Course Structure for **M.Tech in Computer Science and Engineering**

**Objective:**

The proposed M.Tech (CSE) program is intended to fill an existing gap within the industries and academic institution of trained professionals. The prospective students will be well trained in both the theoretical and application aspects of computer science and engineering.

**Eligibility:**

- B.Tech. in C.S.E or equivalent degree with valid GATE score in C.S..
- B.Tech from IITs with CPI > 8.0 (as per the institute rule)
- In addition, candidates sponsored by an industry/R&D organization are also considered. Sponsored candidates must have at least 60% marks in the qualifying examination and at least two years of work experience. GATE score is not mandatory for sponsored candidates.
- Intake capacity: 15 + 5 (sponsored)

**Two Sets of Electives** out of which the electives to be floated in 1st sem from the List-1, while in 2nd sem from the list-2

**Theory Courses:** 8 (4 core and 6 electives); **Lab Courses:** 2

**Total Credits:** 175 2 Years (4 semester programme)
COURSE STRUCTURE

1\textsuperscript{st} Semester

<table>
<thead>
<tr>
<th>Course Id</th>
<th>Course Name</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS541</td>
<td>Foundations of Computer Systems</td>
<td>3-0-0=6</td>
</tr>
<tr>
<td>MA501</td>
<td>Probability, Statistics and Stochastic Process</td>
<td>3-0-0=6</td>
</tr>
<tr>
<td>CS558</td>
<td>Computer Systems Lab-1</td>
<td>0-0-6=6</td>
</tr>
<tr>
<td>CS591</td>
<td>Seminar-I</td>
<td>0-0-4=4</td>
</tr>
<tr>
<td>Elective-I</td>
<td></td>
<td>3-0-0=6</td>
</tr>
<tr>
<td>Elective-II</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elective-III</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total Credit: 40

2\textsuperscript{nd} Semester

<table>
<thead>
<tr>
<th>Course Id</th>
<th>Course Name</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS511</td>
<td>Foundations of Theoretical Computer Science</td>
<td>3-0-0=6</td>
</tr>
<tr>
<td>CS512</td>
<td>Data Structure and algorithms</td>
<td>3-0-0=6</td>
</tr>
<tr>
<td>CS513</td>
<td>Computer Systems Lab-2</td>
<td>0-0-6=6</td>
</tr>
<tr>
<td>CS592</td>
<td>Minor Project / Seminar</td>
<td>0-0-4=4</td>
</tr>
<tr>
<td>Elective-IV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elective-V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elective-VI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total Credit: 40

3\textsuperscript{rd} Semester

<table>
<thead>
<tr>
<th>Course Id</th>
<th>Course Name</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS691</td>
<td>Comprehensive Viva</td>
<td>0-0-0=10</td>
</tr>
<tr>
<td>CS698</td>
<td>Project Thesis -I</td>
<td>0-0-0=40</td>
</tr>
</tbody>
</table>

Total Credit: 50

4\textsuperscript{th} Semester

<table>
<thead>
<tr>
<th>Course Id</th>
<th>Course Name</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS699</td>
<td>Project Thesis - II</td>
<td>0-0-0=45</td>
</tr>
</tbody>
</table>

Total Credit: 45

Total Credits= 40+40+50+45=175
List of Electives

Group-1 (Floated in 1st Sem depending on the instructor's choice. More specialized electives will be added further.)

1. CS561 Artificial Intelligence
2. CS542: Software Testing
3. CS543: Distributed Systems
4. MA511: Large Scale Scientific Computing (Maths)
5. CS 528CAD for VLSI

Group-2 (Likely to be floated in 2nd Sem depending on the instructor's choice. More specialized electives will be added further)

6. CS 548 Wireless Networks
7. CS549 Computer and Network Security
8. CS 508 Formal methods for analysis and verification
9. CS743: Advanced topics on Database
10. CS502: Pattern Recognition
Detailed Syllabus

CS541 Foundations of Computer Systems (3 0 0 6)


Review of concepts of operating systems: Processes, threads, Unix fork-exec model, Unix signals, Interprocess communication, scheduling, memory management.

Review of concepts of computer networks: link layer protocols, local area networks (Ethernet and variants), interconnecting networks with IP, routing, transport layer protocols. Advanced concepts of distributed networked systems: Virtualization, distributed file systems, mass storage systems, recovery and fault tolerance, content networking including multimedia delivery

Texts:

CS511 Foundations of Theoretical Computer Science (3 0 0 6)

Discrete Structures -- Sets, Relations and Functions; Proof Techniques, Algebraic Structures, Morphisms, Posets, Lattices and Boolean Algebras. Logic -- Propositional calculus and Predicate Calculus, Satisfiability and validity, Notions of soundness and completeness. Automata and Languages -- Finite automata and regular expressions, pushdown automata and context-free grammars, pumping lemmas and closure properties of regular and context-free languages, non-context-free languages. Computability theory -- Church-Turing thesis, Hilbert's problem,
decidability, halting problem, reducibility; Complexity theory: time and space complexity, Classes P, NP, NP-complete, PSPACE, and PSPACE-complete.

