<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HF-201</td>
<td>Humanities and Social Sciences</td>
<td>(X-X-X) 2</td>
<td>Please refer to Department of Humanities and Social Sciences.</td>
</tr>
<tr>
<td>CP-223</td>
<td>Data Structures</td>
<td>(3-0-0) 3</td>
<td>Introduction to Data structures. Arrays: Representation – row-major, column-major, sparse matrix – implementation, addition, multiplication; polynomial – Representation, addition, evaluation and multiplication. Strings: Representation, operations, string matching - Brute force or naïve, Robin-Karp, Knuth-Morris-Pratt. Linked List: Static and dynamic implementation, single, double, circular, multiple linked list. Stack: Static and dynamic implementation, expression evaluation, prefix (polish), infix, postfix (inverse polish) expressions, application, multiple stacks, recursion. Queues: Static and dynamic implementation, applications, circular queue, multiple queue. Tree: Binary tree, binary search tree, static and dynamic implementation, tree operations - insertion,</td>
</tr>
</tbody>
</table>
deletion and search, tree traversal, Binary heaps. Introduction to AVL trees and B trees.
Sorting: Insertion sort, selection sort, Bubble sort, quick sort, merge sort, heap-sort, radix sort (bucket sort).
Searching: Linear and binary search, hashing.
Graph: Representation of graphs, BFS, DFS, topological sort.

Text/References:
2. Kruse R.L., Data Structure and Program Design, PHI.
3. Horowitz and Sahni: Data Structure in C++, Glagotia
4. Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structures
5. Aaron M. Tenenbaum, Y. Langsam, Moshe J. Augenstein, Data Structures Using C
7. Sartaj Sahni, Data Structures, Algorithms, and Applications in C++

CP-225 Probability and Statistics (3-0-0)  3

Discrete Distributions: Probability distribution functions and cumulative distribution functions
Mean and variance; moment-generating functions. Marginal and conditional probability distributions. Some specific discrete distributions
Continuous Distributions: Probability density functions and cumulative distribution functions. Mean and variance; moment generating functions. Marginal and conditional probability distributions. Some specific continuous distributions
Functions of Random Variables: Distribution function technique, Transformation technique. Moment-generating function techniques

Text/References:

CP-227 Abstract Algebra (3-0-0)  3

Group Theory: Groups, Semi groups and Monoids, Cyclic semi graphs and sub monoids, Subgroups and cosets, Congruence relations on Semi groups, Factor groups and homomorphisms, Morphisms Normal sub groups. Structure of cyclic groups, Permutation groups, dihedral groups, Sylow theorems, abelian groups; solvable groups, Nilpotent groups; groups of small order, elementary applications in coding theory.
Rings: Rings, Subrings, Morphism of rings, ideal and quotient rings, Euclidean domains, Commutative rings; integral domains, noncommutative examples, Structure of Noncommutative Rings, Ideal Theory
of Commutative Rings

Field Theory: Integral domains and Fields, polynomial representation of binary number, Galois fields, primitive roots, discrete logarithms, split search algorithm.

Modules: Sums and products; chain conditions, Composition series; tensor products.

Text/References:

---

**CP-251 LSD Lab**

The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with CP-221.

1. Truth table verification – NAND gate, NOR gate, OR gate, AND gate, NOT gate.
2. Verifying if NAND gate is a universal gate.
3. Constructing XOR gate using NOR gate only.
4. Realizing given truth table using SOP form.
5. Realizing given truth table using POS form.
7. Design of binary-gray encoder.
8. Design of parity generator and detector.

Text/References:
1. Digital Systems and Hardware and Firmware Algorithms: M. Ercegovac and T. Lang, Pearson.
5. Malvino leech: Digital Electronics

---

**CP-253 Data Structure Lab**

The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with CP-223.

Programs in C or C++ for following:
2. Searching programs: Linear Search, Binary Search.
3. Array implementation of Stack, Queue, Circular Queue, Linked List.
4. Implementation of Stack, Queue, Circular Queue, dynamic memory allocation.
5. Infix to postfix (prefix) conversion.
6. Program for expression evaluation.
8. Program for graph traversal (BFS, DFS).
9. Program for minimum cost spanning tree, shortest path.

Text/References:
2. Kruse R.L., Data Structure and Program Design, PHI.
3. Horowitz and Sahni: *Data Structure in C++*, Glagotia
4. Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structures
5. Aaron M. Tenenbaum, Y. Langsam, Moshe J. Augenstein, Data Structures Using C
   Computation)
7. Sartaj Sahni, Data Structures, Algorithms, and Applications in C++

Open Elective I

Please refer to concerned Department.

B.Tech (Computer Engineering) Semester IV

CP-222 Data Base Management System (3-0-0) 3

Need, purpose and goal of DBMS, Three tier architecture, ER Diagram, data models- Relational,
Network, Hierarchical and Object Oriented.
Data Base Design: Conceptual data base design, Theory of Normalization
Primitive and Composite data types, concept of physical and logical databases, data abstraction and data
independence, data aggregation, Relational Calculus.
SQL : DDL and DML, Relational Algebra.
Application Development using SQL : Host Language interface, embedded SQL programming, Stored
procedures and triggers and views, Constraints assertions.
Internal of RDBMS : Physical data organisation in sequential, indexed random and hashed files. Inverted
and multilist structures, B trees, B+ trees, Query Optimisation, Join Algorithm, Statistics and Cost Base
optimisation.
Transaction Processing, concurrency control, and recovery management. Transaction model properties
and state serialisability. Lock base protocols, two phase locking.

