

DETAILED SYLLABI

DISCRETE STRUCTURES FOR COMPUTER SCIENCE

MCSSE 101

SUBJECT DESCRIPTION AND OBJECTIVES

The foundation of various areas of study in the Computer Science is Discrete Structures included in different modules in this subject. The topics are selected to serve as the base for other subjects essential for a PG course in Computer Science and Systems Engineering. First module is essential subjects like Automata Theory, Algorithms, and Verification etc. Second module is essential for subjects like Formal Verification and Program Analysis etc. Third module is essential for subjects like Algorithms, Cryptography, Security, Computer Vision etc. Fourth module is essential for understanding various algebraic structures used in many other subjects.

Module 1

Sets, Relations, Functions and Lattices: Naive Set Theory, Relations, Functions and operations, Partial orders, Hasse Diagram, Equivalence Relation and Partitions. Knaster-Tarski theorem : Lattices, Complete Lattice, Fixed points, pre and post fixed points, monotonic functions, distributive functions.

Module 2

Propositional Logic : Semantics of Propositional Logic - Meaning of Logical Connectives, Mathematical Induction, Soundness of Propositional Logic, Completeness of Propositional Logic.

Module 3

Counting and Probability : The Rules of Sum and Product, Permutations, Combinations : Binomial Theorem, Combinations with Repetition, Probability, Axioms of Probability, Conditional Probability, Discrete Random Variables. Bernoulli, binomial and geometric random variables, Expectation, Linearity of Expectation.

Module 4

Abstract Algebra: Ring - Definition and Examples, Properties of Rings, Integers Modulo n , Groups – Definition, Examples and Elementary Properties, Homomorphisms, Isomorphisms, and Cyclic Groups, Cosets and Lagrange's Theorem. Ring - Definition and Examples, Properties of Rings, Integers Modulo n , Structure of the ring Z_n and the unit group Z_n^* .

References:

1. Ralph P. Grimaldi, Rose-Hulman Institute of Technology: "*Discrete and Combinatorial Mathematics*", 5/E, Pearson, 2004.
2. Michael Huth, Mark Ryan: "*Logic in Computer Science: Modelling and Reasoning about Systems*", Cambridge University Press, 2004.
3. Nielson, Nielson, and Hankin, "*Principles of Program Analysis*", Springer-Verlag.
4. L.Lovasz, J. Pelikan, K. Vesztergombi. "*Discrete Mathematics*", Springer.
5. Rudolf Lidl, Gunter Pilz. "*Applied Abstract Algebra*". Springer
6. Alfred Tarski, "*A lattice-theoretic fixpoint theorem and its applications*", Pacific J. Mathematics, 5, pages 285--309, 1955

DESIGN AND ANALYSIS OF ALGORITHMS

MCSSE 102

SUBJECT DESCRIPTION AND OBJECTIVES

Algorithms are fundamental to computer science and software engineering. The real-world performance of any software system depends on only two things: (1) the algorithms chosen and (2) the suitability and efficiency of the various layers of implementation. Good algorithm design is therefore crucial for the performance of all software systems. Moreover, the study of algorithms provides insight into the intrinsic nature of the problem as well as possible solution techniques independent of programming language, programming paradigm, computer hardware, or any other implementation aspect.

Specifically, students will be able to achieve the following:

Demonstrate the following capabilities: to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in programming context.

Choose the appropriate data structure for modelling a given problem and Compare and contrast the costs and benefits of different data structure implementations.

Explain the significance of NP-completeness and importance of approximation algorithms.

Explain the use of randomization in the design of an algorithm for a problem where a deterministic algorithm is unknown or much more difficult.

Module 1

Advanced Structures: Binary search trees, B trees, AVL trees, Red black trees, splay trees. Van Emde Boas trees. Randomly built binary search trees. Heaps, Binomial heaps, Fibonacci heaps. Minimum spanning trees, BFS, DFS, Strongly connected components, Biconnected components.

Module 2

Network Flow: Flow networks, The Ford-Fulkerson method, Maximum bipartite matching. **String Matching:** Naive string-matching algorithm, Rabin-Karp algorithm, String matching with finite automata, Knuth-Morris-Pratt algorithm.

Module 3

Approximation algorithms: NP completeness, Reductions, coping with NP completeness, Approximation algorithms: The vertex cover problem, The travelling salesman problem, The set covering problem, The Subset-sum problem. Graph colouring.

Module 4

Randomized algorithms: Las Vegas and Monte Carlo algorithm, Random variables and their expectations. probabilistic analysis and uses of indicator random variables: Birthday paradox, coupon collector's problem, The online hiring problem. Randomized version of quick sort, Miller Rabin randomized primality Test.