**Texts:**

**References:**

**CS 512 Data Structure and Algorithms**

**Problem Solving using Computers** - Abstraction - Abstract data types; Data Representation; Elementary data types; Basic concepts of data Structures; Mathematical preliminaries - big-Oh notation; efficiency of algorithms; notion of time and space complexity; performance measures for data structures.

**ADT array** - Computation on arrays - sorting and searching algorithms. ADT Stack, Queue, list - array, linked list, cursor based implementations of linear structures.

**ADT Tree** - tree representation, traversal of trees;

**ADT Binary tree** - binary trees, threaded binary trees, application of binary trees - Huffman coding; application of threaded binary trees - differentiation;

**Search Tree** - Binary search tree; balanced binary search trees - AVL tree; Applications of Search Trees - TRIE; 2-3 tree, 2-3-4 tree; concept of B-Tree.

**ADT Dictionary** - array based and tree based implementations; hashing - definition and application - LZW encoding. ADT Priority Queue - Heaps; heap-based implementations; applications of heaps - sorting;

**Graphs** - shortest path, minimum spanning tree, DFS, BFS - an application of DFS and BFS.

**Algorithm Design Paradigms** - greedy, divide and conquer, dynamic Programming, backtracking.

**References:**
3. Aho, Hopcroft and Ullmann, "Data structures and Algorithm", Addison Welsey,
MA 501 : Probability, Statistics and Stochastic Processes

Probability : algebra of sets, monotone class, sigma fields, Borel sigma fields, set function, product spaces, measurable transformations, probability measure, notions of probability space and some consequences, Borel-Cantelli Lemma Discrete, continuous and mixed type probability spaces, cumulative distribution functions, probability mass (density) functions, mathematical expectations, general concepts of conditional probability and expectation, conditional expectation given a sigma field, properties of conditional expectation, moments, moment and probability generating functions, moment inequalities: Markov, Chebyshev-Bienayme, Lyapunov.


Sampling Distributions: The Central Limit Theorem, Demoivre theorem, uniform convergence in CLT, characteristics functions, continuity theorems, strong law of large numbers, Sequence of random variables, modes of convergence and some results, Slutsky theorem, distributions of the sample mean and the sample variance for a normal population, Chi - Square, t and F distributions and their distributional properties Point and Interval Estimation : The method of moments and the method of maximum likelihood estimation, large sample properties, concepts of unbiasedness, criteria for choosing estimators, consistency and efficiency of estimates, confidence intervals for the parameters of common distributions, pivotal quantities, confidence intervals for proportions (one and two samples problems).

Testing of Hypotheses : simple and composite hypothesis, Null and alternative hypotheses, critical and acceptance regions, two types of error, level and size of test, error probabilities of a test, the most powerful test and Neyman - Pearson Fundamental Lemma, tests for one sample and two sample problems for normal populations, tests for one sample and two sample proportions, Likelihood ratio tests, Chi square test for goodness of fit.

References:

CS 558 Computer Systems Lab – 1

- Basics of OS programming: process creation and synchronization, shared memory and semaphore, shell programming.
- Socket programming, database creation and update, building large client server applications. Basics of compiler writing using lex and yacc

CS 513 Computer Systems Lab-2

- Object-oriented programming concepts and implementation of abstract data types. Implementation of graph algorithms. Linear programming with applications.
Detailed Syllabus for Elective courses - Group-1

CS 561: Artificial Intelligence (3-0-0-6)

Introduction, Problem Solving: Uninformed search, Informed search, local Search, Online search; Knowledge and Reasoning: Building a Knowledge Base, Semantic Nets, Frames, First order logic, Inference in First Order Logic; Probabilistic Reasoning Systems: Bayes’ Nets; Learning: Learning from examples and analogy, Naive Bayes, Computational Learning Theory, Explanation Based Learning, Neural Networks; Evolutionary Optimization: Genetic algorithms, Multi objective optimization, Differential Evolution, Particle Swarm Optimization; Introduction to NLP; Introduction to Fuzzy sets.

