Scheme & Syllabi

of

B. Tech. (Computer Science and Engineering)

July 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

Contents

Tentative UG(CSE) Scheme

Department of Computer Science and Engineering

First Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credit**	**Type**
		Institute Core Subjects		***19***	***IC***
	22CPT103	Problem Solving using C	2-0-0	2	PC
	22CPT104	Discrete Mathematics	3-0-0	3	PC
	22CPP105	Problem Solving Using C Lab	0-0-2	1	PC
				25	

Second Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credit**	**Type**
		Institute Core Subjects		***18***	***IC***
	22CPT106	Data Structures	3-0-0	3	PC
	22CPT107	Logic System Design	2-0-0	2	PC
	22CPP108	Data Structures Lab	0-0-2	1	PC
	22CPP109	Logic System Design Lab	0-0-2	1	PC
				25	

Third Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credits**	**Type**
	22CPT201	Data Communications	3-0-0	3	PC
	22CPT202	Design and Analysis of Algorithms	3-0-0	3	PC
	22CPT203	Digital Circuits and Microprocessors	3-0-0	3	PC
	22CPT204	Foundation of Learning	3-0-0	3	PC
	22CPT205	Object Oriented Analysis and Design	3-0-0	3	PC
	22CPT206	Technical Writing	1-0-0	1	PC
	22HST241	Social Sciences and Professional Ethics	3-1-0	4	BS

	22CPP207	Design and Analysis of Algorithms Lab	0-0-4	2	PC
	22CPP208	Digital Circuits and Microprocessors Lab	0-0-2	1	PC
	22CPP209	Object Oriented Analysis and Design Lab	0-0-2	1	PC
	22CPP210	Technical Writing Lab	0-0-2	1	PC
				25	

Fourth Semester

S. No	Code	Subject	L-T-P	Credits	Type
	22CPT211	Computer Networks	3-0-0	3	PC
	22CPT212	Computer Organization and Architecture	3-1-0	4	PC
	22CPT215	Theory of Computation	3-1-0	4	PC
	22CPT213	Database Information Systems	3-0-0	3	PC
	22CPT214	Machine Learning	3-0-0	3	PC
	22BMT201	Basics of Managements	3-0-0	3	MM
	22CPP216	Computer Networks Lab	0-0-4	2	PC
	22CPP217	Database Information Systems Lab	0-0-4	2	PC
	22CPP218	Machine Learning Lab	0-0-2	1	PC
				25	

Fifth Semester

S. No	Code	Subject	L-T-P	Credits	Type
	22CPT301	Compiler Design	3-0-0	3	PC
	22CPT302	Cryptography	3-0-0	3	PC
	22CPT303	Operating System	3-0-0	3	PC
	22CPT304	Software Engineering	3-0-0	3	PC
	22CPT305	Emerging Technologies for CS	3-0-0	3	PC
	22CPTxxx	Program Elective-1	3-0-0	3	PE
	22CPP306	Compiler Design Lab	0-0-2	1	PC
	22CPP307	Cryptography Lab	0-0-2	1	PC
	22CPP308	Operating System Lab	0-0-2	1	PC
				21	

Honors

	22CPTxxx	Advance Data Structures and Algorithms		3	
	22CPTxxx	Honors Elective-1*		3	
				6	

Minor CSE

	22CPT106	Data Structures		3	PC
	22CPT303	Operating System		3	PC
				6	

Sixth Semester

S. No	Code	Subject	L-T-P	Credits	Type
	22CPT309	Artificial Intelligence	3-0-0	3	PC
	22CPT310	Computer and Network Security	3-0-0	3	PC
	22CPT311	Digital Image Processing	3-0-0	3	PC
	22CPT312	Parallel and Distributed Computing	3-0-0	3	PC
	22EET313	Smart Grid	3-0-0	3	PLEAS
	22CPTxxx	Program Elective-2	3-0-0	3	PE
	22CPP313	Computer and Network Security Lab	0-0-2	1	PC
	22CPP314	Digital Image Processing Lab	0-0-2	1	PC
	22CPP315	Parallel and Distributed Computing Lab	0-0-2	1	PC
				21	

Honors

	22CPTxxx	Honors Elective-2*		3	
	22CPTxxx	Honors Elective-3*		3	
				6	

Minor CSE

	22CPT211	Computer Networks		3	PC
	22CPT213	Database Information Systems		3	PC
				6	

Seventh Semester

S. No	Code	Subject	L-T-P	Credits	Type
1	22CPS401	Training Seminar	0-0-4	2	PC
2	22CPD402	Minor Project	0-0-6	3	PC
3	22CPTxxx	Program Elective-3	3-0-0	3	PE
4	22CPTxxx	Program Elective-4	3-0-0	3	PE
5	22CPTxxx	Program Elective-5	3-0-0	3	PE
6	22CPTxxx	Program Elective-4 Lab	0-0-2	1	PE
7	22CPTxxx	Program Elective-5 Lab	0-0-2	1	PE
8		Open Elective – 1	3-0-0	3	OE
				19	

Honors

	22CSTxxx	Honors Elective-4*		3	
				3	

Minor CSE

	22CPT304	Software Engineering		3	PC
				3	

Eighth Semester

S. No	Code	Subject	L-T-P	Credits	Type
1	22CPD403	Major Project	0-0-16	8	PC
2	22CPTxxx	Program Elective-6	3-0-0	3	PE
3	22CPTxxx	Program Elective-7	3-0-0	3	PE
4	22CPPxxx	Program Elective-6 Lab	0-0-2	1	PE
5	22CPPxxx	Program Elective-7 Lab	0-0-2	1	PE
6		Open Elective – 2	3-0-0	3	OE
				19	

Honors					
	22CSTxxx	Honors Elective-5*		3	
				3	

Minor CSE					
	22CPT309	Artificial Intelligence		3	PC
				3	

** Honors Elective courses will be taken from PG departmental subject pool*

Semester-wise Scheme and Syllabus

Scheme and Syllabus of 1st Year Institute Core Subjects

Programming with Python				
Prerequisite: Nil	L	T	P	C
Total hours: 28	2	0	0	2

Course Content		Hrs.
Unit 1	Introduction to computer system and binary number systems – addition,subtraction (2's complement), multiplication, left shifting and right shifting.	4
Unit 2	Introduction to Python: Python variables, Python basic Operators, Understanding python blocks. Python Data Types, Declaring and using Numeric data types: int, float etc. Python Program Flow Control Conditional blocks: if, else and else if, Simple for loops in python, for loop using ranges, string, list and dictionaries. Use of while loops in python, Loop manipulation using pass, continue, break and else. Programming using Python conditional and loop blocks.	6
Unit 3	Python Complex data types: Using string data type and string operations, Defining list and list slicing, Use of Tuple data type. String, List and Dictionary.	6
Unit 4	Building blocks of python programs: string manipulation methods, List manipulation, Dictionary manipulation, Programming using string, list and dictionary in-built functions. Python Functions, Organizing python codes usingfunctions, Introduction to classes.	6
Unit 5	Python File Operations: Reading files, Writing files in python, Case study: development of mini projects using libraries like matplotlib, numpy, etc.	6

References	
1.	Wesley J. Chun, "Core Python Applications Programming", 3rd Edition , Pearson Education, 2016.
2.	Charles Dierbach, "Introduction to Computer Science using Python", Wiley, 2015.
3.	Jeeva Jose &P.SojanLal, "Introduction to Computing and Problem Solving with PYTHON", KhannaPublishers, New Delhi, 2016.
4.	Downey, A. et al., "How to think like a Computer Scientist: Learning with Python", John Wiley, 2015.
5.	Mark Lutz, "Learning Python", 5th edition, Orelly Publication, 2013, ISBN 978- 1449355739
6.	John Zelle, "Python Programming: An Introduction to Computer Science", Second edition, CourseTechnology Cengage Learning Publications, 2013, ISBN 978- 1590282410
7.	Michel Dawson, "Python Programming for Absolute Beginers" , Third Edition, Course TechnologyCengage Learning Publications, 2013, ISBN 978-1435455009
8.	David Beazley, Brian Jones., "Python Cookbook", Third Edition, Orelly Publication, 2013, ISBN 978-1449340377

Programming with Python Lab				
Prerequisite:	L	T	P	C
Total hours: 28	0	0	2	1

	Course Content	Hrs.
	The following proposed coverage are broad guiding areas lab. The programs mentioned here just sample programs and they are just for reference purpose. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with concerned theory course. 1. Installation of Python Tool, Introduction to Python programming, and python datatypes [1 Lab] 2. Data types, Input/Output and library imports [1 Lab] 3. Python strings operations, Doc strings [1 Lab] 4. Objects - List, Tuples and Dictionaries [3 Lab] 5. Control flow, functions working and some advanced functions [2 Lab] 6. Python File Operations: Reading files, Writing files in python [1 Lab] 7. Introduction to classes [1 Lab] 8. Numpy, Matlabplotlib utility functions [2 Lab]	

References:	
1.	Core Python Applications Programming: Wesley J. Chun, Pearson Education, 2016.
2.	Introduction to Computer Science using Python: Charles Dierbach, Wiley, 2015
3.	Python for Programmers: Paul J. Deitel, Harvey Deitel , Pearson, 2020 .
4.	Learning Python: Mark Lutz, Orelly Publication, 2013
5.	Python Programming: An Introduction to Computer Science: John Zelle, Course Technology Cengage Learning Publications, 2013.

Scheme and Syllabus of 1st Semester

First Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credit**	**Type**
		Institute Core Subjects		***19***	***IC***
	22CPT101	Discrete Mathematics	3-0-0	3	PC
	22CPT102	Problem Solving using C	2-0-0	2	PC
	22CPP103	Problem Solving Using C Lab	0-0-2	1	PC
				25	

Discrete Mathematics				
Prerequisite: Nil	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Logic: Truth Tables, Conditionals ($P \Rightarrow Q$), and Bi-conditionals ($P \Leftrightarrow Q$), Negation, Converse, and Contrapositive, Existential and Universal Quantifiers ($\forall$,$\exists$, $\exists!$), Proof Techniques (Contrapositive, Contradiction, Induction), Counterexamples, and Proving Statements with Quantifiers, Predicate logic, first order logic, Logical Inferences.	8
Unit 2	Set Theory: Sets and Set Notation, the Empty Set, the Power Set, Cardinality rules and infinite sets, Union, Intersection, Complement, Subsets , Proving sets are equal, Axioms of Naïve Set Theory.	6
Unit 3	Relations: Cartesian Products and Relations, Equivalence Relations and Partitions, Partial Orderings, Lattices.	6
Unit 4	Functions: Definition of a Function, Domains and Co-domains, Composition and Inverses, Well-Defined, Injective, Surjective, and Bijective Functions, Recurrence Relations, Generating functions.	6
Unit 5	Abstract Algebra: Groups-Binary operation, and its properties, Definitionof a group, Groups as symmetries, cyclic, dihedral, symmetric, matrix groups, Subgroups, Cosets, normal subgroups and quotient groups, Conjugacy classes, Lagrange's theorem, Monoid.	8
Unit 6	Number Theory: Prime Numbers, Euclid’s Algorithm for GCD, The GCD-LCM product theorem, Extended Euclid’s Algorithm, Linear DiophantineEquations, Modular Arithmetic, Chinese Reminder Theorem, FastModular Exponentiation, Fermat’s little theorem, Euler’s totient theorem,Euler’s theorem.	8

References	
1.	Ronald L. Graham, Donald E. Knuth, Oren Patashnik ,Concrete Mathematics: A Foundation for ComputerScience (2nd Edition)
2.	K. Rosen, Discrete Mathematics and Its Applications, 7th edition, McGraw-Hill, 2011.
3.	M. Lipson, Schaum’s Outline of Discrete Mathematics, revised 3rd edition, 2009.
4.	D. Velleman, How to Prove it: A Structured Approach. Cambridge University Press, 1994

Problem solving using C				
Prerequisite: :NiL	L	T	P	C
Total hours: 28	2	0	0	2

	Course Content	Hrs
Unit 1	Introduction to Computers, Basic Computer Organization, Computational Thinking and problem solving, Planning the Computer Program - Debugging, Types of errors, Techniques of Problem. Aspects of programming language: Syntax, semantics. System Software, Application Software. Compiler - Compilation process -Compiler and interpreter. Basics: C language introduction, C language Standards, Data Types and Storage Classes: Different data types, Storage Classes – auto, static, extern,register. Reserved words, operators, constants in C, identifiers, printf/scanf (formatted printf/scanf), assignment statement, built-in data types – int, char, float, double; usage of sizeof(), integer arithmetic, typecasting	6
Unit 2	IF/IF..ELSE control construct through maximum of two numbers, ternary operator for maximum of three numbers. SWITCH statement through figure to words problem Swapping of variables, Solving problem of gcd of two numbers. Introduction to 1D arrays in C, implementation of strings as char array,string function implementation: example problem could be palindrome. Loop constructs: significance of initialization, terminating condition and increment/decrement (pre/post increment/decrement operator usage). Usage of FOR/WHILE/DO..WHILE in problems like sum /maximum/deviation of N numbers. Illustration of loops for solving computation of sin of a number	8
Unit 3	Problem Solving: Sorting an array consisting of zeros and ones, Partitioning an array, merging two sorted arrays, computation of square root of a number Recurrence through Factorial problem, binary search to illustrate divide and conquer approach, Fibonacci through recursion and problems with thisapproach, Fibonacci through storing previous values – introduction to dynamic programming, Nested loops through sorting methods; use of break and continue Bit vector implementation of set and usage of bitwise operators for testingmembership (withing set), union and intersection of two sets. Macro & Preprocessor in C	
Unit 4	Structures in C: struct and typedef through implementation of complex numbers Functions: Passing arguments in main() function, Call by value, Call by reference. Function for implementing raising a number to large power (logarithmic complexity) Multi-dimensional array (example problem can be matrix transpose/ addition) Command line arguments in C Passing variable number of arguments	6
Unit 5	Pointers: Introduction to pointers, pointer arithmetic, void *, pointers v/s array, malloc() – case study linked list. Pointer to array versus array of pointers, pointers to structures, array of pointers, Pointer to functions. Enum operator. File Handling in C: Basics of working with text files, File read, write, append and other similar operations.	8

References

1.	Education Solutions Limited, I. T. L. (2004). Introduction to Computer Science. India: Pearson Education.
2.	How to Solve it by Computer, RG Dromey, PHI
3.	The C Programming Language, Brian W. Kernighan and Dennis Ritchie, Latest Edition, Prentice Hall.
4.	Programming in ANSI C, E. Balagurusamy, Latest Edition, McGraw Hill
5.	Let us C, YashavantKanetkar, Latest Edition, BPB Publication

Problem solving using C Lab					
Prerequisite:		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs.
Unit 1	The following proposed coverage are broad guiding areas lab. The programs mentioned here just sample programs and they are just for reference purpose. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with concerned theory course. 1. Basic C commands and First C program-printing hello world on the screen, programs related to basic arithmetic operations, swapping of numbers etc. (2 lab) 2. C Expressions: Programs involving concepts of C expressions like finding roots of quadratic equation, area of circle and simple interest calculation. (1 lab) 3. C operators: Programs requiring in-depth knowledge of various C operators (especially conditional operator, bitwise operators and sizeof operator). (1 lab) 4. Conditional statements: Programs with applications of c conditional statements: if, if else, nested if else, switch-case (1 lab) 5. Arrays and Loops: C programs for performing various operations (finding maximum, second-maximum, minimum, reversing an array etc) on 1-D arrays and Applications of concepts of loops (leap year, palindrome, displaying prime numbers etc). (2 lab) 6. Functions and Recursions: Programs demonstrating use of functions (like adding N numbers, calculator etc) and Recursion (factorial, Fibonacci, GCD, binary search etc). (1 lab) 7. Strings, Pointers and Structures: Programs related to the following concepts: String manipulations, pointer to arrays, and pointer to functions and Structures (3 lab) File Management: Programs related to file handling (Finding the number of characters, words and lines of given text file and File handling programs) (1 lab)				
References:					
1.	Education Solutions Limited, I. T. L. (2004). Introduction to Computer Science. India: Pearson Education.				
2.	How to Solve it by Computer, RG Dromey, PHI				
3.	The C Programming Language, Brian W. Kernighan and Dennis Ritchie, Latest Edition, Prentice Hall.				
4.	Programming in ANSI C, E. Balagurusamy, Latest Edition, McGraw Hill				
5.	Let us C, Yashavant Kanetkar, Latest Edition, BPB Publication				

