Scheme of Study for Master of Computer Science
MCS
2-year programme (4 semesters)

#	Category	# of Courses	Credit Hours
1	**Computing Core Courses** (Comp-Core)	**12**	**43**
2	**Computer Science Courses**	**9**	**27**
	CS Core Courses (CS-Core)	6	18
	CS Elective Courses (CS-Elec)	3	9
3	**University Electives** (Univ-Elec)	**2**	**6**
	Total	**23**	**76**

Semester-I: (20 Credit Hours)

#	*Course Code*	*Category*	*Course Title*	*Credit Hours*	*Prerequisites*
1	CS 301	Comp-Core	Introduction to Computing	4 (3,1)	
2	CS 303	Comp-Core	Programming Fundamentals	4 (3,1)	
3	CS 305	Comp-Core	Discrete Structures	3 (3,0)	
4	CS 307	Comp-Core	Digital Logic Design	3 (3,0)	
5	* 309	Univ-Elec	Univ Elective-1	3 (3,0)	
6	* 311	Univ-Elec	Univ Elective-2	3 (3,0)	

* *Two alphabetic characters (EG or MG or SS) to be used for the respective course from the university elective course list.*

Semester-II: (19 Credit Hours)

#	*Course Code*	*Category*	*Course Title*	*Credit Hours*	*Prerequisites*
1	CS 302	CS-Core	Computer Organization and Assembly Language	3 (2,1)	Digital Logic Design
2	CS 304	CS-Core	Theory of Automata and Formal Languages	3 (3,0)	Discrete Structures
3	CS 306	Comp-Core	Object Oriented Programming	4 (3,1)	Programming Fundamentals
4	CS 308	Comp-Core	Intro. to Database Systems	3 (2,1)	
5	CS 310	Comp-Core	Computer Communications and Networks	3 (3,0)	
6	CS 312	Comp-Core	Intro. to Software Engineering	3 (3,0)	

Semester-III: (19 Credit Hours)

#	*Course Code*	*Category*	*Course Title*	*Credit Hours*	*Prerequisites*
1	CS 401	CS-Core	Computer Architecture	3 (3,0)	Computer Org. and Assembly Language
2	CS 403	CS-Core	Artificial Intelligence	3 (2,1)	Discrete Structures
3	CS 405	Comp-Core	Operating Systems	3 (3,0)	
4	CS 407	Comp-Core	Data Structures and Algorithms	4 (3,1)	Programming Fundamentals
5	CS 409	CS-Elec	CS Elective-1	3	
6	CS 411	CS-Elec	CS Elective-2	3	

Semester-IV: (18 Credit Hours)

#	*Course Code*	*Category*	*Course Title*	*Credit Hours*	*Prerequisites*
1	CS 402	CS-Core	Design & Analysis of Algorithms	3 (3,0)	Data Structures & Algorithms
2	CS 404	CS-Core	Compiler Construction	3 (2,1)	Theory of Automata & Formal Languages
3	CS 406	CS-Elec	CS Elective-3	3	
4	CS 408	Comp-Core	Human Computer Interaction	3 (2,1)	
5	CS 410	Comp-Core	Software Project/Thesis	6 (0,6)	

Electives for MCS

CS Elective Courses:

#	Course Code	Category	Course Title	Credit Hours	Prerequisites
1	CS	CS-Elec	Software Engineering II	3 (3,0)	Intro to Software Engg
2	CS	CS-Elec	Data Communications	3 (3,0)	
3	CS	CS-Elec	Principles of Programming Languages	3 (3,0)	
4	CS	CS-Elec	Computer Graphics	3 (2,1)	
5	CS	CS-Elec	Digital Image Processing	3 (2,1)	
6	CS	CS-Elec	Visual Programming	3 (2,1)	
7	CS	CS-Elec	Distributed Computing	3 (2,1)	
8	CS	CS-Elec	Network Security	3 (3,0)	Computer Comm. and Networks
9	CS	CS-Elec	Computer Vision	3 (3,0)	Data Struc. & Algo.
10	CS	CS-Elec	Systems Programming	3 (2,1)	Operating Systems
11	CS	CS-Elec	Distributed Database Systems	3 (2,1)	Intro. to Database Sys.
12	CS	CS-Elec	Data Warehousing	3 (3,0)	Intro. to Database Sys.
13	CS	CS-Elec	Web Engineering	3 (2,1)	
14	CS	CS-Elec	Artificial Neural Networks	3 (2,1)	Artificial Intelligence
15	CS	CS-Elec	Expert Systems	3 (2,1)	
16	CS	CS-Elec	Operations Research	3 (3,0)	
17	CS	CS-Elec	Network Programming	3 (2,1)	Computer Comm. and Networks
18	CS	CS-Elec	Wireless Networks	3 (3,0)	
19	CS	CS-Elec	Telecommunication Systems	3 (2,1)	
20	CS	CS-Elec	Mobile Computing	3 (2,1)	
21	CS	CS-Elec	Java Programming	3(2,1)	
22	CS	CS-Elec	Android Programming	3(2,1)	Java Programming
23	CS	CS-Elec	Cloud Computing	3(2,1)	Distributed Computing
24	CS	CS-Elec	Cyber Security	3 (3,0)	
25	CS	CS-Elec	Object-Oriented Analysis & Design	3(3,0)	Intro to Software Engg
26	CS	CS-Elec	Ethical Hacking	3(2,1)	

