MODEL CURRICULUM

for

UNDERGRADUATE DEGREE COURSES
IN

COMPUTER SCIENCE & ENGINEERING

(Engineering & Technology)

[January 2018]

ALL INDIA COUNCIL FOR TECHNICAL EDUCATION
Nelson Mandela Marg, Vasant Kunj, New Delhi 110 070
www.aicte-india.org

All India Council for Technical Education
Model curriculum for
Undergraduate Degree Courses in Engineering & Technology

COMPUTER SCIENCE AND ENGINEERING

CONTENTS

All India Council for Technical Education
Model curriculum for
Undergraduate Degree Courses in Engineering & Technology

COMPUTER SCIENCE AND ENGINEERING

Chapter -1
General, Course structure & Theme
&
Semester-wise credit distribution

A. Definition of Credit:

1 Hr. Lecture (L) per week	1 credit
1 Hr. Tutorial (T) per week	1 credit
1 Hr. Practical (P) per week 2 Hours Practical(Lab)/week	0.5 credit 1 credit

B. Range of credits-A range of credits from 150 to 160 for a student to be eligible to get Under Graduate degree in Engineering. A student will be eligible to get Under Graduate degree with Honours or additional Minor Engineering, if he/she completes an additional 20 credits. These could be acquired through MOOCs.

C. Structure of Undergraduate Engineering program:

S. No.	Category	Credit Breakup for CSE students
1	Humanities and Social Sciences including Management courses	12
2	Basic Science courses	24
3	Engineering Science courses including workshop, drawing, basics of electrical/mechanical/computer etc	29
4	Professional core courses	49
5	Professional Elective courses relevant to chosen specialization/branch	18
6	Open subjects – Electives from other technical and /or emerging subjects	12
7	Project work, seminar and internship in industry or elsewhere	15
8	Mandatory Courses [Environmental Sciences, Induction Program, Indian Constitution, Essence of Indian Knowledge Tradition]	(non-credit)
	Total	**159***

**Minor variation is allowed as per need of the respective disciplines.*

D. Credit distribution in the First year of Undergraduate Engineering program:

	Lecture	Tutorial	Laboratory/Practical	Total credits
Chemistry-I	3	1	3	5.5
Physics	3	1	3	5.5
Maths-1	3	1	0	4
Maths-2	3	1	0	4
Programming for Problem solving	3	0	4	5
English	2	0	2	3
Engineering Graphics & Design	1	0	4	3
Workshop/ Practical	1	0	4	3
Basic Electrical Engg.	3	1	2	5
*Biology	2	1	0	3
*Maths-3	3	1	0	4

These courses may be offered preferably in the later semesters

E. Course code and definition:

Course code	Definitions
BSC	Basic Science Courses
ESC	Engineering Science Courses
HSMC	Humanities and Social Sciences including Management courses
PCC-CS	Professional core courses
PEC -CS	Professional Elective courses
OEC-CS	Open Elective courses
LC	Laboratory course
MC	Mandatory courses
SI	Summer Industry Internship
PROJ-CS	Project

HUMANITIES AND SOCIAL SCIENCES INCLUDING MANAGEMENT COURSES

Sl. No	Code No.	Course Title	Hours per week			Total Credits	Semester
			Lecture	Tutorial	Practical		
1	HSMC 201	English	2	0	2	3	2
2	HSMC 301	Humanities – 1	3	0	0	3	3
3	HSMC 401	Management-I (Organizational Behaviour)/ Finance & Accounting	3	0	0	3	4
4	HSMC 501	Humanities – II	3	0	0	3	5
					Total Credits:	**12**	

BASIC SCIENCE COURSE [BSC]

Sl. No	Code No.	Course Title	Hours per week			Total Credits	Semester
			Lecture	Tutorial	Practical		
1	BSC101	Physics (Semi-conductor Physics)	3	1	3	5.5	1
2	BSC 201	Mathematics-II (Probability and Statistics)	3	1	0	4	2
3	BSC 102	Mathematics-I (Calculus and Linear Algebra)	3	1	0	4	1
4	BSC 202	Chemistry-I	3	1	3	5.5	2
5	BSC 701	Biology	2	1	0	3	7
6	BSC 301	Mathematics-III (Differential Calculus)	2	0	0	2	3
					Total Credits:	**24**	

ENGINEERING SCIENCE COURSE [ESC]

Sl. No	Code No.	Course Title	Hours per week			Total Credits	Semester
			Lecture	Tutorial	Practical		
1	ESC 101	Basic Electrical Engineering	3	1	2	5	1
2	ESC 102	Engineering Graphics & Design	1	0	4	3	1
3	ESC 201	Programming for Problem Solving	3	0	4	5	2
4	ESC 202	Workshop/Manufacturing Practices	1	0	4	3	2
5	ESC 301	Analog Electronic Circuits	3	0	4	5	3
6	ESC 302	Digital Electronics	3	0	4	5	4
7	ESC 501	Signals and Systems	3	0	0	3	5
					Total Credits:	**29**	

PROFESSIONAL CORE COURSES [PCC]

Sl. No	Code No.	Course Title	Hours per week			Total Credits	Semester
			Lecture	Tutorial	Practical		
1	PCC CS301	Data Structure & Algorithms	3	0	4	5	3
2	PCC CS302	IT Workshop – (Sci Lab/MATLAB)	1	0	4	3	3
3	PCC CS401	Discrete Mathematics	3	1	0	4	4
4	PCC CS402	Computer Organization and Architecture	3	0	4	5	3
5	PCC CS403	Operating Systems	3	0	4	5	4
6	PCC CS404	Design and Analysis of Algorithms	3	0	4	5	4
7	PCC CS 501	Database Management Systems	3	0	4	5	5
8	PCC CS502	Formal Language, Automats and Complier	3	0	0	3	5
9	PCC CS503	Object Oriented Programming	2	0	4	4	5
10	PCC CS601	Complier Design	3	0	4	5	6
11	PCC CS602	Computer Networks	3	0	4	5	6
					Total Credits	**49**	

PROFESSIONAL ELECTIVE [PEC]

Sl. No	Code No.	Course Title	Hours per week			Total Credits	Semester
			Lecture	Tutorial	Practical		
1	PEC	Elective – I	3	0	0	3	5
2	PEC	Elective-II	3	0	0	3	6
3	PEC	Elective-III	3	0	0	3	6
4	PEC	Elective-IV	3	0	0	3	7
5	PEC	Elective-V	3	0	0	3	7
6	PEC	Elective-VI	3	0	0	3	8
					Total Credits	**18**	

OPEN ELECTIVE COURSES [OEC]

Sl. No	Code No.	Course Title	Hours per week			Total Credits	Semester
			Lecture	Tutorial	Practical		
1	OEC	Open Elective – I	3	0	0	3	6
2	OEC	Open-Elective-II	3	0	0	3	7
3	OEC	Open-Elective-III	3	0	0	3	8
4	OEC	Open-Elective-IV	3	0	0	3	8
					Total Credits:	**12**	

4 year Curriculum structure
Undergraduate Degree in Engineering & Technology

Branch / course: Computer Science and Engineering
Total credits (4 year course): 159

I. Induction Program (Please refer Appendix-A for guidelines)

Induction program (mandatory)	**3 weeks duration** (Please refer Appendix-A for guidelines & also details available in the curriculum of Mandatory courses)
Induction program for students to be offered right at the start of the first year.	• Physical activity • Creative Arts • Universal Human Values • Literary • Proficiency Modules • Lectures by Eminent People • Visits to local Areas • Familiarization to Dept./Branch & Innovations

II Semester-wise structure of curriculum

[L= Lecture, T = Tutorials, P = Practicals & C = Credits]

