morgan Kaufmann Publishers.
2. D. Salomon, "*Data Compression, The Complete Reference,*" Springer, 2nd Edition, 2000.
3. A. Gersho and R.M. Gray "*Vector Quantization and Signal Compression,*" Kluwer Academic Press, 1992.

L	T	P	C
0	0	3	2

MCSSE 207 COMPUTER AIDED SOFTWARE ENGINEERING LAB

- System Requirement Specification (SRS) and related analysis documents as per the guidelines in ANSI/IEEE Std 830-1984.
- Design documents representing the complete design of the software system.
- The design can further be improved by the principles learned in this module.
- Analysis and design for the same problem should be done using Object Oriented approach.
- Test documents as per ANSI/IEEE Std. 829/1983 Software Test Documentation.
- Simple exercises in effort and cost estimation in COCOMO model.
- Application of COCOMO and Function Point (FP) model for the actual project that has been chosen.
- Familiarization of SCM tools with some public domain software like SCCS, CVS.
- Familiarization of some reverse engineering tools available in the public domain.
- At the end of the semester, there should be a presentation of the project with demonstration.
- It is also advisable to have the students present the documents associated with the projects as and when they are ready. This will help the instructor identify pointing out the mistakes to them and the rest of the students.

L	T	P	C
0	0	2	1

MCSSE 208 SEMINAR II

Objective : To provide exposure to recent developments in the field of interest

Each student shall present a seminar on any topic of interest related to the core / elective courses offered in the second semester of the M. Tech. Programme. He / she shall select the topic based on the References: from reputed International Journals, preferably IEEE journals. They should get the paper approved by the Programme Coordinator / Faculty member in charge of the seminar and shall present it in the class.
Every student shall participate in the seminar. The students should undertake a detailed study on the topic and submit a report at the end of the semester. Marks will be awarded based on the topic, presentation, participation in the seminar and the report submitted.