University Electives Courses:

#	*Course Code*	*Category*	*Course Title*	*Credit Hours*
1	EG	Univ-Elec	Business Communications and Technical Writing	3 (3,0)
2	MG	Univ-Elec	Financial Accounting	3 (3,0)
3	MG	Univ-Elec	Financial Management	3 (3,0)
4	MG	Univ-Elec	Human Resource Management	3 (3,0)
5	MG	Univ-Elec	Marketing	3 (3,0)
6	SS	Univ-Elec	Psychology	3 (3,0)
7	SS	Univ-Elec	Foreign/Regional Languages (French, German, Chinese, Japanese, Russian, Sindhi, Punjabi, Balochi, Pashto etc.)	3 (3,0)

Detailed Courses Outline for MCS

Description of Computing-Core and CS-Core Courses

Course Name: Introduction to Computing
Course Structure: Lectures: 3, Labs: 1 **Credit Hours:** 4
Objectives: This course focuses on a breadth-first coverage of computer science discipline, introducing computing environments, general application software, basic computing hardware, operating systems, desktop publishing, Internet, software applications and tools and computer usage concepts; Introducing Software engineering and Information technology within the broader domain of computing, Social issues of computing.
Course Outline: Number Systems, Binary numbers, Boolean logic, History of computer system, basic machine organization, Von Neumann Architecture, Algorithm definition, design, and implementation, Programming paradigms and languages, GUI programming, Overview of Software Engineering and Information Technology, Operating system, Compiler, Computer networks and internet, Computer graphics, AI, Social and legal issues.
Reference Material:
1. "Computers: Information Technology in Perspective, 9/e", Larry Long and Nancy Long, Prentice Hall, 2002 / ISBN: 0130929891
2. "An Invitation to Computer Science", Schneider and Gersting, Brooks/Cole Thomson Learning, 2000
3. " An overview of Computer Science", Sherer

Course Name: Programming Fundamentals
Course Structure: Lectures: 3, Labs: 1 **Credit Hours:** 4
Objectives: The course is designed to familiarize students with the basic structured programming skills. It emphasizes upon problem analysis, algorithm designing, and program development and testing.
Course Outline: Overview of computers and programming. Overview of a computer language, for example, C language. Basics of structured and Modular programming. Basic algorithms and problem solving, development of basic algorithms, analyzing a problem, designing solution, testing designed solution. Fundamental programming constructs, translation of algorithms to programs, data types, control structures, functions, arrays, records, files, testing programs.
Reference Material:
1. "Problem Solving and Program Design in C / 6th Ed.", Hanly & Koffman Addison-Wesley Publisher 2009. ISBN-10: 0321535421 | ISBN-13: 9780321535429
2. "C How to Program 5th Ed", (Harvey & Paul) Deitel & Deitel, ISBN-10: 0132404168 ISBN-13: 9780132404167 Publisher: Prentice-Hall 2007

Course Name: Object Oriented Programming
Course Structure: Lectures: 3, Labs: 1 **Credit Hours:** 4
Prerequisites: Programming Fundamentals
Objectives: The course aims to focus on OO concepts, analysis and software development.
Course Outline: Evolution of Object Oriented (OO) programming, OO concepts and principles, problem solving in OO paradigm, OO program design process, classes, methods, objects and encapsulation; constructors and destructors, operator and function overloading, virtual functions, derived classes, inheritance and polymorphism. I/O and file processing, exception handling.

