



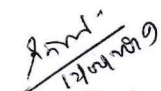
## UNIVERSITY OF CALCUTTA

### Notification No. CSR/ 04 /19

It is notified for information of all concerned that the Syndicate in its meeting held on 10.12.2018 (vide Item No. 10) approved revised syllabus 7<sup>th</sup> Semester B.Tech. in Information Technology course of study under this University, as laid down in the accompanying pamphlet.

The above shall be effective from the academic session 2018-2019.

SENATE HOUSE  
KOLKATA-700073  
The 12<sup>th</sup> February, 2019

  
(Dr. Soumitra Sarkar)  
Registrar (Acting)

  
12/2/19.

## Syllabus for 3<sup>rd</sup> Semester B. Tech in Information Technology

Course/L-T-P	Subject	Credits	Comments
IT-301 (3-1-0)	Data Structures and Algorithms	4	Departmental
IT-302 (3-1-0)	Analog and Digital Communication	4	Departmental
IT-313 (3-1-0)	Computer Architecture and Organization-I	4	Departmental
IT-324 (3-1-0)	Operating Systems-I	4	Departmental
IT 314(2-1-0)	Engineering Mathematics-III	3	Common subject
	Environment Sciences	3	Common subject
IT-361 (0-0-3)	Data Structures Laboratory	3	Practicals
IT-362 (0-0-3)	Communication Laboratory	3	Practicals
IT-364 (0-0-3)	Operating Systems Laboratory	3	Practicals
<b>Total Credits for Semester</b>		<b>30</b>	

## **IT-301: Data Structures and Algorithms [40 lectures]**

**Learning Objective:** *This is a first level introduction to computer technology at the software level. A computer program consists of a step-by-step method (algorithm) of solution of a problem involving data in the form of the input premises as well as the output generated. While algorithms usually built upon existing ones, there are some fundamental ones which a student of IT must know to apply and implement and also analyze in terms of complexity. Similarly, data may be organized in several ways building upon certain fundamental concepts. This course equips a student with basic knowledge of algorithms and architecture. At the end of the course, the student should be able to implement some basic algorithms and data structures and build up on them to create good and efficient solutions for problems.*

### **Algorithms and Analysis [4 lectures]**

- Basic definitions and properties. Concept of algorithmic complexity involving time and space [2 lectures]
- Asymptotic analysis --- Dominant term, loops, nested loops; order notations [2 lectures]

### **Recursion and Iteration [4 lectures]**

- Recursion tree, evaluation of the factorial; generating Fibonacci numbers; finding GCD of two integers; the Tower of Hanoi problem

### **Linear Data Structures [12 lectures]**

- Fundamental data types; arrays; sequential and linked memory allocation [2 lectures]
- Linked list and its implementation; singly and doubly linked lists [4 lectures]
- Abstract Data Types (ADT); List as an ADT – array-based and linked list based implementation [2 lectures]
- Applications of lists --- sparse matrix representation; polynomial representation; addition/subtraction of polynomials; representation and operations on long integers [2 lecture]
- Stack and Queue as ADT --- array and link list implementation (both linear and circular); applications of stacks and queues [2 lectures]

### **Non-linear Data Structures [8 lectures]**

- Tree as an ADT --- basic definitions; terminology (nodes and branches, height or depth, levels, etc.) [1 lecture]
- Binary Trees --- different types (complete, full, skewed); representation (array; linked); properties (nodes, branches, internal and external path lengths, minimum and maximum values, etc.); traversal algorithms (pre, in and post order); threaded trees [2 lectures]
- Binary Tree Variations --- binary search trees; heaps and their applications; B tree; AVL tree [6 lectures]

## Sorting and Searching [8 lectures]

- Search Techniques --- sequential and ordered sequential search; binary search; interpolation search; Fibonacci search [2 lectures]
- Sorting Techniques --- bubble sort; insertion sort; selection sort; quick sort; merge Sort; heap Sort; shell sort; tree sort; radix sort; [4 lectures]
- Hashing --- Definition and utility; collision resolution (open addressing with linear/quadratic probing, simple chaining, value of load factors, etc.) [2 lectures]

## Graphs [4 lectures]

- Definition and computer representation.) [1 lecture]
- Graph Properties [1 lecture]
- Graph Algorithms [2 lectures]

## References

- **M. A. Weiss**, *Data Structures and Algorithm Analysis in C*, Pearson Education Asia, 2009.
- **V. Aho, J. E. Hopcroft**, and **J. D. Ullman**, *Data Structures and Algorithms*, Pearson Education.
- **R. F. Gilberg, B. A. Forouzan**, *Data Structures: A Pseudocode approach with C*, 4ed, Thomson India Edition, 2010.
- **Sara Baase** and **A. Van Gelder**, *Computer Algorithms*, Fourth Edition, Pearson Education, 2009.
- **T. H. Cormen, C. E. Leiserson, R. L. Rivest**, and **C. Stein**, *Introduction to algorithms*, Fifth Edition, Prentice Hall of India Ltd, 2010.
- **Goodrich, M.** and **Tamassia, R.** *Data Structures and Algorithms in Java* 3ed, John Wiley and Sons, Inc.
- **Sahni S.**, *Data Structures, Algorithms and Applications in C++*, McGraw-Hill, 1998.
- **Horowitz E., Sahni S., Rajasekaran**, *Computer Algorithms*, Computer Science Press, 1998.

## IT-302: Analog and Digital Communication [40 lectures]

**Learning Objective:** This is an introductory course of communication technology at the undergraduate level. The primary objective of the course is to provide an idea of the modern communication engineering which has undergone some radical changes during the past decades and continues to do so. The course will provide an in-depth knowledge, starting from the basics of communication to the exposure on the latest advancement in the field. The concept of time domain and frequency domain representation of the signals, basic elements of communication systems, principle of analog communication, analog to digital conversion, data encoding, digital modulation and multiplexing techniques are explained with emphasis on practical applications. In addition, information theory overview is incorporated in the course to provide the basic idea of error control and coding. On completion of the course, students should be capable to realize how message signals are processed for transmission from transmitter to receivers, merits and demerits of digital communication over analog communication and effect of noise and distortion in the transmission channel. They will be conversant with the recent development related to the major field of the study and will be able to contribute effectively in multidisciplinary areas.

### Basic concepts of Signals and Spectra [8 lectures]

- Signals --- classification of signals; concept of spectra, energy and power signal; Fourier series; Fourier transform; inverse Fourier transform; power spectral density; correlation; convolution [4 lectures]
- Transmission of Signals through a Linear System --- introduction to communication systems; transmission media (twisted pair cable, coaxial cable, characteristics of optical fiber); origin of noise and its effect; importance of SNR; distortion of signal over channel; distortion less transmission [4 lectures]

### Principles of Analog Communications [10 lectures]

- Concept of Modulation and Demodulation --- amplitude modulation; DSB-FC, DSB-SC, SSB-SC and VSB; advantages, disadvantages and applications; power and bandwidth requirements; superheterodyne receiver [4 lectures]
- Frequency and Phase Modulation --- basic principles; mathematical expressions, NBFM and WBFM; power and bandwidth requirements; FM radio transmitters and receivers [4 lectures]
- Signal to noise ratio of AM, FM and PM transmissions [2 lectures]

### Introduction to Analog/Digital Conversion [10 lectures]

- Analog to Digital Inter-conversion --- Sampling theorem; Nyquist sampling rate; A/D and D/A converters; aliasing effect; PAM; PWM; PPM; natural and flat top sampling [3 lectures]
- Concept of Quantization --- regenerative repeater; PCM, DPCM, DM, ADM; generation and detection; data encoding (NRZ, RZ, Bipolar-AMI, Manchester coding) [4 lectures]
- Baseband pulse transmission; Inter-symbol interference; Nyquist criterion for distortion less baseband transmission; Eye patterns [3 lectures]

### **Digital Modulation Techniques [8 lectures]**

- Digital Modulation --- ASK; FSK; PSK; DPSK; QPSK; M array PSK; MSK; GMSK and introduction to QAM; probability of error; bandwidth efficiency, and performance comparison. [5 lectures]
- Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM) [ 1 lecture]
- Overview of Spread Spectrum Modulation --- introduction to Code Division Multiplexing (CDM) [2 lectures]

### **Introduction to Information Theory [4 lectures]**

- Information contents; entropy; mutual information; information rate; channel capacity; Shannon's information capacity theorem [2 lectures]
- Error control and coding basic principles [ 2 lectures]

### **References**

## IT-313: Computer Architecture and Organization-I [40 lectures]

**Learning Objective:** This is a first level introduction to computer technology at the hardware level. The syllabus assumes pre-coverage or parallel coverage of sequential logic circuit design in other subject(s). The course will provide basic insight into the design of a computer system by explaining how a typical CPU works internally and how data can be exchanged between a CPU/memory with the external world. At the end of the course, the student should be able, at least on paper, to design a computer system of reasonable complexity and also be able to evaluate a given design of a system.

### Basic Structure of Computers [12 lectures]

- Basic Concepts --- functional units, basic operational concepts, bus structures, software, performance, historical perspective --- how CPU-s have evolved [2 lectures]
- Internal working of a CPU through an early generation example like 8085 [4 lectures]
- The Memory System --- basic concepts; semi-conductor RAM, ROMs and types; speed, size and cost; introduction to cache memories; performance considerations; secondary storage [3 lectures]
- The Input/Output System --- accessing I/O devices; interrupts; DMA; buses and standard I/O interfaces; introduction to timing diagrams for memory and I/O interactions [3 lectures]

### Interfacing Technology [10 lectures]

- More About Bus --- parallel versus serial interfacing; serial interfacing/communication standards; hierarchical memory interfacing (static and Dynamic RAMs, interleaving); CPU chip internal memory; special purpose I/O controllers (8255 PPI, Counter/Timers, Display, Serial I/O, etc.); interrupt and DMA controllers [10 lectures]

### Architecture of a Modern CPU (80386) [12 lectures]

- Introduction to 8086 architecture --- functional diagram; register organization; memory segmentation; physical and logical memory organization; addressing schemes and interrupts; parallelism through maximal/minimal mode of operation [6 lectures]
- Programming 8086 --- instruction set; assembler directives; macros simple programs involving logical, branch and call instructions; sorting; evaluating arithmetic expressions; string manipulations [4 lectures]
- Introduction to 80386 Architecture --- real versus protected mode; built-in memory management; concept of tasks and multi-tasking [2 lectures]

### Introduction to Modern Micro-controllers [6 lectures]

- Overview of 8051 (as an example of a legacy micro-controller), a modern example like Atmega8 or Atmega32 [6 lectures]

## References

1. **Carl Hamacher, Zvonko Vranesic, Safwat Zaky**, *Computer Organization*, McGraw Hill.
2. **Ramesh S Gaonkar**, *Microprocessor Architecture, Programming, and Applications with the 8085* Prentice Hall.
3. **M. M. Mano**, *Computer System Architecture*, Prentice Hall.
4. **B. Govindarajalu**, *Computer Architecture and Organization*, Tata McGraw-Hill Education.
5. **A. K. Ray and K.M. Bhurchandani**; *Advanced Microprocessors and Peripherals* Tata McGraw-Hill Education, 2nd edition 2006.
6. **Liu and G A Gibson**, *Micro Computer System 8086/8088 Family Architecture Programming and Design* PHI.
7. **Ajay. V. Deshmukh**, *Microcontrollers and Application* Tata McGraw-Hill Education.



## IT 324: Operating Systems-I

**Learning Objective:** This is a basic level course of computer technology explaining how a typical modern computer system works. It covers the needs and tenets of Operating Systems (OS), the relationship and the integration of the OS and the architecture of the underlying system. It is expected that this course will run parallel with a first level computer organization and architecture course. Ideas from OS like process-synchronization, memory-management, file-management, I/O-operations, disk-management etc., are covered. Functionalities and the structures of the assembler, linker, loader, compiler etc., is also covered. On completion of the course, a student should be able to appreciate the features of a typical modern OS and some amount of understanding of how several smaller and simpler units together provide the functionality of an OS.