Text/References:
3. C.J. Date: Data Base Design, Addison Wesley
4. Hansen and Hansen : DBM and Design, PHI

CP-224 Microprocessor and Microcontrollers (3-0-0) 3

Introduction to 8085 microprocessor: CPU Architecture, CPU Specifications, CPU Pin Description,
System Timing Diagrams, Instructions, Interrupts etc.
Introduction to 8086/88 microprocessor: CPU Architecture, 8086 CPU Specifications, CPU Pin
Description, System Timing Diagrams, Bus Standards, 8086 Address & Data Buses, Segmentation and
Paging, Addressing Modes, Accessing Memory, RAM & Direct Memory Access, Memory Mapped I/O,
Processor Registers, Data Organization.
Software Architecture: Introduction to Assembly Language Programming, Instruction and timing:
instruction classification, instruction formats, addressing modes, instruction timings and status,
interrupts.
I/O System Design: 8255 Programmable Peripheral Interface, 8259 Programmable Interrupt Controller,
Direct Memory Access: basic concepts of DMA techniques, Description and interfacing of DMA
controller 8257. Introduction to Microcontrollers.

Text/References:
1. Douglas V. Hall : Microprocessors and Interfacing, McGraw Hill
2. Gaonkar ; 8085 Architecture, Programming and interfaces, Penram Press

---

**CP-226 Computer Architecture**

<table>
<thead>
<tr>
<th><strong>Course Code</strong></th>
<th><strong>Course Title</strong></th>
<th><strong>Credits</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-226</td>
<td>Computer Architecture and Organization</td>
<td>(3-0-0) 3</td>
</tr>
</tbody>
</table>

**Introduction to computer architecture and organization:** Digital components, Von Neumann Machine Architecture, Flynn Classification

**Register Transfer Language:** Micro operations - data transfer operations, arithmetic, logic and shift micro operations and their hardware implementations as a simple Arithmetic and logic unit.

**CPU Organization:** Addressing techniques - Immediate, direct, indirect, register, register indirect, index, relative and stack addressing techniques.

**Instruction formats:** Instruction set design, Instruction types: example for zero address, one address, two address and three address machines, Stack, accumulator and general purpose register organization

**Arithmetic Algorithms:** Arithmetic and Logic Unit, Adders - Full adder, Ripple carry adder, Carry look ahead adder, Carry select adder, carry save adder, Multiplication - Add and Shift method, Booth's Multiplier, m -Array Multiplier, Division - Restoring and Non restoring method.

**Pipelining:** Pipeline structure, pipeline performance measures, Pipeline types - Instruction and Arithmetic pipelines.

**Memory Organization:** Memory device characteristics, RAM organization: 1D and 2D organization, Virtual memory - Paging and Segmentation, High speed memories: Associative and Cache memory Control Unit Design, Hardwired and Micro programmed control unit design implementation techniques. Memory hierarchies.

**Input-Output Design:** IO interface, Bus structure, Modes of data transfer, Interrupts, Input Output Processor, Serial Communication

**Text/References:**

---

**CP-228 Theory of Computation**

<table>
<thead>
<tr>
<th><strong>Course Code</strong></th>
<th><strong>Course Title</strong></th>
<th><strong>Credits</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-228</td>
<td>Theory of Computation</td>
<td>(3-1-0) 4</td>
</tr>
</tbody>
</table>

Introduction to automata theory, formal languages, recursive definitions, regular expressions, finite automata, transition graphs and Kleen’s theorem.

Non-determination, finite automata with output, regular languages, minimization of finite automata, pumping lemma for regular languages.

Chomsky classification of languages, regular grammars, context free grammars, simplification of context free grammars, Normal forms of context free grammars.

Push Down Automata Theory: push down automata and languages, push down automata and context free grammars, pumping lemma for context free languages.

Turing hypothesis, Turing machine, Minskey’s theorem, TM variation and encoding, Post machines, computability and acceptability.

Elements of prepositional logic and predicate calculus.

**Text/References:**
1. Hopcroft, Motwani and Ullman: Introduction to Automata Theory, Languages and Computation, Pearson Education.
3. Martin: Introduction to Languages and Theory of Computation, TMH.
The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with CP-222.

1. Conceptual designs using ER diagrams.
2. Design of databases. Based on templates, files and relational basis.
3. Development and implementation of DB system from the fundamentals.
4. Experiments on SQL queries.

The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with CP-224.

Programming the 8085: 8085 instruction set, data transfer instruction, arithmetic, logic & branch operations. Rotate and compare. Instruction related to stack operations. Programming Techniques: looping, counting and indexing, counters and time delays, subroutines. Interfacing microprocessors with the additional devices like ADC/DAC and plotting them onto CRO.

Programming the 8086: Instruction types: Data Transfer, String, Arithmetic, Logical, Bit Manipulation, Program Transfer and Processor Control, the Processor Flags. Interfacing microprocessors with the additional devices like ADC/DAC and plotting them onto CRO. Stepper motor controller and PPI usages.

Text/References:
1. Gaonkar ; 8085 Architecture, Programming and interfaces, Penram Press

Please refer to concerned Department.
linkers and relocating loaders.

**Case study:** UNIX/LINUX, Windows.

**Text/References:**

5. Harvey M. Deitel, An introduction to operating systems. Addison-Wesley.

**CP-323 Compiler Design**

| Translators: Introduction to compilers, translators, and interpreters, compilation process. |
| Lexical Analysis: Finite automata, Regular expressions, Design & implementation of lexical analysers. |
| Syntax directed translation: Syntax directed translation, Attributes, Intermediate codes, Three address codes. |
| Symbol table organization: Hashing, linked list, tree structures. |
| Memory allocation: Static and dynamic structure allocation. |
| Code generation: Compilation of expressions and control structures. Error detection and recovery. |

**Text/References:**

2. Tremblay, Sorenson: The Theory and Practice of Compiler Writing, BSP.
3. Holub, Compiler Design in C, PHI.

**CP-325 Computer Network**

| 1. Computer network architecture, Physical layer: Hardware, topology, data encoding, |
| 2. Data Link Layer: Logical link Control: Error detection and Correction, ARQ protocols, Framing protocols( HDLC, LLC) Medium Access Control: Multiple access protocols, Channel Allocation, contention, reservation, round robin with Examples. |
| 3. Network Inter connection: Generic switches, switch design issues, switching mechanism : virtual Circuit switching, datagram switching, source route switching, Bridge and bridge learning , Global Addressing scheme, fragmentation and reassembly, Address translation: ARP, RARP, ICMP, IP Scalability Issues, sub netting and super netting (CIDR) . IP Routing, EGP, BGP protocols |
| 4. End to End protocols: End to end issues, UDP and TCP segment formats, connection establishment and termination, state transition sliding window protocol, TCP Flow control, Silly window syndrome, TCP retransmission, RTT Estimation, TCP Congestion Control and congestion avoidance protocols |
| 5. Internet applications : Client server paradigm, DNS, SMTP, RPC, NFS and General network security issues. |

**Text/References:**

1. Data Networks: Bertsekas and Gallagher, Phi.
4. Computer Networks and Internet: D.E. Comer, Pearson
The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with CP-321.

1. Simple Unix-C programs: Programs using system calls, library function calls to display and write strings.
2. Concurrent Programming concepts using fork, semaphore and pipes.
3. Programs for error reporting using errno, perror() function.
4. Programs to simulate process scheduling like FCFS, shortest Jobs First and Round Robin.
5. Programs to simulate page replacement algorithms like FIFO, Optimal and LRU.
6. Programs to simulate Free space management.
7. Programs to simulate virtual memory.
8. Programs to simulate deadlock detection.
9. Any other as per curriculum.

Text/References:
1. Unix concepts and applicaions by Sumitbha Das, TMH applications.
2. Unix Programming by stevens, Pearsons Education.
4. Operating System concepts by silberschatz, and Peter Galvin.

The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations.

Programming exercises to implement typical lexical analyzers, parsers, intermediate code generation. Assignments using LEX and YACC tools.
Programming exercises to implement assemblers, editors, debuggers etc.

Text/References:
1. Aho, Ullman and Sethi: Compilers, Pearson Education.
2. Levine, Mason and Brown: Lex and Yacc, O'Reilly.

The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with CP-325.

1. Programming for data encoding, CRC detection and Correction.
2. Estimation of network delay through OS utilities.
4. Packet measurement and observation using network sniffing tools.
5. Use of sniffers for protocol dynamics.
6. Introduction to Socket programming and application development for internet.

Text/References:
1. Computer Networks and Internet: D.E. Comer, Pearson

One/More of the following courses as offered by Department.
CP-371 Object Oriented Programming
CP-373 Logical and Functional Programming
CP-375 Programming in Java
CP-377 Concurrent Programming

Open Elective III (X-X-X)  3/4
Please refer to concerned Department.

B.Tech (Computer Engineering)  Semester VI

CP-322 Software Engg. (3-0-0)  3

System analysis: Feasibility study requirement analysis, Cost benefit analysis, Planning systems, Analysis tools and techniques.
System Design: design fundamentals, Modular Design, Data and procedural design, object oriented design.
Software Maintenance: Maintenance Characteristics, Maintainability, Maintenance tasks and side effects.