Scheme and Syllabus of 2nd Semester

Second Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credit**	**Type**
		Institute Core Subjects		***18***	***IC***
	22CPT104	Data Structures	3-0-0	3	PC
	22CPT105	Logic System Design	2-0-0	2	PC
	22CPP106	Data Structures Lab	0-0-2	1	PC
	22CPP107	Logic System Design Lab	0-0-2	1	PC
				25	

<table>
<tr><th colspan="6">Data Structures</th></tr>
<tr><td colspan="2">Prerequisite: :Nil</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4">Fundamentals of Data Structures, Memory Allocation, Abstract DataTypes, Arrays, Lists Stack Implementation, Stack applications.
Queue Implementation, Sequential, Circular, and Dequeue representation, Dynamic Queue implementation, Queue applications.</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Searching and Sorting: Linear and Binary search, Bubble Sort, Selection Sort, Insertion Sort, Merge sort, Quick sort, Counting sort, Bucket sort, Radix sort, Heap sort, comparisons of sorting algorithms.</td><td>8</td></tr>
<tr><td>Unit 3</td><td colspan="4">Hashing and Hash Tables: Hash functions, Open and closed hashing, Dynamic and extendible hashing, Hash collision, chaining, Hash Tables and Probing Techniques</td><td>8</td></tr>
<tr><td>Unit 4</td><td colspan="4">Trees: Binary Tree and its representations, Tree traversal, Binary Search Tree, Threaded binary trees, Representing list as binary trees, Dynamic implementation of Binary tree and AVL tree, Tree applications, Interval tree, M-way search Tree, B-Tree and its variants , B+ Tree , Heaps and itsapplications</td><td>10</td></tr>
<tr><td>Unit 5</td><td colspan="4">Graphs: Fundamentals of Graph, Adjacency Matrix and List; GraphTraversal using DFS and BFS. Dijkstra and Prims algorithms.</td><td>8</td></tr>
</table>

References	
1.	T. Cormen, C. Lieserson, R. Rivest, and C. Stein, “Introductions to Algorithms”, Prentice-Hall/India,3rd edition, 2009
2.	Aaron M. Tenenbaum, Y. Langsam, Moshe J. Augenstein, Data Structures Using C
3.	Introduction to Algorithms ,Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and CliffordStein,PHI,2nd Edition.
4.	Aho A.V., J.E. Hop croft, J.D. Ullman, Data Structures and algorithms, Addison Wesley
5.	Introduction to design & Analysis of Algorithms,Anany Levitin,2ndEdition,Pearson.

Logic System Design					
Prerequisite:		L	T	P	C
Total hours: 28		2	0	0	2
Course Content					Hrs.
Unit 1	Number Systems and Codes: Representation of Negative Numbers; 1’s Complement and 2’s Complement, Complement Arithmetic, BCD Arithmetic, Digital Codes - Excess-3 code, Gray code, Binary to Excess- code conversion and vice versa, ASCII code, EBCIDIC code, Error Detection Codes				7
Unit 2	Logic Gates, Universal Gates and their characteristic: K-Map, SOP, POS				4
Unit 3	Combinational circuits: Adders, Subtractors, Binary Parallel Adder – Carry look ahead Adder, BCD Adder, Multiplexer, Demultiplexer, Comparator, Decoder and Encoder.				4
Unit 4	Sequential Circuits: Latches, Flip-Flops: RS, D Type, JK, and T Type and their conversion, Master-Salve Flip and Race Conditions.				5
Unit 5	Registers: Design of shift registers and their operations.				4
Unit 6	Counters: Asynchronous and Synchronous counters, Applications of counters.				4
References:					
1.	Herbert Taub, Donald L. Schilling, Digital Integrated Electronics, McGraw-Hill.				
2.	Fredrick J. Hill, Gerald R. Peterson, Computer aided logical design with emphasis on VLSI, Wiley				
3.	M. Morris Mano, Digital Logic and Computer Design, Person Education.				
4.	Malvino & Leach, Digital Principles and Applications				
5.	R P Jain, Modern Digital Electronics				

<table>
<tr><th colspan="6">Data Structures Lab</th></tr>
<tr><td colspan="2">Prerequisite:</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 28</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><th colspan="5">Course Content</th><th>Hrs.</th></tr>
<tr><td>Unit 1</td><td colspan="4">The following topics are broad areas. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with concerned theory courses.
Arrays and Linked List: Programs involving creation of arrays, singly, double and circular linked list and performing various operations on them (updating/adding/deletion an element in the begin/middle/end of the list and linear search). Stack and Queue: Programs involving implementationsand applications of stacks and queues (array and linked list implementations, Dynamic Queue implementation, applications like balanced brackets problem, infix to postfix conversion)</td><td></td></tr>
<tr><td>Unit 2</td><td colspan="4">Comparison based sorting algorithms: Programs requiring in-depth knowledge of various comparison based sorting algorithms (bubble, insertion, merge, quick etc). Linear time sorting algorithms: Programs with applications of linear time sorting algorithms (counting sort, radix sort, bucket sort). Heaps: Programs involving creation of heap from the given list of elements, conversion of min heap to max heap (and vice versa), heap-sort</td><td></td></tr>
<tr><td>Unit 3</td><td colspan="4">Hashing: Programs demonstrating applications of hashing and hash functions (like phonebook problem). Tree traversals and binary search tree: Programs related to in order, pre order and post order traversals, creation of binary search tree, searching/inserting/deletion in binary search tree.</td><td></td></tr>
<tr><td>Unit 4</td><td colspan="4">AVL trees and B+ Trees: Programs in which efficient implementation of various key operations (insertion/deletion/updation/searching) on AVL trees and B+ Trees are required.</td><td></td></tr>
<tr><td>Unit 5</td><td colspan="4">Graphs: Programs demonstrating implementations and applications of graph traversal methods (BFS, DFS) and minimum spanning tree problem (Dijkstra and Prims algorithms)</td><td></td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Introductions to Algorithms, T. Cormen, C. Lieserson, R. Rivest, and C. Stein,, Prentice Hall/India, 3rd edition, 2009.</td></tr>
<tr><td>2.</td><td colspan="5">Introduction to Algorithms, , Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 3rd Edition, 2009</td></tr>
<tr><td>3.</td><td colspan="5">Fundamentals of Data Structures in C++, Ellis Horowitz, Sartaj Sahni and Dinesh P. Mehta, Galgotia Press, 2009</td></tr>
</table>

Logic System Design Lab					
Prerequisite:		L	T	P	C
Total hours: 42		0	0	2	1
Course Content					Hrs.
	Lab 1. Design and test a 2-bit and 4-bit half adder. Lab 2. Design and test a 2-bit and 4-bit adder (ripple, carry look ahead). Lab 3. Design and test of encoder/decoder (binary-gray, self-complementing). Lab 4. Design and test of parity generator and detector. Lab 5. Design and test of one bit error detecting and correcting circuit. Lab 6. Design and test of a 2-bit multiplier. Lab 7. Design and test of n -bit comparator. Lab 8. Design and test of flip flops – RS/JK/D/T. Lab 9. Design and test of SISO and PIPO shift registers. Lab 10. Design and test of counters. Lab 11. Implementation and simplification of k -map (upto 3 variables) Lab 12. Implementation of Quine-Mckluskey's method.				
References:					
1.	Herbert Taub, Donald L. Schilling, Digital Integrated Electronics, McGraw-Hill.				
2.	Fredrick J. Hill, Gerald R. Peterson, Computer aided logical design with emphasis on VLSI, Wiley				
3.	M. Morris Mano, Digital Logic and Computer Design, Person Education.				
4.	Malvino & Leach, Digital Principles and Applications				
5.	R P Jain, Modern Digital Electronics				

Scheme and Syllabus of 3rd Semester

Third Semester					
S. No	Code	Subject	L-T-P	Credits	Type
	22CPT201	Data Communications	3-0-0	3	PC
	22CPT202	Design and Analysis of Algorithms	3-0-0	3	PC
	22CPT203	Digital Circuits and Microprocessors	3-0-0	3	PC
	22CPT204	Foundation of Learning	3-0-0	3	PC
	22CPT205	Object Oriented Analysis and Design	3-0-0	3	PC
	22CPT206	Technical Writing	1-0-0	1	PC
	22HST201	Social Sciences and Professional Ethics	3-1-0	4	BS
	22CPP207	Design and Analysis of Algorithms Lab	0-0-4	2	PC
	22CPP208	Digital Circuits and Microprocessors Lab	0-0-2	1	PC
	22CPP209	Object Oriented Analysis and Design Lab	0-0-2	1	PC
	22CPP210	Technical Writing Lab	0-0-2	1	PC
				25	

Data Communications				
Prerequisite: Nil	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to Data Communication: Overview of communication systems, Analog vs. digital communication, Elements of a digital communication system, Data Representation Communication channels and noise	6
Unit 2	Physical Layer: Signals- representation, sampling, aliasing, quantization, transformations; filters, spectral analysis. Analog and Digital transmissions, conversions. Pulse transmission over Band limited signals, sampling theory; Pulse Modulations, metrics - bit transmission, signalling rate, error probability, S/N ratio, bandwidth requirement. Other Modulation.	8
Unit 3	Data Communication Network: OSI Model Physical layer: Importance, bit, bit rate, Signal Rate, bit Interval, Data Rate: Shannon Capacity, Nyquist bit rate, Line Configuration & Line Coding Schemes, Data Transmission: simplex, half-duplex or full-duplex mode, Topology, Signals: Period, frequency, phase, bandwidth.	10
Unit 4	Transmission Media: Guided and Unguided Media, Transmission Medium, Transmission Impairments; Multiplexing- FDM & TDM; Switching: Circuit, Message, Packet, Datagram, Virtual Networks, and DSL. Fiber Optic Communication Principles, loss, dispersion, light source and detectors. Spread spectrum, Multiple access: reservation based – TDMA, FDMA, CDMA; random access – Aloha, CSMA, ISMA; hybrid schemes; digital modulation schemes. Examples: Token Bus, Token Ring, Ethernet, including Gigabit Ethernet and Wi-Fi (802.11).	10
Unit 5	Data Link Layer: Flow Control, Error, Error Control; Error Detection: Simple parity checking, 2D Parity Checking, arithmetic checksum, CRC. Error Correction Codes: Information theory, Shannon's theorem, Source coding, error control coding, (Block codes, Cyclic codes, Linear code, checksum). Flow Control, Sliding Window and Stop and Wait protocols.	8

References	
1.	Forouzan, Data Communications and Networking, McGraw Hill.
2.	Tanenbaum, Computer Networks, Pearson India.
3.	Haykins, Analog and Digital Communications, Wiley Publications.
4.	Haykins, Digital Communication Systems, Wiley Publications.
5.	B.P.Lathi: Modern Digital Communication, Oxford.

<table>
<tr><th colspan="6">Digital Circuits and Microprocessors</th></tr>
<tr><td colspan="2">Prerequisite: Nil</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><th colspan="5">Course Content</th><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4"><b>Sequential Circuits:</b> Fundamental Mode Circuits Analysis, Synthesis of Flow Tables, Minimization, Transition Tables, Excitation maps, Output Maps, <b>C</b>lock Input Control, Extended State Table, Program Description, Synthesis, vector Operations, Logical Functions of Vectors, Incompletely Specified Sequential Circuits</td><td>20</td></tr>
<tr><td>Unit 2</td><td colspan="4"><b>Microprocessor:</b> Introduction to x86 microprocessor, Addressing Modes, Instruction Set, Code Conversation, Directives, Control Operations, String Manipulation Operations. Programming in assembly Language. Interfacing Devices, 8255 PPI, 8259 PIC, 8237/8257 DMA Controller, 8279 Keyboard and display Controller, A/D converters, USB, DMA, Timing and delay, Stack and Subroutine, interrupts, Assembly Language programming,</td><td>22</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Herbert Taub, Donald L. Schilling, Digital Integrated Electronics, McGraw-Hill,</td></tr>
<tr><td>2.</td><td colspan="5">Fredrick J. Hill, Gerald R. Peterson, Computer aided logical design with emphasis on VLSI, Wiley</td></tr>
<tr><td>3.</td><td colspan="5">Fredrick J. Hill, Gerald R. Peterson, Introduction to Switching Theory and Logical Design, Wiley</td></tr>
<tr><td>4.</td><td colspan="5">M. Morris Mano, Digital Logic and Computer Design, Person Education</td></tr>
<tr><td>5</td><td colspan="5">Lance A. Leventhal, S. Cordes, Assembly language subroutines for the 8086, McGraw-Hill Book Co.</td></tr>
<tr><td>6</td><td colspan="5">Randall Hyde, The Art of Assembly Language, 2nd Edition</td></tr>
<tr><td>7</td><td colspan="5">The 80x86 Family: Design, Programming, and Interfacing, Pearson, John Uffenbeck.</td></tr>
</table>

Design and Analysis of Algorithms

Prerequisite: Data Structures	L	T	P	C
Total Hours: 42	3	0	0	3

	Course Content	Hrs
Unit 1	***Algorithm Analysis:*** Asymptotic notation, model of computation, time and space complexities, average and worst-case analysis, Master's Theorem, solving recurrence equations- interation method, substitution, recursion tree, master method. Amortised Analysis. Linear Search, Insertion Sort, Euclid's Algorithm for finding GCD (Lame's Theorem): Correctness, Best-Case, Average-Case and the Worst-Case Running Time Analysis. Permutation Model for Average-Case Analysis of an Algorithm for Finding Maximum Element in an Array	8
Unit 2	***Divide and Conquer*:** General recurrence and methods for obtaining bounds on given recurrence. Binary Search, Merge Sort, and Maximum Subarray Sum Problem. Quick-sort: Correctness, Running Time Analysis, Order statistics - finding median and Worst-case Linear Time Algorithm for Selection Problem. Max-Min problem, Strassen's Algorithm for Matrix Multiplication, Karatsuba`s Algorithm for Large Integer Multiplication	8
Unit 3	***Dynamic Programming Approach:*** Introduction to dynamic programming - principal of optimality, Optimal substructure. Matrix Chain Multiplication Problem, Optimal Binary Search Tree Problem, Longest Common Subsequence Problem, 0/1 Knapsack Problem. ***Greedy Approach:*** Elements of Greedy Strategy - Greedy choice property, optimal substructure. Example Problems - Activity Selection Problem, Fractional Knapsack Problem, Huffman codes, Travelling Salesman Problem.	9
Unit 4	***Graph Algorithms:*** Graph Traversal Algorithms (BFS, DFS), Shortest path algorithms (Bellman-ford, Dijkstra's, Transitive-Closure, Floyd-Warshall), minimum spanning tree algorithms Kruskal, Prim), Network-flow (ford-fulkerson) , applications of DFS:- bi-connectivity, topological sort, strongly-connected components, Articulation point.	9
Unit 5	***Backtracking:*** Introduction to Backtracking, Enumerating Independent Sets of a Graph, Graph Coloring Problem and N-Queen's Problem. ***Complexity Classes:*** P, NP, NP-Hard and NP-Complete. NP-Complete Examples with Reductions: Satisfiability, Clique, Independent Set, Vertex Cover , Graph Coloring, Dominating Set,	8

References	
1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.
2.	Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
3.	Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.

4.	Michael R. Garey and David S. Johnson, Computers and Intractability: A Guide the theory of NP-Incompleteness, W.H. Freeman & Co., 1979.
5.	Herbert S. Wilf, Algorithms and Complexity, AK Peters Ltd., 2003.
6.	Jon Kleinberg and Eva Tardos. 2005. Algorithm Design. Addison-Wesley Longman Publishing Co., Inc., USA.

Foundations of Learning				
Prerequisite: Some basic set theory (what is a set and elementary set operations), combinatorics (knowing different ways of counting, inclusion-exclusion principle) and calculus (knowing derivatives and integrals)	L	T	P	C
Total hours: 40	3	0	0	3

	Course Content	Hrs
Unit 1	Linear Algebra i. Scalars, Vectors, Matrices and Tensors ii. Multiplying Matrices and Vectors iii. Identity and Inverse Matrices iv. Linear Dependence and Span v. Norms vi. Special Kinds of Matrices and Vectors vii. Eigen decomposition viii. Singular Value Decomposition ix. The Moore-Penrose Pseudoinverse x. The Trace Operator xi. The Determinant xii. Principal Component Analysis	12
Unit 2	Probability and Information Theory i. Random Variables ii. Probability Distributions iii. Marginal Probability iv. Conditional Probability v. The Chain Rule of Conditional Probabilities vi. Independence and Conditional Independence vii. Expectation, Variance and Covariance viii. Common Probability Distributions ix. Useful Properties of Common Functions x. Technical Details of Continuous Variables xi. Information Theory xii. Structured Probabilistic Models	12
Unit 3	Statistical inference: statistical decision theory, statistical assumptions, estimation theory. Methods of estimation: method of moments, method of minimum variance	9
Unit 4	Statistical hypothesis testing, null and alternate hypotheses. Simple and composite hypotheses, Type-I and Type-II errors, Z-tests for difference of means, chi-square test, tests for correlation and regression.	9

References	
1.	Linear Algebra, Gilbert Strang, MIT Cambridge Press
2.	Foundations of Learning, Julie Fisher, Open University Press
3.	Foundations of Learning, Laurie L. Hazard, Jean-Paul Nadeau, Pearson
4.	Deep learning, Ian Goodfellow, MIT Cambridge Press
5.	Probability and Statistics for Machine Learning, Anirban Das Gupta, Springer

Object Oriented Analysis and Design					
Prerequisite: : **Computer Programming skills**		L	T	P	C
Total hours: 40		3	0	0	3

Course Content			Hrs
Part I	Unit 1	Introduction to Object Oriented Programming fundamentals: Object Oriented Programming and Design, Review of abstraction Classes, Objects and Methods: Class, Object, Object reference, Constructor, Constructor Overloading	8
	Unit 2	C++ Programming Basics: Fundamentals, variables and assignments, Input and Output, Data types and expressions, flow of control, subprograms, top-down design, predefined functions, user defined functions, procedural abstractions, local variables, overloading function names, operator overloading, parameter passing, this pointer, destructors, copy constructor, overloading the assignment operator, virtual functions, function calling functions, friend functions, recursive functions, recursive member functions. Static member function.	6
	Unit 3	C++ Object oriented concepts: Objects and classes, use of file for I/O, formatting output with stream functions, Character I/O, inheritance, structures for diverse data, structures as function arguments, initializing structures, defining classes and member functions, public and private members, constructors for initialization, standard C++ classes, derived classes, flow of control, use of Boolean expressions, multiway branches, use and design of loops. Friend function and friend class	8
Part II	Unit 4	Introduction to OOD, Unified Process, UML diagrams, Use Case, Use case Modelling, Relating Use cases – include, extend and generalization – When to use Use-cases, Class Diagram, Elaboration-Domain Model, Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement, Finding conceptual class Hierarchies, Aggregation and Composition – Relationship between sequence diagrams and use cases, When to use Class Diagram	6
	Unit 5	Dynamic Diagrams – UML interaction diagrams, System sequence diagram, Collaboration diagram, When to use Communication Diagrams, State machine diagram and Modelling –When to use State Diagrams, Activity diagram, When to use activity diagrams Implementation Diagrams-UML package diagram – When to use package diagrams, Component and Deployment Diagrams – When to use Component and Deployment diagrams	6
	Unit 6	Design patterns: GRASP: Designing objects with responsibilities – Creator – Information expert – Low Coupling – High Cohesion – Controller Design Patterns – creational – factory method – structural – Bridge – Adapter – behavioural – Strategy – observer –Applying GoF design patterns – Mapping design to cod	6

References	
1.	Deitel and Deitel, C++ How to Program, Third Edition, Pearson Publication.
2.	Robert Lafore, Object Oriented Programming in C++, Fourth Edition, SAMS publications.
3.	Craig Larman, ―Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development‖, Third Edition, Pearson Education, 2005.
4.	Ali Bahrami – Object Oriented Systems Development – McGraw Hill International Edition – 1999

<table>
<tr><th colspan="6">Technical Writing</th></tr>
<tr><td colspan="2">Prerequisite: :NiL</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours:</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td></td><td colspan="4">Introduction to Documentation using Doxygen, Google Docs, Latex/ Overleaf.
Drawing software: inkscape, xfig, open-office, and/or similar.
Presentation using Beamer: Introduction to creating slides, adding frames, dividing the slide into multiple columns, adding different blocks, and similar.
Graph plotting software (e.g., gnuplot). Version control tools - GIT /GitHub/SVN</td><td>7</td></tr>
<tr><td></td><td colspan="4">Introduction: LaTeX, its installation, and different IDEs. The learner creates the first document using LaTeX, organizes content into sections using article and book class of LaTeX.
Styling Pages: Reviewing different paper sizes, examines packages, formats the page by setting margins, customizing header and footer, changing the page orientation, dividing the document into multiple columns. Different types of error messages.
Formatting Content: formatting text (styles, size, alignment), adding colors to text and entire page, and adding bullets and numbered items, the process of writing complex mathematics.
Tables and Images: creating basic tables, adding simple and dashed borders, merging rows and columns, and handling situations where a table exceeds the size of a page. Add an image, explore different properties like rotate, scale, etc..
Referencing and Indexing: the learner learns to add cross-referencing (refer to sections, table, images), add bibliography (references), and create back index.</td><td>7</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Latex - A document preparation system, 2/e, by Leslie Lamport, Addison-Wesley, 1994</td></tr>
<tr><td>2.</td><td colspan="5">https://www.doxygen.nl/</td></tr>
</table>