### Introductory Concepts [4 lectures]

- Introduction to OS --- necessity and functioning of OS; evaluation of OS-es; different types of OS [2 lectures]
- System Structure --- computer system operation; I/O structure; storage structure; storage hierarchy; different types of protections; operating system structure; O/S services; system calls; [2 lectures]

### Process Management [12 lectures]

- Concept of a Process --- introduction; task and the context of a task; process scheduling; operations on processes; co-operating processes; inter-process communication [2 lecture]
- Threads --- overview; lightweight process, benefits of threads, user and kernel threads [2 lectures]
- CPU Scheduling --- scheduling criteria; preemptive and non-preemptive scheduling; scheduling algorithms; multi-processor scheduling [3 lecture]
- Process Synchronization --- background; critical section; critical region; synchronization hardware; classical problems of synchronization; semaphores; locks [3lecture]
- Deadlocks --- system model; deadlock characterization; methods for handling deadlocks; deadlock prevention; deadlock avoidance; deadlock detection; recovery from deadlock [2 lectures]

### Memory Management [6 lectures]

- Basic Concepts of Memory Management --- background, logical versus physical address space, swapping, contiguous memory allocation, paging, segmentation [2 lectures]
- Virtual Memory --- background; demand paging; page replacement and page replacement algorithms; allocation of frames; thrashing [4 lectures]

### File Systems [8 lectures]

- The Concept of a File --- layout on disk; access methods; directory structure [2 lecture]

- Structure of a File System --- allocation methods (FAT, inode, journal); free-space management; directory implementation[6 lectures]

### **Input/Output and Disk Management [6 lectures]**

- I/O Management --- typical I/O hardware; data access (polling, interrupts, DMA); application's interface with I/O; kernel and I/O subsystem; device driver (classes, speed issues, buffering) [4 lectures]
- Disk Management --- structure of a disk; disk I/O scheduling; disk reliability; disk formatting; boot block; page blocks; disks with multiple OS; bad blocks [2 lectures]

### **Systems Programming [4 lectures]**

- Assembler --- basic concept; absolute and relocatable assembly; design of assemblers and macro processors [2 lecture]
- Linkers --- concept and design; loaders; -different types of loaders; debuggers; interpreters [1 lecture]
- Compilers --- various phases; lexical analysis; parsing, error handling; syntax directed translation; intermediate codes [1 lecture]

### **References**

1. Silberschatz, Galvin, Gange "Operating System Concepts" John Wiley & Sons.
2. Milan Milenkovic "Operating Systems: Concepts and Design" McGraw-Hill Companies
3. Andrew S. Tanenbaum "Modern Operating Systems" Pearson-Prentice Hall
4. D.M. Dhamdhare "Systems Programming and Operating Systems" Tata McGraw-Hill
5. John J. Donovan "System Programming" Tata McGraw-Hill

## IT 314 Engineering Mathematics III

Learning objective :

It's a basic algebra , probability , statistic , Graph theory courses . Since it's a basic course , the prerequisite for the course is Set theoretic concept . After compilation of this course student will be able to apply this concept to develop and analyze the computing system.

### Basic Algebra (10 lectures)

- Basic concept of set theory ,Relation, Mapping; Group and Subgroup , Order of a Group , finite subgroup ; Cyclic group, generator of a cyclic group: Homomorphism & Isomorphism of group; Cosets , Lagrange's theorem , Normal subgroup, Isomorphism Theorem(Statement only) Quotient group, direct product. (5 lectures)
- Rings, sub-rings, ideals and quotient rings, integral domains and fields, field of fractions, Euclidean domain and unique factorization domain. (3 lectures)
- Posets, Lattices and Boolean Algebra. (2 lectures)

### Probability (7 lectures)

- Axiomatic definition of probability, Conditional probability, Independent events, Related problems, Bayes' theorem ( Statement only) & its application.(3 lectures)
- One dimensional random variable, Probability distributions-discrete and continuous, Expectation, Variance and covariance ,Moment Generating Function ,Chebyshev's Inequality. Binomial, Poisson, Uniform, Exponential and Normal distribution, Problems on Binomial, Poisson and Normal distribution only. (4 lectures)

### Sampling theory (3 lectures)

- Random sampling, Parameter, Statistic and its Sampling distribution, Standard error of statistic, Sampling distribution of sample mean and variance in random sampling from a normal distribution (statement only) and related problems.(3 lectures)

### Estimation of parameters(2 lectures)

- Unbiased and consistent estimators, Point estimation, Interval estimation, Maximum likelihood estimation of parameters (Binomial, Poisson and Normal), Confidence intervals and related problems.(2 lectures)

### Graph Theory (8 lectures)

- Graph , Digraph , Isomorphism , Walk, Path , Circuit , Shortest Path Problems ,Dijkstra's Algorithm (2 lectures)

● Tree , Properties of Tree , Binary Tree and Fundamental Circuit , Minimal Spanning Tree , Kruskal's Algorithm , Prim's Algorithm , DFS , BFS. Cut Set : Fundamental Cut Set and Cut Vertices.(3 lectures)

● Planar and Dual Graphs , Matrix Representation of Graphs ( Adjacency and Incidence Matrices ) , Network , Flow Augmenting Path , Ford-Fulkerson Algorithm for Maximum Flow , Floyd Algorithm ; Max-Flow and Min-Cut Theorem (Statement only )(3 lectures)

#### References:

1. Liu C.L., "Introduction to Combinatorial Mathematics", McGraw Hill, 1968.
2. S.K Mapa – Higher Algebra (Abstract & Modern)
3. Balakrishnan: Graph Theory (Schaum's Outline Series), TMH.
4. Chakraborty S.K and Sarkar B.K.: Discrete Mathematics, OUP.
5. Deo N: Graph Theory with Applications to Engineering and Computer Science, Prentice Hall.
6. Sen M.K., Ghosh S. and Mukhopadhyay P.: Topics in Abstract Algebra, University Press.

# IT 315: Environment Sciences

## Environmental Science

***Perspectives of environment and society:*** Genesis of environmentalism; Social issues and major environmental movements; Global environmental crisis; Human population and environment; Concepts of ecological footprints; Impact of technology on society; Gender and environment; Environment education and awareness; Global environmental initiatives;

***Segments of Environment:*** Earth, environment and origin of life; Atmosphere, hydrosphere, lithosphere and biosphere; Bio-geo chemical cycles; Weather and climate; Basic instrumental applications in meteorological studies;

***Ecology and biodiversity:*** Ecosystem services and energy flow; Basic concepts of population, community, niche and succession; Indian biogeography and concepts of protected area network; Types of diversity; Biodiversity documentation and bioprospecting; Ecosystem case studies – wetland, forest and coastal; Wildlife biology and trade;

***Concepts of Natural Resources:*** Non-renewable and renewable resources; Wise use of resources; Natural resource availability in India; Environmental tradition and values in ancient India; Environmental and social impacts of natural resource exploitation; Food and water security;

***Energy and Environment:*** Conventional and nonconventional energy resources; Energy budget of the earth; Patterns of global energy use; Alternative energy and environmental impacts; Energy security in India

***Pollution and Mitigation:*** Water (surface and ground), soil and air (outdoor, indoor and vehicular) pollution; Noise pollution; Radioactive pollution; Thermal pollution; Major domestic and industrial pollutants; Societal and health impacts of environmental pollution; Case studies; Introduction to pollution monitoring and control technologies;

***Waste Reduction and Management:*** Classification of wastes; Strategies of waste segregation and disposal; Technologies of waste remediation; Importance of “reduce, reuse, recycle and restore”; Case studies- hazardous waste and electronic waste;

***Disaster Mitigation and Adaptation:*** Nature, Scale and types of Disaster; Environmental impacts; Disaster preparedness and post disaster response; Mitigation strategies; Case studies

***Environmental Law and Ethics:*** Environmental legislation in India; Provisions for environment in Indian Constitution; Existing environmental acts in India; Indian Organization of Standards and Ecomarks; Environmental ethics; Protection of traditional knowledge; Concepts of piracy and patenting;

***Environmental Economics:*** Population growth and carrying capacity; Concepts of sustainable development; Sustainability indices; Concept of consumerism, poverty and globalization; Using economics to assess environmental quality; Urban and rural planning in India and land use pattern; Environmental valuation and decision making; Economic development and Cost-Benefit analysis

## IT-361 Data Structures Laboratory [15 sessions]

This practical session writing of properly written programs covering aspects of what has been taught in the theory class. There will be three types of projects:

- **Familiarization** (2 session with several sub-sessions) --- No submission necessary
- **Program Development** (6-8 sessions) --- there must be at least FIVE submissions per group, in the form of C-code, intermediate and output files, etc.
- **Mini-project** (3-5 sessions) --- full submission including appropriately documented source code, a design document and a presentation by the group consisting of a live demonstration of created software

For this laboratory, the following C declaration will be assumed:

```
typedef struct _Student
{
    char name[32];
    int rollNumber;
    int marks[5];
} Student;
```

There will be a read-only text file containing data for a large number of students (to be called FILE). In experiments involving reading student data from file, this file will have to used 'in place' (i.e., without copying).

### Familiarization Experiments (Programs)

- Read numbers from console, print formatted results on console
- Store the data in in-memory data structures like arrays, stacks, lists, queues and trees
- Read data from text file and save output to files

### Program Development Experiments

- **Experiment-1:** Read data from FILE into an array. Write an implementation of bubble sort that can be used to sort the array. Use library function qsort to sort the array on i) rollNumber ii) name and iii) total of marks and save them in separate output files. Compare performance between your bubble sort implementation and qsort.
- **Experiment-2:** Implement an AVL tree using pointers. Read data from File into your AVL tree, generating a 'verbose' of steps during inserting each student data in the tree. Generate output files for pre-, in- and post-order traversal of your tree.
- **Experiment-3:** Repeat Experiment-2 but with a Heap implemented using an array.
- **Experiment-4:** Demonstrate the efficacy of a stack in solving a) postfix expression evaluation and b) Stock Span problems. Prepare you own input files to prove the correctness of your programs. There should be output files tracing the different parts of your implementation at different debug levels.

- **Experiment-5:** Demonstrate the efficacy of Prim's and Kruskal's minimal spanning tree algorithms. Prepare you own input files to prove the correctness of your programs. There should be output files tracing the different parts of your implementation at different debug levels.
- **Experiment-6:** Demonstrate the efficacy of Dijkstra's minimum path algorithm. Prepare you own input files to prove the correctness of your programs. There should be output files tracing the different parts of your implementation at different debug levels.

### Mini Project

One from the choices below or your own choice provided it is of a similar level

- a) Online library management system with data stored in text file(s)
- b) Solving a linear programming problem using the Simplex method (must demonstrate with at least a 20 variable scenario)
- c) A flight-plan problem using multiple airlines, timings and ticket prices

## IT-362: Communication Laboratory

Experiments on aspects of analog and digital communications

## IT-364: Operating Systems Laboratory [15 sessions]

*This practical subject trains the student in understanding the working of a typical OS, linux, from a user and programmer perspective. It is assumed that the student has had an earlier practical session on writing, editing, computing and executing simple C/C++ programs and in the process has been exposed to basic linux shell commands like 'ls', 'cat', etc.*

1. More shell commands; concept of stdin, stdout and stderr; redirection and pipelining; regular expressions; grep; ps; etc; an experiment involving redirection and pipelining of multiple shell commands to achieve a goal; writing a simple filter program and redirecting input and output from the same
2. Elements of shell programming; shell variable; loops; file wild-characters; command-line handling; capturing return values of programs; getting and setting environment variables
3. Developing programs spanning across multiple C and header files; make and make files; tracing the build process of gcc; exchanging pre-build object modules among student groups
4. File I/O with text files and binary files to understand clearly the format of storage; file I/O using lower level calls like open, read, write; etc; investigating a directory using system calls
5. Overwriting a function (say strcat) from the standard C-library and observing log messages on stderr; top level understanding of executable-linkable format (i.e., elf files); use of gnu debugger
6. Executing programs from within another program using "exec" and "fork"; linux interprocess communication through named pipes, file maps and sockets



## Syllabus for 4<sup>th</sup> Semester B. Tech in Information Technology

<b>Course/L-T-P</b>	<b>Subject</b>	<b>Credits</b>	<b>Comments</b>
IT-401 (3-1-0)	Design and Analysis of Algorithms	4	Departmental
IT-402 (3-1-0)	Object Oriented Programming Principles	4	Departmental
IT-413 (3-1-0)	Computer Architecture and Organization-II	4	Departmental
IT-404 (3-1-0)	Computer Networks	4	Departmental
IT-451 (2-1-0)	Discrete Structures	3	Common subject
IT-461 (0-0-3)	Computer Architecture Laboratory	3	Practicals
IT-462 (0-0-3)	Object Oriented Programming Laboratory	3	Practicals
IT-463 (0-0-3)	Computer Networks Laboratory	3	Practicals
IT-284 (0-0-3)	Seminar	2	Seminar and Viva-voce
<b>Total Credits for Semester</b>		30	

## **IT-401: Design and Analysis of Algorithms [40 lectures]**

**Learning Objective:** *This course introduces the student to the intricacies of algorithm design and analysis. It is expected that the student is conversant with basic data structures and algorithms. The course introduces classical design and analysis techniques that can form the basis for building more complex and efficient algorithms. At the end of the course, the student should be able to apply the knowledge acquired and also appreciate the effectiveness of other techniques and methodologies for solving problems algorithmically.*