Semester I (First year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Course Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Basic Science course	BSC 101	Physics (semi-conductor Physics)	3	1	3	5.5
2	Basic Science course	BSC 102	Mathematics-1 (Calculus & Linear Algebra)	3	1	0	4
3	Engineering Science Course	ESC 101	Basic Electrical Engineering	3	1	2	5
4	Engineering Science Course	ESC 102	Engineeri ng Graphics & Design	1	0	4	3
						Total credits	**17.5**

Semester II (First year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Basic Science course	BSC 202	Chemistry-I	3	1	3	5.5
2	Basic Science course	BSC 201	Mathematics-II (Probability and Statistics)	3	1	0	4
3	Engineering Science Course	ESC 201	Programming for Problem Solving	3	0	4	5
4	Engineering Science Course	ESC 202	Workshop /Manufacturing Practices	1	0	4	3
5	Humanities & Social Sciences including Management courses	HSMC 201	English	2	0	2	3
			Total credits				**20.5**

Semester III (Second year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Engineering Science Course	ESC 301	Analog Electronic Circuits	3	0	4	5
2	Professional Core Courses	PCC-CS301	Data structure & Algorithms	3	0	4	5
3	Professional Core Courses	ESC 302	Digital Electronics	3	0	4	5
4	Professional Core Courses	PCC-CS302	IT Workshop (Sci Lab/MATLAB)	1	0	4	3
5	Basic Science course	BSC 301	Mathematics-III (Differential Calculus)	2	0	0	2
6	Humanities & Social Sciences including Management courses	HSMC 301	Humanities-I	3	0	0	3
			Total credits				**23**

Semester IV (Second year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Professional Core Courses	PCC-CS401	Discrete Mathematics	3	1	0	4
2	Engineering Science Course	PCC-CS 402	Computer Organization & Architecture	3	0	4	5
3	Professional Core Courses	PCC-CS403	Operating Systems	3	0	4	5
4	Professional Core Courses	PCC-CS404	Design & Analysis of Algorithms	3	0	4	5
5	Humanities &Social Sciences including Management courses	HSMC 401	Management 1 (Organizational Behaviour/ Finance & Accounting)	3	0	0	3
6	Mandatory Courses	MC	Environmental Sciences	-	-	-	0
				Total credits			**22**

Semester V (Third year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Engineering Science Course	ESC501	Signals & Systems	3	0	0	3
2	Professional Core Courses	PCC-CS501	Database Management Systems	3	0	4	5
3	Professional Core Courses	PCC-CS502	Formal Language & Automata Theory	3	0	0	3

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
4	Professional Core Courses	PCC-CS503	Object Oriented Programming	2	0	4	4
5	Humanities &Social Sciences including Management courses	HSMC-501	Humanities II	3	0	0	3
6	Professional Elective courses	PEC	Elective-I	3	0	0	3
7	Mandatory Courses	MC	Constitution of India/ Essence of Indian Knowledge Tradition	-	-	-	0
			Total credits				**21**

Semester VI (Third year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Professional Core Courses	PCC-CS601	Complier Design	3	0	4	5
2	Professional Core Courses	PCC-CS602	Computer Networks	3	0	4	5
3	Professional Elective courses	PEC	Elective-II	3	0	0	3
4	Professional Elective courses	PEC	Elective-III	3	0	0	3
5	Open Elective courses	OEC	Open Elective-I (Humanities)	3	0	0	3
6	Project	PROJ-CS601	Project-1	0	0	6	3
			Total credits				**22**

Semester VII (Fourth year] Curriculum
Branch/Course: Computer Science Engineering

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Professional Elective courses	PEC	Elective-IV	3	0	0	3
2	Professional Elective courses	PEC	Elective-V	3	0	0	3
3	Open Elective courses	OEC	Open Elective-II	3	0	0	3
4	Basic Science course	BSC 701	Biology	2	1	0	3
5	Project	PROJ-CS701	Project-II	0	0	12	6
						Total credits	**18**

Semester VIII (Fourth year] Curriculum
Branch/Course: Computer Science Engineering
[Summer Industry Internship]

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	Professional Elective courses	PEC	Elective-VI	3	0	0	3
2	Open Elective courses	OEC	Open Elective-III	3	0	0	3
3	Open Elective courses	OEC	Open Elective-IV	3	0	0	3
4	Project	PROJ-CS801	Project-III	0	0	12	6
						Total credits	**15**

CHAPTER 2

DETAILED 4-YEAR CURRICULUM CONTENTS

Undergraduate Degree in Engineering & Technology

Branch/Course: COMPUTER SCIENCE AND ENGINEERING

Second year (Third semester onwards)

PROFESSIONAL CORE COURSES

PCC-CS301	Data Structure & Algorithms	3L:0T: 4P	5 credits

Pre-requisites	ESC 201

Objectives of the course:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures

Detailed contents:

Module 1:

Introduction: Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. **Searching:** Linear Search and Binary Search Techniques and their complexity analysis.

Module 2:

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.

Module 3:

Linked Lists: Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.

Module 4:

Sorting and Hashing: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

Graph: Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

Suggested books:

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.

Suggested reference books:

1. Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. "How to Solve it by Computer", 2nd Impression by R.G. Dromey, Pearson Education.

Course outcomes

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
2. For a given Search problem (Linear Search and Binary Search) student will able to implement it.
3. For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.
4. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
5. Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.

PCC-CS401	Discrete Mathematics	3L:1T:0P	4 Credits

Objectives of the course

Throughout the course, students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following:

1. Use mathematically correct terminology and notation.
2. Construct correct direct and indirect proofs.
3. Use division into cases in a proof.
4. Use counterexamples.
5. Apply logical reasoning to solve a variety of problems.

Detailed contents:

Module 1:

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.

Principles of Mathematical Induction: The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic.

Module 2:
Basic counting techniques-inclusion and exclusion, pigeon-hole principle, permutation and combination.

Module 3:
Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. **Proof Techniques:** Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency.

Module 4:
Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form

Module 5:
Graphs and Trees: Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Graph Colouring, Colouring maps and Planar Graphs, Colouring Vertices, Colouring Edges, List Colouring, Perfect Graph, definition properties and Example, rooted trees, trees and sorting, weighted trees and prefix codes, Bi-connected component and Articulation Points, Shortest distances.

Suggested books:
1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
2. Susanna S. Epp, Discrete Mathematics with Applications,4th edition, Wadsworth Publishing Co. Inc.
3. C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.

Suggested reference books:
1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and It's Application to Computer Science", TMG Edition, TataMcgraw-Hill
2. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press. Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson,
3. Discrete Mathematics, Tata McGraw - Hill

Course Outcomes
1. For a given logic sentence express it in terms of predicates, quantifiers, and logical connectives
2. For a given a problem, derive the solution using deductive logic and prove the solution based on logical inference

3. For a given a mathematical problem, classify its algebraic structure
4. Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra
5. Develop the given problem as graph networks and solve with techniques of graph theory.

PCC-CS402	**Computer Organization &Architecture**	**3L:0T:4P**	**5 Credits**

Pre-requisites	**ESC 302**

Objectives of the course:

To expose the students to the following:
1. How Computer Systems work & the basic principles
2. Instruction Level Architecture and Instruction Execution
3. The current state of art in memory system design
4. How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism
6. To impart the knowledge on micro programming
7. Concepts of advanced pipelining techniques.

Detailed contents:

Module 1

Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU–registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction sets of some common CPUs.

Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-andadd, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

Module 2:

Introduction to x86 architecture.
CPU control unit design: hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU.
Memory system design: semiconductor memory technologies, memory organization.
Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers–program controlled, interrupt driven and DMA, privileged and

non-privileged instructions, software interrupts and exceptions. Programs and processes–role of interrupts in process state transitions, I/O device interfaces – SCII, USB

Module 3:
Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.
Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

Module 4:
Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Suggested books:

1. "Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
2. "Computer Organization and Embedded Systems", 6th Edition by CarlHamacher, McGraw Hill Higher Education.

Suggested reference books:

1. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
3. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course outcomes

1. Draw the functional block diagram of a single bus **architecture of a computer and describe the function of the** instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.
2. **Write** assembly language program for specified microprocessor for computing 16 bit multiplication, division and I/O device interface (ADC, Control circuit, serial port communication).
3. Write a flowchart for Concurrent access to memory and cache coherency in **Parallel Processors** and describe the process.
4. Given a CPU organization and instruction, design a memory module and analyze its operation by interfacing with the CPU.
5. Given a CPU organization, assess its performance, and apply design techniques to enhance performance using pipelining, parallelism and RISC methodology

---

PCC- CS403	Operating Systems	3L:0T:4P	5 Credits

Pre-requisites	PCC – CS402

Objectives of the course

To learn the fundamentals of Operating Systems.

1. To learn the mechanisms of OS to handle processes and threads and their communication
2. To learn the mechanisms involved in memory management in contemporary OS
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
4. To know the components and management aspects of concurrency management
5. To learn to implement simple OS mechanisms

Detailed contents

Module 1:
Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

Module 2:
Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

Module 3:
Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

Module 4:
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Module 5:
Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.
Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

Module 6:
I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms
File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

Suggested books:

1. Operating System Concepts Essentials, 9th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Suggested reference books:

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Course Outcomes

1. Create processes and threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
3. For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement file management system.

5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

PCC-CS-404	Design and Analysis of Algorithms	3L:0T: 4P	5 Credits

Pre-requisites	ESC 201

Objectives of the course

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Detailed contents:

Module 1:

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

Module 2:

Fundamental Algorithmic Strategies: Brute-Force, Greedy,Dynamic Programming, Branch-and-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving , Bin Packing, Knap Sack TSP. Heuristics – characteristics and their application domains.

Module 3:

Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

Module 4:

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.

Module 5:

Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE

Suggested books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.

Suggested reference books

1. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms—A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Course Outcomes

1. For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms .
2. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and develop the dynamic programming algorithms, and analyze it to determine its computational complexity.
5. For a given model engineering problem model it using graph and write the corresponding algorithm to solve the problems.
6. Explain the ways to analyze randomized algorithms (expected running time, probability of error).
7. Explain what an approximation algorithm is. Compute the approximation factor of an approximation algorithm (PTAS and FPTAS).

PCC-CS501	Database Management Systems	3L:0T:4 P	5 Credits

Pre-requisites	PCC-CS 403

Objectives of the course

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
- To understand and use data manipulation language to query, update, and manage a database

- To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Detailed contents

Module 1
Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).
Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

Module 2:
Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.
Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Module 3:
Storage strategies: Indices, B-trees, hashing.

Module 4:
Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

Module 5:
Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

Module 6:
Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Suggested books:

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Suggested reference books

1 "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
2 "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
3 "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

Course Outcomes

1. For a given query write relational algebra expressions for that query and optimize the developed expressions
2. For a given specification of the requirement design the databases using E-R method and normalization.
3. For a given specification construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2.
4. For a given query optimize its execution using Query optimization algorithms
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
6. Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

PCC-CS502	**Formal Language & Automata Theory**	**3L:0T:0 P**	**3 Credits**

Pre-requisites	**PCC-CS 403**

Objectives of the course

- Develop a formal notation for strings, languages and machines.
- Design finite automata to accept a set of strings of a language.
- Prove that a given language is regular and apply the closure properties of languages.
- Design context free grammars to generate strings from a context free language and convert them into normal forms.
- Prove equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
- Identify the hierarchy of formal languages, grammars and machines.
- Distinguish between computability and non-computability and Decidability and undecidability.

Detailed contents

Module 1:

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Suggested books

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

Suggested reference books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
2. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
4. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

Course Outcomes:

1. Write a formal notation for strings, languages and machines.
2. Design finite automata to accept a set of strings of a language.
3. For a given language determine whether the given language is regular or not.
4. Design context free grammars to generate strings of context free language .
5. Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
6. Write the hierarchy of formal languages, grammars and machines.
7. Distinguish between computability and non-computability and Decidability and undecidability.

PCC-CS503	Object Oriented Programming	2L:0T:4 P	4 Credits

Pre-requisites	PCC-CS 301

Objectives of the course
The course will introduce standard tools and techniques for software development, using object oriented approach, use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests.

Detailed contents

- *Abstract data types and their specification.*
- *How to implement an ADT.* Concrete state space, concrete invariant, abstraction function. Implementing operations, illustrated by the Text example.
- *Features of object-oriented programming.* Encapsulation, object identity, polymorphism – but not inheritance.
- *Inheritance in OO design.*
- *Design patterns.* Introduction and classification. The iterator pattern.
- *Model-view-controller pattern.*
- *Commands as methods* and as objects. • *Implementing OO language features.*
- *Memory management.*
- *Generic types and collections*
- *GUIs.* Graphical programming with Scala and Swing • *The software development process.*

The concepts should be practised using C++ and Java. Pearl may also be introduced wherever possible.

Suggested books

1. Barbara Liskov, *Program Development in Java*, Addison-Wesley, 2001

Suggested reference books

1. Any book on Core Java
2. Any book on C++

Course Outcomes
After taking the course, students will be able to:

1. Specify simple abstract data types and design implementations, using abstraction functions to document them.
2. Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
3. Name and apply some common object-oriented design patterns and give examples of their use.
4. Design applications with an event-driven graphical user interface.

PCC-CS601	Complier Design	3L:0T: 4P	5 Credits

Pre-requisites	PCC-CS 302, PCC-CS 502

Objectives of the course

- To understand and list the different stages in the process of compilation.
- Identify different methods of lexical analysis
- Design top-down and bottom-up parsers
- Identify synthesized and inherited attributes
- Develop syntax directed translation schemes
- Develop algorithms to generate code for a target machine

Detailed contents
Module 1:
The aim is to learn how to design and implement a compiler and also to study the underlying theories. The main emphasis is for the imperative language. Introduction: Phases of compilation and overview. Lexical Analysis (scanner): Regular languages, finite automata, regular expressions, from regular expressions to finite automata, scanner generator (lex, flex).Syntax Analysis (Parser): Context-free languages and grammars, push-down automata, LL(1) gram-mars and top-down parsing, operator grammars, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc, bison)Semantic Analysis: Attribute grammars, syntax directed definition, evaluation and flow of attribute in a syntax tree. Symbol Table: Its structure, symbol attributes and management. Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope. Intermediate Code Generation: Translation of different language features, different types of intermediate forms. Code Improvement (optimization): Analysis: control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc. Architecture dependent code improvement: instruction scheduling (for pipeline), loop optimization (for cache memory) etc. Register allocation and target code generation Advanced topics: Type systems, data abstraction, compilation of Object Oriented features and non-imperative programming languages.

Course Outcomes

1. For a given grammar specification develop the lexical analyser
2. For a given parser specification design top-down and bottom-up parsers
3. Develop syntax directed translation schemes
4. Develop algorithms to generate code for a target machine

PCC-CS602	**Computer Networks**	**3L:0T: 4P**	**5 Credits**

Pre-requisites	PCC-CS - 402 PCC-CS - 403

Objectives of the course

- To develop an understanding of modern network architectures from a design and performance perspective.
- To introduce the student to the major concepts involved in wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs).
- To provide an opportunity to do network programming
- To provide a WLAN measurement ideas.

Detailed contents

Module 1:

Data communication Components: Representation of data and its flow Networks , Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.

Module 2:

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA

Module 3:

Network Layer: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols.

Module 4:

Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

Module 5:

Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography

Suggested books

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.

