Scheme of Study for Master of Computer Science

MCS

2-year programme (4 semesters)

<table>
<thead>
<tr>
<th>#</th>
<th>Category</th>
<th># of Courses</th>
<th>Credit Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computing Core Courses (Comp-Core)</td>
<td>12</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>Computer Science Courses</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>CS Core Courses (CS-Core)</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>CS Elective Courses (CS-Elec)</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>University Electives (Univ-Elec)</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>23</strong></td>
<td><strong>76</strong></td>
</tr>
</tbody>
</table>

Semester-I: (20 Credit Hours)

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

* 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)

<table>
<thead>
<tr>
<th>#</th>
<th>Course Code</th>
<th>Category</th>
<th>Course Title</th>
<th>Credit Hours</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS 302</td>
<td>CS-Core</td>
<td>Computer Organization and Assembly Language</td>
<td>3 (2,1)</td>
<td>Digital Logic Design</td>
</tr>
<tr>
<td>2</td>
<td>CS 304</td>
<td>CS-Core</td>
<td>Theory of Automata and Formal Languages</td>
<td>3 (3,0)</td>
<td>Discrete Structures</td>
</tr>
<tr>
<td>3</td>
<td>CS 306</td>
<td>Comp-Core</td>
<td>Object Oriented Programming</td>
<td>4 (3,1)</td>
<td>Programming Fundamentals</td>
</tr>
<tr>
<td>4</td>
<td>CS 308</td>
<td>Comp-Core</td>
<td>Intro. to Database Systems</td>
<td>3 (2,1)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CS 310</td>
<td>Comp-Core</td>
<td>Computer Communications and Networks</td>
<td>3 (3,0)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CS 312</td>
<td>Comp-Core</td>
<td>Intro. to Software Engineering</td>
<td>3 (3,0)</td>
<td></td>
</tr>
</tbody>
</table>
### Semester-III: (19 Credit Hours)

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

### Semester-IV: (18 Credit Hours)

<table>
<thead>
<tr>
<th>#</th>
<th>Course Code</th>
<th>Category</th>
<th>Course Title</th>
<th>Credit Hours</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS 402</td>
<td>CS-Core</td>
<td>Design &amp; Analysis of Algorithms</td>
<td>3 (3,0)</td>
<td>Data Structures &amp; Algorithms</td>
</tr>
<tr>
<td>2</td>
<td>CS 404</td>
<td>CS-Core</td>
<td>Compiler Construction</td>
<td>3 (2,1)</td>
<td>Theory of Automata &amp; Formal Languages</td>
</tr>
<tr>
<td>3</td>
<td>CS 406</td>
<td>CS-Elec</td>
<td>CS Elective-3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CS 408</td>
<td>Comp-Core</td>
<td>Human Computer Interaction</td>
<td>3 (2,1)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CS 410</td>
<td>Comp-Core</td>
<td>Software Project/Thesis</td>
<td>6 (0,6)</td>
<td></td>
</tr>
</tbody>
</table>
## Electives for MCS

### CS Elective Courses:

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

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

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:

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:

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:

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.
Reference Material:

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:

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:
Reference Material:
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:

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:
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:

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:

**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:**  

**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.  
**Reference Material:**  
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.
Reference Material:

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.
Reference Material:

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:
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:

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 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:**

**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:**

**Course Name: Distributed Computing**
**Course Structure:** Lectures: 3 Labs: 0  Credit Hours: 3
Reference Material:

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:

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:

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:

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.
Reference Material:
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:  

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:  