References:


CS 542: Software Testing (3 0 0 6)


Books:
2. Lessons Learned in Software Testing by Kaner, Bach and Pettichord
CS 543 Distributed Systems (3 0 0 6)


References:


MA 511 : Large Scale Scientific Computation (3 0 0 6)

Introduction to sparse matrices, Storage Schemes, Permutations and Reorderings, Sparse Direct Solution Methods. Iterative methods and Preconditioning Convergence Krylov Subspaces, Arnoldi’s Method, GMRES, Symmetric Lanczos Algorithm, conjugate Gradient Algorithm, Convergence Analysis, Block Krylov Methods, Preconditioned Conjugate Gradient, Preconditioned GMRES, Jacobi, SOR, and SSOR Preconditioners, ILU Factorization Preconditioners, Block Preconditioners, Types of Partitionings, Techniques, Direct Solution and the Schur Complement, Schur Complement Approaches, Full Matrix Methods, Graph Partitioning: Geometric Approach, Spectral Techniques. Newton’s method and some of its variations, Newton method in several dimension, continuation methods, conjugate direction method and Davidon-Fletcher-Powell Algorithms, Introduction to Non-linear Multigrid with applications. HPC kernels (BLAS, multicore and GPU computing)

Texts / References


CS 528  CAD for VLSI  (3-0-0-6)

Introduction: VLSI design flow, challenges.
Verilog/VHDL: introduction and use in synthesis, modeling combinational and sequential logic, writing test benches.
Logic synthesis: two-level and multilevel gate-level optimization tools, state assignment of finite state machines.
Basic concepts of high-level synthesis: partitioning, scheduling, allocation and binding. Technology mapping.
Testability issues: fault modeling and simulation, test generation, design for testability, built-in self-test.
Testing SoC's.
Basic concepts of verification. Physical design automation.
Review of MOS/CMOS fabrication technology.
VLSI design styles: full-custom, standard-cell, gate-array and FPGA.
Physical design auto-mation algorithms: floor-planning, placement, routing, compaction, design rule check, power and delay estimation, clock and power routing, Special considerations for analog and mixed-signal designs

References:
Detailed Syllabus for Elective courses - Group-2

CS 548 Wireless Networks  
(3 0 0 6)

**Wireless technologies:** Antennas and radio propagation. Signal encoding and modulation techniques. Spread spectrum. Coding and error control.


**Texts:**

**References:**
5. Research papers.

CS 549 COMPUTER AND NETWORK SECURITY  
(3 0 0 6)

**Overview:** vulnerabilities, risk assessment, incidents.
**Cryptography:** Classical Cryptography, Symmetric Cryptography, Public Key (Asymmetric cryptography), Modern Cryptography (RSA, ECC), Hash Functions, Digital Signature.

**Authentication and Key Management:** Entity authentication, Key exchange, Key management, Kerberos

**Networking:** Security: Security at application layer (PGP, S/MIME), Security at Transport Layer (SSL and TLS), Security at Network Layer (IPSEC)

**System Security:** Unix Security, Vulnerabilities and Counter Measures (Virtues, worms, Trojan horses, backdoors, buffer overflows, RPC), Exploits (Buffer overflow, Port Scanning etc). Firewall, VPN etc, Secure (commerce) Transaction over a network.

**Current network security issues:**
Texts:


References:

4. Related publications in Journals/Conferences.

CS 508: Formal Methods for Analysis and Verification (3006)

Introduction to formal methods; Analysis Vs. Verification; Correctness and soundness theorem; Formal semantics: operational, denotational, axiomatic; Specification Languages; Various formal methods and their application to verification and analysis: Model Checking, Abstract Interpretation, Shadow semantics, Hoare logic, Theorem Proving.

References:

5. Recent Research Papers relevant to the course.

CS 502: Pattern Recognition (3006)

Syllabus: Introduction to Pattern Recognition: Learning paradigms, Supervised and unsupervised learning; Bayesian decision theory: Minimum error rate classifier; Parameter estimation: Maximum likelihood and Bayesian Estimation; Hidden Markov models; Nonparametric techniques: Nearest neighbor rules, Parzen windows; Decision trees: Axis-parallel, Oblique, Impurity measures; Feature selection:
Forward, backward search; Component analysis and discriminate functions: Principal component analysis, Fisher linear discriminate, Perceptron, Support vector machines; Generalization ability of learning methods: Bias and variance, Regularization; Bootstrapping, Boosting, Bagging; Unsupervised learning and clustering: k-Means methods.

Texts:

References:
1. C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

CS 743 Advanced Topics in Database Systems (3-0-0-6)

Database Computation Models: Page and Object Models

Correctness in databases: Serializability - review of the basic theory, multiversion serializability, semantic serializability, relative atomicity, relative serializability, etc.

Concurrency control methods: Two phase locking, timestamp and optimistic methods, tree locking.

Correctness in Software Transactional Memory (STM): Opacity, Virtual Worlds Consistency, Abort Shielded Consistency


Object model crash and recovery: Unified concurrency control and recovery, compensating transactions, Algorithm for 2-Layered Systems, Algorithm for General Executions

Special Database Systems: Object based, Semi-structured, Active, Deductive, Temporal, Spatial, Multimedia

Database Security: Access Control Models MAC, DAC, RBAC
Datawarehousing: Multidimensional data model, OLAP, Data Warehouse Architecture

This course will draw materials mainly from the books given below. However, there are many research papers that will help understand the course contents. These will be provided on time to time basis.

Suggested Text Books:


Reference Books: