NAGALAND UNIVERSITY

(A Central University Established by an Act of Parliament 1989)

School of Engineering &Technology

Annexure I: Revision of course structure/syllabus for B.Tech in Computer Science & Engineering

1st semester

S.N.	Course Code	Course Name	L	T	P	Total Credits
THEORY						
1	G1T01	Engineering Mathematics –I	3	1	0	4
2	G1T02	Engineering Physics-I	3	0	0	3
3	G1T03	Technical English	2	1	0	3
4	G1T04	Electrical Engineering	3	0	0	3
5	G1T05	Engineering Chemistry	3	0	0	3
6	G1T06	Engineering Graphics	1	0	0	1
PRACTICAL						
1	G1L01	Engineering Physics –I Lab	0	0	2	1
	G1L02	Engineering Chemistry	0	0	2	1
3	G1L03	Engineering Graphics Lab	0	0	4	2
Total Credits			**15**	**2**	**8**	**21**

2nd Semester

S.N.	Course Code	Course Name	L	T	P	Total Credits
1	G2T01	Engineering Mathematics –II	3	1	0	4
2	G2T02	Engineering Physics-II	3	0	0	3
3	G2T03	Fundamentals of Computing	3	0	0	3
4	G2T04	Basic Electronics	3	0	0	3
5	G2T05	Engineering Mechanics	2	1	0	3
6	G2T06	Environmental Science	3	0	0	0 (non credit)
PRACTICAL						
1	G2L01	Workshop Practice	0	0	4	2
2	G2L02	Basic Electronics Lab	0	0	2	1
2	G2L03	Fundamentals of Computing Lab	0	0	2	1
4	G2L04	Engineering Physics-II Lab	0	0	2	1
Total Credits			**17**	**2**	**10**	**21**

3rd Semester

S.N.	Course Code	Course Name	L	T	P	Total Credits
THEORY						
1	CSB301	Object Oriented Programming	3	0	0	3
2	CSB302	Data Structures & Algorithm	3	0	0	3
3	CSB303	Computer Graphics & Virtual Reality	3	0	0	3
4	MAT3T2	Differential Calculus	3	1	0	4
5	EC3T03	Digital Electronics & Logic Design	2	1	0	3
PRACTICAL						
1	CSB311	Object Oriented Programming Lab	0	0	3	1.5
2	CSB312	Data Structures & Algorithm Lab	0	0	3	1.5
3	EC3L02	Digital Electronics & Logic Design Lab	0	0	3	1.5
Total Credits			**14**	**2**	**9**	**20.5**

4th Semester

S.N.	Course Code	Course Name	L	T	P	Total Credits
THEORY						
1	CSB401	Design & Analysis of Algorithms	3	0	0	3
2	CSB402	Web & Internet	3	0	0	3
3	CSB403	Formal Language & Automata Theory	3	0	0	3
4	CSB404	Computer Organization & Architecture	4	0	0	4
5	MAT4T2	Discrete Mathematics	3	1	0	4
PRACTICAL						
1	CSB411	Design & Analysis of Algorithms Lab	0	0	3	1.5
2	CSB412	Web & Internet Lab	0	0	3	1.5
3	CSB413	Computer Organization & Architecture Lab	0	0	2	1
Total Credits			**16**	**1**	**8**	**21**

5th Semester

S.N.	Course Code	Course Name	L	T	P	Total Credits
THEORY						
1	CSB501	Graph Theory	3	0	0	3
2	CSB502	Operating System	3	0	0	3
3	CSB503	Database Management Systems	3	0	0	3
4	CSB504	Software Engineering	3	0	0	3
5	MAT5T1	Numerical Analysis & Probability	3	1	0	4
PRACTICAL						
1	CSB511	Software Engineering Lab	0	0	3	1.5
2	CSB512	Operating System Lab	0	0	3	1.5
3	CSB513	Database Management Systems Lab	0	0	3	1.5
Total Credits			**15**	**1**	**9**	**20.5**

6th Semester

S.N.	Course Code	Course Name	L	T	P	Total Credits
1	CSB601	Compiler Design	4	0	0	4
2	CSB602	Computer Networks	3	0	0	3
3	CSB62X	Elective-I	3	0	0	3
4	CSB62X	Elective-II	3	0	0	3
5	HSB601	Project Management & Entrepreneurship	2	0	0	2
PRACTICAL						
1	CSB611	Computer Networks Lab	0	0	3	1.5
2	CSB612	Compiler Design Lab	0	0	3	1.5
3	CSB613	Application Programming Lab	0	1	3	2.5
Total Credits			**15**	**1**	**9**	**20.5**

7th Semester

S.N.	Course Code		L	T	P	Total Credits
THEORY						
1	CSB701	Distributed System	3	0	0	3
2	CSB702	Machine Learning	3	0	0	3
3	CSB72X	Elective-III	3	0	0	3
4	CSB72X	Elective-IV	3	0	0	3
PRACTICAL						
1	CSB711	Project-I #	0	0	12	6
2	CSB712	Machine Learning Lab	0	0	3	1.5
3	CSB713	Colloquium-I*	0	0	0	0 (No credit)
Total Credits			**12**	**0**	**15**	**19.5**

8th Semester

S.N.	Course Code		L	T	P	Total Credits
THEORY						
1	CSB82X	Elective-V	3	0	0	3
2	CSB83X	Open Elective-I	3	0	0	3
3	G8T01	Constitution of India	3	0	0	0 (No credit)
4	HSB801	Human Relations at work	2	0	0	2
PRACTICAL						
1	CSB811	Project-II	0	0	16	8
Total Credits			**11**	**0**	**16**	**16**

*****The student will give presentation (Colloquium-I) on the summer/winter/industrial training (**6 – 8 weeks**) that She / He underwent during the vacation period after 5th or 6th semester. The credit(Pass or Fail) will be awarded in the 7th Semester under Colloquium-I. Presentation will be conducted in the beginning of 7th semester.

#The student will submit a synopsis for their Project at the beginning of the semester in a specified format which should be approved by the departmental committee. The student will also have to present the progress of their project through seminars and progress reports.

Total no. of Credits from 1st to 8th Semester: 160 (One Hundred Sixty)

List of Elective

Sl. No.	Course Code	Semester	Course Title	L	T	P	Total Credits
1	CSB621	6	Data Mining	3	0	0	3
2	CSB622	6	Data Analytics	3	0	0	3
3	CSB623	6	Information Retrieval	3	0	0	3
4	CSB624	6	Multimedia Technology	3	0	0	3
5	CSB625	6	Software Testing	3	0	0	3
6	EC6T06	6	Information Theory & Coding	3	0	0	3
7	EC8T01	6	Digital Image Processing	3	0	0	3
8	MAT6T1	6	Operations Research	3	0	0	3
1	CSB721	7	Design & Management of Computer Network	3	0	0	3
2	CSB722	7	Human Computer Interaction	3	0	0	3
3	CSB723	7	Cloud Computing	3	0	0	3
4	CSB724	7	Wireless Sensor Networks	3	0	0	3
5	CSB725	7	Internet-of- Things	3	0	0	3
6	CSB726	7	Real Time Systems	3	0	0	3
7	CSB727	7	Advanced Computer Architecture & parallel programming	3	0	0	3
8	EC7EL1	7	Embedded Systems & Design	3	0	0	3
1	CSB821	8	Distributed Database	3	0	0	3
2	CSB822	8	Artificial Intelligence	3	0	0	3
3	CSB823	8	Speech & Natural Language Processing	3	0	0	3
4	CSB824	8	Neural Networks & Deep Learning	3	0	0	3

List of Open Elective

1	CSB831	8	Cryptography & Network Security	3	0	0	3
2	CSB832	8	Mobile Applications & Services	3	0	0	3
3	CSB833	8	Cyber Law & Ethics	3	0	0	3
4	CSB834	8	Linux Internal	3	0	0	3

The following courses will be offered through MOOCs/SWAYAM/NPTEL with in-house examination.

1. Embedded Systems
2. Software Testing
3. Artificial Intelligence
4. Internet-of-Things
5. Cryptography & Network Security

The following practical lab will be conducted under virtual lab.

1. Computer Organization & Architecture Lab
2. Software Engineering Lab

Details of the course contents for B.Tech in Computer Science & Engineering

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB301	Object Oriented Programming	3	0	0	3

-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.

Textbooks:

1. Barbara Liskov, Program Development in Java, Addison-Wesley, 2001.
2. A.R. Venugopal, Rajkumar, T. Ravishanker " Mastering C++", TMH, 1997.
3. R.Lafore," Object Oriented Programming with C++', BPB publication,2004.
4. Schildt Herbert,"C++ Programming", 2nd Edition, Wiley Dream Tech.

Reference books:

1. Herbert Schildt, "Java - The Complete Reference 7th Edition"**,** Mcgraw Higher Ed 2007.
2. D. Parasons, " Object Oriented Programming using C++", BPB Publications, 1999.
3. Steven C. Lawlor, " The art of Programming Computer Science with C++", Vikas publication, 2002.

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.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB302	Data Structures & Algorithm	3	0	0	3

Unit 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.
Unit 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.
Unit 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.
Unit 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.

Textbooks:
1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
2. R.L.Kruse, B.P. Leung, C.L. Tondo, " Data Strucfture and program design in C", PHI,2000.

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.
3. Schaum's outline series, " Data Structure", TMH, 2002.

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.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB303	Computer Graphics & Virtual Reality	3	0	0	3

UNIT-1: Introduction to Computer Graphics, Display devices, Bitmap and Vector based Graphics, Overview of coordinate system, scan conversion of: point, line using differential analyzer and Breshenham's algoritm, circle using midpoint approach. Curve Generation: Bezier and B-Spline curves. Introduction to fractals: generation procedure, classification, dimension and Koch Curve.

UNIT-2: Area Filling:Inside/Outside test, scanline polygon fill algorithm, Boundary fill and floor fill algorithm. Basic 2D transformations: Translation, Rotation, Scaling, Reflection, Shear – Homogeneous Matrix representation and composite transformation.

UNIT-3: Introduction to 2D viewing, viewing pipeline, view coordinate reference frame, window to viewport transformation, point clipping. Line clipping: Cohen Sutherland Algorithm, Liang Barsky algorithms. Polygon clipping: Sutherland Hodgeman polygon clipping and Weiler Atherton. Text Clipping.

UNIT-4: Three Dimensional Transformations: Translation, Scaling, Rotation and Composite. 3D object representation: Polygon surfaces, Tables, Meshes. 3D viewing pipeline, viewing transformation, Projections: Parallel (oblique and orthographic), Perspective (one point); visibility – Z-Buffer.

UNIT-5: Introduction to Animation, key frame animation, animation sequence, Motion control methods, morphing, mesh warping.

UNIT-6: Virtual Reality : Basic Concepts , Classical Components of VR System , Types of VR Systems, Three Dimensional Position Trackers, Navigation and Manipulation Interfaces, Gesture. Interfaces, Graphical Display, Sound displays, and Haptic Feedback . Input Devices ,Graphical Rendering Pipeline , Haptic Rendering Pipeline, Open GL rendering pipeline.Applications of Virtual Reality.

UNIT-7: Geometric Modeling: Virtual Object Shape, Object Visual Appearance. Kinematics Modeling: Object Position, Transformation Invariants, Object Hierarchies, Physical Modeling: Collision Detection, Surface Deformation, Force Computation. Behavior Modeling.

UNIT-8: Programming through VRML : Defining and Using Nodes and Shapes , VRML Browsers , Java 3D :Visual Object Definition by Shape 3D instances , Defining personal visual object class, ColorCube Class, Geometric – Utility Classes, Geometry Classes , Attributes.

Textbooks:

1. Donald Hearn and M. Pauline Baker, "Computer Graphics", Pearson Education.
2. R. K Maurya, "Computer Graphics with Virtual Reality", Wiley India.

Reference Books:

1. Grigore Burdea, Philippe Coiffet, "Virtual Reality Technology", Wiley.
2. Steven Harrington, "Computer Graphics", McGraw Hill.
3. Vince, "Virtual Reality Systems", Pearson Education.
4. F.S. Hill , Stephen M. Kelley , "Computer Graphics using Open GL" Prentice Hall

Course Outcomes: After completion of course, students would be:

1.To list the basic concepts used in computer graphics.

2. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.

3. To describe the importance of viewing and projections.

4. To define the fundamentals of animation and its related technologies.

5. Develop design and problem solving skills with application to computer graphics.

6. Gain experience in constructing interactive computer graphics programs using OpenGL

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
MAT3T2	Differential Calculus	3	1	0	4

Unit 1: Sequences and series*:* Convergence of sequence and series, tests for convergence, power series, Taylor's series. Series for exponential, trigonometric and logarithmic functions.

Unit 2: Multivariable Calculus (Differentiation)*:* Limit, continuity and partial derivatives, directional derivatives, total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers; Gradient, curl and divergence.

Unit 3: Multivariable Calculus (Integration): Multiple Integration: double and triple integrals (Cartesian and polar), change of order of integration in double integrals, Change of variables (Cartesian to polar). Theorems of Green, Gauss and Stokes, orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds.

Unit 4: First order ordinary differential equations

Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

Unit 5:Ordinary differential equations of higher orders: Second order linear differential equations with variable coefficients, method of variation of parameters, Cauchy-Euler equation; Power series solutions; Legendre polynomials, Bessel functions of the first kind and their properties.

Text books/Reference books:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
2. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
5. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
6. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
7. W. E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
8. S. L. Ross, Differential Equations, 3rd Ed., Wiley India, 1984.
9. E. A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.
10. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1958.
11. G.F. Simmons and S.G. Krantz, Differential Equations, Tata McGraw Hill, 2007.