Reference Material:
1. "C++ How to Program, 6th Ed.", (Harvey & Paul) Deitel & Deitel ISBN-10: 0136152503 ISBN-13: 9780136152507 Publisher: Prentice Hall
2. "Java How to Program, 7th Ed.", (Harvey & Paul) Deitel & Deitel ISBN-10: 0132222205 ISBN-13: 9780132222204 Publisher: Prentice Hall.

Course Name: Discrete Structures

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Objectives: This course introduces the foundations of discrete mathematics as they apply to Computer Science, focusing on providing a solid theoretical foundation of the subject matter. Further, this course aims to develop understanding and appreciation of the finite nature inherent in most Computer Science problems and structures through study of combinatorial reasoning, abstract algebra, iterative procedures, predicate calculus, tree and graph structures. In this course emphasis is given to statistical and probabilistic formulation with respect to computing aspects.

Course Outline: Introduction to logic and proofs: Direct proofs; proof by contradiction, Sets, Combinatorics, Sequences, Formal logic, Prepositional and predicate calculus, Methods of Proof, Mathematical Induction and Recursion, loop invariants, Relations and functions, Pigeonhole principle, Trees and Graphs, Elementary number theory, Optimization and matching. Fundamental structures: Functions; relations (more specifically recursions); pigeonhole principle; cardinality and countability, probabilistic methods.

Reference Material:
1. "Discrete Mathematics and Its Applications, 6TH edition", Kenneth H. Rosen, McGraw Hill Book Co. 2006
2. "Discrete Mathematics, 7TH edition", Richard Johnsonbaugh, Prentice-Hall, 2008.
3. "Discrete Mathematical Structures, 4th edition", Kolman, Busby & Ross, Prentice-Hall Publishers 2000.
4. "Discrete and Combinatorial Mathematics: An Applied Introduction", Ralph P. Grimaldi, Addison-Wesley Pub. Co., 1985.

Course Name: Operating Systems

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Objectives: To help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with the layered approach that makes design, implementation and operation of the complex OS possible.

Course Outline: History and Goals, Evolution of multi-user systems, Process and CPU management, Multithreading, Kernel and User Modes, Protection, Problems of cooperative processes, Synchronization, Deadlocks, Memory management and virtual memory, Relocation, External Fragmentation, Paging and Demand Paging, Secondary storage, Security and Protection, File systems, I/O systems, Introduction to distributed operating systems. Scheduling and dispatch, Introduction to concurrency. Assignments involving different single and multithreaded OS algorithms.

Reference Material:

1. "Operating Systems Concepts, 7th Edition", Silberschatz A., Peterson, J.L., & Galvin P.C. 2004.

2. "Modern Operating Systems, 3rd Edition", Tanenbaum A.S., 2008.

Course Name: Introduction to Database Systems

Course Structure: Lectures: 2, Labs: 1 **Credit Hours:** 3

Objectives: The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design methods. The course primarily focuses on relational data model and DBMS concepts.

Course Outline: Basic database concepts; Entity-Relationship modeling, Relational data model and algebra, Structured Query language; RDBMS; Database design, functional dependencies and normal forms; Physical database design: Storage and file structure; indexed files; b-trees; files with dense index; files with variable length records; database efficiency and tuning. Transaction processing and optimization concepts; concurrency control and recovery techniques; Database security and authorization. Small *Group Project* implementing a database.

Reference Material:

1. "Introduction to Database Systems", C.J. Date, Addison Wesley Pub. Co. 2004.
2. "Database Systems: A Practical Approach to Design, Implementation and Management", R.Connolly and P.Begg, Addison-Wesley Pub. Co. 2009.
3. "Fundamentals of Database Systems", Elmasri and Navathe, Addison-Wesley, ISBN: 0-201-74153-9.

Course Name: Introduction to Software Engineering

Course Structure: Lectures: 3, Lab: 0 **Credit Hours:** 3

Objective: To study various software development models and phases of software development life cycle (SDLC). The concepts of project management, change control, process management, software development and testing are introduced through hands-on Team Projects. The students will study techniques for software verification, validation and testing. They would also study reliability and performance issues in software design and development. Upon successful completion of this course the student will be to understand the importance of software engineering to computer science and the most important general approaches to structuring the software production process, analyze the requirements for a software system and produce a software design from requirements, assess software productivity using metrics, use different testing techniques used in software engineering to test software systems, manage the important issues for planning a project.