### **Introduction [7 lectures]**

- Preliminaries --- Definition of algorithm; notations for representations [2 lectures]
- Some Basic Data Structures --- Hashing; analysis of hashing complexities; disjoint subsets (union-find algorithm); binary and n-ary trees review [2 lectures]
- Algorithmics --- Problems and instances; efficiency; average and worst case analysis; asymptotic notations (big-Oh, big-Theta, big-Omega, etc); methods of solutions [3 lectures]

### **Sorting and Searching [8 lectures]**

- Sorting --- Basic sorting algorithms (Heap sort with analysis of Heapify, radix sort); DAG-s and topological sort [5 lectures]
- Searching --- Linear and binary search; finding maximum and minimum; k largest elements in order; breadth-first and depth-first tree traversals on undirected and directed graphs [3 lectures]

### **Classical Solutions [15 lectures]**

- Divide and Conquer --- Principles and examples; quick sort (with analysis) merge sort (merge sort); Strassen's matrix multiplication [4 lectures]
- Backtracking --- the n-queen problem [1 lecture]
- Greedy Method --- Depth and breadth first searches; minimal spanning trees (Kruskal's and Prim's algorithms with implementation and analysis); Dijkstra's shortest path algorithm; the knaprack problem; Huffman code [6 lectures]
- Dynamic Programming --- Matrix chain multiplication; multi-level graphs; the travelling salesman problem; [4 lectures]

### **Other Techniques and Concepts [10 lectures]**

- String Matching Algorithms --- Naive string matching; Knuth-Morris-Pratt algorithm with analysis; Rabin-Karp and Boyer Moore's algorithm (brief introduction) [5 lectures]
- Some More Algorithmic Concepts --- Branch-and-bound; approximation algorithms; Randomized algorithms; Monte Carlo technique; parallel and distributed algorithms [3 lectures]
- Abstract Concepts --- Tractable versus intractable problems; classes P and NP; NP-Hard; NP-Complete [2 lectures]

## References

## IT-402: Object Oriented Programming Principles [40 lectures]

**Learning Objective:** This course covers object-oriented programming principles and techniques using Java. Topics cover classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, exceptions, container classes, etc. This course also covers basic concepts for software design and reuse. After completion this course student will be able to use Eclipse as an Integrated Development Environment, take a problem and develop the structures to represent objects and the algorithms to perform operations, apply standards and principles to write effective as well as truly readable code, design a class to serve as a program module, understand and demonstrate the concepts learnt in the course.

### Basic Concepts of Object-Orientation [16 lectures]

- Introduction to OOP Principles --- Data abstraction; encapsulation; inheritance; aggregation [2 lectures]
- Object Model --- Evolution of the object model; element of the object model and its applications [5 lectures]
- Fundamentals of OOP --- Classes and objects; the nature of an object; relationship among objects; the nature of classes; relationships among classes; interplay of classes and objects; methods and messages; inheritance; dynamic binding and polymorphism; virtual functions; virtual function table; different perspectives of inheritance; interface vs implementation; single vs multiple inheritance [7 lectures]
- Object-oriented Programming Languages --- Evolution of OOPs; Simula; Smalltalk; Eiffel; C++; Java; C# [2 lectures]

### Object-oriented Programming Using Java [8 lectures]

- Introduction; operators; data types, variables; arrays; control statements; classes and methods; inheritance; package and interface; exception handling; strings; event handling; multi-threaded programming; programming for networks

### Advanced Object Oriented Concepts [16 lectures]

- Prototype Based OOPs --- Class vs prototype; cloning; delegation; the 'self' reference; object behavioural evolution; dynamic inheritance [2 lectures]
- Theory of Objects --- types and subtypes; the substitution principle; covariance and contravariance; overloading; parametric polymorphism; inclusion polymorphism; lambda calculus; object calculus; denotational semantics [7 lectures]
- Unified Modelling Language (UML) --- Class diagrams; object diagrams; sequence diagrams; use case diagrams; collaboration diagrams; state-chart diagrams; activity diagrams; component diagram; deployment diagram [7 lectures]

## References

1. **Grady Booch**, Object Oriented Analysis and Design, Addison-Wesley.
2. **Brad. J. Cox** , Object Oriented Programming - An Evolutionary Approach
3. **Scott Ambler**, The Object Primer -The Application Developer's Guide to Object Orientation and the UML
4. **Bertrand Meyer**, Object-Oriented Software Construction

## IT-413: Computer Architecture and Organization-II [40 lectures]

**Learning Objective:** This is a follow-up course on computer organization and architecture. The idea is to enlighten the student with the advancements in computer design that has happened over the last 3/4 decades. At the end of the course, the student should be able to judge a digital system from the perspective of modernity, performance, expandability, domain of use, etc. It will be good if a course on Operating Systems runs in parallel or in the immediately following semester.

### Review of Basic Organization and Architectural Techniques [12 lectures]

- RISC processors and their characteristics; RISC vs CISC; coverage of one RISC processor like ARM [4 lectures]
- Classification of instruction-set architectures [2 lectures]
- Performance measurements [2 lectures]
- Digital Signal Processors (DSP-s) and their use [2 lectures]
- Architecture and OS interrelationship [2 lectures]

### Memory and Bus [12 lectures]

- Cache memory and its use cache optimizations cache performance, etc [2 lectures]
- Virtual memory design and implementation [2 lectures]
- Secondary memory; RAID; memory hierarchy in the ARM Cortex and Intel Core i7 [2 lectures]
- Modern storage devices (flash drive, SD drives, etc) [2 lectures]
- Buses and arbitration; hierarchical bus system; backplane bus specification; bus arbitration and control [2 lectures]
- Plug and Play interfacing; PCI bus; USB [2 lectures];

### Parallel Computation [16 lectures]

- Instruction Level Parallelism --- pipelining; arithmetic pipelines; instruction pipelines; hazards in a pipeline; hazard resolution techniques; dynamic instruction scheduling; branch prediction; parallelism using software approaches; superscalar techniques [6 lectures]
- Data-Level Parallelism --- vector architecture; SIMD Instruction Set Extensions for Multimedia (Intel MMX); GPU-s [3 lectures]
- Thread Level Parallelism --- centralized vs distributed shared memory; interconnection topologies; symmetric and asymmetric multiprocessing; cache coherence problem; synchronization [2 lectures]
- Review of modern multi-core architectures [1 lecture]

### References

1. **J. L. Hennessy and D. A. Patterson**, Computer Architecture: A Quantitative Approach. Morgan Kaufmann.

2. **William Stallings**, Computer Organization and Architecture. Pearson.
3. **J.P.Hayes**, Computer Architecture And Organization, McGraw-Hill
4. **Kai Hwang**, Advanced Computer Architecture.
5. **Kai Hwang**, Scalable Parallel Computing: Technology, Architecture, Programming.
6. **Anoop Gupta**, Parallel Computer Architecture: A Hardware/software Approach.

## IT-404: Computer Networks [40 lectures]

**Learning Objective:** This is the principal course on computer networking covering salient aspects of the different 'layers' of network protocols --- how devices and/or algorithms perform to create the connected world that we live in today. The learner should have basic understanding of computer architecture, data communication fundamentals and knowledge of data structures and algorithms. At the end of the course, the learner is expected to have to a reasonable depth, an understanding of the working of the Internet, especially how different pieces work in harmony to make the Internet a reality.

### Introductory Concepts [10 lectures]

- Introduction to Networking --- Network categories; devices; layered network architecture; protocols and standards [2 lectures]
- Error Detection and Correction --- Theory; parity; checksum; Hamming distance; Hamming codes; block codes; Cyclic Redundancy Check (CRC) [6 lectures]
- Networking Devices --- Switches; repeaters; bridges; routers; gateways; concept of Quality of Service [1 lecture]
- Overview of the TCP/IP Protocol Suite [1 lecture]

### Link Layer Concepts [10 lectures]

- Local Area Networks --- IEEE LAN Standards; MAC protocols; slotted and unslotted Aloha, CSMA, CSMA/CD, collision free protocols, limited contention protocols [4 lectures]
- Data Link Control --- Line configurations (Point-to-Point, Multi-Point full/half duplex); link control protocols (Stop-and-Wait; sliding Window); error control (Stop-and-Wait ARQ, Go-Back-N ARQ, Selective-Reject ARQ) [4 lectures]
- Concept of Switching --- Circuit switching; space and time division switches; time and frequency division multiplexing; packet switching and the datagram approach; virtual circuit switching [2 lectures]

### Network Layer Concepts [10 lectures]

- Basic Concepts --- responsibilities of the network layer; network layer protocols [1 lecture]
- The Internet Protocol (IP) --- IPv4 and IPv6; IP addresses and classes; subnetting and subnet masks; classless subnetting; IP layer protocols (ICMP, ARP, BOOTP); static vs dynamic IP addresses; DHCP [5 lectures]
- Concepts of Routing --- Basic concept of IP routing; routing tables; the optimality principle; flooding; shortest path routing; distance vector routing; link state routing; broadcast and multicast routing; mobile host routing; exterior routing; routing protocols (RIP, OSPF, BGP) [4 lectures]



### Transport Layer Concepts [6 lectures]

- Basic Concepts --- responsibilities of transport layer; connection-less vs connection oriented protocols; transport layer protocols (UDP and TCP) [2 lectures]
- Transmission Control Protocol (TCP) --- TCP responsibilities; ports; connection; handshaking; congestion and its control [2 lectures]
- Elements of programming with sockets [2 lectures]

### Other Concepts [4 lectures]

- Some Application Layer Protocols --- FTP; HTTP; HTTP and the world wide web; DHCP [2 lectures]
- Network Security --- Encryption/decryption—Character Level encryption, Bit level encryption. RSA. Data Authentication—digital signature [4 lectures]

### References:

1. **Behrouz A. Forouzan**, Data Communication and Networking, McGraw-Hill.
2. **Andrew S. Tanenbaum**, Computer Networks, Prentice Hall.
3. **Douglas Comer**, Internetworking with TCP/IP, Volume 1, Prentice Hall of India.
4. **W. Richard Stevens**, TCP/IP Illustrated, Volume 1, Addison-Wesley.
5. **William Stallings**, Data and Computer Communication, Prentice Hall of India.
6. **J.F.Kurose and K.W.Ross**, Computer Networking: A Top-Down Approach Featuring the Internet, Pearson Education.
7. **Murthy and Manoj**, Ad Hoc Wireless Networks, Pearson.

## IT -451 Discrete Structures

### Learning Objective:

This course covers the topics typically covered in a first level combinatorics course. It introduces the elementary notions in combinatorics and presents the most elementary techniques in combinatorics - pigeon hole principle, inclusion-exclusion principle, recurrence relations and generating functions etc.

Pre-requisite : -Basic familiarity of sets, relation, function , partial order , concept of group theory.

After completion of this course ,students will become familiar with fundamental combinatorial structures that naturally appear in various other fields of mathematics and computer science. They will learn how to use these structures to represent mathematical and applied questions, and they will become comfortable with the combinatorial tools commonly used to analyze such structures.

### **Basic Counting [4 lectures]**

- Introduction(Addition Principle , Multiplication Principle), Permutation , Combinations . [1 lecture]
- Bijections between two finite sets , The Binomial Theorem , Newton's Binomial Theorem , Pascal Identity, Applications .[2 lectures]
- counting object with repetitions, Multinomial Theorem , Applications .[1 lecture]

### **Listing Combinatorial Objects [2 lectures]**

- permutations on a set , inversion of a permutation, number of inversion ,inversion sequence , Applications ;[1 lecture]
- Listing combinations , order relations on a set , immediate predecessor , immediate successor , Lex order , Colex order , problems.[1 lecture]

### **Permutations [2 lectures]**

- Combinatorial representation of a permutation , Guard representation, Leader representation, Applications ;descent of a permutations, reverse permutation, Eulerian number; Tree representation for a permutation , Applications ;

### **The inclusion-exclusion property [4 lectures]**

- The inclusion-exclusion principle , some applications, derangement; use of Rook polynomial , expansion formula , applications; [2 lectures]
- Euler functions , Mobius function, Mobius inversion formula, applications.[2 lecture]

### **PIGEONHOLE PRINCIPLE [2 lectures]**

- Introduction , Applications , Ramsay theorem .

### **Advanced counting number [2 lectures]**

- Stirling number, Bell number, Catalan number, application.

### **Recurrence relations[6 lectures]**

- Introduction , Hanoi problem ; Fibonacci recurrence relation, Fibonacci sequence , Fibonacci number , Fibonacci expression , Zeckendorfs theorem, weight of a number , Cassini's identity , Applications.[3 lectures]
- Linear recurrence relation, homogeneous recurrence relation, linear homogeneous recurrence relation, linear homo generous recurrence relation with constant coefficient , associated

characteristic polynomial of recurrence relations , case of repeated roots , problems;  
Difference table and sum of polynomials, Applications.[3 lectures]

### **Generating functions [3 lectures]**

- Introduction ,generating function of a sequence; Money exchange problem , Schur's theorem ;  
exponential generating function , E.G.F of the sequence of Bell numbers; Bernoulli numbers;

### **Basic Probability [2 lectures]**

- Introduction, Ballot Problem, Conditional Problem, Bayes' theorem.