Text Book

T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein. "*Introduction to Algorithms*," Third edition ,Prentice Hall India, 2011

References:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford stein, "*Introduction to Algorithms*", second edition, Prentice Hall India, 2009.
2. Sara. Basse, Allen Van Gelder, "*Computer Algorithms: Introduction to Design and Analysis*", Pearson, 2000.
3. R. Motwani and P. Raghavan, "*Randomized Algorithms*," Cambridge University Press, 1995.
4. Dexter C .Kozen, "*The Design and Analysis of Algorithms*," Springer, 1992
5. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C++*," Third edition, Pearson 2007.
6. Michael Sipser, "*Introduction to theory of computation*", Thomson Course technology, 2006.
7. Alfred V Aho, Jeffrey D Ullman, John E Hopcroft, "*Data Structures and Algorithms*", Pearson, 1983
8. Sartaj Sahni, "*Data Structures, algorithms & applications In C++*," university press, 2008.

OPERATING SYSTEM DESIGN

MCSSE 103

SUBJECT DESCRIPTION AND OBJECTIVES

The aim of this subject is to give a clear understanding of the concept of Operating System Design. First module deals with an overview of the UNIX system followed by an in depth study of its file subsystems, processes and memory management policies. Upon completion of this subject, the student will get a thorough knowledge of operating system concepts and will be familiar with the implementation of some of the concepts.

Module 1

General Overview of the System: History – System structure – User perspective – Operating system services – Assumptions about hardware. Introduction to the Kernel : Architecture of the UNIX operating system – Introduction to system concepts – Kernel data structures – System administration. The Buffer Cache: Buffer headers – Structure of the buffer pool – Scenarios for retrieval of a buffer – Reading and writing disk blocks – Advantages and disadvantages of the buffer cache.

Module 2

File Subsystems: INODES – Structure of a regular file – Directories – Conversion of a path name to an INODE – Super block – INODE assignment to a new file – Allocation of disk blocks. System Calls for the file system: Open – Read – Write – File and record locking – Adjusting the position of file I/O – lseek – close – File creation – Creation of special files – Changing directory, root, owner, mode – stat and fstat – Pipes – Dup – Mounting and unmounting file systems – link – unlink – File system abstraction and maintenance

Module 3

Processes: Process states and transitions – Layout of system memory – The context of a process – Saving the context of a process – Manipulation of the process address space - Sleep. Process Control : Process creation – Signals – Process termination – Awaiting process termination – Invoking other programs – user id of a process – Changing the size of a process - Shell – System boot and the INIT process– Process Scheduling.

Module 4

Memory Management Policies: Swapping – Demand paging – Hybrid System. The I/O Subsystem : Driver Interface – Disk Drivers – Terminal Drivers– Streams – Inter process communication: Process tracing – System V IPC – Network Communications – Sockets.

Text Book

Maurice J. Bach, “*The Design of the Unix Operating System*”, First Edition, Prentice Hall of India, 1986.

References:

1. Uresh Vahalia, "*Unix Internals – The new Frontiers*", Pearson Education, 2006
2. B. Goodheart, J. Cox, "*The Magic Garden Explained*", Prentice Hall of India, 1986.
3. S. J. Leffler, M. K. McKusick, M. J. Karels and J. S. Quarterman., "*The Design and Implementation of the 4.3 BSD Unix Operating System*", Addison Wesley, 1998

AUTOMATA THEORY AND COMPUTABILITY

MCSSE 104

SUBJECT DESCRIPTION AND OBJECTIVES

This subject is the corner stone of many other core subjects like Compiler, Algorithms, Formal Verification and Program Analysis in Computer Science. The topics are also important for Digital Circuit Design for state minimisation and analysis. The subject introduces rigorous formal methods for specifying various aspect of software which is a promising factor as far the software Industry is concerned. In spite of these, this subject is an indispensable area in Computer Science and Systems Engineering.

Module 1

Finite Automata : DFA, NFA – Examples, Equivalence (subset construction), Closure Properties (proof), Homomorphisms, Limitations of Finite Automata, Pumping Lemma and Ultimate Periodicity, DFA State Minimization (quotient construction), Myhill-Nerode Relations, Myhill-Nerode Theorem; problems

Module 2

Regular Expressions and Logical Characterization of Regular Languages : Pattern Matching, Pattern Matching and Regular Expressions, Regular Expressions and Finite Automata, Buchi's Logical Characterisation of Regular Language,. Proof.