Course Outlines:

Introduction to Software Engineering, Software Process Framework, Process Models, Agile Software Process, Software Engineering Practices, System Engineering, Requirements Engineering, Analysis Modelling, Design Engineering, Architectural Design, Component Design, User Interface Design, Testing Strategies, Testing Tactics, Product and Process, Metrics, Project Management, Project Estimation, Project Scheduling, Risk Management, Quality Management, Change Management.

Reference Material:

1. "Software Engineering: A Practitioner's Approach", Roger Pressman, McGraw-Hill, 6th

edition 2005.
2. “Software Engineering”, Ian Sommerville. Addison-Wesley, 2001 (7th edition).
3. “UML Distilled”.

Course Name: Computer Communication and Networks

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Objectives: To introduce students to the concept of computer communications. Analog and digital transmission. Network Layers, Network models (OSI, TCP/IP) and Protocol Standards. Emphasis is given on the understanding of modern network concepts.

Course Outline: Analog and digital Transmission, Noise, Media, Encoding, Asynchronous and Synchronous transmission, Protocol design issues. Network system architectures (OSI, TCP/IP), Error Control, Flow Control, Data Link Protocols (HDLC, PPP). Local Area Networks and MAC Layer protocols (Ethernet, Token ring), Multiplexing, Switched and IP Networks, Inter-networking, Routing, Bridging, Transport layer protocols TCP/IP, UDP. Network security issues. Programming exercises, labs or projects involving implementation of protocols at different layers.

Reference Material:
1. “Introduction to Computer Networks”, A. S. Tanenbaum, Prentice Hall 2003
2. “Computer Networks and Internets”, Douglas E. Comer, Purdue University ISBN-10: 0136061273 ISBN-13: 9780136061274 Publisher: Prentice Hall 2008
3. “Data and Computer Communications”, W. Stallings, Macmillan Pub. , 8th Ed., 2006
4. “Data Communications and Networking” (4th edition), Behrouz A. Forouzan, McGraw-Hill, 2006. **ISBN-13:** 978-0073250328

Course Name: Human Computer Interaction

Course Structure: Lectures: 2, Labs:1 **Credit Hours:** 3

Objectives: This course introduces the human issues of usability and its importance. It considers the implications of human understanding on the usability of computer systems and the importance of understanding the context of use. It describes guidelines for use of different media and interface styles. Topics include Usability Design principals, standards and models, evaluation techniques. Groupware, pervasive and ubiquitous applications.

Course Outlines: The Human, Computer and Interaction, Usability paradigm and principles, Introduction to design basics, HCI in software process, Design rules, prototyping, evaluation techniques, task analysis, Universal design and User support and Computer Supported Cooperative Work. Introduction to specialized topics such as Groupware, pervasive and ubiquitous applications.

Reference Material:
1. “Human-Computer Interaction, 3/E”, Alan Dix, Janet E. Finlay, Gregory D. Abowd, Russell Beale, ISBN-10: 0130461091 ISBN-13: 9780130461094 Prentice Hall
2. “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, 4/E Ben Shneiderman, Catherine Plaisant, ISBN-10: 0321197860 ISBN-13: 9780321197863 Publisher: Addison-Wesley
3. “Designing Interfaces: Patterns for Effective Interaction Design”, (2nd Ed.) Jennifer Tidwell. Publisher: O'Reilly Media, 2011

Course Name: Computer Organization and Assembly Language
Course Structure: Lectures: 2, Labs: 1 **Credit Hours:** 3
Prerequisites: Digital Logic Design
Objectives: The main objective of this course is to introduce the organization of computer systems and usage of assembly language for optimization and control. Emphasis is given to expose the low-level logic employed for problem solving while using assembly language as a tool. At the end of the course the students should be capable of writing moderately complex assembly language subroutines and interfacing them to any high level language.
Course Outline: Microprocessor Bus Structure: Addressing, Data and Control, Memory Organization and Structure (Segmented and Linear Models), Introduction to Registers and Flags, Data Movement, Arithmetic and Logic, Program Control, Subroutines, Stack and its operation, Peripheral Control Interrupts, Interfacing with high level languages, Real-time application.
Objectives and Perspectives of Assembly Language, Addressing Modes, Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code, Describe actions inside the processing chip, Discuss operations performed by an instruction set, Write a fully documented program, Using an assembler of choice.
Reference Material:
1. "Computer Organization & Architecture", 7th ed, W. Stallings, Prentice Hall, 2006.
2. “Assembly Language for Intel-based Computers”, 5th ed, Irvine, Prentice Hall, 2007.
3. “Computer Organization and Design, The Hardware/Software Interface”, 4th ed, David A. Patterson and John L. Hennessy, 2008. Elsevier Publishers.

Course Name: Theory of Automata and Formal languages
Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3
Prerequisites: Discrete Structures
Objectives: The course aims to develop an appreciation of the theoretical foundations of computer science through study of mathematical & abstract models of computers and the theory of formal languages. *Theory of formal languages* and use of various abstract machines as ‘recognizers’ and parsing will be studied for identifying/validating the synthetic characteristics of programming languages. Some of the abstract machines shall also study as ‘Transducers’.
Course Outline: *Finite State Models:* Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene’s theorem, Transducers (automata with output), Pumping lemma and non-regular language. *Grammars and PDA:* Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs , Normal form grammars and parsing, Decidability, Chomsky’s hierarchy of grammars *Turing Machines Theory:* Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Context sensitive Grammars, Defining Computers by TMs.
Reference Material:
1. “An Introduction to Formal Languages and Automata”, 4th ed., Peter Linz, Jones & Bartlett Publishers, 2006
2. “Theory of Automata, Formal Languages and Computation”, S. P. Eugene, Kavier, 2005, New Age Publishers, ISBN-10: 81-224-2334-5, ISBN-13: 978-81-224-2334-1.
3. “Introduction to Automata Theory, Languages, and Computation”, 2nd ed., John Hopcroft and Jeffrey Ullman, 2001, Addison-Wesley.

4. "Introduction to Languages and the Theory of Computation", 3rd ed., John C. Martin 2002, McGraw-Hill Professional.

Course Name: Design and Analysis of Algorithms
Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3
Prerequisites: Data Structures and Algorithms
Objectives: Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis to be on the structure, complexity, and efficiency of algorithms.
Course Outline: Introduction; Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms.
Reference Material:
1. "Introduction to Algorithms", T. H. Cormen, C. E. Leiserson, and R. L. Rivest, MIT Press, McGraw-Hill, New York, NY, 2001.
2. "Algorithms in C++", Robert Sedgewick

Course Name: Artificial Intelligence
Course Structure: Lectures: 2, Labs: 1 **Credit Hours:** 3
Prerequisites: Discrete Structures
Objectives: This course studies four main objectives of AI. Modeling the environment by constructing computer representations of the real world. Perception and reasoning - obtaining and creating information/*knowledge* to populate a computational representation. Taking actions by using the knowledge of the environment and desired goals to plan and execute actions. Learning from past experience.
Course Outline: Artificial Intelligence: Introduction, Intelligent Agents. Problem-solving: Solving Problems by Searching, Informed Search and Exploration, Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation. Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning, Statistical Learning Methods, Reinforcement Learning. Communicating, perceiving, and acting: Communication, Probabilistic Language Processing, Perception and Robotics. Introduction to LISP/PROLOG and Expert Systems (ES) and Applications.
Reference Material:
1. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", George F. Luger, 6th edition: Pearson Education, 2008.
2. "Artificial Intelligence: A Modern Approach", Stuart Jonathan Russell, Peter Norvig, John F. Canny, 2nd Edition, Prentice Hall, 2003.

Course Name: Computer Architecture
Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3
Prerequisites: Computer Organization and Assembly Language
Objectives: To get a deeper understanding of how computers work, working knowledge of various subsystems and the general principles that affect their performance, analyze the performance of systems and quantify the performance measurements, fundamentals of all technologies, and advanced architectural features that boost the performance of computers.
Course Outlines: Fundamentals of Computer Design including performance measurements & quantitative principles, principles of Instruction Set Design, Operands, addressing modes and encoding, pipelining of Processors: Issues and Hurdles, exception handling features, Instruction-Level Parallelism and Dynamic handling of Exceptions, Memory Hierarchy Design, Cache Design, Performance Issues and improvements, Main Memory Performance Issues, Storage Systems, Multiprocessors and Thread Level Parallelism. Case Studies.
Reference Material:

1. "Computer Architecture: A Quantitative Approach", Hennessy & Patterson, Morgan & Kauffman Series (2006) Fourth Edition.
2. "Computer Organization and Design: The Hardware/Software Interface", Patterson & Hennessy, Morgan & Kauffman Series 2008.