Course Outcomes:

The objective of this course is to familiarize the prospective engineers with techniques in multivariate integration, ordinary and partial differential equations and complex variables. It aims to equip the students to deal with advanced level of mathematics and applications that would be essential for their disciplines.

The students will learn:

1. The mathematical tools needed in evaluating multiple integrals and their usage.
2. The effective mathematical tools for the solutions of differential equations that model physical processes.
3. The tools of differentiation and integration of functions of a complex variable that are used in various techniques dealing engineering problems.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
EC3T03	Digital Electronics & Logic Design	2	1	0	3

Unit 1: Introduction To Digital System: Number systems, binary arithmetic and codes: positional number system; binary, octal and hexadecimal number systems; representation of signed numbers; binary arithmetic –addition, subtraction, multiplications and division; fixed and floating point numbers ; binary coded decimal codes; Gray codes; error detection and correction codes-parity check codes and Hamming code.

Unit 2: Boolean Algebra And Switching Functions: Boolean algebra; basic postulates and fundamental theorems of Boolean algebra; truth tables; basic logic operations and gate symbols ; algebraic forms of switching functions- SOP and POS forms, minterms and maxterms; derivation of canonical forms; minterms and maxterms; simplification of switching functions- K-map and quine-McCluskey tabular minimization methods; synthesis of combinational logic circuits-NAND and NOR networks.

Unit 3: Logic Families: Introduction to different logic families; operational characteristics of BJT in saturation and cut-off regions; operational characteristics of MOSFET as switch; TTL inverter- circuit description and operation; CMOS inverter-circuit description and operation; other TTL and CMOS gates; electrical behavior of logic circuits-noise margins, fanout, transmission time, propagation delay, power dissipation.

Unit 4: Combinational Logic Modules: Decoders, encoders, multiplexers, de-multiplexers and their applications; three state devices and buses; code converter; binary adders: half adder and full adder, ripple carry adder, carry-look-ahead adder; subtractors ; multipliers; ALU; comparators; parity circuits; circuit timing-timing diagrams and specifications ; combinational circuit design examples.

Unit 5: Sequential Logic Devices And Circuits : Latches; flip- flops; registers, shift-registers; counters ripple counters , synchronous counters , up-down counters, BCD counters, ring counters, timing diagrams and specifications; state machine models-synchronous state machines; state machine design examples design examples; design using ASM charts ; timing hazards and races ; design and analysis of asynchronous sequential circuits: pulse mode and fundamental mode.

Unit 6: Programmable Logic Devices (PLDs): PROMs, PLAs, PAL, Semiconductor memory: organization, Operation, and classification.

Textbooks:

1. M. Mano, Digital Design, 2nd Ed. PHI, 1997.
2. J.F. Wakerly, Digital Design-principles and practices, 3rd Ed, Pearson Education; 2001.

Reference Books:

3. M.D. Ercegovac, T. Lang and J.H. Moreno, Introduction to Digital Systems, John Wiley, 2000.
3. V.P.Nelson, H.T.Nagle, B.D. Carroll and J.D. Irwin, Digital Logic Circuit Analysis and Design, Prentice-Hall,1995.
4 D.D. Gajski; Principles of Digital Design, Prentice Hall, 1996.

Course Outcomes: At the end of this course, students will demonstrate the ability to

1. Understand working of logic families and logic gates.
2. Design and implement Combinational and Sequential logiccircuits.
3. Understand the process of Analog to Digital conversion and Digital to Analog conversion.
4. Be able to use PLDs to implement the given logical problem.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB311	Object Oriented Programming Lab	0	0	3	1.5

1. To write a C++ program to calculate income tax using default arguments.
2. To write a C++ program to categorize employees based on designation using static data members.
3. To write a C++ program to add two private data members using friend functions.
4. To write a C++ program to implement matrix vector multiplication using friend functions.
5. To write a C++ program to manipulate complex numbers using operator overloading and type conversions.
6 Write a program to implement inheritance feature .
7. To write a program to perform calculate student marks by overloading new and delete operators.
8. To write a program to develop a template for linked list class and its methods.
9. Write a program to implement ADT.
10. Write a program to show the use of access specifier.
11. Write a program to implement function overloading and overriding.
12. Write a program to implement memory management.

Teacher may add more programs based on any concept taught in object oriented programming.

The concepts taught in object oriented programming should be practiced using C++ and Java.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB312	Data Structures & Algorithm Lab	0	0	3	1.5

1. Write a C program to perform the following searching technique
   a. Sequential Search b. Binary Search
2. Write a C program to perform the following operation on singly linked list and double linked list
   a. insertion b. deletion c. searching
3. Write C programs to implement the Stack ADT and Queue ADT using an array.
4. Write C programs to implement the Stack ADT and Queue ADT using a singly linked list
5 Write a C program to perform the following operations:
   a. Insert an element into a binary search tree.
   b. Delete an element from a binary search tree.
   c. Search for a key element in a binary search tree.
7.Write a C program that use recursive functions to traverse the the given binary tree in
   a. Preorder b. Inorder c. Postorder
8. Write a C program that use non recursive functions to traverse the the given binary tree in
   a. Preorder b. Inorder c. Postorder
9. Write a program to implement the following technique used represent graph.
   a. Adjacency list b. Adjacency Matrix
10. Write a C program to perform the following sorting technique
   a. Selection Sort b. Bubble Sort c. Insertion Sort
11. Write a C program to perform the following operations on AVL-trees:
   a. Insertion
   b. Deletion.

Teacher may add more programs based on any concept taught in Data Structures & Algorithm .

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
EC3L02	Digital Electronics & Logic Design Lab	0	0	3	1.5

1. Simplification, realization of Boolean expressions using logic gates/Universal gates.

2. Realization of Half/Full adder and Half/Full Subtractors using logic gates.

3. (i) Realization of parallel adder/Subtractors using 7483 chip
   (ii) BCD to Excess-3 code conversion and vice versa.

4. Realization of Binary to Gray code conversion and vice versa

5.MUX/DEMUX – use of 74153, 74139 for arithmetic circuits and code converter.

6.Realization of One/Two bit comparator and study of 7485 magnitude comparator.

7. Use of a) Decoder chip to drive LED display and b) Priority encoder.

8.Truth table verification of Flip-Flops: (i) JK Master slave (ii) T type and (iii) D type.

9. Realization of 3 bit counters as a sequential circuit and MOD – N counter design (7476, 7490, 74192, 74193).

10. Shift left; Shift right, SIPO, SISO, PISO, PIPO operations using 74S95.

11. Wiring and testing Ring counter/Johnson counter.

12. Wiring and testing of Sequence generator.

Teacher may add more programs based on any concept taught in Digital Electronics & Logic Design .

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB401	Design & Analysis of Algorithms	3	0	0	3

Unit 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.

Unit 2: Fundamental Algorithmic Strategies: Brute-Force, Greedy,Dynamic Programming, Branchand-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.

Unit 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.

Unit 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.

Unit 5: Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE

Textbooks:

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.

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

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB402	Web & Internet	3	0	0	3

Unit-1: Basic of core Java: Data type, classes, objects, features of java, control statement, file handling, exception handling, packages, constructor, inheritance class, array, strings, access protection.

Unit 2: Working with User Interfaces – JAVA AWT Package, Window fundamentals, Basic User Interface Components (Labels, buttons, Check boxes, Radio buttons, choice Menu or Choice Lists, Text fields, Text areas, scrolling list, scroll bars, panels and frames), Layouts (Flow, Grid, Border, Card). Event-driven programming-Event driven programs, Event handling process, Java‟s event types. JAVA Swings- Comparison between Swing and AWT, Java swing packages, Swing basic containers, Swing components, event handling using Java swing, using dialogs, Joptionpane class, input dialog boxes, Timers and Sliders, Tables, Borders for components.

Unit 3: JAVA database connectivity- JDBC/ODBC bridge, JAVA.SQL package, Connecting to remote data base, Data manipulation and Data navigation JAVA Servlets – Introduction- Servlet API, Lifecycle of Java Servlet, Creating Servlets, Running servlets, Cookie class. Networking with java- Java.net package, Implementation of client-server application using TCP/IP and UDP.

Unit 4: Introduction to HTML- HTML tags, Frames and forms, Java Script- Introduction to scripting, control statements, Functions, Arrays, Objects. DHTML – Object model and Collections, Event model, Filters and Transitions, Data binding with tabular data control. XML – XML vocabularies, Document Object Model, SAX, Simple Object Access Protocol (SOAP), Extensible Style sheet Language(XSL)

Unit 5: Server side scripting Languages- JSP- Introduction to JSP, JSP Architecture, Scripting components, Standard actions, JSP with JDBC – case study of a simple online application. PHP – Introduction (variables, control statements etc), String processing and regular expression, Form processing and business logic, Connecting to a database, Cookies, Dynamic content in PHP-case study of an online application

Module 6: I/O – AWT – Event handling – Introduction to Threads - Basics of Networking – TCP and UDP sockets – Connecting to the Web

Textbooks:

1. Deitel & Deitel, JAVA : How to Program, Pearson education , 7e (2008) 2. Deitel & Deitel, Internet and World Wide Web How to Program, Pearson education, 3e ,(2005)

Reference Books:

1. Ivan BayRoss, Web Enabled Commercial Application using Java 2, bpb publication (1998)
2. David Flanagan , Java Script The Definitive Guide, O‟relly, 5e (2006)

Course Outcomes:

1. Students are able to develop a dynamic webpage by the use of java script and HTML
2. Students will be able to write a well formed / valid XML document.
3. Students will be able to write a server side java application called Servlet to catch update and delete operations on DBMS table.
4. Students will be able to write a server side java application called JSP to catch form form data sent from client, process it and store it on database. data sent from client and store it on database.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB403	Formal Language & Automata Theory	3	0	0	3

Unit 1: Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Unit 2: 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.

Unit 3: 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.

Unit 4: Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

Unit 5: Turing machines: The basic model for Turing machines (TM), Turingrecognizable (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.

Unit 6: Un-decidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Textbooks

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

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.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB404	Computer Organization & Architecture	4	0	0	4

Unit 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-and add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.
Unit 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
Unit 3: Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards. Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.
Unit 4: Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Textbooks:
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.

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

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
MAT4T2	Discrete Mathematics	3	1	0	4

Unit 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.

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

Unit 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.

Unit 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

Unit 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.

Textbooks:

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.

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.
3. Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson, 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.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB411	Design & Analysis of Algorithms Lab	0	0	3	1.5

1. Program for Merge sort using Divide and Conquer method.
2. Program for Strassen's Matrix multiplication using Divide and Conquer method.
3. Program for Binomial and Fibonacci heap.
4. Program for Matrix chain multiplication using Dynamic programming.
5. Program for Longest common subsequence (LCS) using Dynamic programming.
6. Program for 0/1 Knapsack problem using Dynamic programming.
7. Program for Activity-Selection problem using Greedy algorithm.
8. Program for Huffman coding using Greedy algorithm.
9. Program for Fractional Knapsack Problem using Greedy algorithm.
10. Program for BFS & DFS using graph.
11. Program based on quick sort and heap sort.
12. Program for MST using Prim's and Kruskal's algorithm.
13. Program for Bellman-Ford Algorithm using graph
14. Program for Floyd-Warshall Algorithm using graph.
15. Program for Maximum flow using Ford-Fulkerson algorithm.

Teacher may add more programs based on any concept taught in Design & Analysis of Algorithms.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB412	Web & Internet Lab	0	0	3	1.5

1. Java program with multiple classes
2. Java program to experiment control statements
3. Java program on application of classes and objects
4. Java program on application of constructor and destructor
5. Java program to define and illustrate the use of static members
6. Java program on application of single inheritance with super keyword
7. Java program for alphabetical ordering of strings using compareTo String method
8. Java program to sort a list of numbers using length of the array
9. Java program to implement sting manipulation using StringBuffer class
10. Java program to implement the use of wrapper class methods
11. Java program to implement working with vectors and arrays
12. Java program to implement the use and working of interfaces
13. Java program to implement the concept of multiple inheritance using interfaces
14. Java program to illustrate importing classes from other packages
15. Java program to implement try and catch for exception handling
16. Java program to implement multiple catch blocks and finally
17. Java program to throw own exception
18. Java program to implement Thread methods like yield(), stop(), sleep
19. Simple java applet code to display a simple message
20. Java Applet to display a rainbow with multiple colors
21. Java applet that implements ActionListener inside an HTML file
22. Java applet to illustrate interactive input to an applet within the HTML file
23. Java applet to implement graphics programming by drawing a human face
24. Java applet to implement the various drawing methods of the Graphics class
25. Java program to implement synchronization

Teacher may add more programs based on any concept taught in Web & Internet.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB413	Computer Organization & Architecture Lab	0	0	2	1

1. Carry look ahead adder
2. Registers and Counters
3. Wallace Tree Adder
4. Combinational Multipliers
5. Booth's Multiplier
6. Arithmetic Logic Unit
7. Memory Design
8. Associative cache Design
9. Direct Mapped cache Design
10. CPU Design

Teacher may add more programs based on any concept taught in Computer Organization & Architecture.

Reference: virtual lab http://cse10-iitkgp.virtual-labs.ac.in/

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB501	Graph Theory	3	0	0	3

Unit 1: Graphs, Sub graphs, some basic properties, various example of graphs & their sub graphs, walks, path & circuits, connected graphs, disconnected graphs and component, euler graphs, various operation on graphs, Hamiltonian paths and circuits the traveling sales man problem.

Unit 2: Trees and fundamental circuits, distance diameters, radius and pendent vertices, rooted and binary trees, on counting trees, spanning trees, fundamental circuits, finding all spanning trees of a graph and a weighted graph, algorithms of primes Kruskal and Dijkstra Algorithms.

Unit 3: Cuts sets and cut vertices, some properties, all cut sets in a graph, fundamental circuits and cut sets , connectivity and separability, network flows Planer graphs, combinatorial and geometric dual: Kuratowski graphs, detection of planarity, geometric dual, Discussion on criterion of planarity, thickness and crossings.

Unit 4: Vector space of a graph and vectors, basis vector, cut set vector, circuit vector, circuit and cut set subspaces, Matrix representation of graph – Basic concepts; Incidence matrix, Circuit matrix, Path matrix, Cut-set matrix and Adjacency matrix. Coloring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, matching, covering, four color problem

Unit 5: Discussion of Graph theoretic algorithm wherever required.

Textbooks :

1. Narsingh Deo, "Graph Theory: With Application to Engineering and Computer Science", Prentice Hall of India, 2003.
2. R.J. Wilson, "Introduction to Graph Theory", Fourth Edition, Pearson Education, 2003

Course Outcomes:

1. To learn the basic terminology and some of the theory associated with graphs.
2. To learn to model problems using graphs and to solve these problems algorithmically.
3. Modern applications of graph theory will be explored.

School of Engineering &Technology

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB502	Operating System	3	0	0	3

Unit 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.

Unit 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.

Unit 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.

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

Unit 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).

Unit 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

Textbooks:

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

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.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB503	Database Management Systems	3	0	0	3

Unit 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.
Unit 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.
Unit 3: Storage strategies: Indices, B-trees, hashing.
Unit 4: Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.
Unit 5: Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.
Unit 6: Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

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

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.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB504	Software Engineering	3	0	0	3

UNIT-1: Introduction : Software crisis, software processes & characteristics, software life cycle models, waterfall, prototype, evolutionary and spiral models.
Software project planning : size estimation like lines of code & function count, cost estimation models, COCOMO, COCOMO-II, Putnam resource allocation model, risk management.
UNIT-2: Requirements Engineering: Requirements Engineering, Initiating the process, Eliciting Requirements, Building the Requirements Model, Negotiating, Validating requirements, Requirements Analysis, Scenario-Based Analysis, Requirements Modeling strategies, Flow-Oriented Modeling, Class based modeling, requirement analysis using DFD, Data dictionaries & ER diagrams, requirements documentation SRS, , nature of SRS, characteristics & organization of SRS.
UNIT-3: Design Process, Design Concepts, The Design Model: Data Design, Architectural, interface Design Elements.
Architectural Design: Software Architecture, Architectural Styles, Architectural Design, User Interface Design: Rules, User Interface Analysis and Design, Applying Interface Design Steps, Issues, Web App Interface Design Principles
UNIT-4: Software testing : Testing process, design of test cases, functional testing: Boundary value analysis, equivalence class testing, decision table testing, cause effect graphing, structural testing, path testing, data flow and mutation testing, unit testing, integration and system testing, debugging, alpha & beta testing, testing tools & standards.
UNIT-5: Software metrics: software measurements : What & Why , token count , halstead software science measures, design metrics, data structure metrics, information flow metrics.
Software reliability : importance, hardware reliability & software reliability, failure and faults, reliability models, basic model, logarithmic poisson model, software quality models, CMM & ISO 9001.
UNIT-6: Software maintenance: Management of maintenance, maintenance process, maintenance models, regression testing, reverse engineering, software re-engineering, configuration management, documentation.

Textbooks:
1. K.K.Aggarwal & Yogesh Singh, " Software engineering ", 2nd Ed. , New age international,2005.
2. R.S. Pressman, " Software engineering- A practioner's approach ", 5th Ed., Mcgraw Hill Int. Ed., 2001.
3. Pressman R., "Software Engineering, A Practitioners Approach", 7th Edition, Tata MCGraw Hill Publication, 2010, ISBN 978-007-126782-3

Reference books:
1. Stephen R. Schach, " classical & object oriented software engineering ", IRWIN,TMH,1996.
2. James peter, W. Pedrycz, " Software engineering : An engineering approach ", John Wiley & Sons, 2004.
3. I. Sommerville, " Software engineering ", Addision Wesley, 2004.
4. K. Chandrasehakhar, " Software engineering & quality assurance ",BPB,2005.

Course Outcomes:
1. Acquire strong fundamental knowledge in science, mathematics, fundamentals of computer science, software engineering and multidisciplinary engineering to begin in practice as a software engineer.
2. Design applicable solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal and economic concerns.
3. Deliver quality software products by possessing the leadership skills as an individual or contributing to the team development and demonstrating effective and modern working strategies by applying both communication and negotiation management skill.
4. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
MAT5T1	Numerical Analysis & Probability	3	1	0	4

Unit 1: Basic Probability: Probability spaces, conditional probability, independence; Discrete random variables, Independent random variables, the multinomial distribution, Poisson approximation to the binomial distribution, infinite sequences of Bernoulli trials, sums of independent random variables; Expectation of Discrete Random Variables, Moments, Variance of a sum, Correlation coefficient, Chebyshev's Inequality

Continuous Probability Distributions: Continuous random varibales and their properties, distribution functions and densities, normal, exponential and gamma densities.

Unit 2: Bivariate Distributions: Bivariate distributions and their properties, distribution of sums and quotients, conditional densities, Bayes' rule.

Unit 3: Introduction: Numbers and their accuracy, Computer Arithmetic, Mathematical preliminaries, errors and their computation, General error formula, Error in series approximation.

Solution of Algebraic and Transcendental Equation:

Bisection Method, Iteration method, Method of false position, Newton-Raphson method, Methods of finding complex roots, Muller's method, Rate of convergence of Iterative methods, polynomial Equations.

Unit 4: Interpolation: Finite Differences, Difference Tables

Polynomial Interpolation: Newton's forward and backward formula.

Cental Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula.

Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation.

Unit 5: Numerical Integration and Differentiation: Introduction, Numerical differentiation Numerical Integration: Trapezoidal rule, Simpson's 1/3 and 3/8 rule, Boole's rule, Waddle's rule.

Unit 6: Solution of differential equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta Methods, Predictor Corrector Methods, Automatic Error Monitoring and Stability of solution.

Unit 7: Statistical Computation : Frequency chart, Curve fitting by method by least squares fitting of straight lines, polynomials, exponential curves etc, Data fitting with Cubic splines, Regression Analysis, Linear and Non linear Regression, Multiple regression, statistical Quality Control methods.

Textbooks/References books:

1. P. Kandasamy, K. Thilagavathy, K. Gunavathi, Numerical Methods, S. Chand & Company, 2nd Edition, Reprint 2012.
2. S.S. Sastry, Introductory methods of numerical analysis, PHI, 4th Edition, 2005.
3. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint).
5. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.

Course Outcomes:

Students will be able to

1. To solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration by programming
2. To solve ordinary and partial differential equations using programming languages like C and softwares like MATLAB.

Course Code	*Course name*	**Lectures**	**Tutorials**	**Practicals**	**Total Credits**
CSB511	Software Engineering Lab	0	0	3	1.5

1. Identifying the Requirements from Problem Statements

Requirements | Characteristics of Requirements | Categorization of Requirements | Functional Requirements | Identifying Functional Requirements

2. Estimation of Project Metrics

Project Estimation Techniques | COCOMO | Basic COCOMO Model | Intermediate COCOMO Model | Complete COCOMO Model | Advantages of COCOMO | Drawbacks of COCOMO | Halstead's Complexity Metrics

3. Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

Use case diagrams | Actor | Use Case | Subject | Graphical Representation | Association between Actors and Use Cases | Use Case Relationships | Include Relationship | Extend Relationship | Generalization Relationship | Identifying Actors | Identifying Use cases | Guidelines for drawing Use Case diagrams

4. E-R Modeling from the Problem Statements

Entity Relationship Model | Entity Set and Relationship Set | Attributes of Entity | Keys | Weak Entity | Entity Generalization and Specialization | Mapping Cardinalities | ER Diagram | Graphical Notations for ER Diagram | Importance of ER modeling

5. Identifying Domain Classes from the Problem Statements

Domain Class | Traditional Techniques for Identification of Classes | Grammatical Approach Using Nouns | Advantages | Disadvantages | Using Generalization | Using Subclasses | Steps to Identify Domain Classes from Problem Statement | Advanced Concepts

6. Statechart and Activity Modeling

Statechart Diagrams | Building Blocks of a Statechart Diagram | State | Transition | Action | Guidelines for drawing Statechart Diagrams | Activity Diagrams | Components of an Activity Diagram | Activity | Flow | Decision | Merge | Fork | Join | Note | Partition | A Simple Example | Guidelines for drawing an Activity Diagram

7. Modeling UML Class Diagrams and Sequence diagrams

Structural and Behavioral aspects | Class diagram | Elements in class diagram | Class | Relationships | Sequence diagram | Elements in sequence diagram | Object | Life-line bar | Messages

8. Modeling Data Flow Diagrams

Data Flow Diagram | Graphical notations for Data Flow Diagram | Explanation of Symbols used in DFD | Context diagram and leveling DFD

9. Estimation of Test Coverage Metrics and Structural Complexity

Control Flow Graph | Terminologies | McCabe's Cyclomatic Complexity | Computing Cyclomatic Complexity | Optimum Value of Cyclomatic Complexity | Merits | Demerits

10. Designing Test Suites

Software Testing | Standards for Software Test Documentation | Testing Frameworks | Need for Software Testing | Test Cases and Test Suite | Types of Software Testing | Unit Testing | Integration Testing | System Testing | Example |

Reference: http://vlabs.iitkgp.ernet.in/se/

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB512	Operating System Lab	0	1	3	2.5

1. Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close, stat, opendir, readdir.
2. Write programs using the I/O system calls of UNIX operating system (open, read, write, etc)
3. Write C programs to simulate UNIX commands like ls, grep, etc.
4. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)
6. Developing Application using Inter Process communication (using shared memory, pipes or message queues)
7. Implement the Producer – Consumer problem using semaphores (using UNIX system calls).
8. Implement some memory management schemes – I
9. Implement some memory management schemes – II
10. Implement any file allocation technique (Linked, Indexed or Contiguous)
11. Write a C/C++ script to display all logged in users
12. C++ program to identify the available memory in the system.
13. Write a program to implement producer consumer problem using semaphore.

Teacher may add more programs based on any concept taught in Operating System.

NAGALAND UNIVERSITY

(A Central University Established by an Act of Parliament 1989)

School of Engineering &Technology

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB513	Database Management Systems Lab	0	0	3	1.5

1.Consider the insurance database given below. The primary keys are made bold and the data types are specified.
PERSON(driver_id:string , name:string , address:string) CAR(regno:string , model:string , year:int)
ACCIDENT(report_number:int , accd_date:date , location:string) OWNS(driver_id:string , regno:string)
PARTICIPATED(driver_id:string , regno:string , report_number:int , damage_amount:int)
1) Create the above tables by properly specifying the primary keys and foreign keys.
2) Enter at least five tuples for each relation.
3) Demonstrate how you
a.Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.
b.Add a new accident to the database.
4)Find the total number of people who owned cars that were involved in accidents in the year 2008.
5)Find the number of accidents in which cars belonging to a specific model were involved.

2.Consider the following relations for a order processing database application in a company.
CUSTOMER(custno:int , cname:string , city:string) ORDER(orderno:int , odate:date , custno:int , ord_amt:int)
ORDER_ITEM(orderno:int , itemno:int , quantity:int) ITEM(itemno:int , unitprice:int)
SHIPMENT(orderno:int , warehouseno:int , ship_date:date) WAREHOUSE(warehouseno:int , city:string)
1) Create the above tables by properly specifying the primary keys and foreign keys.
2) Enter at least five tuples for each relation.
3) Produce a listing: custname , No_of_orders , Avg_order_amount , where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.
4) List the orderno for orders that were shipped from all the warehouses that the company has in a specific city.
5) Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contains this particular item.

3.Consider the following database of student enrollment in courses and books adopted for that course.
STUDENT(regno:string , name:string , major:string , bdate:date) COURSE(courseno:int , cname:string , dept:string)
ENROLL(regno:string , courseno:int , sem:int , marks:int) BOOK_ADOPTION(courseno:int , sem:int , book_isbn:int)
TEXT(book_isbn:int , book_title:string , publisher:string , author:string)
1) Create the above tables by properly specifying the primary keys and foreign keys.
2) Enter atleast five tuples for each relation.
3) Demonstrate how you add a new text book to the database and make this book to be adopted by some department.
4) Produce a list of text books (includes courseno , book_isbn , book_title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
5)List any department that has all its books published by a specific publisher.

4. The following are maintained by abook dealer.
AUTHOR(author_id:int , name:string , city:string , country:string) PUBLISHER(publisher_id:int , name:string , city:string , country:string) CATALOG(book_id:int , title:string , author_id:int , publisher_id:int , category_id:int , year:int , price:int)
CATEGORY(category_id:int , description:string) ORDER_DETAILS(order_no:int , book_id:int , quantity:int)
1)Create the above tables by properly specifying the primary keys and foreign keys.
2)Enter at least five tuples for each relation.
3)Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
4)Find the author of the book that has maximum sales.
5)Demonstrate how you increase the price of books published by a specific publisher by 10%.

5.Consider the following database for a banking enterprise.
BRANCH(branch_name:string , branch_city:string , assets:real) ACCOUNT(accno:int , branch_name:string , balance:real)
DEPOSITOR(customer_name:string , accno:int) CUSTOMER(customer_name:string , customer_street:string , customer_city:string)
LOAN(loan_number:int , branch_name:string , amount:real) BORROWER(customer_name:string , loan_number:int)
1)Create the above tables by properly specifying the primary keys and foreign keys.
2)Enter at least five tuples for each relation.
3)Find all the customers who have at least two accounts at the main branch.
4)Find all the customers who have an account at all the branches located in a specific city.
5)Demonstrate how you delete all account tuples at every branch located in a specific city.

6. Write a program to store data into database from web page and display data on web page from database using PHP, HTML, CSS, Apache and MYSQL.

Teacher may add more programs based on any concept taught in Database Management Systems.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB601	Compiler Design	4	0	0	4

Unit 1: 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
Unit 2: 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)
Unit 3: 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.
Unit 4: Intermediate Code Generation: Translation of different language features, different types of intermediate forms. Code Improvement (optimization):
Unit 5: 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
Unit 6: Advanced topics: Type systems, data abstraction, compilation of Object Oriented features and non-imperative programming languages.

Textbooks:
1. Tremblay, et al., "The Theory and practice of compiler Writing", McGraw Hill, New York,1985.
2. A. Holub, "Compiler Design in C",PHI, 2004.
3. Aho, Ullman & Ravi Sethi, " Principles of Compiler Design", Pearson Education, 2002.

Reference books:
1. Andrew L. Appel, " Modern Compiler Implementation in C", Delhi, Foundation Books,2000.
2. Dick Grune et. Al., " Modern Compiler Design ", Wiley Dreamtech, 2000.

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

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB602	Computer Networks	3	0	0	3

Unit 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.

Unit 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

Unit 3: Network Layer: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols.

Unit 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.

Unit 5: Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography

Textbooks:

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.

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.

NAGALAND UNIVERSITY

(A Central University Established by an Act of Parliament 1989)

School of Engineering &Technology

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
HSB601	Project Management & Entrepreneurship	2	0	0	2

Unit 1: Project Planning: Software pricing, Plan driven development, project scheduling, Agile Planning, Estimation Technique: COCOMO model. Quality Management: Review and inspection, software measurement and matrics Concept of Quality, Control Charts, Latest Trend in Quality Management, TQM, ISO 9000 Series Maintenance Management: Différent types, Latest Trend in Maintenance Management, TPM Project Management: Definition of Project, Network Analysis, PERT and CPM Inventory Management: Concept, EOQ Model Safety in Workplace, Fire and Safety.

Unit 2: Risk Management: type of risk; Risk identification, risk analysis, risk planning and risk monitoring. People Management: Motivating people, human needs hierarchy, People Capability Maturity Model(P-CMM), Con; Motivation Theories; Leaderships, its Qualities and Styles.

Unit 3: Team Management: Selecting group member, group organization, Group Communication, Types of groups; Group Behaviour and Group role; Group Decision-Making; power and politics; Conflicts and types of conflicts; Sources of Conflict; Managing Conflict, Emotional Intelligence. Configuration management: change management, Version management, system building, release management. Process Improve management**:** process improvement process, process measurement, process analysis, process change, The CMMI process improvement framework.

UNIT 4: Entrepreneur, Entrepreneurship & Intrapreneurship**:** Definitions, importance, characteristics, Difference between entrepreneur and manager; Factors affecting entrepreneurial Growth, Qualities and Role of entrepreneurship in economic development; Concept of Intrapreneurship; McClelland Model

UNIT 5: Women and Rural Entrepreneurship: Role & Importance of women entrepreneurship, Problems of Women Entrepreneurs, women entrepreneurship in India; Rural Entrepreneurship, need and problems. Agri-Preneurship & Social Entrepreneurship: Agri-Preneurship-Meaning-Need-Opportunities-challenges, suggestions for development for agri-preneurship. Social entrepreneurship-meaning-the perspective of social entrepreneurship-Social entrepreneurship in practice.

UNIT 6: Project Formulation: Elements of project formulation; feasibility Analysis-Economic feasibility, financial feasibility, Technological Feasibility; Social Cost-benefit analysis; preparation of feasibility report**.** Entrepreneurial Development Agencies**:** Role of Government and supporting agencies/ institutions such as IDBI, District Industries Centre (DIC), Small Industries Development Organizations (SIDO), Small Institutions Service Institutions (SISI), Commercial banks EDI etc.

Text Books:

1. Ian sommerville, Software Engineering, Pearson
2. Koontez, Odoneel: Essentials of Management, Tata McGraw Hill
3. Vasant Desai*: Dynamics of Entrepreneurship Development*.
4. David H. Holt*: Entrepreneurship: New Venture Creation*
5. Poornima M Charantimath : *Entrepreneurship Development & Small Business Enterprises,* Pearson
6. S.SKhanka: *Entrepreneurial Development*: S.Chand

Course Outcomes:

1. Manage the selection and initiation of individual projects and of portfolios of projects in the enterprise.
2. Conduct project planning activities that accurately forecast project costs, timelines, and quality. Implement processes for successful resource, communication, and risk and change management.
3. Demonstrate effective project execution and control techniques that result in successful projects.
4. Demonstrate effective organizational leadership and change skills for managing projects, project teams, and stakeholders
5. To encourage students to take up entrepreneurship as a career option and to impart skills related to it.
6. Provide knowledge about various theories related to entrepreneur and entrepreneurship.
7. Entrepreneurship programme provides knowledge and skills on how to successfully develop captivating products and services to solve challenging problems in a highly uncertain environment, often under considerable time constraints with very limited resources. You will be able to apply these skills in the context of both new ventures as well as in established companies.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB611	Computer Networks Lab	0	0	3	1.5

1. To give IP Address of different classes in given Network id and Subnet
2. To Construct LAN with the help of various topology like Star Topology and etc.
3. To practise the colour code for different cables and Observe the LAN Tester and make the decision accordingly.
4. Trace the route that is taken when you try to access: a) google.com Record the number of hops required for accessing each of the URLs. Also provide an ordered list of the geographical locations corresponding to the machines through which the packets travelled in each case.
5. Packet tracing and analysis for TCP/UDP connection between client and server using Wireshark software
6. Conversion URL name into IP address using DNS with the help of wireshark
7. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP)
HTTP, SNMP, Bluetooth, Firewalls and DHCP using open source available software and tools.
8. client server applications using cocurrent server
9. Implement a chat and mail server
10. Write a simple program for both client and server, in which client will send "How are u" message to server and server will reply "I am fine" using Socket Programming with connection oriented protocol TCP.
12. Write a simple program for both client and server, in which client will send "How are u" message to server and server will reply "I am fine" using Socket Programming with connectionless oriented protocol UDP.
13. Modify the client/server programs to allow exchange of multiple messages.
14. Simulation of network routing algorithm

Teacher may add more programs based on any concept taught in Computer Networks.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB612	Compiler Design Lab	0	0	3	1.5

1. Write a lex Program to identify a simple and a compound statement.
2. Write a lex Program to count the number of keywords and identifiers in a sentence.
3. Imagine the syntax of a programming language construct such as *while-loop* --

while (*condition*)
begin
    statement ;
        :
end

where *while, begin, end* are keywords; *condition* can be a single comparision expression (such as *x == 20*, etc.); and *statement* is the assignment to a location the result of a single arithmatic operation (eg., *a = 10 * b*).

Write a program that verifies whether the input follows the above syntax. Use *flex* to create the lexical analyser module and write a C program that performs the required task using that lexical analyser

4. Write a lexical analyser for the C programming language using the gramar for the language given in the book "The C Programming Language", 2e, by B Kernighan and D Ritchie.
5. Write a YACC Program to check whether given string a^nb^n is accepted by the grammar
6. Write a YACC program to check the validity of an arithmetic expression.
7. Write a YACC Program to identify an input for the grammar a^nb (n>=10)
8. Write a C program for implementing *shift-reduce* parsing using a given grammar. Firstly, define the data structures for representing the given CFG in BNF, the stack for the parsing, and the parse tree to be created.
9. Take a common programming language construct of an HLL, such as the for-loop construct of the C language. Use LEX and YACC to create a translator that would translate input into three-address intermediate code. The output of the translator should finally be in a file. Assume a simple structure for "statements" that may appear inside the construct, and make necessary assumptions for the intermediate code format.

Teacher may add more programs based on any concept taught in Compiler Design.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB613	Application Programming Lab	0	1	3	2.5

1. Write a program to print hello world using scripting language.

2. Write a program to show the use of variables, control statements, arrays, strings, operators, functions using scripting language.

3. Write to program to create FORM with validation operation using DHTML.

NOTE: Dynamic HTML(DHTML) is a collection of technologies used together to create interactive and animated websites by using a combination of a static markup language (such as HTML), a client-side scripting language (such as JavaScript), a presentation definition language (such as CSS), and the Document Object Model (DOM).

4. Write a program to store data into database server and retrieve data from database server using scripting language. Data should be collected from FORM page and user must enter those data.

5. Write a program to store images into database/file system and retrieve data from database/file system using scripting language. User should be able to upload file from graphical user interface(FORM).

6. Write a program to perform update operation on data stored in database using scripting language. User should be able to update data from graphical user interface(FORM).

7. Write a program to show the use of sessions, cookies, Error handling using scripting language. Web pages may be created using HTML.

8. Develop responsive website using appropriate language.
   Note: Web pages can be viewed using many different devices: desktops, tablets, and phones. It is called **responsive** web design when you use CSS and **HTML** to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.
9. Write a Android program to print hello world.
10. Write a Android program for general activity like phone call, messaging, email, notifications, check box, finger prints, touch button, sensor, drag & drop, audio, video, camera, clipboard(copy & paste), images, login, navigation, session, spelling check and integration.
11. Develop an android app to work with database locally.
12. Develop an android app to fetch data from web server and store data into web server.
13. Write a program based on the concept of Internet-of-Things.

Teacher may add more programs based on any concept related to recent trends in application programming.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB701	Distributed System	3	0	0	3

Unit 1: Characterization of Distributed Systems: Introduction, Examples of distributed Systems, Issues in Distributes Operating Systems, Resource sharing and the Web Challenges. System Models: Architectural models, Fundamental Models Theoretical Foundation for Distributed System: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport‟s & vectors logical clocks, Causal ordering of messages, global state, termination detection. Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non token based algorithms, performance metric for distributed mutual exclusion algorithms.

Unit 2: Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms. Agreement Protocols: Introduction, System models, classification of Agreement ProblemInteractive consistency Problem, Applications of Agreement algorithms.

Unit 3: Distributed Objects and Remote Invocation: Communication between distributed objects, Remote procedure call, Events and notifications, Java RMI case study. Transactions and Concurrency Control: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control

Unit 4: Distributed Transactions: Introduction, Flat and nested distributed transactions, Atomic commit protocols, concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Distributed shared memory – Design and Implementation issues, consistency models, CORBA Case Study: CORBA RMI, CORBA services.

Unit 5: File service components, design issues, interfaces, implementation techniques, Sun Network File System – architecture and implementation, other distributed file systems – AFS, CODA. Name services – SNS name service model.

Text Books:

1. "Advanced Concepts in Operating Systems", by Mukesh Singhal & Niranjan G Shivaratri,Tata McGraw Hill (2001).
2. “Distributed System: Concepts and Design”, by Coulouris, Dollimore, Kindberg , Pearson Education (2006)

Reference Books:

1.Tanenbaum S , “Distributed Operating Systems”, Pearson Education (2005). 2. P K Sinha, “"Distributed System: Concepts and Design”, PHI (2004).

Course Outcomes:

1. To demonstrate knowledge of the basic elements and concepts related to distributed system technologies;
2. To demonstrate knowledge of the core architectural aspects of distributed systems;
3. To Design and implement distributed applications;
4. To Demonstrate knowledge of details the main underlying components of distributed systems (such as RPC, file systems);
5. To use and apply important methods in distributed systems to support scalability and fault tolerance;
6. To demonstrate experience in building large-scale distributed applications.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB702	Machine Learning	3	0	0	3

Unit 1: Supervised Learning (Regression/Classification)
- Basic methods: Distance-based methods, Nearest-Neighbours, Decision
- Trees, Naive Bayes
- Linear models: Linear Regression, Logistic Regression, Generalized Linear Models
-Support Vector Machines, Nonlinearity and Kernel Methods
-Beyond Binary Classification: Multi-class/Structured Outputs, Ranking

Unit 2: Unsupervised Learning Clustering:
-K-means/Kernel K-means
-Dimensionality Reduction: PCA and kernel PCA
-Matrix Factorization and Matrix Completion
-Generative Models (mixture models and latent factor models)

Unit 3: Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, Random Forests)
Unit 4: Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning
Unit 5: Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference
Unit 6: Recent trends in various learning techniques of machine learning and classification methods for IOT applications. Various models for IOT applications.

Reference books:
1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online)
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

Course Outcomes: After completion of course, students would be able to:
1. Extract features that can be used for a particular machine learning approach in various IOT applications.
2. To compare and contrast pros and cons of various machine learning techniques and to get an insight of when to apply a particular machine learning approach.
3. To mathematically analyse various machine learning approaches and paradigms.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB712	Machine Learning Lab	0	0	3	1.5

1. Write a program to implement and demonstrate the **FIND-S algorithm** for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the **Candidate-Elimination algorithm** to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based **ID3 algorithm**. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the **Back propagation algorithm** and test the same using appropriate data sets.
5. Write a program to implement the **naïve Bayesian classifier** for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the **naïve Bayesian Classifier** model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a **Bayesian network** considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply **EM algorithm** to cluster a set of data stored in a .CSV file. Use the same data set for clustering using ***k*-Means algorithm**. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement ***k*-Nearest Neighbour algorithm** to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric **Locally Weighted Regression algorithm** in order to fit data points. Select appropriate data set for your experiment and draw graphs.

The programs can be implemented in either JAVA, R or Python.

Teacher may add more programs based on any concept taught in Machine Learning.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
G8T01	Constitution of India	3	0	0	0 (Non Credit)

The Constitution of India is the supreme law of India. Parliament of India cannot make any law which violates the Fundamental Rights enumerated under the Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368, however, it cannot use this power to change the "basic structure" of the constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments. The Constitution of India reflects the idea of "Constitutionalism" – a modern and progressive concept historically developed by the thinkers of "liberalism" – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state. The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of "constitutionalism" in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India's legacy of "diversity". It has been said that Indian constitution reflects ideals of its freedom movement, however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values. No law can be "static" and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950. The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it "as one of the strongest court in the world".

Course content

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India
4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India
9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
11. Emergency Provisions : National Emergency, President Rule, Financial Emergency
12. Local Self Government – Constitutional Scheme in India
13. Scheme of the Fundamental Right to Equality
14. Scheme of the Fundamental Right to certain Freedom under Article 19
15. Scope of the Right to Life and Personal Liberty under Article 21

Textbooks:

1. Introduction of Constitution by Durga Das Basu
2. The Indian Constitution: Cornerstone of A Nation by Austin Granville

Course Outcomes:

Enhance knowledge about the salient features of the constitution of India, the fundamental rights of citizen of India, role of constitution in a democratic society and the need of constitution.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
HSB801	Human Relations at work	2	0	0	2

Unit 1: Understanding and Managing Yourself: Human Relations and You: Self-Esteem and Self-Confidence: Self-Motivation and Goal Setting; Emotional Intelligence, Attitudes, and Happiness; Values and Ethics and Problem Solving and Creativity.
Unit 2: Dealing Effectively with People: Communication in the Workplace; Specialized Tactics for Getting Along with Others in the Workplace; Managing Conflict; Becoming an Effective Leader; Motivating Others and Developing Teamwork; Diversity and Cross-Cultural Competence.
Unit 3: Staying Physically Healthy: Yoga, Pranayam and Exercise: Aerobic and anaerobic.
Unit 4: Staying Psychologically Healthy: Managing Stress and Personal Problems, Meditation.
Unit 5: Developing Career Thrust: Getting Ahead in Your Career, Learning Strategies, Perception, Life Span Changes, Developing Good Work Habits.

Textbooks:
1. Dubrien, A. J. (2017). Human Relations for Career and Personal Success: Concepts, Applications, and Skills, 11th Ed. Upper Saddle River, NJ: Pearson.

Reference books:
1. Greenberg, J. S. (2017). Comprehensive stress management (14th edition). New York: McGraw Hill.
2. Udai, Y. (2015). Yogasan aur pranayam. New Delhi: N.S. Publications.

Course Outcomes:
1. Students will develop skills required to interact with people at work.
2. Students will develop psychological and physical health in maintaining human relations at work and progressing in career.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB621	Data Mining	3	0	0	3

Prerequisites: Linear Algebra and Probability, Database Management Systems

Unit l: Introduction: Basic concepts of data mining, including motivation and definition; different types of data repositories; data mining functionalities; concept of interesting patterns; data mining tasks; current trends, major issues and ethics in data mining

Unit 2: Data: Types of data and data quality; Data Preprocessing: data cleaning, data integration and transformation, data reduction, discretization and concept hierarchy generation; Exploring Data: summary statistics, visualization, multidimensional data analysis

Unit 3: Association and Correlation Analysis: Basic concepts: frequent patterns, association rules - support and confidence; Frequent itemset generation - Apriori algorithm, FP-Growth algorithm; Rule generation, Applications of Association rules; Correlation analysis.

Unit 4: Clustering Algorithms and Cluster Analysis: Concept of clustering, measures of similarity, Clustering algorithms: Partitioning methods - k-means and k-medoids, CLARANS, Hierarchical methods - agglomerative and divisive clustering, BIRCH, Densitybased methods - Subspace clustering, DBSCAN; Graph-based clustering - MST clustering; Cluster evaluation; Outlier detection and analysis.

Unit 5: Classification: Binary Classification - Basic concepts, Bayes theorem and Naive Bayes classifier, Association based classification, Rule based classifiers, Nearest neighbour classifiers, Decision Trees, Random Forest; Perceptrons; Multi-category classification; Model overfitting, Evaluation of classifier performance - cross validation, ROC curves.

Unit 6: Applications: Text mining, Web data analysis, Recommender systems.

Textbooks:

1. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Introduction to Data Mining. Pearson (2005), India. ISBN 978-8131714720
2. Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 3rd edition (July 2011). 744 pages. ISBN 978-0123814791
3. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 3rd edition (January 2011). 664 pages. ISBN 978-0123748560.

Reference books:

1. T. Hastie, R. Tibshirani and J. H. Friedman, The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Springer, 2nd Edition, 2009. 768 pages. ISBN 978-0387848570
2. C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 1st edition, 2006. 738 pages. ISBN 978-0387310732

Course Outcomes: After completion of course, students would be:

1. Study of different sequential pattern algorithms
2. Study the technique to extract patterns from time series data and it application in real world.
3. Can extend the Graph mining algorithms to Web mining
4. Help in identifying the computing framework for Big Data

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB622	Data Analytics	3	0	0	3

Pre- Requisites: Linear algebra, and calculus, probability theory, statistics, and programming

Unit 1: Descriptive Statistics : Introduction to the Data Analytics ,Descriptive Statistics, Probability Distributions
Inferential Statistics: Inferential Statistics through hypothesis tests, Permutation & Randomization Test

Unit 2: Regression & ANOVA: Regression, ANOVA(Analysis of Variance)
Machine Learning: Introduction and Concepts, Differentiating algorithmic and model based frameworks, Regression : Ordinary Least Squares, Ridge Regression, Lasso Regression, K Nearest Neighbours Regression & Classification

Unit 3: Supervised Learning with Regression and Classification techniques -1: Bias-Variance Dichotomy, Model Validation Approaches, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Regression and Classification Trees, Support Vector Machines
Supervised Learning with Regression and Classification techniques -2: Ensemble Methods: Random Forest, Neural Networks, Deep learning

Unit 4: Unsupervised Learning and Challenges for Big Data Analytics: Clustering, Associative Rule Mining, Challenges for big data anlalytics
Prescriptive analytics: Creating data for analytics through designed experiments, Creating data for analytics through Active learning, Creating data for analytics through Reinforcement learning

Reference books:

1. Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: springer, 2009.
2. Montgomery, Douglas C., and George C. Runger. Applied statistics and probability for engineers. John Wiley & Sons, 2010

Course Outcomes: After completion of course, students would be:

1. Explain NoSQL big data management
2. Install, configure, and run Hadoop and HDFS
3. Perform map-reduce analytics using Hadoop
4. Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics

Reference https://onlinecourses.nptel.ac.in/noc17_mg24/preview

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB623	Information Retrieval	3	0	0	3

Unit 1: Information retrieval model, Information retrieval evaluation, Searching the Web
Unit 2: Document Representation, Query languages and query operation, Meta-data search,
Unit 3: Indexing and searching, Scoring and ranking feature vectors,
Unit 4: Ontology, domain specific search, parallel and distributed information retrieval,
Unit 5: Text and multimedia languages, Social networks.
Unit 6: Recent trends in Web search and Information retrieval techniques.

Reference books:

1. C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008 (available at http://nlp.stanford.edu/IR-book).
2. Chakrabarti, S. (2002). Mining the web: Mining the Web: Discovering knowledge from hypertext data. Morgan-kaufman.
3. B. Croft, D. Metzler, T. Strohman, Search Engines: Information Retrieval in Practice, AddisonWesley, 2009 (available at http://ciir.cs.umass.edu/irbook/).
4. R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 2011 (2nd Edition).

Course Outcomes: After completion of course, students would be:
1. To identify basic theories and analysis tools as they apply to information retrieval.
2 To develop understanding of problems and potentials of current IR systems.
3 To learn and appreciate different retrieval algorithms and systems.
4 To apply various indexing, matching, organizing, and evaluating methods to IR problem.
5 To become aware of current experimental and theoretical IR research.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB624	Multimedia Technology	3	0	0	3

Prerequisites: Computer Graphics & Virtual Reality

Unit 1: Introductory Concepts: Multimedia-Definitions, CD-ROM and the Multimedia Highway, Uses of Multimedia, Introduction to making multimedia- The Stages of project, the requirements to make good multimedia, Multimedia skills and training, Training opportunities in Multimedia. Motivation for multimedia usage, Frequency domain analysis, Application Domain & ODA etc.

Unit 2: Multimedia-Hardware and Software: Multimedia Hardware- Macintosh and Windows production Platforms, Hardware peripherals – Connections, Memory and storage devices, Media software- Basic tools, making instant multimedia, Multimedia software and Authoring tools, Production Standards.

Unit 3: Multimedia- making it work- multimedia building blocks- Text, Sound, Images, Animation and Video, Digitization of Audio and Video objects, Data Compression: Different Compression algorithms concern to text, audio, video and images etc., Working Exposure on Tools like Dream Weaver, 3D Effects, Flash etc.

Unit 4: Multimedia and Internet: History, Internet working, Connections, Internet Services, The world Wide Web, Tools for the WWW- web Servers, Web Browsers, Web Page makers and editors, Plug-Ins and Delivery Vehicles, HTML , VRLM, Designing for the WWW-Working on the web , Multimedia Applications- Media Communication, Media Consumption, Media Entertainment, Media games.

Unit 5: Multimedia-looking towards future: Digital Communication and New Media, Interactive Television, Digital Broadcasting, Digital Radio, Multimedia Conferencing,Assembling and delivering a project-planning and costing, Designing and Producing, content and talent, Delivering , CD-ROM technology.

Textbooks:

1. Steve Heath,"Multimedia & Communication Systems",Focal Press,UK,1999.
2. TAy Vaughan,"Multimedia Making it Work", TMH,1999.
3. K. Andleigh And K. Thakkar,"Multimedia System Design",PHI, PTR,2000.

Reference books:

1. Keyes,"Multimedia Handbook",TMH,2000.
2. Ralf Steinmetz and Klara Naharstedt,"Multimedia: Computing, Communication & Applications",Pearson,2001.
3. Steve Rimmer,"Advanced Multimedia Programming", MHI,2000.

Course Outcomes: After completion of course, students would be:

1. To identify a range of concepts, techniques and tools for creating and editing the interactive multimedia applications.
2. To identify the current and future issues related to multimedia technology.
3. To identify both theoretical and practical aspects in designing multimedia systems surrounding the emergence of multimedia technologies using contemporary hardware and software technologies.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB625	Software Testing	3	0	0	3

Prerequisites: Software Engineering, Data Structures & Algorithm

UNIT 1: Introduction: What is software testing and why it is so hard? Error, Fault, Failure, Incident, Test Cases, Testing Process, Limitation of Testing, No absolute proof of correctness, Overview of Graph Theory.

UNIT 2: Introduction, Need of black box testing, Black box testing Concept, Requirement Analysis, Test case design criteria, Testing Methods, requirement based testing, Positive & negative testing, Boundary value analysis, Equivalence Partitioning class, state based or graph based, cause effect graph based, error guessing, documentation testing & domain testing, design of test cases. Case studies of Black-Box testing.

UNIT 3: Introduction, Need of white box testing, Testing types, Test adequacy criteria, static testing by humans, Structure - logic coverage criteria, Cyclomatic Complexity, Basis path testing, Graph metrics, Loop Testing, Data flow testing, Mutation Testing, Design of test cases. Testing of Object oriented systems, Challenges in White box testing, Case-study of White-Box testing

UNIT 4: Testing Activities: Unit Testing, Levels of Testing, Integration Testing, System Testing, Debugging, Domain Testing. **Reducing the number of test cases:** Prioritization guidelines, Priority category, Scheme, Risk Analysis, Regression Testing, slice based testing.

UNIT 5: Software quality, Quality attribute, Quality Assurance, Quality control & assurance, Methods of quality management, Cost of quality, Quality management, Quality factor, Quality management & project management, Software quality metrics-TQM, Six Sigma, ISO, SQA Model.

UNIT 6: Test organization, Structure of testing, Measurement tools, Testing metrics: Type of metric – Project, Progress, Productivity, Metric plan, Goal Question metric model, Measurement in small & large system. Other Software Testing: GUI testing, Validation testing, Regression testing, Scenario testing, Specification based testing, Adhoc testing, Sanity testing,

Textbooks:

1. William Perry, "Effective Methods for software Testing ", John Wiley & Sons, New York, 1995.
2. Louise Tamres, " Software Testing", Pearson Education Asia, 2002.
3. Software Testing, Second Edition By: Ron Patton, Pearson Education ISBN-13: 978-0-672-32798-8
4. Software testing Principle and Practices By Ramesh Desikan, Pearson Education, ISBN 81-7758-121-X

Reference books:

1. Effective methods for software testing by William Perry , Willey Publication, ISBN 81-265-0893-0
2. Metric and Model in Software Quality Engineering , By Stephen H Kan, Pearson Education ISBN 81-297-0175-8
3. K.K Aggarwal & Yogesh Singh, " Software Engineering",2nd Ed., New Age International publishers, New Delhi
4. Boris Beizer, " Software Testing Techniques", Second Volume, Second Edition, Van Nostrand Reinhold, New York
5. Boris Beizer, " Black- Box Testing- Techniques for Functional Testing of Software and System", John wiley & Sons Inc., New York, 1995.

Course Outcomes: Upon successful completion of this course you should be able to:

1. Demonstrate knowledge of the fundamentals of software testing
2. Use fundamental techniques to implement techniques to extrapolate fundamental techniques in the framework of real world scenarios
3. Demonstrate competence in using software designed to assist in the software testing life cycle for given portions of the testing cycle
5. Methods of test generation from requirements
6.Test adequacy assessment using: control flow, data flow, and program mutations
5. The use of various test tools and Application of software testing techniques in commercial environments

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
EC6T06	Information Theory & Coding	3	0	0	3

UNIT - 1: INFORMATION THEORY: Introduction, Measure of information, Average information content of symbols in long independent sequences, Average information content of symbols in long dependent sequences. Markoff statistical model for information source, Entropy and information rate of mark-off source.

UNIT - 2: SOURCE CODING: Encoding of the source output, Shannon's encoding algorithm. Communication Channels, Discrete communication channels,
Continuous channels.

UNIT - 3: FUNDAMENTAL LIMITS ON PERFORMANCE: Source coding theorem, Huffman coding, Discrete memory less Channels, Mutual
Information, Channel Capacity.

UNIT - 4: Channel coding theorem, Differential entropy and mutual information for continuous ensembles, Channel capacity Theorem.

UNIT - 5: INTRODUCTION TO ERROR CONTROL CODING: Introduction, Types of errors, examples, Types of codes Linear Block Codes: Matrix description, Error detection and correction, Standard arrays and table look up for decoding.

UNIT - 6: Binary Cycle Codes, Algebraic structures of cyclic codes, Encoding using an (n-k) bit shift register, Syndrome calculation. BCH codes.

UNIT - 7: RS codes, Golay codes, Shortened cyclic codes, Burst error correcting codes. Burst and Random Error correcting codes.

UNIT - 8: Convolution Codes, Time domain approach. Transform domain approach

Textbooks:

1. K. Sam Shanmugam, John Wiley 'Digital and Analog Communication Systems', India Pvt. Ltd., 2008.
2. Taub & Schilling.'Principles of Communication system' .
3. John G.Proakis 'digital Communications' Tata Mc Graw Hill
4. Digital Communication, Simon Haykin, John Wiley India Pvt. Ltd, 2008.

Reference books:

1. Communication Systems, Sanjay Sharma, Fifth Revised Ed June 2011
2. ITC and Cryptography, Ranjan Bose, TMH, II edition, 2007.
3. Digital Communications - Glover and Grant; Pearson Ed. 2nd Ed 2008

Course Outcomes: After completion of course, students would be:

1. The aim of this course is to introduce the principles and applications of information theory.
2. The course will study how information is measured in terms of probability and entropy.
3 The students learn coding schemes, including error correcting codes, The Fourier perspective; and extensions to wavelets, complexity, compression, and efficient coding of audio-visual information

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
EC8T01	Digital Image Processing	3	0	0	3

Unit 1: Introduction: Digital Image representation, fundamental steps in image processing, Elements of digital image processing systems.

Unit 2 :Digital Image Fundamentals: Elements of Visual perception, A simple Model, Image Sensing and Acquisition, Image Sampling and quantization, some basic relationships between pixels.

Unit 3: Image Transformation and enhancement: Some basic Intensity Transformation functions, Histogram Processing, Smoothing and Sharpening Spatial Filters, Smoothing and Sharpening using Frequency Domain Filters.

Unit 4: Image Restoration: Degradation Model, Noise Models, Restoration in the presence of Noise – Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Inverse Filtering, Minimum Mean square error filter.

Unit 5 :Image Segmentation: Point Detection, Line Detection, Edge Detection, Thresholding, Region – Based Segmentation, Color Image Processing.

Textbooks:

1. Digital Image Processing - Rafael C Gonzalez - Addison Wesley

Reference books:

1. Digital Image Processing - Richard E Woods -'" Addison Wesley
2. Fundamentals of Digital Image Processing - A.K Jain – PHI

Course Outcomes: After completion of course, students would be:

1. Review the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques.
5. Interpret Image compression standards.
6. Interpret image segmentation and representation techniques.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
MAT6T1	Operations Research	3	0	0	3

Unit 1: Optimization Techniques, Model Formulation, models, General L.R Formulation, Simplex Techniques, Sensitivity Analysis, Inventory Control Models
Unit 2: Formulation of a LPP - Graphical solution revised simplex method - duality theory - dual simplex method - sensitivity analysis - parametric programming
Unit 3: Nonlinear programming problem - Kuhn-Tucker conditions min cost flow problem - max flow problem - CPM/PERT
Unit 4: Scheduling and sequencing - single server and multiple server models - deterministic inventory models - Probabilistic inventory control models - Geometric Programming.
Unit 5: Competitive Models,Single and Multi-channel Problems, Sequencing Models, Dynamic Programming, Flow in Networks, Elementary Graph Theory, Game Theory Simulation

Reference books:
1. H.A. Taha, Operations Research, An Introduction, PHI, 2008
2. H.M. Wagner, Principles of Operations Research, PHI, Delhi, 1982.
3. J.C. Pant, Introduction to Optimisation: Operations Research, Jain Brothers, Delhi, 2008
4. Hitler Libermann Operations Research: McGraw Hill Pub. 2009
5. Pannerselvam, Operations Research: Prentice Hall of India 2010
6. Harvey M Wagner, Principles of Operations Research: Prentice Hall of India 2010

Course Outcomes: At the end of the course, the student should be able to
1. Apply the dynamic programming to solve problems of discreet and continuous variables.
2. Students should able to apply the concept of non-linear programming
3. Students should able to carry out sensitivity analysis
4. Student should able to model the real world problem and simulate it.

Course Code	*Course name*	**Lectures**	**Tutorials**	**Practicals**	**Total Credits**
CSB721	Design & Management of Computer Network	3	0	0	3

Prerequisites: Computer Networks

Unit 1: Fundamentals. Communication networks, network elements, networking principles, switched networks and shared media networks, datagrams and virtual circuits, multiplexing, switching, error and flow control, congestion control, layered architecture, QoS.

Unit 2: Requirements and flow analysis. Requirement analysis, requirement specification and map, requirement analysis process - developing the requirements specification, flows, identifying and developing flows, flow models, prioritization and specification.

Unit 3: Network architecture. Architecture and design – component architectures, reference architecture, architecture models, system and network architecture, addressing and routing architecture – addressing and routing fundamentals, network management architecture, performance mechanisms, security and privacy architecture – planning security and privacy mechanisms.

Unit 4: Network design. Design concepts, design process, network layout, design traceability, design metrics, logical network design – topology design, switching and routing protocols, physical network design – selecting technologies and devices for campus and enterprise networks, optimizing network design.

Unit 5: Network management. Monitoring and Control – Network management systems – abstract syntax notation – CMIP – SNMP communication model – SNMP MIB group – functional model – major changes in SNMPv2 and SNMPv3 – remote monitoring – RMON SMI and MIB

Reference books:

1. James D. McCabe and Morgan Kaufmann, "Network Analysis, Architecture, and Design", 3rd Edition, 2007.

2. Larry L. Peterson and Bruce S. Davie "Computer Networks: A Systems Approach", Elsevier, 2007.

3. Priscilla Oppenheimer, "Top-down Network Design: A Systems Analysis Approach to Enterprise Network Design", 3rd Edition, Cisco Press

4. Heinz-Gerd Hegering, Sebastian Abeck, and Bernhard Neumair "Integrated Management of Networked Systems: Concepts, Architectures, and Their Operational Application" Morgan Kaufmann Series in Networking, 1999.

5. Steven T.Karris, "Network Design and Management", 2nd Edition, Orchard Publications, 2009.

6. Teresa C. Mann-Rubinson and Kornel Terplan, "Network Design, Management and Technical Perspective", , CRC Press, 1999.

7. Gilbert Held, "Ethernet Networks-Design, Implementation, Operation and Management", 4th Edition, John Wiley and sons.

8. James Kurose and Keith Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 1999.

9. William Stallings, 'High Speed Networks: Performance and Quality of Service', 2nd Edition, Pearson Education, 2002.

Course Outcomes: At the end of the course, the student should have

1. A comprehensive knowledge of applicable methods and techniques and their limitations.
2. A deep knowledge and understanding of the principles of their specialties.
3. The ability to critically analyze the network.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB722	Human Computer Interaction	3	0	0	3

Unit 1: Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms.

Unit 2: Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules– principles, standards, guidelines, rules. Evaluation Techniques – Universal Design.

Unit 3: Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.

Unit 4: Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.

Unit 5: Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies.

Unit 6: Recent Trends: Speech Recognition and Translation, Multimodal System

Reference books:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human Computer Interaction", 3rd Edition, Pearson Education, 2004 (UNIT I , II & III)
2. Brian Fling, "Mobile Design and Development", First Edition , O□Reilly Media Inc., 2009 (UNIT – IV)
3. Bill Scott and Theresa Neil, "Designing Web Interfaces", First Edition, O□Reilly, 2009.(UNIT-V)

Course Outcomes: After completion of course, students would be able to:

1. Understand the structure of models and theries of human computer interaction and vision.\
2. Design an interactive web interface on the basis of models studied.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB723	Cloud Computing	3	0	0	3

Prerequisites: Operating Systems, Computer Networks

Unit 1: Introduction to Cloud Computing: Online Social Networks and Applications, Cloud introduction and overview, Different clouds, Risks, Novel applications of cloud computing

Unit 2: Cloud Computing Architecture: Requirements, Introduction Cloud computing architecture, On Demand Computing**,** Virtualization at the infrastructure level, Security in Cloud computing environments, CPU Virtualization, A discussion on Hypervisors Storage Virtualization Cloud Computing Defined, The SPI Framework for Cloud Computing, The Traditional Software Model, The Cloud Services Delivery Model Cloud Deployment Models
Key Drivers to Adopting the Cloud, The Impact of Cloud Computing on Users, Governance in the Cloud, Barriers to Cloud Computing Adoption in the Enterprise

Unit 3: Security Issues in Cloud Computing: Infrastructure Security, Infrastructure Security: The Network Level, The Host Level, The Application Level, Data Security and Storage, Aspects of Data Security, Data Security Mitigation Provider Data and Its Security Identity and Access Management Trust Boundaries and IAM, IAM Challenges, Relevant IAM Standards and Protocols for Cloud Services, IAM Practices in the Cloud, Cloud Authorization Management

Unit 4: Security Management in the Cloud: Security Management Standards, Security Management in the Cloud, Availability Management: SaaS, PaaS, IaaS

Privacy Issues: Privacy Issues, Data Life Cycle, Key Privacy Concerns in the Cloud, Protecting Privacy, Changes to Privacy Risk Management and Compliance in Relation to Cloud Computing, Legal and Regulatory Implications, U.S. Laws and Regulations, International Laws and Regulations

Unit 5: Audit and Compliance: Internal Policy Compliance, Governance, Risk, and Compliance (GRC), Regulatory/External Compliance, Cloud Security Alliance, Auditing the Cloud for Compliance, Security-as-a-Cloud

Unit 6: ADVANCED TOPICS: Recent devlopments in hybrid cloud and cloud security.

Text Books:

1. Anthony T.Velte, Toby J.Velte and Robert E, Cloud Computing – A Practical Approach, TMH , 2010
2. Michael Miller, Cloud Computing – Web based Applications, Pearson Publishing, 2011

Course Outcomes:

After completion of course, students would be:

1. Identify security aspects of each cloud model
2. Develop a risk-management strategy for moving to the Cloud
3. Implement a public cloud instance using a public cloud service provider
4. Apply trust-based security model to different layer

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB724	Wireless Sensor Networks	3	0	0	3

Unit 1: Introduction to Wireless Sensor Networks: Course Information, Introduction to Wireless Sensor Networks: Motivations, Applications, Performance metrics, History and Design factors.
Network Architecture: Traditional layered stack, Cross-layer designs, Sensor Network Architecture
Hardware Platforms: Motes, Hardware parameters.
Unit 2: Introduction to ns-3: Introduction to Network Simulator 3 (ns-3), Description of the ns-3 core module and simulation example.
Unit 3: Medium Access Control Protocol design: Fixed Access, Random Access, WSN protocols: synchronized, duty-cycled
Introduction to Markov Chain: Discrete time Markov Chain definition, properties, classification and analysis
MAC Protocol Analysis: Asynchronous duty-cycled. X-MAC Analysis (Markov Chain)
Unit 4: Security: Possible attacks, countermeasures, SPINS, Static and dynamic key
distribution
Unit 5: Routing protocols: Introduction, MANET protocols
Routing protocols for WSN: Resource-aware routing, Data-centric, Geographic Routing, Broadcast, Multicast
Opportunistic Routing Analysis: Analysis of opportunistic routing (Markov Chain) Advanced topics in wireless sensor networks.
Unit 6: ADVANCED TOPICS Recent development in WSN standards, software applications.

References:

1. W. Dargie and C. Poellabauer, "Fundamentals of Wireless Sensor Networks –Theory and Practice", Wiley 2010
2. KazemSohraby, Daniel Minoli and TaiebZnati, "wireless sensor networks -Technology, Protocols, and Applications", Wiley Interscience 2007
3. Takahiro Hara,Vladimir I. Zadorozhny, and Erik Buchmann, "Wireless Sensor Network Technologies for the Information Explosion Era", springer 2010

Course Outcomes: After completion of course, students would be able to:
1. Describe and explain radio standards and communication protocols for wireless sensor networks.
2. Explain the function of the node architecture and use of sensors for various applications.
3. Be familiar with architectures, functions and performance of wireless sensor networks systems and platforms.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB725	Internet-of-Things	3	0	0	3

Prerequisites: Operating Systems, Computer Networks

Unit 1: Introduction and Applications:smart transportation, smart cities, smart living, smart energy, smart health, and smart learning. Examples of research areas include for instance: Self-Adaptive Systems, Cyber Physical Systems, Systems of Systems, Software Architectures and Connectors, Software Interoperability, Big Data and Big Data Mining, Privacy and Security

Unit 2: IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints- Introduction, Technical Design constraintshardware, Data representation and visualization, Interaction and remote control.

Unit 3: Industrial Automation- Service-oriented architecture-based device integration, SOCRADES: realizing the enterprise integrated Web of Things, IMC-AESOP: from the Web of Things to the Cloud of Things, Commercial Building Automation- Introduction, Case study: phase one-commercial building automation today, Case study: phase two- commercial building automation in the future.

Unit 4: Hardware Platforms and Energy Consumption, Operating Systems, Time Synchronization, Positioning and Localization, Medium Access Control, Topology and Coverage Control, Routing: Transport Protocols, Network Security, Middleware, Databases

Unit 5: IOT Physical Devices & Endpoints: What is an IOT Device, Exemplary Device Board, Linux on Raspberry , Interface and Programming & IOT Device

Unit 6: Recent trends in sensor network and IOT architecture, Automation in Industrial aspect of IOT

References:

1. Yasuura, H., Kyung, C.-M., Liu, Y., Lin, Y.-L., Smart Sensors at the IoT Frontier, Springer International Publishing

2. Kyung, C.-M., Yasuura, H., Liu, Y., Lin, Y.-L., Smart Sensors and Systems, Springer International Publishing

3. Mandler, B., Barja, J., Mitre Campista, M.E., Cagáová, D., Chaouchi, H., Zeadally, S., Badra, M., Giordano, S., Fazio, M., Somov, A., Vieriu, R.-L., Internet of Things. IoT Infrastructures, Springer International Publishing

Course Outcomes: On completion of the course the student should be able to

1. Understand the vision of IoT from a global context.
2. Determine the Market perspective of IoT.
3. Use of Devices, Gateways and Data Management in IoT.
4. Application of IoT in Industrial and Commercial Building Automation and Real World Design Constraints. Building state of the art architecture in IoT.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB726	Real Time Systems	3	0	0	3

Prerequisites: Programming and Data Structures, Operating Systems, Computer Architecture and Organization, Computer Network, Database Management Systems

Unit 1: Introduction. Real-Time Applications, Basic model of Real-Time systems, characteristics, safety and reliability, types of real-time tasks, timing constraints, modelling of timing constraints.
Unit 2: Real-Time Scheduling. Types of real-time tasks and their characteristics, task scheduling, clock-driven scheduling, event-driven scheduling, hybrid scheduling.
Unit 3: Handling resource sharing and dependencies among real-time tasks. Priority inversion, priority inheritance protocol, highest locker protocol, priority ceiling protocol, issues in using a resource sharing protocol, handling task dependencies.
Unit 4: Scheduling real-time tasks in multiprocessor and distributed systems. Multiprocessor task allocation, dynamic allocation of tasks, fault tolerant scheduling of tasks, clocks in distributed real-time systems, centralized and distributed clock synchronization.
Unit 5: Commercial real-time operating systems. Time services, features of a real-time operating system, Unix as a real-time operating system, Unix based real-time operating system, windows as a real-time operating system, POSIX, a survey of contemporary real-time operating system, bench marking real-time systems.
Unit 6: Real-time communication. Some examples, basic concepts, real-time communication in a LAN, soft and hard real-time communication in a LAN, bounded access protocols for LAN, real-time communication over packet-switched networks, QoS framework, routing, resource reservation, rate control, QoS models.
Unit 7: Real-time databases. Examples, real-time databases, characteristics of temporal data, concurrency control in real-time database, commercial real-time databases.

Text books:

1. Real-Time systems: Theory and Practice by Rajib Mall, Pearson.
2.Jane W. Liu, "Real-Time Systems" Pearson Education, 2001.
3. Krishna and Shin, "Real-TIme Systems," Tata McGraw Hill. 1999.

Additional readings:

1. Alan C. Shaw, Real-Time Systems and Software, Wiley, 2001.
2. Philip Laplante, Real-TIme Systems Design and Analysis, 2nd Edition, Prentice Hall of India.

Course Outcomes:

After completion of course, students would be:

1. Characterise real time systems and describe their functions.
2. Analyse, design and implement a real-time system.
3. Apply formal methods to the analysis and design of real-time systems.
4. Apply formal methods for scheduling real-time systems.
5. Characterise and debug a real-time system.
6. Explain fundamental principles for programming of real time systems with time and resource limitations.
7. Describe the foundation for programming languages developed for real time programming.
8. Use real time system programming languages and real time operating systems for real time applications.
9. Analyse real time systems with regard to keeping time and resource restrictions.

NAGALAND UNIVERSITY

(A Central University Established by an Act of Parliament 1989)

School of Engineering &Technology

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB727	Advanced Computer Architecture & parallel programming	3	0	0	0

Prerequisites: Computer Organization & Architecture

UNIT-1: Parallel computer models: The state of computing, Multiprocessors and multi computers, Multi vector and SIMD computers, Architectural development tracks. Vector processing principles: Vector instruction types, Vector-access memory schemes. Parallel Processing: SIMD Architecture and Programming Principles, SIMD parallel Algorithms, SIMD Computers and performance Enhancement .
UNIT-2: Program and network properties: Conditions of parallelism, Data and resource dependences, Hardware and Software parallelism, Program partitioning and Scheduling, Grain size and latency, Program flow mechanisms, Control flow versus data flow, Data flow architecture, Demand driven mechanisms, Comparisons of flow mechanisms.
UNIT-3: Interconnect Architectures: Network properties and routing, Static interconnection networks, Dynamic interconnection Networks, Multiprocessor system interconnects, Hierarchical bus systems, Crossbar switch and multiport memory, Multistage and combining network.
UNIT-4: Processors and Memory Hierarchy: Advanced processor technology, Instruction-set Architectures, CISC Scalar processors , RISC Scalar Processors, Superscalar Processors, VLIW Architectures, Vector and Symbolic processors.
UNIT-5: Memory Technology: Hierarchical memory technology, Inclusion, Coherence and Locality, Memory capacity planning, Virtual Memory Technology. Backplane Bus Systems: Backplane bus specification, Addressing and timing protocols, Arbitration transaction and interrupt, Cache addressing models, Direct mapping and associative caches.
UNIT-6: Pipelining: Principles and implementation of Pipelining, Classification of pipelining processors, Pipeline Architecture, Linear pipeline processor, Nonlinear pipeline processor, Instruction pipeline design, Mechanisms for instruction pipelining, Dynamic instruction scheduling, Branch handling techniques, Arithmetic pipeline Design, Computer arithmetic principles, Static arithmetic pipeline, Multifunctional arithmetic pipelines.
Unit 7: Application of parallel computing, Parallel programming using Open MP: private variable, shared variable, scope, distributing for loops, multithreading, inter thread communication using shared memory, race condition, critical section, lock, synchronization, matrix vector operation, sections, task, barrier.

Text books:
1. Kai Hwang, "Advanced computer architecture",TMH,2000.

Reference books:
1. J.P. Hayes, " Computer Architecture and organization", MGH,1998.
2. M.J.Flynn, " Computer Architecture, Pipelined and parallel processor Design", Narosa publishing,1998.
3. D.A. Patterson, J.L. Hennessy, " Computer Architecture: A quantitative approach", Morgan Kauffmann,2002.
4. Hwang and Briggs, " Computer Architecture and Parallel Processing", MGH, 2000.
5. Introduction to Parallel Computing (Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar)

Course Outcomes: After completion of this course, students should be able to
1. Describe the principles of computer design.
2. Describe the operation of performance enhancements such as pipelines, dynamic scheduling, branch prediction, caches, virtual memory and vector processors.
3. Describe modern architectures such as RISC, Super Scalar, VLIW (very large instruction word), multi-core and multi-cpu systems.
4. Compare the performance of different architectures.
5. Understand the concepts and programming principles involved in developing scalable parallel/HPC applications.
6. Write scalable parallel program using OpenMP for multicore architecture.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
EC7EL1	Embedded Systems & Design	3	0	0	3

Unit 1 : Introduction to Embedded System: Introducing Embedded Systems, Philosophy, Embedded Systems, Embedded Design and Development Process.

Unit 2: The Hardware Side: An Introduction, The Core Level, Representing Information, Understanding Numbers, Addresses, Instructions, Registers-A First Look, Embedded Systems-An Instruction Set View, Embedded Systems-A Register View, Register View of a Microprocessor The Hardware Side: Storage Elements and Finite-State Machines (2 hour) The concepts of State and Time, The State Diagram, Finite State Machines- A Theoretical Model.

Unit 3: Memories and the Memory Subsystem: Classifying Memory, A General Memory Interface, ROM Overview, Static RAM Overview, Dynamic RAM Overview, Chip Organization, Terminology, A Memory Interface in Detail, SRAM Design, DRAM Design, DRAM Memory Interface, The Memory Map, Memory Subsystem Architecture, Basic Concepts of Caching, Designing a Cache System, Dynamic Memory Allocation.

Unit 4 : Embedded Systems Design and Development : System Design and Development, Life-cycle Models, Problem Solving-Five Steps to Design, The Design Process, Identifying the Requirements, Formulating the Requirements Specification, The System Design Specification, System Specifications versus System Requirements, Partitioning and Decomposing a System, Functional Design, Architectural Design, Functional Model versus Architectural Model, Prototyping, Other Considerations, Archiving the Project.

Unit 5 : Real-Time Kernels and Operating Systems: Tasks and Things, Programs and Processes, The CPU is a resource, Threads – Lightweight and heavyweight, Sharing Resources, Foreground/Background Systems, The operating System, The real time operating system (RTOS), OS architecture, Tasks and Task control blocks, memory management revisited.

Unit 6 : Performance Analysis and Optimization: Performance or Efficiency Measures, Complexity Analysis, The methodology, Analyzing code, Instructions in Detail, Time, etc. – A more detailed look, Response Time, Time Loading, Memory Loading, Evaluating Performance, Thoughts on Performance Optimization, Performance Optimization, Tricks of the Trade, Hardware Accelerators, Caches and Performance.

Text books:

1. Embedded Systems – A contemporary Design Tool, James K. Peckol, John Weily India Pvt. Ltd, 2008.

Reference books:

1. Embedded Systems: Architecture and Programming, Raj Kamal, TMH. 2008.
2. Embedded Systems Architecture – A Comprehensive Guide for Engineers and Programmers, Tammy Noergaard, Elsevier Publication, 2005.
3. Programming for Embedded Systems, Dream tech Software Team, John Wiley India Pvt. Ltd, 2008

Course Outcomes: After completion of this course, students should be able to

1. Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
2. Describe the differences between the general computing system and the embedded system, also recognize the classification of embedded systems.
3. Become aware of the architecture of the ATOM processor and its programming aspects (assembly Level)
4. Become aware of interrupts, hyper threading and software optimization.
5. Design real time embedded systems using the concepts of RTOS.
6. Analyze various examples of embedded systems based on ATOM processor

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB821	Distributed Databases	3	0	0	3

Pre-Requisites: Distributed Systems

Unit 1: Introduction: Distributed Data processing, Distributed database system (DDBMS), Promises of DDBMSs, Complicating factors and Problem areas in DDBMSs, Overview Of Relational DBMS Relational Database concepts, Normalization, Integrity rules, Relational Data Languages, Relational DBMS.
Unit 2: Distributed DBMS Architecture: DBMS Standardization, Architectural models for Distributed DBMS, Distributed DBMS Architecture. Distributed Database Design: Alternative design Strategies, Distribution design issues, Fragmentation, Allocation. Semantic Data Control: View Management, Data security, Semantic Integrity Control.
Unit 3: Overview of Query Processing: Query processing problem, Objectives of Query Processing, Complexity of Relational Algebra operations, characterization of Query processors, Layers of Query Processing. Introduction to Transaction Management: Definition of Transaction, Properties of transaction, types of transaction. Distributed Concurrency Control: Serializability theory, Taxonomy of concurrency control mechanisms, locking bases concurrency control algorithms.
Unit 4: Parallel Database Systems: Database servers, Parallel architecture, Parallel DBMS techniques, Parallel execution problems, Parallel execution for hierarchical architecture.
Unit 5: Distributed Object Database Management systems: Fundamental Object concepts and Object models, Object distribution design. Architectural issues, Object management, Distributed object storage, Object query processing. Transaction management. Database Interoperability: Database Integration, Query processing.
Unit 6: Recent approaches, models and current trends in improving the performance of Distributed Database.

References:
1. Principles of Distributed Database Systems, Second Edition, M. Tamer Ozsu Patrick Valduriez
2. Distributed Databases principles and systems, Stefano Ceri, Giuseppe Pelagatti, Tata McGraw Hill.

Course Outcomes: After completion of course, students would be:
1. Able to understand relational database management systems, normalization to make efficient retrieval from database and query.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB822	Artificial Intelligence	3	0	0	3

Prerequisites: Data Structures & Algorithm

UNIT-1: Introduction: Introduction to Artificial intelligence, Simulation of sophisticated & Intelligent Behaviour in different area, problem solving in games, natural language, automated reasoning, visual perception, heuristic algorithm versus solution guaranteed algorithms.

UNIT-2: Understanding Natural Languages: Parsing techniques, context free and transformational grammars, transition nets, augmented transition nets, Fillmore's grammars, Shanks conceptual Dependency, grammar free analyzers, sentence generation, and translation.

UNIT-3: Knowledge Representation: First order predicate calculus, Horn Clauses, Introduction to PROLOG, Semantic Nets, Partitioned Nets, Minskey Net frames Case Grammar Theory, Production Rules Knowledge Base, The Interface System, Forward & Backward Deduction.

UNIT-4: Expert System: Existing Systems (DENDRAL, MYCIN), domain exploration, Meta Knowledge, Expertise Transfer, Self Explaining System.

UNIT-5: Pattern Recognition: Introduction to pattern Recognition, Structured Description, Symbolic Description, Machine perception, Line Finding, Interception, Semantic & Model, Object Identification, Speech Recognition. Programming Language: Introduction to programming Language, LISP, PROLOG.

Reference books:

1. Char nick, " Introduction to Artificial Intellegence", Addision Welsey.
2. Rich & Knight," Artificial Intellengence" Winston, " LISP", Addision Wesley
3. Marcellous, " Expert System Programming", PHI.
4. Elamie," Artificial Intellengence", Academic Press.
5. Lioyed, " Foundation of logic Programming", Springer Verlag.

Course Outcomes: After completion of course, students would be:

1. Knowledge of what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.
2. Explain how Artificial Intelligence enables capabilities that are beyond conventional technology, for example, chess-playing computers, self-driving cars, robotic vacuum cleaners.
3. Implement classical Artificial Intelligence techniques, such as search algorithms, minimax algorithm, neural networks, tracking, robot localisation.
4. Ability to apply Artificial Intelligence techniques for problem solving.
5. Explain the limitations of current Artificial Intelligence techniques.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB823	Speech & Natural Language Processing	3	0	0	3

Prerequisites: Compiler Design, Formal Language & Automata

Introduction- Human languages, models, ambiguity, processing paradigms; Phases in natural language processing, applications.

Text representation in computers, encoding schemes.

Linguistics resources- Introduction to corpus, elements in balanced corpus, TreeBank, PropBank, WordNet, VerbNet etc. Resource management with XML, Management of linguistic data with the help of GATE, NLTK.

Regular expressions, Finite State Automata, word recognition, lexicon. Morphology, acquisition models, Finite State Transducer. N-grams, smoothing, entropy, HMM, ME, SVM, CRF. Part of Speech tagging- Stochastic POS tagging, HMM, Transformation based tagging (TBL), Handling of unknown words, named entities, multi word expressions.

A survey on natural language grammars, lexeme, phonemes, phrases and idioms, word order, agreement, tense, aspect and mood and agreement, Context Free Grammar, spoken language syntax.

Parsing- Unification, probabilistic parsing, TreeBank. Semantics- Meaning representation, semantic analysis, lexical semantics, WordNet Word Sense Disambiguation- Selectional restriction, machine learning approaches, dictionary based approaches.

Discourse- Reference resolution, constraints on co-reference, algorithm for pronoun resolution, text coherence, discourse structure.

Applications of NLP- Spell-checking, Summarization Information Retrieval- Vector space model, term weighting, homonymy, polysemy, synonymy, improving user queries.

Machine Translation– Overview.

Textbooks:

1. Daniel Jurafsky and James H Martin. Speech and Language Processing, 2e, Pearson Education, 2009

Reference Books:

1. James A.. Natural language Understanding 2e, Pearson Education, 1994
2. Bharati A., Sangal R., Chaitanya V.. Natural language processing: a Paninian perspective, PHI, 2000 3. Siddiq

Course Outcomes: After completion of course, students would be able to:

1. Compose key NLP elements to develop higher level processing chains
2. Assess / Evaluate NLP based systems
3. Choose appropriate solutions for solving typical NLP subproblems (tokenizing, tagging, parsing)
4. Describe the typical problems and processing layers in NLP
5. Analyze NLP problems to decompose them in adequate independant components

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB824	Neural Networks & Deep Learning	3	0	0	3

Prerequisites: Machine Learning

- Introduction: Various paradigms of earning problems, Perspectives and Issues in deep learning framework, review of fundamental learning techniques.
- Feedforward neural network: Artificial Neural Network, activation function, multi-layer neural network.
- Training Neural Network: Risk minimization, loss function, backpropagation, regularization, model selection, and optimization.
-Conditional Random Fields: Linear chain, partition function, Markov network, Belief propagation, Training CRFs, Hidden Markov Model, Entropy.
-Deep Learning: Deep Feed Forward network, regularizations, training deep models, dropouts, Convolutional Neural Network, Recurrent Neural Network, Deep Belief Network.
-Probabilistic Neural Network: Hopfield Net, Boltzman machine, RBMs, Sigmoid net, Autoencoders.
-Deep Learning research: Object recognition, sparse coding, computer vision, natural language processing.
-Deep Learning Tools: Caffe, Theano, Torch.

Textbooks:

1. T1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016..
2. T2. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books:

1. R1. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
2. R2. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.
3. R3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Books on Optimization Techniques:

1. A. Ravindran, K. M. Ragsdell , and G. V. Reklaitis , ENGINEERING OPTIMIZATION: Methods and Applications , John Wiley & Sons, Inc. , 2016..
2. A. Antoniou, W. S. Lu, PRACTICAL OPTIMIZATION Algorithms and Engineering Applications, Springer , 2007**.**

Course Outcomes: After completion of course, students would be able to:

1. Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains.
2. Implement deep learning algorithms and solve real-world problems
3. Explain different network architectures and how these are used in current applications
4. Implement, train, and evaluate neural networks using existing software libraries
5. Present and critically assess current research on neural networks and their applications
6. Relate the concepts and techniques introduced in the course to your own research
7. Plan and carry out a research project on neural networks within given time limits

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB831	Cryptography & Network Security	3	0	0	3

Prerequisites: Operating Systems, Computer Networks

UNIT-1: Introduction to security attacks, services and mechanisms, Introduction to cryptography. Conventional Encryption: Conventional Encryption model, classical encryption techniques-substitution ciphers & transpositions ciphers, cryptanalysis, stereography, stream and block ciphers. Modern Block Ciphers: Block Ciphers

UNIT-2: Principles, Shannon's theory of confusion and diffusion, fiestal structure, Data Encryption Standards (DES), Strength of DES, Differential & Linear Cryptanalysis of DES, Block Cipher modes of operation, Triple DES, Confidentiality using Conventional Encryption, traffic confidentiality, key distribution, random number generation.

UNIT-3: Introduction to graph, ring and field, Prime and relative prime numbers, modular arithmetic, Fermat's & Euler's Theorem, primality testing, Euclid's Algorithm, Chinese remainder theorem, Discrete logarithms. Principles of public key cryptosystems, RSA algorithm, security of RSA, key management, Diffie- Heilman key exchange algorithm

UNIT-4: Message Authentication & Hash functions : Authentication requirements, Authentication functions, Message Authentication codes, Hash functions, Birthday attacks, Security of Hash function & MACS, MD5 message digest algorithm, Secure Hash algorithm (SHA). Digital Signatures: Digital Signatures, Authentication protocol, Digital Signature Standards (DSS), proof of digital signature algorithm.

UNIT-5: Authentication Applications: directory authentication service, Electronic Mail security- Pretty Good Privacy (PGP), S/ MIME.IP
Security : Architecture, Authentication Header, Encapsulating security payloads, Combining security associations, Secure Electronic Transaction (SET) .
System Security : Intruders, Viruses and related threads, Firewall design principles, trusted systems.

Textbooks:
1. William Stallings," Cryptography and Network Security:
2. Principles and Practice" Prentice Hall,New Jersey.
3. Johannes A. Buchmann," Introduction to cryptography", Springer Verlag. Bruce Schiener, " Applird Cryptography

Course Outcomes: After completion of course, students would be able to:
1. Describe network security services and mechanisms.
2. Symmetrical and Asymmetrical cryptography.
3. Data integrity, Authentication, Digital Signatures.
4. Various network security applications, IPSec, Firewall, IDS, Web security, Email security, and Malicious software etc.

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB832	Mobile Applications & Services	3	0	0	3

Pre-Requisites: Wireless Communication and Mobile Computing

Unit 1: Introduction: Introduction to Mobile Computing, Introduction to Android Development Environment, Factors in Developing Mobile Applications, Mobile Software Engineering, Frameworks and Tools, Generic UI Development Android User

Unit 2: More on Uis: VUIs and Mobile Apps, Text-to-Speech Techniques, Designing the Right UI, Multichannel and Multimodal Uis, . Storing and Retrieving Data, Synchronization and Replication of Mobile Data, Getting the Model Right, Android Storing and Retrieving Data, Working with a Content Provider

Unit 3: Communications via Network and the Web: State Machine, Correct Communications Model, Android Networking and Web, Telephony Deciding Scope of an App, Wireless Connectivity and Mobile Apps, Android Telephony Notifications and Alarms: Performance, Performance and Memory Management, Android Notifications and Alarms, Graphics, Performance and Multithreading, Graphics and UI Performance, Android Graphics

Unit 4: Putting It All Together : Packaging and Deploying, Performance Best Practices, Android Field Service App, Location Mobility and Location Based Services Android Multimedia: Mobile Agents and Peer-to-Peer Architecture, Android Multimedia

Unit 5: Platforms and Additional Issues : Development Process, Architecture, Design, Technology Selection, Mobile App Development Hurdles, Testing, Security and Hacking , Active Transactions, More on Security, Hacking Android

Unit 6: Recent trends in Communication protocols for IOT nodes, mobile computing techniques in IOT, agents based communications in IOT

Reference books:

1. Wei-Meng Lee, Beginning Android™ 4 Application Development, 2012 by John Wiley & Sons

Course Outcomes: On completion of the course the student should be able to

1 Identify the target platform and users and be able to define and sketch a mobile application

2. Understand the fundamentals, frameworks, and development lifecycle of mobile application platforms including iOS, Android, and PhoneGap

3. Design and develop a mobile application prototype in one of the platform (challenge project)

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB833	Cyber Law & Ethics	3	0	0	3

Cyber laws and rights in today's digital age; IT Act, Intellectual Property Issues connected with use and management of Digital Data The similar Acts of other countries Information Warfare: Nature of information warfare, including computer crime and information terrorism; Threats to information resources, including military and economic espionage, communications eavesdropping, computer break-ins, denial-of-service, destruction and modification of data, distortion and fabrication of information, forgery, control and disruption of information How, electronic bombs, and sops and perception management. Countermeasures, including authentication, encryption, auditing, monitoring, intrusion election, and firewalls, and the limitations of those countermeasures. Cyberspace law and law enforcement, information warfare and the military, and intelligence in the information age. Information warfare policy and ethical Issues.

References:

1.Hon C Graff, Cryptography and E-Commerce - A Wiley Tech Brief, Wiley Computer Publisher, 2001

2.Michael Cross, Norris L Johnson, Tony Piltzecker, Security, Shroff Publishers and Distributors Ltd.

http://cse.nitk.ac.in/course/cyber-law-and-ethics

Course Outcomes: On completion of the course the student should be able to

1. Describe laws governing cyberspace and analyze the role of Internet Governance in framing policies for Internet security
2. Discuss different types of cybercrimes and analyze legal frameworks of different countries to deal with these cybercrimes.
3. Explain the importance of jurisdictional boundaries and identify the measures to overcome cross jurisdictional cyber crimes.
4. Illustrate the importance of ethics in the legal profession and determine the appropriate ethical and legal behaviour according to legal frameworks
5. Identify intellectual property right issues in the cyberspace and design strategies to protect your intellectual property.
6. Assess the legal issues with e-commerce
7. Recognize the importance of digital evidence in prosecution and compare laws of different countries

Course Code	Course name	Lectures	Tutorials	Practicals	Total Credits
CSB834	Linux Internal	3	0	0	3

Prerequisites: Operating Systems

UNIT 1: Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities, sed – scripts, operation, addresses, commands, applications, awk – execution, fields and records, scripts, operation, patterns, actions, functions, using system commands in awk.

UNIT 2: Working with the Bourne again shell(bash): Introduction, shell responsibilities, pipes and input Redirection, output redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

UNIT 3: Files: File Concept, File System Structure, Inodes, File Attributes, File types, Library functions, the standard I/O and formatted I/O in C, stream errors, kernel support for files, System calls, file descriptors, low level file access – File structure related system calls(File APIs), file and record locking, file and directory management – Directory file APIs, Symbolic links & hard links.

UNIT 4: Process – Process concept, Kernel support for process, process attributes, process control - process creation, waiting for a process, process termination, zombie process, orphan process, Process APIs. Signals– Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.

UNIT 5: Interprocess Communication : Introduction to IPC, Pipes, FIFOs, Introduction to three types of IPC-message queues, semaphores and shared memory. Message Queues- Kernel support for messages, Unix system V APIs for messages, client/server example.

UNIT 6: Semaphores-Kernel support for semaphores, Unix system V APIs for semaphores. Shared Memory- Kernel support for shared memory, Unix system V APIs for shared memory, semaphore and shared memory example.

UNIT 7: Multithreaded Programming: Differences between threads and processes, Thread structure and uses, Threads and Lightweight Processes, POSIX Thread APIs, Creating Threads, Thread Attributes, Thread Synchronization with semaphores and with Mutexes, Example programs.

UNIT 8: Sockets: Introduction to Sockets, Socket Addresses, Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs.

Textbooks:

1. Unix System Programming using C++, T.Chan, PHI.(UNIT 3 to UNIT 8)
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
3. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones,Wrox, Wiley India Edition.

Reference books:

1. Linux System Programming, Robert Love, O'Reilly, SPD.
2. Advanced Programming in the Unix environment, 2nd Edition, W.R.Stevens, Pearson Education.
3. Unix Network Programming ,W.R.Stevens,PHI.
4. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education.

Course Outcomes: On completion of the course the student should be able to

1. comfortably use basic UNIX/Linux commands from the command line (from a terminal window);
2. organize and manage their files within the UNIX/Linux file system;
3. organize and manage their processes within UNIX/Linux;
4. usefully combine UNIX/Linux tools using features such as filters, pipes, redirection, and regular expressions;
5. customize their UNIX/Linux working environment;
6. be knowledgeable enough about basic UNIX/Linux shell scripting to be able to successfully read and write bash shell scripts;
7. know how to use UNIX/Linux resources to find additional information about UNIX/Linux commands