Module 3

Pushdown Automata and Context-Free Languages : Context-Free Grammars and Languages, examples (proving correctness with mathematical induction), Chomsky Normal Form, Pumping Lemma for CFL, Parikh's theorem, Pushdown Automata, Final State Versus Empty Stack, PDAs and CFGs, Simulating NPDA by CFGs, Deterministic Pushdown Automata.

Module 4

Turing Machines and Effective Computability : Turing machine, examples, Equivalent Models, Universal Turing Machine and Diagonalization, Decidable and Undecidable Problems, Reduction-problems, Rice's Theorem -Proof, Undecidable problems about Context Free Languages.

References:

1. Dexter Kozen: “*Automata and Computability*”. Springer 1999.
2. Hopcroft J.E. and Ullman J.D.: “*Introduction to Automata, Languages and Computation*”. Addison Wesley, 1979.
3. H. R. Lewis and C. H. Papadimitriou:” *Elements of the Theory of Computation*”. Prentice Hall, 1981.
4. Note on Buchi's MSO characterisation of regular languages
(<http://drona.csa.iisc.ernet.in/~deepakd/atc-2011/atc.html>)

EMBEDDED SYSTEMS DESIGN

MCSSE 105-1

SUBJECT DESCRIPTION AND OBJECTIVES

This subject provides an opportunity to gain an in-depth understanding of embedded system development. The architecture and instruction set of Intel 8051 micro controller will be discussed. Students will have the opportunity to learn the basic building blocks of a typical RTOS and their design.

Module 1

Introduction to Embedded systems. Classification of Embedded Systems, Major application areas of Embedded Systems, Purpose of Embedded systems. Examples of Embedded Systems. Advanced Hardware Fundamentals. Watchdog Timers.

Module 2

The 8051 Architecture : Introduction, 8051 Micro controller.8051 Assembly language programming. I/O port Programming. Addressing modes in 8051. Arithmetic Operations. Data Transfer and Logical Instructions.

Module 3

Introduction to Real – Time Operating Systems: Tasks and Task States, Tasks and Data, Semaphores, and Shared Data; Message Queues, Mailboxes and Pipes, Timer Functions, Events, Memory Management, Interrupt Routines in an RTOS Environment.

Module 4

Basic Design Using a Real-Time Operating System: Principles, Semaphores and Queues, Hard Real- Time Scheduling Considerations, Saving Memory and Power, An example RTOS like uC-OS (Open Source); Embedded Software Development Tools: Host and Target machines, Linker/Locators for Embedded Software, Getting Embedded Software into the Target System; Debugging Techniques: Testing on Host Machine, Using Laboratory Tools, An Example System.

References:

1. David E.Simon, “*An Embedded Software Primer*”, Pearson Education Asia, First Indian Reprint 2000.
2. Kenneth J.Ayala, Thomson. “*The 8051 Microcontroller* “ , Third Edition
3. Rajkamal,” *Embedded Systems Architecture, Programming and Design*”, TATA McGraw Hill, First reprint 2003.

DIGITAL IMAGE COMPUTING

MCSSE 105-2

SUBJECT DESCRIPTION AND OBJECTIVES

Digital image computing is one of the most rapidly growing fields in engineering. Towards the beginning, the subject gives an overview of computer imaging systems and processing images in spatial and frequency domains. Later modules will touch upon the topics such as colour image processing, wavelets, multi-resolution processing, image compressions, image segmentation and mathematical morphology. Sound knowledge of this subject will help the students to do research works in growing engineering fields.

Module 1

Overview of Computer Imaging Systems: Fundamental Steps in Digital Image Processing. Sampling and Quantization. Components of Image. Image Enhancement in Spatial Domain: Basic Gray level Transformations-Histogram Processing-Enhancements using Arithmetic/logic Operations-Basics of Spatial Filtering-Smoothing Filters-Sharpening Filters.

Module 2

Processing Images in Frequency Domain : Fourier Transform and the Frequency Domain. Smoothing Filters-Sharpening Filters- Homomorphic Filters - Image Restoration: Degradation model –Noise Models-Restoration in the presence of Noise- Wiener Filter. Color Image Processing : Color Fundamentals-Color Models-Basics of full Color image Processing-Color Transformations.

Module 3

Wavelets and Multi-resolution Processing: Multiresolution Expansions. Wavelet Transforms in One Dimension. The Fast Wavelet Transform. Image Compressions: Fundamentals – Image Compression models – Elements of Information Theory – Error – Free Compression – Lossy compression – Image Compression standards.

Module 4

Image Segmentation: Detection of discontinuities – Edge linking and Boundary detection – Thresholding – Region Based Segmentation. Mathematical Morphology. Operations on Binary and Grayscale Images. Basic Morphological Algorithms.