Course Name: Compiler Construction
Course Structure: Lectures: 2, Labs: 1 **Credit Hours:** 3
Prerequisites: Theory of Automata and Formal Languages
Objectives: To understand the overall structure of compilers and to know significant details of a number of important techniques commonly used in parsing. The students will get awareness of the way in which language features raise challenges for compiler builders.
Course Outline: Contrast between compilers and interpreters. Organization of compilers. Compiler techniques and methodology. Lexical and syntax analysis. Parsing techniques. Object code generation and optimization, detection and recovery from errors.
Reference Material:

1. "Compilers: Principles, Techniques, and Tools", Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Addison-Wesley Pub. Co., 2nd edition,1987
2. "Modern Compiler Implementation in C", Andrew W. Appel, Maia Ginsburg, Cambridge University Press, 2004.
3. "Modern Compiler Design", Dick Grune, Henri E. Bal, Ceriel J. H. Jacobs, Koen G. Langendoen, 2003, John Wiley & Sons.

Course Name: Software Project
Course Structure: Lectures: 0, Labs: 6 **Credit Hours:** 6
Objectives: The software project involves research, conceive, plan and develop a real and substantial project related to computer science. It provides an opportunity to the students to crystallize their acquired professional competence in the form of a demonstrable software product. Includes oral and written project presentations.
Reference Material:

1. "Software Project Management in Practice", Pankaj Jalote. Addison-Wesley 2002.

Description of Elective Courses

Course Name: Computer Graphics
Course Structure: Lectures: 2, Labs: 1 **Credit Hours:** 3
Objectives: Study of various algorithms in computer graphics and their implementation in any programming language.
Course Outline: Graphics hardware. Fundamental algorithms. Applications of graphics. Interactive graphics programming — graph plotting, windows and clipping, and segmentation. Programming raster display systems, Differential Line Algorithm, panning and zooming. Raster algorithms and software — Scan-Converting lines, characters and circles. Scaling, Rotation, Translation, Region filling and clipping. Two and three dimensional imaging geometry (Perspective projection and Orthogonal projection) and transformations. Curve and surface design, rendering, shading, colour and animation.
Reference Material:

1. "Computer Graphics, Principles and Practice," J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, Addison-Wesley ISBN: 0-201-12110-7.
2. "Computer Graphics", F.S.Hill, Maxwell MacMillan ISBN: 0-02-354860-6.
3. "Interactive Computer Graphics: Functional, Procedural and Device-level methods", Peter Burger and Duncan. F. Gillies; Addison-Wesley, (2003)

Course Name: Digital Image Processing
Course Structure: Lectures: 2 Labs: 1 **Credit Hours:** 3
Objective: The aim of this course is to understand the main terms & concepts of image processing systems & their techniques.
Course Outline: Digital Imaging. Image analysis and filtering. Restoration in the Presence of Noise Only–Spatial Filtering, Mean Filters, Order-Statistics Filters, Adaptive Filters, Periodic Noise Reduction by Frequency Domain Filtering, Bandreject Filters, Bandpass Filters, Notch Filters. Estimating the Degradation Function, Estimation by Image Observation, Estimation by Experimentation, Estimation by Modeling, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering. Image Segmentation, Detection of Discontinuities, Point Detection, Line Detection, Edge Detection, Edge Linking and Boundary Detection, Local Processing, Global Processing via the Hough Transform. Thresholding, The Role of Illumination, Basic Global Thresholding, Basic Adaptive Thresholding, Local Thresholding, Thresholds Based on Several Variables. Region-Based Segmentation, Region Growing, Region Splitting and Merging.

Course Name: Computer Vision
Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3
Prerequisites: Data Structures and Algorithms
Objectives: By the end of this course Students will be able to explain the concepts behind computer based recognition and the extraction of features from raster images. Students will also be able to illustrate some successful applications of vision systems and will be able to identify the vision systems limitations.

Course Outlines: Concepts behind computer-based recognition and extraction of features from raster images. Applications of vision systems and their limitations. Overview of early, intermediate and high level vision, Segmentation: region splitting and merging; quadtree structures for segmentation; mean and variance pyramids; computing the first and second derivatives of images using the isotropic, Sobel and Laplacian operators; grouping edge points into straight lines by means of the Hough transform; limitations of the Hough transform; parameterization of conic sections. Perceptual grouping: failure of the Hough transform; perceptual criteria; improved Hough transform with perceptual features; grouping line segments into curves. Overview of mammalian vision: experimental results of Hubel and Weisel; analogy to edge point detection and Hough transform; Relaxation labeling of images: detection of image features; Grouping of contours and straight lines into higher order features such as vertices and facets. Depth measurement in images.