### **Random Variable [2 lectures]**

- Random Variables, means, variance, Chebyshev Inequality.

### **References:**

- Discrete and Combinatorial mathematics – An applied introduction.
- R.P. Grimaldi, B.V. Ramana ;Pearson Education (2007) .
- Introductory Combinatorics ;Richard A Brnaldi ;Pearson Education, Inc. (2004).
- Introduction to Enumerative Combinatorics; Miklos Bona ;Mc Graw Hill (2007).
- Combinatorial Techniques ;Sharad S. Sane. ;Hindustan Book Agency.

## **IT-461: Computer Architecture Laboratory**

This practical item has the following objectives:

- Familiarization with microprocessor/microcontroller kits by trying to program for interfacing with different types of devices both parallel and serial; understand interrupts (and DMA) and program for interrupt service routines
- Familiarization with more complex systems (Raspberry PI, PC running linux) and interact with the OS through system calls; writing rudimentary device drivers
- Designing ASICs using FPGA kits

## **IT-462: Object Oriented Programming Laboratory**

The primary objective of this practical item is to encourage and stimulate programming ability and object oriented thinking. A student should be familiar with classes, objects, interfaces, etc, designed by him or available from elsewhere; development environments like Eclipse (for java and/or gnu C++) and Microsoft Developer Studion (for C++, C#) . A student should also learn to model an assigned problem using object oriented modeling and then use the model to develop a software, possibly by sharing components with other students or groups.

## IT-463: Computer Networks Laboratory

This practical item has the following objectives:

- Familiarization of students with basic networking components and devices --- cables; connectors; switches and routers (both wired and wireless); configuration of switches and routers; monitoring network traffic with tools like Wireshark; mapping observed traffic pattern to theory items
- Simulating and manipulating a network --- use of simulator tools like NS2; configuring NS2 to alter traffic pattern; gaining insights on network traffic engineering
- Creating traffic programmatically --- sockets programming; STREAM sockets (client and server); DATAGRAM sockets; programming an ftp client; programming to make DNS queries; etc.

## **IT-484: Seminar**

This item of the curriculum is aimed at preparing the students for making presentations on technical topics. Concerted attempt will made to break the student's fear and natural shyness towards making presentations. Basic tenets of a good presentation will be inculcated at this early stage of their career to prepare them for making much better presentations in later years.

There will be two presentations by each student:

- An early presentation on a topic that has been taught in the class
- A final presentation on a topic from a subject included in the syllabus but has not yet been covered in class

## Syllabus for 5<sup>th</sup> Semester B. Tech in Information Technology

Course/L-T-P	Subject	Credits	Comments
IT-501 (3-1-0)	Database Management Systems	5	Departmental
IT-502 (3-1-0)	Digital Signal Processing	4	Departmental
IT-503 (3-1-0)	Formal Language and Automata Theory	4	Departmental
IT-504 (3-1-0)	Operating Systems-II	4	Departmental
IT-505 (3-1-0)	Multimedia and Data Compression Techniques	4	Common subject
IT-561 (0-0-3)	Database Management Systems Laboratory	3	Practicals
IT-562 (0-0-3)	Advanced Operating Systems Laboratory	3	Practicals
IT-563 (0-0-3)	Multimedia Laboratory	3	Practicals
<b>Total Credits for Semester</b>		30	



## IT-501: Database Management Systems [40 lectures]

**Learning Objective:** Modern day systems must deal with huge amount data. Organization must use some types of database to store the data. The major objective of this course is to give the student the idea about different data models and how to use and deploy them in practical applications. As several organizations use the concepts of RDBMS and therefore apply SQL to implement, the thrust of this syllabus is to give the details idea of RDBMS for theoretical learning and SQL for the practical deployment. Moreover transactional processing is taught to clarify how to handle huge data in real time. After the end of this course the learner would be able to design the relational database model for any application and could implement them using SQL.

### Introductory Concepts [3 lectures]

- Data base system concepts --- Architectural components; data models; schema and instances; DDL, DML and DCL

### Data Modeling Concepts [28 lectures]

- Entity-relationship model --- Concepts and notations for ER diagrams mapping constraints [2 lectures]
- Keys --- The concept of super key, candidate key, primary key, etc.; entity and attribute; generalization and specialization; mapping cardinality; aggregation [2 lectures]
- From ER diagrams to tables --- Network model; hierarchical model; relational model; object based database model; object oriented database model [3 lectures]
- The relational data model --- Concepts, integrity constraints, keys, domain constraints, referential integrity, etc; Codd's Rule; relational algebra; relational calculus; domain and tuple calculus; SQL queries [7 lectures]
- Query optimization --- Query processing blocks; equivalence of expressions; different query optimization techniques [3 lectures]
- Database design --- Functional dependencies; normal forms (first, second, third and Boyce-Codd normal forms); multi-valued dependencies and fourth normal forms; join dependencies and fifth normal form; inclusion dependencies; lossless join decompositions; normalization using functional and multi-valued dependencies [7 lectures]
- File management for DBMS --- Indexing (primary and secondary Index); B-Tree; B+ Tree; hashing extensible hashing; heaps; sequential, indexed sequential and hashed file organization [4lectures]

### Transactions and Concurrency [9 lectures]

- Transaction processing concepts --- Transaction processing systems; ACID properties; schedule and recoverability; testing of serializability; serializability of schedules; conflicts and view; locking; recovery from transaction failures; deadlock handling [6lectures]
- Concurrency control techniques --- Locking techniques for concurrency control; time stamping protocols for concurrency control; etc [3 lectures]

### References

1. Ramez Elsmari, S.K Navate "Fundamentals of Database System" Addison Wesley
2. Avi Silberschatz, Henry F. Korth, S. Sudarshan "Database System Concepts", McGraw-Hill
3. C.J.Date "An Introduction to Database Systems" O'Reilly Media
4. Alexis Leon & Mathews Leon "Database Management Systems" Vikash Publishing

## IT-502: Digital Signal Processing [40 lectures]:

**Learning Objective:** This course will enable the students to enhance their analytical ability in facing the challenges posed by growing trends in communication, control and signal processing areas. It will develop the ability for problem formulation, system design and solving skills through the basic knowledge of Digital Signal Processing and understanding various transformations. Many discrete-time signals classes, shift-invariant systems, convolution, frequency domain transformations, etc will be covered. At the end of this course the students will be well trained to design systems with digital networks using adders, delay elements, coefficient multipliers, etc.

### Introduction to Discrete-time Signals and Systems [10 lectures]

- Classification of Discrete time signals and sequences --- Linear time-invariant (LTI) systems, (BIBO) stability, and causality; linear convolution in time domain; graphical approach [5 lectures]
- The concept of z-Transforms --- Region of convergence; properties; inverse z-transform; realization of digital filter structures (direct forms I and II, transposed form, cascaded form, parallel form) [5 lectures]

### Discrete-time Signals in Transform Domain [12 lectures]

- Discrete Fourier Series (DFS) and Discrete-time Fourier Transforms (DTFT) [4 lectures]
- Discrete Fourier Transform (DFT) --- Properties of DFT, linear convolution using DFT; circular convolution; fast Fourier transforms (FFT); radix-2 decimation in time and decimation in frequency; FFT algorithms; inverse FFT [8 lectures]

### Digital Filters [18 lectures]

- Infinite Impulse-response (IIR) filters ---analog filter approximations (Butterworth and Chebyshev); impulse invariant transformation; bilinear transformation; design of IIR filters from analog filters [8 lectures]
- Finite Impulse-response (FIR) Filters ---Characteristics of FIR filters; frequency response; design of FIR filters using window techniques; comparison of IIR and FIR filters [8 lectures]
- Multi-rate Processing --- Decimation; interpolation; sampling-rate conversion; implementation of sampling rate conversion [2 lectures]

## References

1. **Sanjit K. Mitra**, Digital Signal Processing, 2<sup>nd</sup> Edition, TATA McGraw Hill
2. **John G. Proakis and Dimitris G. Manolakis**, Digital Signal Processing, Principles, Algorithms, and Applications, Pearson Education/PHI, 2007
3. **Alan V. Oppenheim and Ronald W. Schaffer**, Digital Signal Processing, PHI Ed., 2006
4. **Andreas Antoniou**, Digital Signal Processing, TATA McGraw Hill , 2006
5. **MH Hayes**, Digital Signal Processing, Schaum's Outlines, TATA Mc-Graw Hill, 2007

## IT 503: Formal Language and Automata [40 lectures]

**Learning Objective:** This subject covers the fundamental mathematical properties of computer hardware and programming. The course is designed to let the learner determine what can and cannot be computed, how quickly, with how much memory, and on which type of computational model. The concept of 'formal' languages and their relation to automata is covered with a preliminary introduction to more advanced concepts like 'decidability'. The subject has obvious connections with engineering practice and also has purely philosophical aspects. The learner should have a background in programming and discrete mathematics and at the end of the course should be able to design better programming artifacts based on established models and techniques.

### Introductory Concepts [2 lectures]

- Sets, Alphabets, Languages and Grammars; Concept of Production and Derivation, Chomsky hierarchy of languages

### Finite Automata and Regular Languages [12 lectures]

- Regular Expressions and Regular Languages [2 lectures]
- Deterministic and Non-Deterministic Finite Automata --- Mutual equivalence; equivalence with regular expressions and regular grammars (i.e., three-way equivalence) [2 lectures]
- Properties of regular languages --- Closure properties; the Pumping Lemma; the Myhill-Nerode Theorem and minimization of DFA [8 lectures]

### Context-free Languages and Push-down Automata [10 Lectures]

- Context-Free Grammars (CFG) and Context-Free Languages (CFL) --- definition; parse trees; ambiguity; some basic algorithms; Chomsky and Greibach normal forms; pumping lemma and Ogden's lemma for CFL-s [6 lectures]
- Pushdown automata (PDA) --- definition; equivalence with CFG; deterministic PDA; closure properties of CFL-s [4 lectures]

### Context-sensitive Languages and Turing Machine [11 Lectures]

- Context-sensitive grammars (CSG) and languages --- Introduction; linear bounded automata and equivalence with CSG [2 lectures]
- The Turing Machine --- Basic Turing machine (TM) model; computable languages and functions (recursive sets, etc.); TM variants; nondeterministic-TMs and equivalence with deterministic TMs; TMs as generators, acceptors and enumerators; Algorithms; the Universal Turing Machine [9 lectures]

### Undecidability [6 Lectures]

- Decidable languages; the diagonalization method; Rice's theorem and undecidability of problems about languages.

## References

1. **Michael Sipser**, Introduction to the Theory of Computation, 3<sup>rd</sup> Edition, CENAGE Learning.
2. **John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman**, Introduction to Automata Theory, Languages, and Computation, Narosa Publishing House.
3. **John Martin**, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.
4. **Harry R. Lewis and Christos H. Papadimitriou**, Elements of the Theory of Computation, Pearson Education Asia.
5. **Dexter C. Kozen**, Automata and Computability, Undergraduate Texts in Computer Science, Springer.

## IT 504: Operating Systems-II [40 Lectures]

**Learning Objective:** This is a second level course on Operating Systems and the student is expected to have a reasonable background of OS and modern machine architectures. This course gets into details of fundamental issues of an OS like processes and their scheduling, synchronization, memory and I/O management, etc. More advanced concepts like distributed OS, mobile OS, etc, are also introduced. Linux is covered through an in-depth case study. At the end of the course, the student should become aware of the internal workings of most common OS-es and may even contribute to kernel level programming.