Text/References:
2. Sommerville I: Software Engineering, Addison Wesley

CP-324 Design and Analysis of Algorithms (3-0-0)  3

Algorithm Analysis: Asymptotic notation, solution of recurrence, model of computation, time and space complexities, average and worst case analysis, Amortized analysis.
Algorithm Design Techniques: Greedy algorithm, dynamic programming, divide and conquer, backtracking, branch and bound.
Graph Algorithms: Shortest path algorithms, Disjoint set operations, minimum spanning tree algorithm, network flow, matching, coverings, applications of DFS:- bi-connectivity, Euler circuits, strongly connected components, topological sort, and articulation point.
Dynamic Programming: Chained matrix multiplication, longest common subsequence.
Divide and Conquer: Order Statistics – finding the median, exponentiation, matrix multiplication, LCS.
Computational Geometry: Line segments, Optimal polygon triangulation.
Approximate Algorithm: Travelling Salesman Problem, vertex-cover problem.
Primality testing, Integer factorization, Randomized algorithms, Probabilistic algorithms.
String Matching algorithms: Rabin Karp, KMP, Boyer Moore.
Matrix Algorithms – Strassen Matrix multiplication, LUP decomposition.
Construction of codes: Shannon Fano and Huffman codes.
Introduction to problem classes – NP, NPC, NP-Hard.

Text/References:
4. Brassard: *Fundamental of Algorithmics*, PHI.
5. W.W. Peterson and E. J. Weldon: *Error correcting codes*.

**CP-352 Algorithms Lab**

The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations.

1. Implementation of graph algorithms – DFS, Shortest Path, MST, articulation point, topological sorting.
2. Implementation of Network Flow, matching, covering.
3. Implementation of pattern matching algorithms.
4. Determination of $k^{th}$ shortest number in a given sequence.
5. Dynamic programming – Applications to problem solving.
6. Approximation algorithms for NP problems.
7. Randomized algorithms implementation.

**Text/References:**

4. Brassard: *Fundamental of Algorithmics*, PHI.
5. W.W. Peterson and E. J. Weldon: *Error correcting codes*.

**CP-354 Seminar**

The topics selection covering the latest and relevant topics related to the emerging areas. Ideally, some recent reputed journal papers abstraction and presentation shall be encouraged for presentation. The evaluation shall be continuous and through components evaluation viz. content, coverage, depth, presentation, response to the queries, and seminar report. In case of unsatisfactory performance, an X grade can be awarded for extension work during summer term.

**Program Elective II**

One/More of the following courses as offered by Department.

- CP-372 Digital Signal Processing
- CP-374 Wireless Communications
- CP-376 VHDL
- CP-378 Neural Networks

**Open Elective IV**

(X-X-X) 3/4

Please refer to concerned Department.

**Open Elective V**

(X-X-X) 3/4

Please refer to concerned Department.
CP-451 Industrial Training (0-2-0) 2
The evaluation shall be continuous and through components evaluation viz. content, coverage, depth, presentation, demonstration, response to the queries, and training report. In case of unsatisfactory performance, and failure extra credit course from the department equivalent to CP-451 can be permitted through consent of DUGC.

CP-453 Free and Open Source Lab (0-0-3) 2
The following proposed coverage are broad guiding areas lab. The instructor offering the course in consultation with the theory offered can adopt further variations.
1. Linux basics and installation and management of the Linux.
2. Different types of software development environment (Eclipse)
3. make and other software construction utilities on Linux.
4. Version control and managing project in open source.
5. Managing large software development through wiki or alike project management tools.
6. Introduction to scripting for system management.

Program Elective III (3-0-2) 4
One/More of the following courses as offered by Department.
CP-471 Computer Graphics
CP-473 Speech Recognition
CP-475 Mobile Computing
CP-477 Software Project Management
CP-479 Data Compression

Program Elective IV (3-0-2) 4
One/More of the following courses as offered by Department.
CP-481 Advances in Compiler Design
CP-483 Natural Language Processing
CP-485 Programming Paradigms
CP-487 Wireless and Ad hoc Networks
CP-489 Real Time Systems

Program Elective V (3-0-2) 4
One/More of the following courses as offered by Department.
CP-491 Cryptography
CP-493 Embedded Systems
CP-495 Data Mining
CP-497 VLSI Algorithms
CP-455 Project Lab (0-2-3) 5
Objective of this elective is to facilitate transfer of knowledge acquired by a student to a field of his own choice for application to solving a problem. Student is expected to collect and study relevant material under mentorship of a faculty member working in similar area; identify a suitable problem and propose methodology towards its solution. Alternately a student can explore hardware implementation of existing solution(s). This elective shall act as prequel to project work for next semester. The project coordinator(s) from the department for continuity shall coordinate this course. Grouping and division shall be applicable as defined in the major project of final semester.

Open Elective VI (X-X-X) 3/4
Please refer to concerned Department.

B.Tech (Computer Engineering) Semester VIII

CP-452 Major Project (0-10-0) 10
The major project covers lab component of the final semester work. The evaluation of project shall be continuous and will be done through project coordinator(s). The evaluation mechanism shall be evolved based on the existing practices through DUGC rectified from time to time. Ideally the project should comprise with group size of two students shall be limited to maximum 4 students and the groups shall be evenly distributed among faculty through coordinator(s). Internal and external components shall not exceed 40% each of the overall marks.

CP-454 Group Discussions (0-0-3) 2
This lab will remain only if Industrial training is given weightage of 02 credits.

Program Elective VI (3-0-2) 4
One/More of the following courses as offered by Department.
CP-470 Network Security
CP-472 Biometrics
CP-474 High Level Synthesis of Digital Systems
CP-476 Selected Topics in Cryptography
CP-478 Digital Image Processing

Program Elective VII (3-0-2) 4
One/More of the following courses as offered by Department.
CP-480 Evolving Architectures
CP-482 Topics in Advanced Computing
CP-484 Software Testing and Validation
CP-486 Machine Learning
CP-488 Modeling and Simulation
Please refer to concerned Department.

Program Elective I  (Semester V)

CP-371 Object Oriented Programming  
(3-0-2)  4

**Object Oriented Programming and Design:** Review of abstraction, objects and other basics, Encapsulation, Information hiding, method, Signature, Classes and Instances, Polymorphism and inheritance.

**C++ Programming Basics:** Fundamentals, variables and assignments, Input and Output, Data types and expressions, flow of control, subprograms, top-down design, predefined functions, user defined functions, procedural abstractions, local variables, overloading function names, operator overloading, parameter passing, this pointer, destructors, copy constructor, overloading the assignment operator, virtual functions, function calling functions, friend functions, recursive functions, recursive member functions. Static member function.

**C++ Object oriented concepts:** Objects and classes, use of file for I/O, formatting output with stream functions, Character I/O, inheritance, structures for diverse data, structures as function arguments, initializing structures, defining classes and member functions, public and private members, constructors for initialization, standard C++ classes, derived classes, flow of control, use of Boolean expressions, multiway branches, use and design of loops. Friend function and friend class.

**C++ Data structures and Advanced Topics:** Arrays – programming with arrays, arrays of classes, arrays as function arguments, strings, Multidimensional arrays, Arrays of strings, Pointers Dynamic arrays, Classes and dynamic arrays, Base classes, access control, Templates- generic classes and functions, namespaces. Standard Template Library.

**Text/References:**
2. Robert Lafore: C++ Programming
3. Ashok N. Kamthane: Object Oriented with C++, Pearson Education

CP-373 Logical and Functional Programming  
(3-0-2)  4

**Introduction:** Logic programming, Prolog - Lists, cut operator, and sorting, Data structures, text strings, operators - extensions of SWI Prolog, Searching state space, clause management, and parsing in Prolog

**Introduction to functional programming:** Lambda calculus, Programming language Haskell - introduction, lists, User-defined data types, type classes, and arrays in Haskell. Input/Output in Haskell - type classes IO and Monad, Simple applications/programs in Haskell.

**Text/References:**

CP-375 Programming in Java  
(3-0-2)  4

**Introduction:** Internet, Java as a tool for internet applications, Byte Code and its advantages.

**Object Oriented Programming and Design:** Review of Abstraction, Objects and other basics, Encapsulation, Information hiding, Method, Signature, Classes and Instances, Polymorphism, Inheritance, Exceptions and Exception Handling with reference to object modeling, Coupling and Cohesion in object oriented software. Object Oriented Design – Process, Exploration and Analysis.

**Java Programming Basics:** Fundamentals: Variables and assignments, Input and Output, Data Types and Expressions, Flow of control, Local variables, Overloading Parameter passing, this pointer.

**Java Object Oriented Concepts:** Objects and Classes: Use of file for I/O, Formatting output with
stream functions, Character I/O, Inheritance, Public and private members, Constructors for initializations, Derived classes, Flow of Control

**Java Data Structures and Advanced Topics**

Arrays – Programming with arrays, arrays of classes, arrays as function arguments, Strings, Multidimensional arrays, Arrays of strings, vectors, Base classes.

Introduction to Java Applets

**Text/References:**

1. Herbert Schildt: JAVA 2 - The Complete Reference, TMH, Delhi

---

**CP-377 Concurrent Programming**

(3-0-2) 4


**Text/References:**


---

**Program Elective II (Semester VI)**

---

**CP-372 Digital Signal Processing**

(3-0-2) 4

Introduction to Continuous time Systems, idea about Linear Time Invariant System LTI systems. Fourier Transforms.

Discrete Time Systems: Sampling and aliasing, LTIs, Representation of Sequences by Fourier Transform and properties of Fourier Transform.

Z-Transform, Structures for discrete system, DFT, Computation of DFT. FIR Filters, frequency response of FIR filters. IIR Filters, spectrum analysis. FFT Algorithms

**Text/References:**

1. *Discrete Time Signal Processing* by Alan V Oppenheim, Ronald W Schafer.- PHI
3. *Signal and Systems*: S.Haykin and Veen, Wiley

---

**CP-374 Wireless Communications**

(3-0-2) 4


OFDM principles: Comparison of OFDM and CDMA. WLAN and Bluetooth.

Text/References:
2. K. Pahlavan & P. Krishnamurthy, Principles of Wireless Networks, Prentice Hall:
4. 

CP-376 VHDL (3-0-2) 4

1. Overview of VHDL, fundamentals of VHDL, Lexical elements Data types and objects
2. Data Flow style: Conditional and selected Concurrent assignment, block assignment If and wait statement, Design for synthesizability
3. Structural style: Instantiation and component declaration, statement configuration declaration, generate statement, examples of structural design
4. Behavioural Style : Signal assignment, statement like case, process and wait loop, exit etc., concurrent signal assignment statements, function and procedures, file I/O operations and Testbenches.

Text/References:
2. SS Limaye," Digital Design with VHDL", CMR
3. Douglas Parry, “ VHDL Programming by Example”, MGH
5. Xilinx,” A CPLD VHDL Introduction Application Notes"

CP-378 Neural Networks (3-0-2) 4

Neural Architecture: Neuron model, transfer function, hamming and Hopfield network, perceptron, learning rule, recurrent networks.

Back propagation: generalized delta rule, limitations, modifications – momentum, variable learning rate, conjugate gradient.

Learning: Supervised, associative, competitive, unsupervised learning.


Neural network applications: Pattern classification, function approximation.

Text/ References:
2. Christopher M. Bishop: Neural Networks for Pattern Recognition
### Program Elective III  (Semester VII)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-471</td>
<td>Computer Graphics</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>CP-473</td>
<td>Speech Recognition</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

#### CP-471 Computer Graphics
**Introduction to computer graphics:** Vector and Raster graphics, Graphic primitives and attributes. Computer graphics devices - CRT, plasma, LCD, plotters, Scan conversion of line, circle and ellipse.

**Viewing Transformations:** Coordinate system - world, device and normalized device coordinates, Window and Viewport and viewing transformations

**Filling and Clipping:** Flood fill and seed fill algorithms and scan line polygon filling algorithms, Cohen Sutherland clipping algorithms for Polygon

**Geometric transformations:** - 2D and 3D transformations: Translation, Scaling, rotation, Shearing, reflection etc., Transformations about an arbitrary axis

**Projections:** Parallel - Orthographic, Plans and Elevations, Axonometric - Isometric, Diometric, trimetric, Perspective - One point, two point, three point.

**Hidden surface removal:** Object space and image space algorithms, Back space removal, Z-buffer, scan line, area subdivision, painters, BSP tree, Floating horizon and ray tracing methods

**Curves:** Spline representations, Curve representation techniques, Continuity constraints, Hermite Interpolation, Bezier curves, B-Spline curves

**Text/References:**

#### CP-473 Speech Recognition
**Overview of Speech Recognition:** What is Speech; Why is it important; Applications and issues. Speech Production; Mechanism of speech production; Categories of sounds; Sound units in indian languages. Nature of Speech Signal; Source-system characteristics; Segmental and suprasegmental features; Temporal and spectral parameters for sound units in indian languages. Basics of Digital Signal Processing; Signals and systems; Discrete fourier transform; Digital filtering; Stochastic processes. Speech Signal Processing Methods: Short-time spectrum analysis; Spectrograms; Linear prediction analysis; Cepstrum analysis. Speech Recognition; Isolated word recognition; Connected word recognition Continuous Speech Recognition; Speech recognition problem; Hidden markov models. Other Applications: Word spotting; Speaker recognition; Speech enhancement; Speech synthesis; Practical issues in speech Recognition.

**Text/References:**
### CP-475 Mobile Computing (3-0-2) 4

Introduction to mobile computing: principles, classification & overview of devices, operating systems.
Wireless transmission: brief overview, multipath propagation, hidden & exposed terminals. Medium access control & protocols: SDMA, FDMA, TDMA, DAMA, FAMA, PRMA, Reservation TDMA, polling, CSMA/CA, CDMA etc.
Wireless LAN: infrastructure & ad-hoc networks, IEEE 802.11, HIPERLAN.
Mobile network layer: mobile IP, DHCP, infrastructure & Ad-hoc routing. Mobile transport layer: indirect TCP, snooping TCP, mobile TCP etc. mobile support, WWW & mobility, WAP.

**Text/References:**
1. Principles of mobile computing Hansmann & Merk., Springer
2. Mobile communications Jochen Schiller, Pearson
3. 802.11 wireless networks Matthew S.Gast, O’REILLY.
5. Mobile Communications Handbook by Jerry D. Gibson
6. Mobile Communications Handbook by Raymond Steel

### CP-477 Software Project Management (3-0-2) 4

Software Project Planning: Objectives, Decomposition Techniques and Empirical Estimation Model.
Project Scheduling & Tracking, Software Quality Assurance, Software Configuration Management.

**Text/References:**
1. R. S. Pressman, Software Engineering
2. P. Jalote, Software Project Management in Practice.
3. B. Hughest & M. Cotterell, Software Project Management.

### CP-479 Data Compression (3-0-2) 4

**Compression:** Need, Lossless v/s lossy compression, review of information theory, prefix codes, uniquely decodable code.