Reference Material:

1. “Computer Vision: A Modern Approach”, David Forsyth, Jean Ponce, Prentice Hall, 2003.
2. “Computer Vision”, Linda G. Shapiro, George C. Stockman, Prentice Hall, 2001.
3. “Handbook of Mathematical Models in Computer Vision”, Nikos Paragios, Yunmei Chen, Olivier Faugeras, Birkhäuser, 2006.

Course Name: Data Communications
Course Structure: Lectures: 3 **Credit Hours:** 3
Objectives: To provide knowledge of Data Communication and its different mechanisms.
Course Outlines: Introduction, Data and Network, Layers, OSI Model, Introduction to Signals, Transmission Media, Digital Transmission, PAM, PCM, ASK, FSK, PSK, QAM, Data Communication Techniques and technologies, Modulation, Multiplexing, Types of errors, Data Communication Protocols, Current technologies being used for data communication.
Reference Material:

1. “Data Communication and Networking”, 4th Ed, Behrouz A. Forouzan. McGraw-Hill
2. “Business Data Communication”, William Stallings.

Course Name: Distributed Computing
Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3
Course Outlines: Why use parallel and distributed systems? Why not use them? Speedup and Amdahl's Law, Hardware architectures: multiprocessors (shared memory), networks of workstations (distributed memory), clusters (latest variation). Software architectures: threads and shared memory, processes and message passing, distributed shared memory (DSM), distributed shared data (DSD). Parallel Algorithms, Concurrency and synchronization, Data and work partitioning, Common parallelization strategies, Granularity, Load balancing, Examples: parallel search, parallel sorting, etc. Shared-Memory Programming: Threads, Pthreads, Locks and semaphores, Distributed-Memory Programming: Message Passing, MPI, PVM. Other Parallel Programming Systems, Distributed shared memory, Aurora: Scoped behavior and abstract data types, Enterprise: Process templates. Research Topics.

Reference Material:

1. "Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 1/e", B. Wilkinson and M. Allen. Prentice Hall, 1999.
2. "Advanced Programming in the Unix Environment", W. Stevens, Addison Wesley, 1993.

Course Name: Network Security
Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3
Course Outlines: Introduction; Cryptology and simple cryptosystems; Conventional encryption techniques; Stream and block ciphers; DES; More on Block Ciphers; The Advanced Encryption Standard. Confidentiality & Message authentication: Hash functions; Number theory and algorithm complexity; Public key Encryption. RSA and Discrete Logarithms; Elliptic curves; Digital signatures. Key management schemes; Identification schemes; Dial-up security. E-mail security, PGP, S-MIME; Kerberos and directory authentication. Emerging Internet security standards; SET; SSL and IPsec; VPNs; Firewalls; Viruses; Miscellaneous topics.
Reference Material:

1. "Cryptography and Network Security", W. Stallings, Prentice Hall PTR, Upper Saddle River, NJ, 2003.
2. "Network Security: Private Communication in a Public World", Kaufman, R. Perlman, M. Speciner, Prentice Hall PTR, Upper Saddle River, NJ, 2002.
3. "Computer Security: Art and Science", M. Bishop, Addison-Wesley, 2003.
4. "Cryptography: Theory and Practice", Stinson, CRC Press, Boca Raton, FL, 1995.
5. "An Introduction to Cryptography", Richard A. Mollin, Chapman and Hall/CRC, 2001.
6. "Applied Cryptography", B. Schneier, John Wiley and Sons, NY, 1996.
7. "Handbook of Applied Cryptography", A. Menezes, P. Oorshcot, and S. Vanstone, CRC Press, Boca Raton, FL, 1997.
8. "Cryptography and Network Security", Behrouz A. Forouzan. McGraw-Hill

Course Name: Wireless Networks
Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3
Course Outlines: This course covers fundamental techniques in design and operation of first, second, and third generation wireless networks: cellular systems, medium access techniques, radio propagation models, error control techniques, handoff, power control, common air protocols (AMPS, IS-95, IS-136, GSM, GPRS, EDGE, WCDMA, cdma2000, etc), radio resource and network management. As an example for the third generation air interfaces, WCDMA is discussed in detail since it is expected to have a large impact on future wireless networks. This course is intended for graduate students who have some background on computer networks.

Reference Material:

1. "Wireless Communications", Theodore S Rappaport.
2. "Fundamentals of Wireless Communications", David Tse.
3. "Wireless Communications and Networks", W. Stallings, Prentice Hall, 2002.
4. "Wireless Communications: Principles & Practice", T.S. Rappaport, Second Edition, Prentice Hall, 2002.

5. "Mobile Communications", J. Schiller, Addison Wesley, 2000.
6. "IS-95 CDMA and cdma 2000", V.K. Garg, Prentice Hall PTR, 2000.
7. "The UMTS Network and Radio Access Technology - Air Interface Techniques for Future Mobile Systems", J.P. Castro, Wiley, 2001.
8. "WCDMA for UMTS Radio Access for Third Generation Mobile Communications", H. Holma and A. Toskala, John Wiley & Sons, 2001.

Course Name: Software Engineering II
Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3
Objectives: To study various software development models and phases of software development life cycle. The concepts of project management, change control, process management, software development and testing are introduced through hands-on Team Projects.
Course Outline: Introduction to Computer-based System Engineering; Project Management; Software Specification; Requirements Engineering, System Modeling; Requirements Specifications; Software Prototyping; Software Design: Architectural Design, Object-Oriented Design, UML modeling, Function-Oriented Design, User Interface Design; Quality Assurance; Processes & Configuration Management; Introduction to advanced issues: Reusability, Patterns; Assignments and projects on various stages and deliverables of SDLC.
Reference Material:
1. "Software Engineering", Sommerville, Addison-Wesley, 2006
2. "Software Engineering: A Practitioner's Approach", Roger Pressman, McGraw-Hill, 2009

Course Name: Systems Programming
Course Structure: Lectures: 2 Labs: 1 **Credit Hours:** 3
Prerequisites: Operating Systems
Objectives: Demonstrate mastery of the internal operation of Unix system software including assemblers, loaders, macro-processors, interpreters, inter-process communication.
Course Outline: System Programming overview: Application Vs. System Programming, System Software, Operating System, Device Drivers, OS Calls. Window System Programming for Intel386 Architecture: 16 bit Vs 32 bit, Programming, 32 bit Flat memory model, Windows Architecture. Virtual Machine(VM), System Virtual Machine, Portable Executable Format, Ring O Computer, Linear Executable format, Virtual Device Driver (V + D), New Executable format, Module Management, COFF obj format 16 bit. (Unix) other 32-bit O.S Programming for I 386; Unix Binaryble format (ELF), Dynamic shared objects, Unix Kernel Programming (Ring O), Unix Device Architecture (Character & Block Devices), Device Driver Development, Enhancing Unix Kernel.
Reference Material:
1. "The UNIX Programming Environment", B. Kernighan & R. Pike Prentice-Hall.
2. "System Software", Leland L. Beck, Addison-Wesley Longman, 1990, ISBN: 0-201-50945-8.

Course Name: Distributed Database System
Course Structure: Lectures: 2, Lab:1 **Credit Hours: 3**
Prerequisites: Introduction to Database Systems
Objectives: To understand difference of Centralized database and Distributed database and to enable the students to design/model a distributed database.
Course Outline: Introduction, Overview of relational DBMS and Normalization, Distributed DBMS architecture, Distributed database design and Data Distribution Strategies, Replication/Fragmentation, Distributed Transaction Management, Distributed Query Processing, Distributed Concurrency Control, Distributed Data Security, Distributed Database Recovery.
Reference Material:

1. "Principals of Distributed Database Systems", Ozsu Tamer.
2. "Database Systems", Thomas Connolly.

Course Name: Data Warehousing
Course Structure: Lecture: 3 **Credit Hours:** 3
Prerequisite: Introduction to Database Systems
Objective: Introduction of Data warehouse and its purpose; to enable students to understand different features / issues in data warehousing and its designing.
Course Outline: Introduction to Data Warehouse and Data Marts, Comparison of OLTP Systems & Data Warehousing, Data Warehouse Architecture, Dimensional Modeling, Comparison Of DM & ER Models, Extraction, Cleansing and Loading process and techniques, Designing a Data warehouse, End user tools, OLAP.
Reference Material:

1. "Data Warehouses and OLAP Concepts, Architecture and Solutions", Wrembel
2. "Data Warehousing: Architecture and Implementation", Humphries
3. "Data Warehousing: Design, Development & Best Practices", Mohanty