### Operating Systems Revisited [10 lectures]

- Processes and Threads --- process life-cycle and scheduling; scheduling algorithms and data structures (heap, binomial and Fibonacci heap); processes and threads; threading models [4 lectures]
- Synchronization Mechanisms --- Semaphores; locks; deadlocks (detection, prevention and recovery) [2 lectures]
- Memory Management Techniques --- Allocation strategies; compaction; paging and segmentation strategies; virtual memory [4 lectures]

### Distributed Operating Systems [10 lectures]

- Issues in Distributed OS --- Architecture; communication primitives; Lamport's Logical clocks; causal ordering of messages; distributed mutual exclusion algorithms; Centralized and distributed deadlock detection algorithms; agreement protocols [4 lectures]
- Distributed File Systems ---- Design issues [2 lectures]
- Distributed Shared Memory ---- Algorithms for distributed shared memory; issues; load distribution; scheduling algorithms; synchronous and asynchronous check-pointing [2 lectures]
- Fault Tolerance --- Two-phase commit; nonblocking commit; security and protection [2 lectures]

### Real Time and Mobile OS [10 lectures]

- Real Time OS --- Basic model (hard, soft); characteristics; applications [2 lectures]
- Real Time Task Scheduling --- Resource sharing; priority based scheduling [2 lectures]
- Mobile OS --- Issues in mobile OS; micro-kernels; processes and threads in mobile OS; memory and file management in mobile OS; case study of Android as OS (media, services and core-OS layers and typical Android SDK) [6 lectures]

### Case Study of Linux [10 lectures]

- Unix Fundamentals --- Assumptions of architecture; kernel and user modes; file handling through inodes; I/O management (device driver architecture) [6 lectures]

- Linux Specificities --- How linux handles the x86 architecture; Linux inter-process communication; dynamic linking; kernel level programming [4 lectures]

## References

1. Pradeep K. Sinha, "Distributed Operating Systems: Concepts and Design", PHI
2. Rajib Mall, " Real-Time Systems: Theory and Practice, 1/e", Pearson Education, 2007, ISBN 9788131700693
3. Maurice J. Bach, "The Design of the UNIX Operating System 1st Edition ", Pearson, ISBN: 9780132017992,



## IT 505: Multimedia and Data Compression Techniques [40 Lectures]

**Learning Objective:** As more text data, images and audio data are required to be processed, data compression techniques will continue to grow and evolve. This course tries to provide students with a good understanding of the more important principles and issues in the field of data and multimedia compression, as well as some of the practical techniques in reliable data compression. Some existing compression standards and compression utilities are examined from mathematical and algorithmic standpoint. At the completion of the course the students should know about the basics of image, video, and audio formation and processing, the basics of multimedia compression and representation and appreciate the impact of classical and modern mathematical concepts that are used.

### Introduction to Multimedia [7 lectures]

- Historical Overview --- Multimedia representations; software tools; authoring tools [2 lectures]
- Basics of Image Formation --- Cameras and lenses; basic camera models and geometry; digitization; standard image formats; colors in images and videos [5 lectures]

### Processing Digital Images [10 lectures]

- Image Computing --- Binary image analysis, processing 2D images; thresholding; convolution; edge and corner detection; mathematical morphology; shape descriptors [3 lectures]
- Case Study --- implementation of a simple Optical Character Recognition (OCR) System [7 lectures]

### Multimedia Compression [23 lectures]

- Fundamental Concepts of Multimedia Compression --- Lossless versus lossy compression; fixed versus variable length coding; dictionary based coding; the Fourier transform; Discrete Cosine Transform (DCT); application of FT and DCT to image compression (JPEG compression) [10 lectures]
- Video Processing Concepts --- Fundamentals of video, image versus video compression; MPEG video coding (MPEG4, MPEG7 and beyond) [5 lectures]
- Audio Processing Concepts --- Fundamentals of digital audio; quantization and transmission of audio signals; audio compression techniques; linear predictive coefficients (LPC), audio codec standards (MP3, AAC, AAC3, etc) [6 lectures]
- Multimedia Applications --- Content-based retrieval in digital libraries; appropriate case studies; HTML and associated concepts [2 lectures]

## References

1. **Steinmetz Ralf**, Multimedia Fundamentals: Media Coding And Content Processing, Nahrstedt Klara, PHI.
2. **Li Ze-Nian and Drew Mark S.**, Fundamentals of Multimedia, PHI
3. **Irina Bocharova**, Compression For Multimedia, Cambridge University Press.
4. **Khalid Sayood**, Introduction to Data Compression, Morgan Kaufmann Series in Multimedia Information and Systems.

## **IT-561: Database Management Systems Laboratory [15 sessions]**

This laboratory would cover experiments and projects around RDMS tools (SQL) and No-SQL databases. The major topics that are to be covered:

1. Writing DDL in SQL
2. Writing DML queries extensively to cover majority of the basic and complex queries
3. Basics of PL/SQL
4. Basics of No-SQL based tools

## IT-562: Advanced Operating Systems Laboratory [15 sessions]

This is a follow-up laboratory session to Operating Systems Lab (IT-364). Experiments will cover:

1. Multithreaded programming on linux using the pthread library; writing an rudimentary http client; echo server; chat application (client+server)
2. Implementing a memory management back-end that serves "malloc" like class in a test program making random malloc calls; extending the concept to block-based memory allocation (from pre-allocated disk blocks) simulating a file system;
3. Implement a priority based scheduler using different data structures (arrays of queues; heaps; binomial/Fibonacci heaps; etc)
4. Build linux from an open-source distribution; write a output-only character device driver that 'does nothing'
5. Implement a serial character device driver that reads writes bytes from COM over USB (the other end being a Arduino program feeding/consuming them)
6. Writing a linux device driver for ethernet --- learning to handle PCI bus; memory-mapped I/O, interrupts and DMA, linux device driver structure, etc. --- a few old PC-s with NIC cards for which driver codes are available open-source, can be used

## **IT-563: Multimedia Laboratory [15 sessions]**

This practical sessions will cover:

- Implementation of some basic image processing algorithms with and without MatLab
- Implementation of some basic audio compression algorithms with and without MatLab
- Implementation of some basic image/video compression algorithms with and without MatLab
- Advanced HTML programming using javascript (json), CSS, HTML5
- Animations using flash and HTML

## Syllabus for 6<sup>th</sup> Semester B. Tech in Information Technology

Course/L-T-P	Subject	Credits	Comments
IT-601 (3-1-0)	Embedded Systems	4	Departmental
IT-602 (3-1-0)	Software Engineering	4	Departmental
IT-603 (3-1-0)	Compiler Design	4	Departmental
IT-604 (3-1-0)	Web and Android Technology	4	Departmental
IT-605 (2-1-0)	Institutional Elective (Economics for Engineers)	3	Common subject
IT-661 (0-0-3)	Embedded Systems Laboratory	3	Practicals
IT-662 (0-0-3)	Software Engineering Laboratory	3	Practicals
IT-663 (0-0-3)	Web and Android Technology Laboratory	3	Practicals
IT-684 (0-0-3)	Seminar	2	Seminar and Viva-voce
<b>Total Credits for Semester</b>		30	

## IT-601 Embedded Systems [40 lectures]

**Learning Objective:** *Embedded computing systems now far outnumber any other type of computing system. Computing devices are being added to an increasingly wide range of everyday products including smart phones, automobiles, toys, home appliances etc. The ability to design, implement and analyze embedded computing systems is in demand in industry as well as academia. This course will enable the students to understand the embedded computer system design problem including multiple constraints, create a design that satisfies the constraints, implement the design in hardware and software, and measure performance against the design constraints. Issues of modern embedded systems development will be covered with appropriate case studies. The student taking this course should be conversant with CPU architecture, fundamental aspects of hardware design, program development (compiling, linking, dynamic linking), debugging, etc, principles of working of OS-es including the concept of multi-tasking, synchronization, etc. At the end of the course, a student should be able to technically appreciate and evaluate a given embedded system and be able to contribute in conception, design and development of new systems.*

### Introduction to Embedded Systems [14 lectures]

- Range of embedded systems --- CPU size and complexity; memory size; I/O handling variations; use of OS-es with single or multitasking [2 lectures]
- Interfacing in embedded systems --- Parallel interfacing and parallel bus with study of the PCI bus; serial interfacing standards like I2C, SPI, etc; serial communication using Rs232C/RS485; the USB concept [6 lectures]
- Embedded systems design cycle --- From idea conception to marketing; reference designs; detailed design; hardware bring-up; flashing code into CPU; debugging; in-circuit emulation; boundary scanning; introduction to JTAG [6 lectures]

### Microcontrollers used in embedded systems [12 lectures]

- Some 8-bit microcontrollers --- 8051 as a historical study; ATMEGA 328 [4 lectures]
- Some 16/32-bit microcontrollers --- ARM cortex, DSP-s; combos (OMAP, Qualcomm multi-cores, etc) [2 lectures]
- Input and Output --- Interrupts and service routines; DMA; 'deferred' interrupts; case study with serial I/O [4 lectures]

### Aspects of Real-time Operating Systems (RTOS) [5 lectures]

- Introduction to RTOS --- Soft versus hard real-time; concept of deadline; coding discipline under RTOS [1 lecture]
- Study of some RTOS-es --- Some suitable examples like eCos, RT-Linux, WindowsCE, etc [4 lectures]

## Software Development for Embedded Systems (under and OS) [9 lectures]

- Device Driver Concepts --- Introduction to device drivers; module versus port drivers; handling interrupts and MA; virtual to physical address translation; case study with a serial driver [7 lectures]
- Introduction to firmware, middleware, code-libraries, etc. [2 lectures]

## References

1. **Wayne Wolf**, Computers as Components-principles of Embedded computer system design, Elsevier.
2. **Kenneth J. Ayala**, The 8051 Microcontroller, Third Edition, Thomson.
3. **Labrosse**, Embedding system building blocks, via CMP publishers.
4. **Michael Barr and Anthony Massa**, Programming Embedded Systems: With C and GNU Development Tools, O'Reilly Media.
5. **Jonathan Corbet, Alessandro Rubini and Greg Kroah-Hartman**, Linux Device Drivers, 3rd Edition, O'Reilly Media.
6. **Jean J. Labrosse**, MicroC OS II: The Real Time Kernel (With CD-ROM) by Micrium Press.



## IT-602: Software Engineering [40 Lectures]

**Learning Objective:** *Software Engineering is a foundation and state-of-the art to build and deploy software in real life environment. This involves analysis, designing, modeling, testing and maintaining of software. This course is designed to introduce the concepts to the students along with the practical ideas of how to deploy in a real life system. At the end of the course the student would have a clear idea of theoretical and practical aspects software related issues. The course also covers different mathematical and analytical aspects of software engineering. After the completion of this course, students should be able to design the software model for the small applications would be in a position to participate in larger software projects with a good understanding of modeling and analysis of real life software.*

- Introduction: System, Software, Engineering, Benefits, Problems, Errors, SDLC, Documentation, Structured Programming, Coupling, Cohesion
- Tools: DFD, Structure Chart, Decision Table, Decision Tree
- Design: Top-Down, Bottom-Up, Hybrid.
- Model: Water Fall, Iterative Waterfall, Prototype, Iterative Enhancement, Spiral.
- Cost Model: Function Point Metric, Estimation Model, Heuristic, COCOMO, Analytical.
- Testing: Unit, Integration, Function, Performance, Acceptance.
- Maintenance: Corrective, Preventive, Perfective, Adaptive.
- Quality: Metrics, SQA activities, Standard, Verification, Validation, Review, Inspection, Walkthrough.
- Risk Analysis: Identification, Projection, Refinement, Monitoring & Management.
- Management Activities: Software Project Management, Software Configuration Management, Staffing, Scheduling.
- Software Reliability & Failures: Introduction, Usefulness, Basic Model.
- Advanced Methodologies: Agile, Reverse Engineering, Re-Engineering, Six Sigma#

### References

1. Roger S. Pressman "Software Engineering: A Practitioner's Approach" McGraw-Hill
2. Pankaj Jalote "Software Engineering: A Precise Approach" Wiley
3. Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli "Fundamentals of Software Engineering" Pearson

4. Richard Fairley "SOFTWARE ENGINEERING CONCEPTS" McGraw-Hill

## IT-603: Compiler Design [40 lectures]

**Learning Objective:** *This course explores the principles, algorithms, and data structures involved in the design and construction of compilers. Topics include underlying theories like finite-state machines, lexical analysis, context-free grammars, push-down parsers, LR and LALR parsers and other parsing techniques, symbol tables, error recovery, and an introduction to intermediate code generation. Students should have completed a course on Formal Languages and Automata and the course will cover aspects of how theories have been put to very effective practical use. At the end of the course, a student should be able to develop compilers targeting any platform and possibly have developed a compiler for a subset of C.*

### Introduction [1 lecture]

- Phases of compilation and overview

### Details of Compiler Phases [33 lectures]

- Lexical Analysis --- Regular languages and regular expressions; finite automata; from regular expressions to finite automata; scanner generator (lex/flex) [2 lectures]
- Syntax Analysis --- Context-free languages and grammars; push-down automata; LL(1) grammars and top-down parsing; Operator and operator precedence grammars; introduction to LR parsing; automatic construction of LR parsers --- SLR(1), LR(1), LALR(1) grammars and bottom-up parsing; ambiguity and LR parsing; LALR(1) parser generator (yacc, bison) [12 lectures]
- Semantic Analysis --- Attribute grammars; syntax directed definition; evaluation and flow of attribute in a syntax tree [2 lectures]
- Symbol Table and its structure --- symbol attributes and management [2 lectures]
- Run-time Environment --- Procedure activation, parameter passing, value return, memory allocation, and scope [2 lectures]
- Intermediate Code Generation --- Translation of different language features; different types of intermediate forms; generation of 3-address codes from parser [6 lectures]
- Code Improvement and Optimization --- Control and data-flow analysis; data-flow dependence etc.; code improvement; local and global optimization; loop optimization; peep-hole optimization; architecture dependent code improvement; instruction scheduling (for pipeline), loop optimization for cached environment [6 lectures]
- Register allocation and target code generation [1 lecture]

## Advanced topics [6 lectures]

- Type systems [2 lectures]
- Data abstraction [2 lectures]
- Compilation of Object Oriented features and non-imperative programming languages [2 lectures]

## Reference

1. **A. V. Aho, Monica S. Lam, R. Sethi, & J. D. Ullman**, Compilers Principles, Techniques, and Tools, 2nd ed., Pearson Ed., 2008
2. **Keith D. Cooper & Linda Troczon**, Engineering a Compiler, Morgan Kaufmann, Elsevier, 2004
3. **Santanu Chattopadhyay**, Compiler design, PHI, 2005
4. **Michael L. Scott**, Programming Language Pragmatics, Morgan Kaufmann, Elsevier, 2006

## IT-604: Web and Android Technology [40 Lectures]

**Learning Objective:** This course covers fundamental design patterns and philosophies associated with modern web application architectures, along with their major components. This course also covers the core components of the Android platform that underlie all Android applications, advanced components and concepts provided by the Android platform. After completion of this course students are able to design, develop and deploy a modern web application (This course is not about how to build a pretty web page), understand the major architectural components in web apps, and how they fit together and build a complete android app from scartch.

### Essential of Web Technology [26 lectures]

- Web Essentials --- Clients and servers and their inter-communication; basic protocols for the web (HTTP); the request-response regime; markup languages (HTML, XHTML, etc); X/HTML syntax and fundamental elements, lists, tables, forms; XML [2 lectures]
- Server-Side Programming --- Java servlets (architecture, overview, generating dynamic content, sessions, cookies, etc); data storage in servlets; concurrency; case study [5 lectures]
- Web Data Modeling --- XML documents; namespaces; Javascript and XML (Ajax); javascript DOM; XML processing (SAX and DOM parsing); transforming XML Documents; stylesheetsjava server pages (JSP); JSP and servlets;- JSP tag libraries and files support; the model-view-controller paradigm; introduction to HTML5 [10 lectures]
- Web Services --- The concept of a Java web service; writing a web service; WSDL; data types; XML schema; communicating object; SOAP technology; REST APIs [9 lectures]

### Essentials of Android Technology [14 lectures]

- Introduction to Android OS --- History; architecture; Android stack; directory structure; boot process; security architecture; aspects of low power programming [1 lecture]
- Exploring the Android SDK --- Documentation; application life cycles; Android APIs; client side basics; building a multi page application; common controls; offline access; string translation; local storage; encrypted cache; JSON store [1 lecture]
- The Android UI Framework ---Working with jQuery mobil, Dojo mobile, Sencha Touch, Apache Cordova (overview, creating plug-ins, etc); the native plugin class for Android; native plugin class for iOS; plugin Javascript wrapper; WebViewOverlay plugin [8 lectures]

- Integration Adapter --- Overview; SQL adapters; HTTP adapters; HTTP adapter with SOAP services; cast iron adapter; JMS adapter; invoking adapter procedures from java code; server side scripting [2 lectures]
- Security --- Adapter based authentication; server-side authentication; custom login modules; WebSphere; LTPA-based authentication [2 lectures]

## References

1. **Jeffrey C. Jackson**, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006.
2. **Robert. W. Sebesta**, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007.
3. **Jerome (J.F.) DiMarzio**, "Android Programming Guide", TaTa McGraw-Hill

### **IT-661: Embedded System Laboratory [15 sessions]**

Experiments relevant to different Embedded systems and tools

### **IT-662: Software Engineering Laboratory [15 sessions]**

The practical sessions will cover:

1. Analyzing a Problem and structuring it towards IT based Solution
2. Writing SRS
3. Designing aspects such as DFD, UML etc.
4. Integration with Database Design
5. Orientation towards product development

### **IT-663: Web Android Technology Laboratory [15 sessions]**

Experiments relevant to different web based software and upcoming technologies with a major emphasis on Android.

### **IT-684: Seminar**

## Syllabus for 7<sup>th</sup> Semester B. Tech in Information Technology

Course/L-T-P	Subject	Credits	Comments
IT-701 (4-1-0)	Data Engineering	5	Departmental
IT-702 (4-1-0)	Information Security	5	Departmental
IT-753 (3-1-0)	Elective-1	4	Departmental
IT-754 (3-1-0)	Elective-2	4	Departmental
IT-755 (2-1-0)	Institutional Elective	3	Common subject
IT-785 (0-0-3)	Seminar	2	Seminar and Viva-voce
IT-796 (0-0-3)	Mini Project	8	Project
<b>Total Credits for Semester</b>		30	

### Elective-1

MOBILE COMPUTING  
WIRELESS SENSOR NETWORK  
HIGH PERFORMANCE COMPUTING  
INTERNET OF THINGS  
SYSTEM SIMULATION AND MODELLING

### Elective-2

ARTIFICIAL INTELLIGENCE  
COMPUTATIONAL GEOMETRY  
VLSI  
NATURAL LANGUAGE PROCESSING  
HUMAN COMPUTER INTERACTION

### IT-701 Data Engineering [40 Lectures]

#### Course Objective:

To understand different components of data engineering life cycle like extraction of the data, storage, modeling and visualization. To develop an understanding of big data ecosystem and it's relation with the data engineering life cycle

#### Introduction: [2 Lectures]

- Data Lifecycles, Different components of Data lifecycle

#### Data Extraction and Transformation [4 Lectures]

- Data Extraction
- Push versus Pull, APIs , Web Scraping
- Data Scaling and other Transformation

**Data Warehouse [10 Lectures]**

- Architectures
- Measures, Codd's guideline, Data Mart, Virtual Data Warehouse
- Schema, Lattice of Cuboids, Concept Hierarchy,
- ETL Tools

**Big Data Ecosystem: [10 Lectures]**

- Definition and 5 Vs
- Hadoop and Mapreduce
- Other Important components Hbase, Hive etc.
- No SQL Database

**Data Mining and Machine Learning: [12 Lectures]**

- Definition
- Different types of tasks Classification, Clustering, Regression, Association rule Mining etc.
- Algorithms Naïve Bayes, Decision Tree, K- Means, DBSCAN, Linear and logistic Regression, Apriori

**Visualization [2 Lectures]**

- Different types of reports
- exploratory techniques like Scatter Plot, Box Plot, Heat map

**Reference Books:**

1. Han J, Pei J, Kamber M. Data mining: concepts and techniques.
2. Tony Ojeda, Sean Patrick Murphy, Benjamin Bengfort, Abhijit Dasgupta, "Practical Data Science Cookbook", Packt Publishing Ltd., 2014.
3. Hadoop – The Definitive Guide 4<sup>th</sup> Edition Tom White , O'Reilly Media
4. Machine Learning 1st Edition (English, Paperback, Tom M. Mitchell)

**IT 702: INFORMATION SECURITY [40 Lectures]****Course Objectives:**

1. To understand the fundamentals of Cryptography.
2. To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
3. To understand the various key distribution and management schemes.
4. To understand how to deploy encryption techniques to secure data in transit across data networks.
5. To design security applications in the field of Information technology.

**Learning Outcome:**

1. Analyse the vulnerabilities in any computing system and hence be able to design a security solution.
- 2: Identify the security issues in the network and resolve it.



3: Evaluate security mechanisms using rigorous approaches, including theoretical

### **Introduction and Symmetric Key Encipherment [15L]**

*Introduction:* Security goals, Attacks, Services and Mechanism, Techniques.

*Symmetric Key Mathematics:* Integer arithmetic, Modular arithmetic, Residue Matrices, Linear Congruence, Algebraic Structures,  $GF(2^n)$  fields.

*Traditional Symmetric Key Ciphers:* Substitution ciphers- monoalphabetic, polyalphabetic. Transposition ciphers- keyless, keyed. Kirchoff's Principle, Crypto-analysis attacks- cipher text only, known plaintext, chosen plaintext, chosen cipher text.

*Block and Stream ciphers:* Operational principal, Electronic Codebook Mode (ECB), Chain Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), Counter Mode (CTR).

*Data Encryption Standard (DES):* Structure, Key generation, Security, Multiple DES.

### **Asymmetric Key Encipherment [15L]**

*Symmetric Key Mathematics :* Primes, Euler's Phi function, Fermat's theorem, Euler's Theorem, Chinese Remainder Theorem, Quadratic Congruence, Fast Exponentiation , Discrete Logarithm.

*Asymmetric Key Cryptography :* General Principal, Trapdoor One Way Function, Knapsack Cryptosystem, RSA Cryptosystem, Rabin Cryptosystem, El-Gamal Cryptosystem, *Elliptic Curve Cryptography*

### **Integrity, Authentication, and Key Management [10L]**

*Message Integrity :* Document and Finger print, Message and Message Digest, Difference, integrity Check.

*Message Authentication:* Modification Detection Code (MDC), Message Authentication Code (MAC), Cryptographic Hash Functions , Digital Signature.

*Entity Authentication:* Password, Challenge Response, Zero Knowledge, Biometrics.

*Key Management:* Symmetric Key Distribution, Public Key Distribution.

*Key Management and Distribution:* Symmetric key distribution using symmetric encryption, Diffie-Hellman Exchange, Hierarchical Key Control, Session Key Lifetime, Transparent Key Control Scheme, Decentralized Key Control, Controlling Key Usage, Symmetric Key Distribution Using Asymmetric Encryption, Simple Secret Key Distribution Scheme, Distribution Of Public Keys, Public Announcement Of Public Keys, Publicly Available Directory, Public Key Authority, Public Keys Certificates.

### **Reference Books:**

1. Handbook of Applied Cryptography, Alfred J. Menezes, Paul C. Van Oorschot and Scott A. Vanstone, CRC Press.
2. Cryptography and Network Security: Principles and Practice, William Stallings, Prentice Hall of India.
3. A course in number theory and cryptography, Neal Koblitz, , Springer.
4. Introduction to Cryptography, Undergraduate Text in Mathematics, Johannes A. Buchmann, Springer.
5. Cryptography Theory and Practice, Doug Stinson, CRC Press.
6. Public-Key Cryptography: Theory and Practice, A. Das and C. E. VeniMadhavan, Pearson Education Asia.

7. Foundations of Security: What Every Programmer Needs to Know (Expert's Voice), Neil Daswani, Christoph Kern and Anita Kesavan: A Press, 2007.

## **IT-753: MOBILE COMPUTING – CHOICE 1** [40 Lectures]

### **Course Objectives:**

1. Learn about the concepts and principles of mobile computing;
2. To explore both theoretical and practical issues of mobile computing;
3. To develop skills of finding solutions and building software for mobile computing applications.

### **Learning Outcomes:**

1. Grasp the concepts and features of mobile computing technologies and applications.
2. To have a good understanding of how the underlying wireless and mobile communication networks work, their technical features, and what kinds of applications they can support.
3. Identify the important issues of developing mobile computing systems and applications.
4. Organize the functionalities and components of mobile computing systems into different layers and apply various techniques for realizing the functionalities.
5. To develop mobile computing applications by analysing their characteristics and requirements, selecting the appropriate computing models and software architectures, and applying standard programming languages and tools.
6. To organize and manage software built for deployment and demonstration.

**Introduction to Mobile Communications and Computing:** Applications, limitations and architecture. Radio Communications: power model, path loss model, interference and frequency reuse. Cellular Networks: Generations, GSM, GPRS, UMTS and future architectures. [5L]

**Wireless Medium Access Control:** Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA. IEEE 802.11 architecture and services. Distributed Coordination Function (DCF) and Point Coordination Function (PCF). [5L]

**UMTS and LTE system:** UTRAN Architecture, Core NW, NW Database, Radio Bands/coding, Initial procedures, Paging, Authentication/Encryption, Logical/transport/Physical Channels, Radio Bearers, Timers, CS Operations, PS Operations, Call establishment, Mobility and Handovers (hard/soft/softer). Session Management, PDP contexts, Movement to LTE: Simplified RAN, frequency bands, IP only, EPC nodes and interaction [6L]

**Mobile Network Layer:** Basic Mobile IP, Mobile IP Type-MIPv4 and MIPv6, Mobile IP: Concept and Operations. MIPv4 Reverse Tunnelling, MIPv4 Triangular Routing, Problems and Limitations of MIP, MIPv4 Route Optimization. [3L]

**Mobile Transport Layer:** Factors affecting TCP performance in wireless and mobile environment. Indirect TCP (I-TCP), Snooping TCP(S-TCP), Mobile TCP (TCP) and Transaction oriented TCP (T-TCP). [3L]

**Mobile Ad hoc Networks (MANETs):** Overview, Properties of a MANET, spectrum of MANET applications, routing and various routing algorithms, security in MANETs. [3L]

#### **5G (15L):**

**5G Overview/Drivers (2L):** 5G Introduction – Features/Need/Use Cases, 5G System Architecture: RAN Architecture/Core Architecture/Service Based Architecture;

**Air Interface Overview (5L):** 5G RAN Deployment Options:EN-DC, RRC diversity, Split Bearer ; Introduction to the 5G Air Interface,5G Spectrum Framework / Access Schemes / Coding schemes /Dynamic TDD/Massive MIMO, mmWave ;

**5GNAS Operation (3L):** Initial Procedures/ EN-DC addition/RRC-Reconfig/ENDC modify-release.

**5G security (1L):** Authentication/NAS-RRC signaling;

**Mobility Procedures (2L):** DRX changes/Xn and N2 Handover ;

**Software-Defined 5G Architecture (2L):** Network Evolution Towards 5G System; Flexible and Software-Based Network ; Network Slicing ; 5G UDN, UCN and Multi-Connectivity ; 5G Device-to-Device and Vehicular Communications; 5G QoS Framework

#### **Reference Books:**

1. MartynMallick, “Mobile and Wireless Design Essentials”, Wiley Publishing, 2003.
2. Yu-Kwong Ricky Kwok,Vincent K.N. Lau, “Wireless Internet and Mobile Computing: Interoperability and Performance”, Wiley-IEEE Press, 2007.
3. D.P. Agrawal and Q.-A. Zeng, “Introduction to Wireless and Mobile Systems”, 2nd edition, Thomson Learning, 2006.
4. RezaB’Far, “Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML”, Cambridge University Press, 2005.
5. J. Schiller, “Mobile Communications”, 2nd edition, Pearson Education, 2003.

### **IT-753: WIRELESS SENSOR NETWORK – CHOICE 2 [40 Lectures]**

#### **Course Objectives:**

1. To understand the fundamentals of wireless sensor network from the viewpoint of application scenario, design issues and constraints.
2. To understand the working principle different MAC protocols in wireless sensor network.
3. To understand the concept of clustering and chain topology and associated data routing methods.
4. To understand the different embedded operating systems in a sensor node.
5. To understand the different application areas of wireless sensor network.

#### **Learning Outcome:**

1. To analyse an application scenario and design the wireless sensor network to implement it.

2: Maximising network lifetime by choosing an appropriate data routing protocol satisfying the required QoS for the envisaged application.

**Introduction:** commercially available sensor nodes - Imote, IRIS, Mica Mote, EYES nodes, BTnodes, TelosB, Sunspot, node – architecture, sensing and communication range, design issues, energy consumption, clustering of sensors, applications. [8L]

**Medium Access Control Protocols:** Fundamentals of MAC protocols - Low duty cycle protocols and wakeup concepts - Contention-based protocols - Schedule-based protocols - SMAC - BMAC - Traffic-adaptive medium access protocol (TRAMA) - The IEEE 802.15.4 MAC protocol. [10L]

**Routing And Data Gathering Protocols :** Routing Challenges and Design Issues in Wireless Sensor Networks, Flooding and gossiping – Data centric Routing – SPIN – Directed Diffusion – Energy aware routing - Gradient-based routing - Rumour Routing – COUGAR – ACQUIRE – Hierarchical Routing - LEACH, PEGASIS – Location Based Routing – GF, GAF, GEAR, GPSR – Real Time routing Protocols – TEEN, APTEEN, SPEED, RAP - Data aggregation - data aggregation operations - Aggregate Queries in Sensor Networks - Aggregation Techniques – TAG, Tiny DB. [15L]

**Embedded Operating Systems:** Introduction to Tiny OS – NesC – Interfaces and Modules- Configurations and Wiring - Generic Components -Programming in Tiny OS using NesC, Emulator TOSSIM. [7L]

#### **Reference Books:**

1. Kazem Sohraby, Daniel Minoli and Taieb Znati, “ Wireless Sensor Networks Technology, Protocols, and Applications“, John Wiley & Sons, 2007.
2. Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks”, John Wiley & Sons, Ltd, 2005.
3. K. Akkaya and M. Younis, “A survey of routing protocols in wireless sensor networks”, Elsevier Ad Hoc Network Journal, Vol. 3, no. 3, pp. 325--349
4. Philip Levis, “ TinyOS Programming”
5. Holger Karl & Andreas Willig, " Protocols And Architectures for Wireless Sensor Networks" , John Wiley, 2005.
6. Feng Zhao & Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.
7. Kazem Sohraby, Daniel Minoli, & Taieb Znati, “Wireless Sensor Networks - Technology, Protocols, And Applications”, John Wiley, 2007.
8. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.

**Course Objective:** Provide basic knowledge in High Performance Computing with focus on practical aspects and include examples which are relevant to the current industry requirements like Compiler Optimization and Open MP Parallelization, MPI Programming, Vectorization, Usage of Performance Libraries and Profiling.

**Learning Outcome:**

After successful completion of the course, student will be able to:

- Understand the architecture of modern CPU's and how this architecture influences the way programs should be written.
- Optimize all aspects in the processes of programming: from compilation, starting and running program by OS, executing (parallel) instructions by CPU, to writing output to disk.
- Write numerical software that exploits the memory hierarchy of a CPU, to obtain a code with close to optimal performance.
- Analyze an existing program for OpenMP and MPI parallelization possibilities.
- Evaluate the possibilities of accelerators to speed up computational work.

**Syllabus:** [40 lectures]

**Parallel architecture:** [6]

Inter-process communication, Synchronization, Mutual exclusion, Basics of parallel architecture, Introduction to MPI, Portable Extensible Toolkit for Scientific Computation, and other software frameworks for HPCParallel programming with message passing using MPI.

**Parallel Processing Concepts:** [6]

Levels of parallelism (instruction, transaction, task, thread, memory, function), Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc), Architectures: N-wide superscalar architectures, multi-core, multi-threaded Processor Architecture, Multicore-Interconnection.

**High performance computing architectures:** [10]

BM CELL BE, Nvidia Tesla GPU, Intel Larrabee Micro architecture and Intel Nehalem micro-architecture.

**Design Issues in Parallel Computing:** [12]

Memory hierarchy and transaction specific memory design: Thread Organization, Synchronization, Scheduling, Job Allocation, Job Partitioning, Dependency Analysis, Mapping Parallel Algorithms onto Parallel Architectures, Open MP, Performance Analysis of Parallel Algorithms Limitations Facing Parallel Computing, Bandwidth Limitations, Latency Limitations, Latency Hiding/Tolerating Techniques and their limitations.

**Parallel Programming Languages:** [6]

OpenCL and CUDA programming

**Reference Books:**

1. Kai Hwang & Briggs: Computer Architecture and Parallel Processing.

2. Kai Hwang: Advanced Computer Architecture Parallelism, Scalability, Programmability.
3. J. L. A. Patterson, Morgan Kaufmann: Computer Architecture: A Quantitative Approach.
4. A. Silberschatz, P. B. Galvin, G. Gagne, Operating System Concepts, John Wiley.
5. Charles Severance and Kevin Dowd, O Reilly: High Performance Computing (RISC Architectures, Optimization & Benchmarks)
6. K. A. Dowd, Michael Kosta Loukides, O Reilly: High Performance Computing.
7. Ananth Grama et. al, Addison-Wesley: Introduction to Parallel Computing.
8. David E. Culler, Jaswinder Pal Singh, Anoop Gupta: Parallel Computer Architecture: A Hardware/Software Approach, Gulf Professional Publishing.
9. R. E. Bryant and D. R. O'Hallaron, Computer Systems: A Programmer's Perspective, Prentice Hall.
10. Jason Sanders and Edward Kandrot: CUDA by example - An introduction to General-Purpose GPU programming.

## **IT-753: INTERNET OF THINGS – CHOICE 4** [40 Lectures]

### **Course Objective:**

Today we live in an era of connected devices (mobile phones, computers etc.), the future is of connected things like - home appliances, vehicles, personal accessories, and other equipments. Internet of Things (IoT) is a term given to the attempt of connecting objects to the internet and also to each other - allowing people and objects themselves to analyze data from various sources in real-time and take necessary actions in an intelligent fashion. This course gives an introduction to IoT, with examples to understand the IoT market perspective. It teaches the students how to implement efficient data and knowledge management gathered from IoT devices. It also enables the students to understand the current state-of-the-art IoT architectures.

### **1. Introduction to IoT: [2 lectures]**

Overview of IoT, IoT and M2M, Business models for the IoT

### **2. Basic IoT architecture: [8 lectures]**

Hardware, SoC, sensors, device drivers, Sensor Network

### **3. IoT Standards and importance of cloud computing: [2 lectures]**

IoT standards, Cloud computing for IoT, W3C Social Web WG, WoT

### **4. Communication Protocols for IoT: [10 lectures]**

CoAP, XMPP, CoSIP, Zigbee, IoT bridge/gw

### **5. Services and Attributes: [4 lectures]**

Dependability, Security, Maintainability, Domain specific applications of IoT, Introduction to different IoT tools, Developing applications through IoT tools.

## 6. Fog Computing and programming with Python: [14 lectures]

Fog Computing, Introduction to Python Programming. Implementation of IoT using Raspberry Pi

### Reference Books:

1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases, by Pethuru Raj and Anupama C. Raman (CRC Press)
2. Internet of Things: A Hands-on Approach, by Arshdeep Bahga and Vijay Madisetti (Universities Press).
3. Architecture and Technologies for the Internet of Things, Luca Davoli, [dspace-unipr.cineca.it/bitstream/1889/3423/4/Luca\\_Davoli\\_PhD\\_Thesis\\_DSpace.pdf](http://dspace-unipr.cineca.it/bitstream/1889/3423/4/Luca_Davoli_PhD_Thesis_DSpace.pdf).

## IT-753: SYSTEM SIMULATION AND MODELLING – CHOICE 5

[40 Lectures]

### Course Objectives:

The course aims to teach the generic (*i.e.*, tool and application domain independent) concepts of modelling and simulation. By the end of this course, you should have a deep understanding of the concepts of modelling and simulation of dynamic systems using a variety of formalisms. You should be able to build modelling and simulation systems. This will give you ample background to understand and use existing modelling and simulation systems. The course presents general modelling and simulation principles by applying them to concrete problems.

Applications (software process modelling and simulation, reactive systems design such as complex graphical user interfaces, population dynamics analysis, traffic analysis, supermarket queueing, *etc.* ) are used to illustrate the different modelling formalisms.

<b>Course Introduction. What is Modelling and Simulation?</b>	<b>[1 Lectures]</b>
<b>Advantages, Disadvantages and Pitfalls of Simulation.</b>	<b>[1 Lectures]</b>
<b>Model Syntax and Semantics.</b>	<b>[3 Lectures]</b>
<b>Review of Basic probability and statistics</b>	<b>[4 Lectures]</b>
<ul style="list-style-type: none"><li>• Random variables</li><li>• Stochastic processes</li></ul>	
<b>Object-oriented Modelling of Physical Systems.</b>	<b>[3 Lectures]</b>
<b>Process Interaction (GPSS).</b>	<b>[3 Lectures]</b>
<b>Pseudo-random generators, input/output analysis.</b>	<b>[2 Lectures]</b>

**Continuous-time models, solvers, sorting.** [5 Lectures]

**Population Dynamics, System Dynamics.** [5 Lectures]

**System Specification Hierarchy, Model Classification.** [3 Lectures]

**Mini Project:** [10 Hours]

**Reference Books:**

1. A plethora of modelling formalisms: Paul A. Fishwick. Simulation Model Design and Execution, building digital worlds: Prentice Hall, 1995.
2. The foundations of modelling and simulation: Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim
3. Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press

**IT-754: ARTIFICIAL INTELLIGENCE – CHOICE 1**

[40 Lectures]

**Course Objective:**

The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches. Develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning. The course also provides an overview of deep learning and its application.

**Introduction: [2 Lectures]**

Introduction to AI and Intelligent Agents

**Problem Solving: [12 Lectures]**

- Solving Problems by Searching, heuristic search techniques
- constraint satisfaction problems
- stochastic search methods, Best First search, Hill Climbing, Simulated Annealing, Genetic Algorithm, A \*

**Game Playing: [6 Lectures]**

minimax, alpha beta pruning

**Knowledge and Reasoning: [8 Lectures]**

- Building a Knowledge Base:
- Propositional logic, first order logic
- Theorem Proving in First Order Logic Planning, partial order planning.
- Uncertain Knowledge and Reasoning,
- Probabilities, Bayesian Networks.