References:

1. Rafael C.Gonzalez and Richard E.Woods, “*Digital Image Processing*”, Third Edition, Pearson Education, 2008.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, “*Image Processing, Analysis and Machine Vision*”, Third Edition, Brooks Cole, 2008.
3. Anil K.Jain, “*Fundamentals of Digital Image Processing*”, Prentice- Hall India, 2007
4. Frank Y. Shih “*Image Processing and Mathematical Morphology: Fundamentals and Applications*” CRC Press, 2009.

ADVANCED DATABASES

MCSSE 105-3

SUBJECT DESCRIPTION AND OBJECTIVES

Database plays an inevitable part in computer system engineering. This subject helps to develop an overall understanding of advanced Databases like Object-Oriented Databases, Distributed Databases, Web Databases and Mobile Databases. The ultimate purpose is to expose the students, in such a way as to enable them design and implement database solutions for general applications.

Module 1

Object-Oriented Databases: Overview of Object-Oriented concepts, Object identity, Object structure, and type constructors, Encapsulation of operations, Methods, and Persistence, Type hierarchies and Inheritance, Type extents and queries, Complex objects; Database schema design for OODBMS; OQL, Persistent programming languages; OODBMS architecture and storage issues; Transactions and Concurrency Control, Example of ODBMS.

Module 2

Distributed Databases: Distributed Database Concepts, Types of Distributed Database systems, Distributed Database Architectures, Data Fragmentation, Replication and Allocation Techniques for Distributed Database systems, Query Processing and Optimization in Distributed Databases, Overview of Transaction Management, Concurrency Control and Recovery in Distributed Databases, Distributed Catalog Management.

Module 3

Databases on the Web and Semi Structured Data: Web interfaces to the Web, Overview of XML; Structure of XML data, Document schema, Querying XML data; Storage of XML data, XML applications; The semi structured data model, Implementation issues, Indexes for text data.

Module 4

Emerging Database Technologies and Applications - Mobile Databases, Multimedia Databases, Geographic Information Systems (GIS), Temporal Databases, Spatial Databases, Genome Data Management.

References:

1. Korth, Silberschatz, " *Database System Concepts* " , 4th Ed., TMH, 2000.
2. Ramez Elmasri & Shamkant B.Navathe, " *Fundamentals of Database Systems* ", Fourth Edition , Pearson Education ,2004.
3. M.Tamer Ozsu , Patrick Ualdurriel, " *Principles of Distributed Database Systems* ", Second Edition, Pearso nEducation, 2003.
4. C.S.R.Prabhu, " *Object Oriented Database Systems* ", PHI, 2003.
5. Peter Rob and Coronel, " *Database Systems, Design, Implementation and Management* " , Thomson Learning.
6. C.J.Date, Longman, " *Introduction To Database Systems* ", Pearson Education
7. Chhanda Ray " *Distributed Database Systems* ", Pearson Education, 2009

GRAPH THEORY

MCSSE 105-4

SUBJECT DESCRIPTION AND OBJECTIVES

Graph theory is an essential course for a PG course in Computer Science. Different types of graphs, theorems and techniques selected have profound applications in different areas of study in Computer Science and Engineering. Examples include Algorithms, Complexity Theory, Networks, etc. Graph theory has enormous application in modelling problems of Computer Science. Solution to such problems can be easily obtained by appeal to the known results in Graph Theory.

Module 1

Matching : Review of Basic Concepts, Bipartite Matching - Hall's Matching Condition, Konig's Min-Max theorems, Gallai's Min-Max Theorem, Maximum Bipartite Matching using augmenting path algorithm. Matchings in general graphs – Tutte's 1-factor theorem.

Module 2

Connectivity : Cuts and Separators, Edge connectivity, Vertex Connectivity, Block-Cut Point Graph, Menger's theorem, Applications, Properties of 2- connected and 3-connected graphs.

Module 3

Coloring : Vertex Coloring – Greedy Coloring, Brook's Theorem. Edge Coloring – Vizing's Theorem. List Coloring. Perfect Graphs – Weak Perfect Graph Theorem, Statement of Strong Perfect Graph Theorem.

Module 4

Planarity : Euler's Formula, Minors, Kuratowski's Theorem. Planar Graph Coloring - Six Color Theorem, Kempe's Attempt of 4 Color Theorem, Heawood's 5 Color Theorem, Thomassen's List Color Theorem.

References:

1. D. B. West: “*Introduction to Graph Theory*”, Prentice Hall of India, 2001.
2. J. A. Bondy and U. S. R. Murty: “*Graph Theory*”, Springer 2008.
3. Reinhard Diestel: “*Graph Theory*”, Electronic Edition 2010.
4. B. Bollobas: “*Modern Graph Theory*” Springer, 1998.