Suggested reference books

1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Course Outcomes

1. Explain the functions of the different layer of the OSI Protocol.
2. Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
3. For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component
4. For a given problem related TCP/IP protocol developed the network programming.
5. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

PROFESSIONAL ELECTIVE COURSES

Additional Courses for B.Tech (Hons.)

Branch/Course: Computer Science Engineering

In order to have an Honours degree, a student choose 19-20 credits from the following courses in addition. The professional electives may be selected **excluding** these.

Sl. No.	Type of course	Code	Course Title	Hours per week			Credits
				Lecture	Tutorial	Practical	
1	PEC	PEC-CS-T<number>	Graph Theory	3	0	0	3
2	PEC	PEC-CS-S<number>	Software Engineering	3	0	4	5
3	PEC	PEC-CS-S<number>	Embedded Systems	3	0	4	5
4.	PEC	PEC-CS-D<number>	Artificial Intelligence	3	0	0	3
5.	PEC	PEC-CS-A<number>	Cryptography & Network Security	3	0	0	3
6.	PEC	PEC-CS-S<number>	Internet-of-Things	3	0	0	3
7.	PEC	PEC-CS-D<number>	Data Analytics	3	0	0	3
8.	PEC	PEC-CS-D<number>	Machine Learning	3	0	0	3

Electives

Electives will be introduced in 4 threads besides the Open Elective. There are 6 slots for Electives and 4 slots for Open Electives. The department may permit students to take 50% of these (electives + open electives) from other disciplines, based on the choices of the students and consent of course advisors.

A. **Theory** B. **Systems** C. **Data Science** D.**Applications** and E.**Open Electives**

The students will have options of selecting the electives from the different threads depending on the specialization they wish to acquire. **There should be at least two electives from the open elective choices; the rest two can be taken from the other threads, if intended.**

Pls. see the Table.

The Electives are shown in different threads.

The list is suggestive.

The actual list of electives will depend on the availability of faculty and their research interests. **However, there should be courses available in each thread.**

On-line MOOC courses may contribute upto 20% of the credits, with in-house examination being conducted.

Theory and Algorithms **Code: PEC-CS-T<number>**	**Systems** **Code: PEC-CS-S<number>**	**Data Science and Machine Intelligence** **Code: PEC-CS-D <number>**	**Applications** **Code: PEC-CS-A<number>**	**Open Electives** **OEC-CS<number>**
Theory of Computation	Advanced Computer Architecture	Artificial Intelligence	Image Processing	Soft Skills and Interpersonal Communication
Graph Theory	Software Engineering			
Advanced Algorithms	Distributed Systems	Machine Learning	Digital Signal Processing	Human Resource Development and Organizational Behavior
Parallel and Distributed Algorithms	Embedded Systems	Data Mining	Cloud Computing	Cyber Law and Ethics
Computational Complexity	Advanced Operating Systems	Soft Computing	Human Computer Interaction	Introduction to Philosophical Thoughts
Computational Geometry	Low Power Circuits and Systems	Speech and Natural Language Processing	Electronic Design Automation	Comparative Study of Literature
Queuing Theory and Modeling	Fault Tolerant Computing	Data Analytics	Computer Graphics	Indian Music System

Computational Number Theory	Real Time Systems	Information Retrieval	VLSI System Design	History of Science & Engineering
Quantum Computing	Ad-Hoc and Sensor Networks	Neural Networks and Deep Learning	Optimization Techniques	Introduction to Art and Aesthetics
Information Theory and	Signals and Networks	Multi-agent Intelligent	Web and Internet	Economic Policies in India

Coding		Systems	Technology
	Internet-of-Things	Data Analytics	Cryptography and Network Security

MODEL CURRICULUM

for

UNDERGRADUATE DEGREE COURSES
IN

ELECTRONICS & COMMUNICATION ENGINEERING

(Engineering & Technology)

[January 2018]

ALL INDIA COUNCIL FOR TECHNICAL EDUCATION
Nelson Mandela Marg, Vasant Kunj, New Delhi 110 070
www.aicte-india.org