**Learning: [12 Lectures]**

- Overview of different forms of learning,
- Learning Decision Trees,
- Neural Networks.
- Introduction to Natural Language Processing

**Reference Books:**

1. Russell SJ, Norvig P. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, 2016.
2. Khemani D. A first course in artificial intelligence. McGraw-Hill Education; 2013.
3. Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. Cambridge: MIT press; 2016 Nov 18.

**IT-754: COMPUTATIONAL GEOMETRY – CHOICE 2****[40 Lectures]**

**Course Objective:** This course introduces geometric data structures design and analysis of multi-dimensional search for the problems in computational geometry, notions of convex hull, Voronoi diagrams and Delaunay Triangulations, and facilitates the students to apply geometric techniques to real-world problems.

**Multi-dimensional search [15 lectures]**

Dynamic unweighted and weighted trees, k-d tree, range tree, priority search tree, finger-tree, interval trees, amortized analysis of dynamically maintaining weight balanced binary trees [8 lectures];

Dynamic search trees in secondary storage; Orthogonal range searching, order decomposable problems, multi-dimensional divide and conquer [7 lectures].

**Algorithms on polygons and related techniques [17 lectures]**

Point location in monotone subdivision, convex polygons, convex hull of point set and polygon in 2 and 3 dimensions [5 lectures];

Voronoi diagram, Delaunay triangulation, Arrangement and duality, triangulation of polygons, binary space partitioning [8 lectures];

Visibility, simplex range searching, isothetic geometry, matrix searching [4 lectures].

**Applications of computational geometry [8 lectures]**

Applications in different geometric optimization problems, few applications in GIS and robot motion planning, design in VLSI.

**Reference Books:**

1. F. P. Preparata and M. I. Shamos: Computational Geometry: An Introduction, Springer-Verlag, Berlin, 1985.
2. M. De Berg, M. Van Kreveld, M. Overmars and O. Schwarzkopf: Computational Geometry: Algorithms and Applications, Springer-Verlag, Berlin, 1997,
3. J. O'Rourke: Computational Geometry in C, Cambridge University Press, London, 1995.
4. K. Mehlhorn and St. Naher: The LEDA Platform of Combinatorial and Geometric Computing, Cambridge University Press, London, 1999.

### **IT-754: VLSI – CHOICE 3**                      **[40 Lectures]**

**Course Objective:** This course give a basic idea of the theories and techniques of design and testing of VLSI circuits implemented using CMOS technology. In this course, we will learn the fundamental concepts and structures of designing digital VLSI systems include CMOS devices and circuits, standard CMOS fabrication processes, CMOS design rules, static and dynamic logic structures, interconnect analysis, CMOS chip layout, simulation and testing, low power techniques, design tools and methodologies, VLSI architecture. The course gives an understanding of the different steps in designing a vlsi circuit and the different techniques for testing them.

1. **Introduction to VLSI Physical Design : [1 Lecture]**  
VLSI design cycles, Design and system packaging style, design rules.
2. **Fabrication of VLSI Devices : [4 Lectures]**  
Transistor fundamentals, Fabrication material, fabrication of vlsi circuits, layout of basic devices, Scaling method.
3. **Partitioning : [4 lectures]**  
Kernighan-Lin Algorithm, Stimulated Annealing.
4. **Floor Planning and Pin-Assignment: [6 lectures]**  
Constraint based floor planning, Integer programming based floor Planning, Rectangular Dualization, Floorplanning for Mixed block and cell design, Timing driven Floorplanning, General Pin Assignment, Channel Pin Assignment.
5. **Placement : [5 Lectures]**  
Simulated Annealing, Force Directed Placement, Sequence-Pair Technique, Breuer's algorithm, Terminal Propagation Algorithm.
6. **Routing : [10 lectures]**

(i).Global Routing - Maze Routing Algorithms, Line-Probe Algorithms, Shortest Path Based Algorithms, Steiner Tree based Algorithms.

(ii). Detailed Routing – Routing Models, routing problems, single layer routing algorithms, double layer channel routing algorithms, three layer channel routing algorithms.

#### **7. Compaction : [2 lectures]**

One-Dimensional compaction, 1 1/2-Dimensional compaction, two-Dimensional compaction, Hierarchical compaction.

#### **8. Basics of Testing and Fault Modelling : [8 lectures]**

Introduction to Testing, Faults in digital circuits, Modeling of Faults, Logical Fault Model, Fault Detection, Fault Location, Stuck-at-fault Models, Fault Equivalence, fault collapsing, fault Dominance, Checkpoint Theorem.

#### **Reference Books :**

1. Algorithms for VLSI Physical Design Automation, Naveed Sherwani.
2. An introduction to VLSI physical design, Majid Sarrafzadeh, C.K. Wong
3. Essential of Electronic Testing for Digital, Memory and Mixed Signal VLSI circuits : Micheal L. Bushnell, Vishwani Agarwal.
4. VLSI Physical Design, Andrew B. Kahng, Jens Liening, Igor L. Markov, Jin Hu.
5. VLSI Design, K. Lal. Kishore, V.S.V Prabhakar.
6. Digital Systems and Testable Design, M. Abramovici, M.A. Breuer, A.D. Friedman, Jaico Publishing House.

### **IT-754: NATURAL LANGUAGE PROCESSING – CHOICE 4**

**[40 Lectures]**

#### **Course Objective:**

To understand natural language processing and to learn how to apply basic algorithms in this field. To get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora. To understand the techniques of retrieval, discovering relationship, topic modeling etc. To develop an understanding of word embedding and LSTM.

#### **Introduction: [2 Lectures]**

- NLP tasks in syntax, semantics, and pragmatics.
- Applications such as information extraction, question answering, and machine translation.
- The problem of ambiguity
- The role of machine learning. Brief history of the field

#### **Regular Expressions and Context Free Grammars [4 Lectures]**

- Regular languages, and their limitations.
- Finite-state automata. Practical regular expressions for finding and counting language phenomena.

- Constituency, CFG definition
- use and limitations.
- Top-down parsing, bottom-up parsing, and the problems with each.
- The desirability of combining evidence from both directions

### **Language modeling and Naive Bayes: [8 Lectures]**

- Probabilistic language modeling and its applications.
- Markov models. N-grams.
- Estimating the probability of a word, and smoothing.
- Generative models of language.
- Their application to building an automatically-trained email spam filter

### **Part of Speech Tagging and Sequence Labeling: [4 Lectures]**

- Lexical syntax.
- Hidden Markov Models (Forward and Viterbi algorithms and EM training).

### **Text Retrieval [8 Lectures]**

- Text Retrieval, Text Retrieval vs. Database Retrieval
- Document Selection vs. Document Ranking
- Common Form of a Retrieval Function
- Vector Space Retrieval Models

### **Word Association Mining [4 lectures]**

- General idea of word association mining
- Discovery of paradigmatic relations
- Discovery of Syntagmatic Relations
- Evaluation of Word Association Mining

### **Topic Analysis [8 lectures]**

- Topics as Terms
- Topics as Word Distributions
- Mining One Topic from Text
- Probabilistic Latent Semantic Analysis
- Extension of PLSA and Latent Dirichlet Allocation
- Evaluating Topic Analysis
- Summary of Topic Models

### **LSTM Recurrent Neural Networks Introduction [2 Lectures]**

- Any basic introduction to perceptron and back propagation
- Word Embedding
- Understanding LSTM network

## Reference Books:

1. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
2. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
3. Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining ChengXiangZhai, Sean Massung, ACM books

## IT-754: HUMAN COMPUTER INTERACTION – CHOICE 5

[40 Lectures]

**Course Objective:** Acquire the knowledge and skills needed to create highly usable software systems and should develop understanding in the key areas of Human Perception, Ergonomics, Cognition and Psychology, Task Analysis, User Interface Design, Interface Programming, System Evaluation.

### Introduction to HCI: [3 Lectures]

- Why Design for Usability?
- Historical Perspective: machinery, computers, PCs and GUIs, networks, mobile, Possible Futures

### Foundations: [4 Lectures]

- Human Perception, Information Presentation and Layout
- Information Visualization

### The Human Body and Device Design: [3 Lectures]

Input Devices and Ergonomics, Virtual Reality

### Low-Level Human Cognition: [5 Lectures]

- GOMS Keystroke-Level Modelling
- Time-scales and the Illusion of Multi-Tasking,
- Hypothesis Testing and Statistical Significance.

### Higher Cognition: [8 Lectures]

- Metaphor, Direct Manipulation
- Widget Survey, Command Languages,
- Other Interaction Styles, Choosing Among Interaction Styles, Command Languages
- Other Interaction Styles, Choosing Among Interaction Styles

### Usability Analysis: [8 Lectures]

- Error Handling, Error Prevention
- Cognitive Walkthroughs,

- Heuristic Evaluation,
- Usability Guidelines, Choosing Among Usability Methods.

### **Specifying and Prototyping: [5 Lectures]**

- Low-Fidelity Prototyping
- Transition Diagrams,
- Task Analysis, User-Centered Design

### **Interface Implementation: 4 Lectures]**

Events and Handlers, Development Tools, Responsiveness Issues

### **Reference Books:**

Designing the User Interface, 6th Edition. Ben Shneiderman, Catherine Plaisant, et al., Addison Wesley, 2017.

## **IT-755: ETHICS AND VALUES FOR ENGINEERS [40 Lectures]**

### **Course Objectives:**

- ❖ To inculcate Ethics and Human Values into the young minds.
- ❖ To develop moral responsibility and mould them as best professionals.
- ❖ To create ethical vision and achieve harmony in life.

**Learning outcome:** By the end of the course student should be able to understand the importance of ethics and values in life and society.

### **UNIT – I (6 Lectures)**

#### **1. Introduction (6 Periods)**

- ❖ Why study Values and Ethics?
- ❖ Definition of Human Values
- ❖ Sources/Formulation of Values
- ❖ Definition of Ethics
- ❖ Sources/Formulation of Ethics
- ❖ Nature and Scope of Ethics
- ❖ Uses of Ethics

### **UNIT-II (4 Lectures)**

#### **❖ Types/Classification of Values**

- Nature of Values: Value Spectrum of a Good Life
- Psychological Values: Integrated Personality, Mental Health
- Societal Values:, Values in Indian Constitution
- Aesthetic Values: Perception and enjoyment of beauty, simplicity, clarity

- Moral and ethical values: Nature of moral judgements, Canons of Ethics, Ethics of Virtue, Ethics of Duty, Ethics of Responsibility

#### **UNIT –III (5 Lectures)**

- ❖ **Branches of Ethics**
  - Descriptive Ethics
  - MetaEthics
  - Normative Ethics
  - Applied Ethics
- ❖ **Introduction to Ethical Theories:**
  - Utilitarianism
  - Justice
  - Rights
  - Egoism

#### **UNIT –IV (15 Lectures)**

- ❖ **Professional Ethics**
  - Defining a Profession
  - Importance of Values and Ethics in profession
- ❖ **Ethics in Engineering Profession:** Nature of Engineering Ethics, Profession and Professionalism, Professional
  - Introduction: Engineering Profession
  - Ethical issues in engineering practice
  - Codes of professional ethics
  - Conflicts between business demands and professional ideals
  - Social and ethical responsibilities of Technologists
  - Ethics, Code of Ethics, Sample Codes – IEEE, ASCE, ASME and CSI.
- ❖ **Whistle Blowing**
  - Types
  - Is it justified: Criteria justifiable for Whistle Blowing
  - Whistleblower and his plight
- ❖ **Engineers' Responsibility for Safety: Case studies**

#### **UNIT V (10 Lectures)**

- ❖ **Global Issues**
  - Current scenario: Technology Revolution
  - Corporate Social Responsibility: Definition and Concept, Case Studies
  - Environmental Ethics: Sustainable Development, Pollution

#### **Reference Books:**

1. Ethics and Values for Engineers and Managers, Subrata Chattopadhyay & Saumaya Singh, Himalaya Publishing House
2. Professional Ethics, R. Subramanian, Oxford Univ Press
3. A Textbook on Professional Ethics and Human Values, R.S. Naagarazan, New Age International (P) Limited Publishers ( PDF available)



### Syllabus for 8<sup>th</sup> Semester B. Tech in Information Technology

Course/L-T-P	Subject	Credits	Comments
IT-801 (3-1-0)	To be determined	4	Departmental
IT-852 (3-1-0)	Elective-3	3	Departmental
IT-853 (3-1-0)	Elective-4	3	Departmental
IT-884	Grand Viva	8	Seminar and Viva-voce
IT-895	Project	12	Project
<b>Total Credits for Semester</b>		<b>30</b>	