COMPLEXITY THEORY

MCSSE 106-1

SUBJECT DESCRIPTION AND OBJECTIVES

Complexity Theory is an important study/research area in Computer Science. It deals with analysing the cost of finding solutions to problems in Computer Science. The applicability of solutions (programs) drastically depends on its cost in terms of time and memory usage. The purpose of this subject is to introduce different complexity classes and techniques for analysing the complexities. This subject is an essential part for a PG course in Computer Science & Systems Engineering.

Module 1

Time and Space Complexity : Time Complexity - P, NP, NP-Completeness, Cook-Levin Theorem and its proof, Examples of NP-Complete problems and reductions. Space Complexity – Savitch's Theorem, PSPACE, PSPACE Completeness of TQBF, L, NL, Logspace Reductions, NL-Completeness of PATH, NL=coNL.

Module 2

Intractability : Time and Space Hierarchy Theorems, Relativization, Baker-Gill-Solovay Theorem, Polynomial Hierarchy.

Module 3

Probabilistic Computation : Probabilistic Turing Machines, BPP, Amplification Lemma, RP, co-RP, ZPP, Probabilistic Algorithm for Primality Testing, IP=PSPACE.

Module 4

Advanced Topics : Parallel Computation- Circuit Complexity, NC and NC-hierarchy. P-Completeness. Karp-Lipton Theorem. Adleman's Theorem, Sipser-Gacs Theorem.

References:

1. Michael Sipser: "*Theory of Computation*". Course Technology 2007.
2. S. Arora, B. Barak, "*Computational Complexity: A Modern Approach* ", Cambridge University Press, 2009.
3. Papadimitriou C. H. "*Computational Complexity*", Addison Wesley, First Edition, 1993.

INFORMATION RETRIEVAL , DATA MINING AND DATA WAREHOUSING

MCSSE 106-2

SUBJECT DESCRIPTION AND OBJECTIVES

The first part of the subject aims to give a good overview of the ideas and techniques of Information retrieval which covers the tasks of indexing, searching, and recalling data, particularly text or other unstructured forms. The module II deals with ideas which are behind recent development in the data warehousing and online analytical processing (OLAP) fields. The rest of the modules cover data mining and clustering techniques. The objective is designing and managing data storages, data warehousing, OLAP, data mining techniques and to study theoretical aspects as well as implementation issues of classical and modern retrieval problems.

Module I

Study some basic concepts of information retrieval and data mining, such as the concept of relevance, association rules, and knowledge discovery. Understand the conceptual models of an information retrieval and knowledge discovery system. Indexing : Introduce various indexing techniques for textual information items, such as inverted indices, tokenization, stemming and stop words. Retrieval Methods: Study popular retrieval models. Other commonly-used techniques include relevance feedback, pseudo relevance feedback, and query expansion. Evaluation of Retrieval Performance : Measurements: Average precision, NDCG, etc. Cranfield paradigm and TREC conferences.

Module 2

Evolution of Decision Support Systems- Data warehousing Components–Data warehouse, Data Warehouse and DBMS, Data marts, Metadata, Multidimensional data model, OLAP ,OLTP, Data cubes, Schemas for Multidimensional Database: Stars, Snowflakes and Fact constellations. Types of OLAP servers, 3–Tier data warehouse architecture, distributed and virtual data warehouses. Data warehouse implementation, tuning and testing of data warehouse. Data Staging (ETL) Design and Development, data warehouse visualization, Data Warehouse Deployment, Maintenance, Growth, Business Intelligence Overview- Data Warehousing and Business Intelligence Trends - Business Applications- tools-SAS.

Module 3

Data mining-KDD versus data mining, Stages of the Data Mining Process-task primitives, Data Mining Techniques -Data mining knowledge representation – Data mining query languages, Integration of a Data Mining System with a Data Warehouse – Issues, Data pre-processing – Data cleaning, Data transformation, Feature selection, Dimensionality reduction, Discretization and generating - Mining frequent patterns-association and correlation

Module 4

Decision Tree Induction - Bayesian Classification – Rule Based Classification by Back propagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Clustering techniques – , Partitioning

methods- k-means- Hierarchical Methods - distance- based agglomerative and divisible clustering, Density-Based Methods – expectation maximization -Grid Based Methods – Model-Based Clustering Methods – Constraint – Based Cluster Analysis – Outlier Analysis, Practical retrieval and data mining applications. Currently available tools.

Text Books:

1. Jiawei Han and Micheline Kamber, “*Data Mining: Concepts and Techniques*”, Morgan Kaufmann Publishers, third edition 2011, ISBN: 1558604898.
2. Alex Berson and Stephen J. Smith, “*Data Warehousing, Data Mining & OLAP*”, TataMc Graw Hill Edition, Tenth Reprint 2007.
3. G. K. Gupta, “*Introduction to Data Mining with Case Studies*”, Easter Economy Edition, Prentice Hall of India, 2006.
4. MargaretH. Dunham, S.Sridhar, “*Data Mining : Introductory and Advanced Topics*”, Pearson Education.
5. Baeza-Yates and Ribeiro-Neto “*Modern Information Retrieval,*” Addison Wesley, 1999.

SOFT COMPUTING TECHNIQUES

MCSSE 106-3

SUBJECT DESCRIPTION AND OBJECTIVES

Soft Computing refers to a collection of computational techniques in computer science, artificial intelligence and engineering disciplines. This subject thoroughly discusses Soft Computing Techniques such as Artificial Neural Networks, Fuzzy Systems and Genetic Algorithms.

Module 1

Basics of Artificial Neural Networks: Characteristics of Neural Networks – Terminology – Models of neuron – Topology – Basic Learning Laws , Feed forward Neural Networks. Feedback Neural Networks.

Module 2

Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition -Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Module 3

Adaptive networks based Fuzzy interface systems - Classification and Regression-Trees - Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls - Simulated annealing – Evolutionary computation.

Module 4

Genetic Algorithms (GA) – Applications of GA in Machine Learning - Machine Learning Approach to Knowledge Acquisition. Support Vector Machines for Learning – Linear Learning Machines – Support Vector Classification – Support Vector Regression - Applications.

References:

1. Yegnanarayana B, "*Artificial Neural Networks*", PHI.
2. Jang J.S.R., Sun C.T. and Mizutani E, "*Neuro-Fuzzy and Soft computing*", Pearson Education 2003.
3. Timothy J.Ross, "*Fuzzy Logic with Engineering Applications*", McGraw Hill, 1997.
4. Mitchell Melanie, "*An Introduction to Genetic Algorithm*", Prentice Hall,1998.
5. David E. Goldberg, "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison Wesley, 1997.

COMPUTER GRAPHICS AND VISION

MCSSE 106-4

SUBJECT DESCRIPTION AND OBJECTIVES

Computer graphics is the art and science of communicating information using images that are generated and presented through computation. This requires (i) the design and construction of models that represent information in ways that support the creation and viewing of images (ii) the creation of techniques for rendering the model. The goal of computer vision is to deduce the properties and structure of the three dimensional world from one or more two-dimensional images. Knowledge of Computer graphics, Computer vision and Image processing as well as familiarity with OpenGL and OpenCV library will enable students to design and implement software for graphics and visual systems.

Module 1

Review of basic concepts: Antialiasing techniques, 2D viewing, 2D Transformations, Clipping : Line clipping ,Polygon Clipping, 3D display\$\$\$ methods, 3D Object Representation ,Three-Dimensional Viewing , Projections , 3D Transformations.

Module 2

Fractals – Classification of Fractals, Calculation of fractal dimension. Shape grammars. Visible surface detection algorithms, Surface Rendering Methods, Basic illumination Models, Polygon rendering Methods, Ray-Tracing Methods, Texture Mapping.

Module 3

Computer vision, applications, photometric image formation, digital camera. Feature-based alignment: 2D and 3D feature-based alignment, Pose estimation, Geometric intrinsic calibration. Structure from motion: Triangulation, Two-frame structure from motion, Factorization, Bundle adjustment, constrained structure and motion.

Module 4

Stereo correspondence: Epipolar geometry, sparse correspondence, dense correspondence, Local methods, Global optimization, Multi-view stereo. Recognition: Object detection, Face recognition, Instance recognition, Category recognition, Context and scene understanding.

References:

1. D Hearn and M. P. Baker, “*Computer Graphics C version*”, Pearson Education, 2004.
2. Richard szeliski ,”*Computer Vision: Algorithms and applications* “, Springer,2011
3. David A Forsynth and Jean Ponce, “*Computer Vision- A modern approach*”, Pearson education series, 2003.

P.T.O

4. Milan Sonka, Vaclav Hlavac and Roger Boyle,” *Digital image processing and computer vision*”, Cengage learning, 2008.
5. Schalkoff R. J., “*Digital Image Processing and Computer Vision*,” John Wiley, 2004.
6. E. S. Angel, “*Interactive Computer Graphics, A top-down approach with OpenGL*,” (5e), Pearson Education, 2009.
7. J. Foley, A. Van Dam, Feiner and Hughes; “*Computer Graphics; Principles and practice*”; 2nd edn. In C; Addison Wesley, 1997.
8. Gary Bradski, Adrian Kaehler , “*Learning OpenCV: computer vision with the OpenCV library* “ O’Reilly,2008
9. R. Hartley, and A. Zisserman, “*Multiple View Geometry in Computer Vision*,” (2e), Cambridge University Press, 2004.
10. Xiang Z., Plastock R ,Schaum’s outline series: “*Computer Graphics*” (2nd Ed) (TMH),2000

OPERATING SYSTEM DESIGN LAB

MCSSE 107

1. Introduction to Linux – booting – login - simple commands
2. Wild card characters – grep - pipe- tee - command substitution - shell variables - subshells, filters - head, tail. cut, paste, sort, uniq, nl, join
3. Editors-Vi and Emacs
4. Shell Programming
5. Use of system calls – files – processes – I/O – IPC
6. Implement the following: Dining philosopher Problem, Producer Consumer problem, Binary Search Implementation using shell scripting, quick sort implementation using shell scripting, Message queue, Kernel compilation, System call implementation.
7. C/C++ programming on Unix/Linux – use of make, version control.
8. Unix / Linux sources – build, run kernel – small modifications.

SEMINAR I

MCSSE 108

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 first 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 Co-ordinator / 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.

OBJECT ORIENTED SYSTEMS ENGINEERING

MCSSE 201

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.

COMPUTER SYSTEM DESIGN AND ARCHITECTURE

MCSSE 202

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 indispensable 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.

COMPUTER SYSTEM ENGINEERING

MCSSE 203

SUBJECT DESCRIPTION AND OBJECTIVES

Computer System Engineering allows to develop the skills and expertise needed to design computer systems. To acquaint the students with the concepts of Systems Engineering and the techniques for controlling complexity; strong modularity using client-server design and impact of computer systems on society with case studies of working systems.

Module 1

Systems Engineering Basics: Origins of Systems Engineering. Power of Systems Engineering. Software Systems Engineering. Complexity in computer systems. Sources of Complexity. Coping with Complexity.

Module 2

Enforcing Modularity with Clients and Services: Client/Service Organization.RPC. Case Study: The Internet Domain Name System (DNS).

Enforcing Modularity with Virtualization. Enforcing Modularity in Memory. Virtualizing Memory. Multilevel Memories. Case Study: Evolution of Enforced Modularity in the Intel x86.

Module 3

Networks: Three-layer Model. Congestion Control. Case Study: Mapping the Internet to the Ethernet.**Fault tolerance:** Reliable Systems from Unreliable Components. Case Study: Fault tolerance in memory systems. **Atomicity:** All-or-nothing and Before-or-after. Case Study: Machine Language Atomicity

Module 4

Consistency: Constraints and Interface Consistency. Cache Coherence. Durable Storage. Reconciliation.

Information Security: Secure Systems. Authenticating Principals. Security Protocols. Case Study: Secure Socket Layer (SSL) protocol.

Text Book

Jerome H. Saltzer and M. Frans Kaashoek” *Principles of Computer System Design: An Introduction*” Morgan Kaufmann Publishers. 2009.

References:

1. 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.
2. Buede, Dennis M., John " *The Engineering Design of Systems : Models and Methods*", John Wiley & Sons, Inc., 2000

AUTOMATED VERIFICATION

MCSSE 204

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 indispensable 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.

PROGRAM ANALYSIS AND VERIFICATION

MCSSE 205-1

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,

Test Book:

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

ENGINEERING SYSTEM ARCHITECTURES

MCSSE 205-2

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/Sub-system 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.

INFORMATION SYSTEMS SECURITY

MCSSE 205-3

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

Review of number theory and algebra: Integers, Divisibility and Division Algorithm, Euclidean Algorithm, Fundamental Theorem of Arithmetic, Modular Arithmetic, Congruences and Residues, Euler's Phi Function. The Theorems of Fermat, Euler and Lagrange. Quadratic Residues, Chinese Remainder Theorem, Primitive Roots and Discrete Logarithm, Primes, Miller-Rabin primality Test.

Module 2

Public-Key Cryptography: Concept of Public-Key Cryptography; RSA System – Key Generation and Encryption, Digital Signatures, Attacks; Elgamal's Encryption; Elgamal's Encryption Schemes; Digital Signature Algorithm.

Module 3

Cryptographic Protocols: *Key Exchange and Authentication* – Kerberos, Diffie-Hellman Key Agreement, Key Exchange and Mutual Authentication; *Identification Schemes* – Interactive Proof System, Simplified Fiat-Shamir Identification Scheme, Zero-Knowledge, Fiat-Shamir Identification Scheme, Fiat-Shamir Signature Scheme.