**Lossless Compression:** Huffman coding – minimum variance, optimal, non-binary, extended, adaptive. Applications and limitations of Huffman codes, Run length encoding, Arithmetic coding, Predictive coding – Burrows-Wheeler transform, Delta modulation, Adaptive delta modulation

**Dictionary based compression** - Lempel-Ziv-Welch, LZ77 and LZ-78

**Lossy Compression Techniques** – JPEG and its application

**Error detection and correction:** Parity, 1,2,n dimensions, Hamming codes, p-out-of-q codes

**Quantization:** Scalar and Vector Quantization.

**Texts/References:**
1. Khalid Sayood, Introduction to Data Compression, Morgan Kauffman
Program Elective IV  (Semester VII)

CP-481 Advances in Compiler Design  
(3-0-2)  4

Text/References:
2. Steven Muchnick, Advanced Compiler Design & Implementation, Morgan Kaufmann.
3. Keith Cooper and Linda Torczon, Engineering a Compiler, Morgan Kaufmann.

CP-483 Natural Language Processing  
(3-0-2)  4

Text/References:
2. Readings in Natural Language Processing, Morgan Kaufmann, 1986. Winograd T.

CP-485 Programming Paradigms  
(3-0-2)  4
Introduction: History of Programming Languages, Syntax and Semantics, Paradigms.
    Case Study: C language.
Object-Oriented Programming: Objects and Classes, Encapsulation, Information hiding, Method and Signature, Polymorphism and Inheritance.
    Case Study: C++ language.
Aspect-Oriented Programming: Crosscutting concerns, Aspect, Joinpoint and Pointcuts, Advice and static crosscutting.
    Case Study: AspectJ language.
Functional programming: Introduction, lists, User-defined data types, type classes, and arrays, Input/Output and Monad, Simple applications/programs.
    Case Study: Haskell language.

Text/References:
1. Kerninghan and Ritchie: The 'C' programming language, Pearson Education.
CP-487 Wireless & Ad-hoc Networks (3-0-2) 4


Text/References:

CP-489 Real Time Systems (3-0-2) 4


Text/References:

Program Elective V (Semester VII)

CP-491 Cryptography (3-0-2) 4

Review of Number theory: Prime numbers, modular arithmetic, Fermat’s theorem, Euler’s theorem,
Chinese remainder theorem, Discrete logarithms, Random number generation, factoring, prime number generation.

**Cryptography:** Need, conventional techniques, stream ciphers, block cipher, steganography. Public v/s private key cryptography.

**Stream Ciphers:** Caesar Cipher, mono-alphabetic and poly-alphabetic ciphers, Playfair Cipher, Hill Cipher, Rotor machines, One time pad.

**Random Number Generation:** Pseudo Random Number, PRNG, LFSR, Blum-Blum Shub generator

**Private-key cryptography:** Feistel structure, DES (Data encryption standard), design of S-boxes, AES, Triple DES.

**Public key cryptography:** Key management, Key exchange – Diffie-Hellman, El-Gamal, Merkle's Puzzle, Authentication, Signatures, Deniability, RSA.

**Threshold Cryptography:** Sharing Secrets.

**Digital Signature:** DSA and its variants, discrete logarithm based digital signatures.

**One-way hash functions** – MD5, SHA (Secure Hash Algorithm).

**Cryptanalysis:** Differential and linear cryptanalysis - cracking DES.

**Text/References:**


**CP-493 Embedded Systems**

Introduction to embedded systems., design representations, level of abstractions, design methodologies, Models and architectures, Taxonomy of models and architectures, brief descriptions of specification languages, Specification requirement for embedded systems, Spec Chart and Spec Chart Description. Design challenges & issues, hardware and software design, co-design of software and hardware, ASIC. Design quality estimation : Quality matrix, software and hardware estimation.

Introduction
Sample design Specification of Answering machine/ Microcontroller 8051.

**Text/References:**

1. Denial D. Gajski, frank Vahid: Specification and design of embedded systems, PH
3. Myke Predko: Programming and Customizing the 8051 Micro Controller, TMH
4. Ayala: 8051 Micro controllers, Penram Press

**CP-495 Data Mining**

Introduction: Basic Data Mining Tasks, Data Mining Issues, Data Mining Metrics, Data Mining from a Database Perspective.

Data Mining Techniques: A Statistical Perspective on Data Mining, Similarity Measures, Decision Trees, Neural Networks, Genetic Algorithms.

Classification: Statistical-Based Algorithms, Distance-Based Algorithms, Decision Tree-Based Algorithms, Neural Network-Based Algorithms, Rule-Based Algorithms, Combining Techniques.

Clustering: Similarity and Distance Measures, Hierarchical Algorithms, Partitional Algorithms, Clustering Large Databases, Clustering with Categorical Attributes.

Advanced Techniques: Web Mining, Spatial Mining, Temporal Mining.

Text/References:

CP-497 VLSI Algorithms

1. Introduction of VLSI Technology, VLSI design cycle, design styles, basic Layout rules and circuit abstraction, introduction to standard Cell, Gate array, FPGA
2. Overview of basic graph algorithms, Graph algorithms for physical Design
3. Partitioning: Classification of partitioning algorithms, Kernighan-Lin Algorithm, FM Algorithm, Ratio cut algorithm
4. Floor-planning: Rectangular dual graph approach of floor-planning, hierarchical tree based approach, Integer programming based floor-planning.
5. Placement: placement by simulated annealing and force directed method

Text/References:
2. Sarrafzadeh and Wong "An introduction to VLSI Physical design" MGH
3. Sze: VLSI Technology

Program Elective VI (Semester VIII)

CP-470 Network Security


Text/References:
2. Kurose & Ross, Computer Networking, Pearson Education.

CP-472 Biometrics
Biometrics: DNA, fingerprint, Iris, Retinal scan, Face, hand geometry, human gait, speech, ear.
Handwriting, Keystroke dynamics, Signature
Multimodal biometrics: Combining biometrics, scaling issues.
Biometric template security.

Texts/References:
2. Davide Maltoni (Editor), et al, Handbook of Fingerprint Recognition
3. L.C. Jain (Editor) et al, Intelligent Biometric Techniques in Fingerprint and Face Recognition
4. John Chirillo, Scott Blaul, Implementing Biometric Security

CP-474 High Level Synthesis of Digital Systems (3-0-2) 3
Overview. Design methodologies. Abstractions and views.
Review of basic concepts in algorithms and graph theory
Design representation and modeling, Modeling languages, Abstract models
Synthesis at higher levels of abstraction
Scheduling, Resource sharing
Structural synthesis: Module selection. Pipeline. Control
Synthesis at lower levels of abstraction, Logic synthesis

Text/Reference:
4. Recent papers from journals and conferences.

CP-476 Selected Topics in Cryptography (3-0-2) 4
Elliptic Curve Cryptography
Secret Sharing, Threshold cryptography – Robust ElGamal system
Visual Cryptography.
Interactive zero knowledge proofs, witness hiding protocols.
Group encryption, decryption. Group signatures, ring signatures.
E Voting: requirements, issues and challenges, existing solutions, write-in ballots.
Pair based cryptography – Weil and Tate pairing.
Contemporary Issues.

Text/References:
2. Selected paper and online reference material.

CP-478 Digital Image Processing (3-0-2) 4
Digital Image Fundamentals: Image Model, Sampling, Quantization, Neighborhood, connectivity of pixels, Labeling of connected components, Distance measures
Image Transforms: Fourier Transform, Discrete Fourier Transform, Properties of 2D Discrete Fourier Transform, The fast Fourier Transform and its algorithm, number of operations, the inverse FFT.
Discrete Cosine Transform and its applications, KL Transform, Convolution and correlation
Image Enhancement: Enhancement by point processing, spatial filtering, enhancement in frequency domain, generation of spatial masks from frequency domain specifications
Image Segmentation: Detection of discontinuities, edge linking and boundary detection, thresholding, region oriented segmentation
Representation and Description: Representation schemes, boundary descriptors, regional descriptors.
Morphology: Dilation, erosion, opening, closing, Hit-or-Miss Transform, some basic morphological algorithms like pruning, thinning and thickening

Text/References:

Program Elective VII     (Semester VIII)

CP-480 Evolving Architectures     (3-0-2)     4
Special, and emerging advanced topics in different areas of Computer Engineering will be covered under this course.

Text and References:
1. Research reports and papers from journals.

CP-482 Topics in Advanced Computing     (3-0-2)     4

Fault Tolerant Computing: Fault Tolerance Strategies - Fault detection, masking, containment, location, reconfiguration, and recovery. Fault Tolerant Design Techniques - Hardware redundancy, software redundancy, time redundancy, and information redundancy.

Text and References:
6. Research papers and internet resources.

CP-484 Software Testing and Validation     (3-0-2)     4
Basic software testing principles - Software Quality, Software testing, test generation and test management.
Acceptance Testing: User acceptance testing, alpha and beta testing. Verification and Validation, Functional and Non-functional system testing.
Static and dynamic testing, Black-box or functional testing, structural, white box or glass box testing.
Integration testing, component testing.
Software testing tools.
Books/References:
1. Recent papers from conferences and journals.
2. A. P. Mathur, Fundamentals of software testing.

CP-486 Machine Learning (3-0-2) 4
Ensemble Learning Using committees of multiple hypotheses. Bagging, boosting, and DECORATE. Active learning with ensembles.
Instance-Based Learning: Constructing explicit generalizations versus comparing to past specific examples. k-Nearest-neighbor algorithm. Case-based learning.

Text/References:

CP-488 Modelling and Simulation (3-0-2) 4
Analytical v/s simulation modeling, performance measurement and benchmarking, Workload modeling, random variables, commonly used distributions, Stochastic Processes, Performance evaluation methods, Evaluation Metrics'
Markov chains, Birth and Death Processes, Markov chain models of Computer systems, Steady-state and transient analysis
Queuing models, M/M systems and their steady state analysis, Single server and multi-server queues, open and closed queuing networks
Petri Net based Performance Modeling : Classical Petri Nets, Timed Petri Nets, Discrete Petri Nets,
Modeling multiprocessor systems
Discrete event simulation – Simulation languages, random number generation and testing, model verification and validation, analysis of simulation results, confidence intervals, variance reduction techniques, Case studies of analytical and simulation studies of computer systems

Text/References:
1. Law and Kelton, Simulation Modeling and Analysis, Mcgraw Hill
3. K.S.Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, PHI
4. Kant, Introduction to Computer System Performance Evaluation, Mcgraw Hill