Module 4

Authentication Protocols: Introduction, Authentication and Refined Notions, Convention, Basic Authentication Techniques, Password-based Authentication, Authenticated Key Exchange Based on Asymmetric Cryptography, Typical Attacks on Authentication Protocols.

References:

1. W. Mao, “*Modern Cryptography: Theory & Practice*”, Pearson Education, 2004.
2. W. Stallings, “*Cryptography and Network Security Principles and practice*,” 3/e, Pearson Education Asia, 2003
3. H. Delfs and H. Knebl, “*Introduction to Cryptography: Principles and Applications*”, Springer-Verlag, 2002.

SEMANTIC WEB

MCSSE 205-4

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 introduces 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, languages and tools for knowledge management.

Module 2

Ontology-based Systems: Ontology based knowledge management, ontology construction, generating, storing, aligning and maintaining ontologies for semantic web. Languages to represent ontology XML, RDF, OIL, DAML and OWL

Module 3

Information retrieval from natural language based documents; ontology evolution; ontological indexing and searching techniques for searching web

Module 4

Tools for ontology construction, 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.

PROGRAMMING LANGUAGES AND TYPE SYSTEM

MCSSE 206-1

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, extension and Type Inference : Function Types, Typing Relation, Properties of Typing, Base Types, Unit type, Sequencing and 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.

Text Book:

Benjamin C. Pierce, “*Types and Programming Languages*”, MIT Press, 2002

References:

1. David A. Schmidt. “*Programming Language Semantics*”., Handbook of Computer Science and Engineering. CRC Press, 1996.
2. Michael L. Scott. “*Programming Language Pragmatics*” Elsevier, 2004

SYSTEMS MODELING AND SIMULATION

MCSSE 206-2

SUBJECT DESCRIPTION AND OBJECTIVES

This subject covers modelling 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.

ADVANCED SOFTWARE TESTING

MCSSE 206-3

SUBJECT DESCRIPTION AND OBJECTIVES

This subject provides the knowledge of a variety of ways to test software and understanding of some of the trade offs 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

DATA COMPRESSION TECHNIQUES

MCSSE 206-4

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 performance, 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 Huffman 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 Quantization: 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.

COMPUTER AIDED SOFTWARE ENGINEERING LAB

MCSSE 207

- 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.

SEMINAR II

MCSSE 208

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 Co-ordinator / 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.

SEMESTER III

INDUSTRIAL TRAINING AND MINI PROJECT

MCSSE 301

Objective: To provide exposure to industrial environment and develop practical ability and knowledge about practical tools/techniques in order to solve real life problems.

The student shall undergo industrial training for one month or 20 number of days and a Mini Project. Industrial training should be carried out in an industry / company approved by the institution and under the guidance of a staff member in the concerned field. At the end of the training he / she has to submit a report on the work being carried out. A detailed project report duly approved by the guide in the prescribed format should be submitted for end semester assessment. Marks will be awarded based on the report and their performance during presentations and demonstrations. Equal weightage for Mini Project and Industrial Training.

MASTER'S THESIS (PHASE – I)

MCSSE 302

The master's thesis (Phase – I) shall consist of research work done by the candidate or a comprehensive and critical review of any recent development in the subject or a detailed report of project work consisting of experimentation / numerical work, design and or development work that the candidate has executed.

In Phase – I of the thesis it is expected that the student should decide a topic of thesis, which is useful in the field or practical life. It is expected that students should refer national and international journals, proceedings of national and international seminars. Emphasis should be given to the introduction to the topic, literature review, and scope of the proposed work along with some preliminary work / experimentation carried out on the thesis topic.

Student should submit Phase – I thesis report in two copies covering the content discussed above and highlighting the features of work to be carried out in part – I of the thesis. Student should follow standard practice of thesis writing.

The candidate will deliver a talk on the topic and the assessment will be made on the basis of the term work and talks there on by a panel of internal examiners one of which will be the internal guide. These examiners should give suggestions in writing to the student to be incorporated in thesis work Phase – II.

SEMESTER IV

MASTER'S THESIS EVALUATION

MCSSE 401

In the fourth semester the student has continue the thesis work and present the report. At the end of successfully finishing the work he / she has to submit a detailed report and has to present for a viva – voce.

MASTER'S COMPREHENSIVE VIVA

MCSSE 402

A Comprehensive viva-voce examination will be conducted at the end of the fourth semester by an internal and external examiners appointed by the university to assess the candidates overall knowledge in the field of specialization.