Social Sciences and Professional Ethics				
Prerequisite: Nil	L	T	P	C
Total hours: 56	3	1	0	4

Course objectives:

- Augmenting the understanding of society, societal issues and problems
- To provide the students an insight into the multifaceted economic and financial environment
- Development of a positive character, empathetic human being, responsible citizen
- Inculcating a positive work culture respecting professional ethics

	Course Content	Hrs
Unit 1	**Introducing Sociology** Meaning, scope and evolution of Sociology, Key theoretical trajectories. Society, community, Social Institutions, Social Groups, Socialisation and Culture, Norms and Values, Agency and structure	10
Unit 2	**Social Change** Social Change, development and progress; Globalisation, Industrialisation, urbanisation and modernisation; Social mobility and social stratification	8
Unit 3	**Social Issues** Science technology and society; Digital divide, Appropriate technology, Gender inequality; Substance abuse, Consumerism, Environmental degradation and climate crisis, Nation building	10
Unit 4	**Socio-economic environment** Overview of Socio-economic policy environment; PESTLE analysis. Economic growth & development; primary, secondary and tertiary sectors; structural changes & emerging sectors of the Indian economy. Design and strategy of economic reforms and liberalization: India’s growth post liberalization.	10
Unit 5	**Finance and banking** Banking and Financial Sector; Reforms & Challenges; Monetary & Fiscal Policies; meaning, importance & instruments. Global economic environment and opportunities. Intellectual property rights and R & D environment.	6
Unit 6	**Ethics and values** Professional Ethics: Need, importance and principles of Professional ethics, Ethics in relation with use of technology and technology development, diversity inclusion and equity; Social responsibility. Constitutional values: Preamble and DPSP, Rights and duties	12

References

1. Haralambos, Michael & Holborn, Martin. Sociology: Themes and Perspective, Harper Collins. Eighth edition. 2014.
2. Ritzer, George. Sociological Theories, McGraw-Hill; Fifth edition. 2011
3. Lillie, William. An introduction to Ethics Allied Publishers Pvt. Ltd.; 1st edition (1967)
4. Lama, Dalai. Ethics for the New Millennium by the. Riverhead Books; Reissue edition (2001)
5. Uma Kapila, Indian Economy Performance and Policies, Academic Foundation, New Delhi
6. Ahluwalia, I.J. & IMD Little, India’s Economic Reform and Development, Oxford University Press.

Design and Analysis of Algorithms Lab					
Pre-requisite: C Programming, Data Structures		L	T	P	C
		0	0	4	2
Course Content					
	1. Implementation of various sorting and searching algorithms (Revision) 2. Implement quick sort with three different positions of pivot element- first, last, random 3. Implement Tree traversal, and graph traversal (recursive algorithms) 4. Implement deterministic and randomized selection problem 5. Implement maximum subarray sum problem 6. Implement Karatsuba`s Algorithm for Large Integer Multiplication 7. Implement matrix chain multiplication, longest common sub-sequences, 0/1 knapsack 8. A program to obtain the topological ordering of vertices in a given digraph. 9. Implement travelling salesman problem. 10. Print all the nodes reachable from a given starting node in a digraph using BFS method. 11. Check whether a given graph is connected or not using DFS method. 12. Find minimum cost spanning tree of a given undirected path using a Prim's algorithm. 13. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.				
References					
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.				
2	Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.				
3	Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.				

Digital Circuits and Microprocessors Lab

Prerequisite: Basics of Digital Logic	L	T	P	C
	0	0	2	1

Course Content

1. Design and synthesis of pulse mode and clock mode circuits
2. Design and synthesis of clock mode circuits
3. Design and Simplification of incompletely specified sequential circuits
4. SPICE simulator based design and evaluation
5. X86 based Assembly language Programming
    a. Athematic and logical operations
    b. String manipulation
    c. Number Conversion
    d. Implementing a real-time clock.
    e. Writing a program to control and monitor external devices through parallel port communication.
    f. Developing a program to interface with a APC DAC.
    g. Creating a program to implement a basic graphical user interface (GUI) using assembly language.
    h. Implementing a program to perform image processing operations, such as image filtering or edge detection.
    i. Designing a program to control and monitor a robotic arm using assembly language.
    j. Developing a program to interface with an external memory device, such as EEPROM or flash memory.
    k. Writing a program to implement a basic text-based adventure game using assembly language.
    l. Developing a program to interface with a graphical LCD display using the 8086 microprocessor.
    m. Creating a program to perform digital signal processing (DSP) operations, such as filtering or Fourier transforms.

References

1.	Computer Aided Logical Design with Emphasis on VLSI, 4th Edition, Frederick J. Hill, Gerald R. Peterson, by Wiley.
3.	The 80x86 Family: Design, Programming, and Interfacing, 3rd Edition by John E. Uffenbeck, Pearson Education

Object Oriented Analysis and Design Lab				
Prerequisite: : Basic Computer Programming, Data Structures	L	T	P	C
Total hours: 42	0	0	2	1

	Course Content	Hrs
	The laboratory course will be in two parts: Part I: Object oriented Programming (OOP). In this part, students will be given OOP assignments to cover the practical on: - Programs Using Functions - Simple Classes for understanding objects, member functions and Constructors - Classes with primitive data members - Classes with arrays as data members - Classes with pointers as data members – String Class - Classes with constant data members - Classes with static member functions - Compile time Polymorphism - Operator Overloading including Unary and Binary Operators - Function Overloading - Inheritance and - Runtime Polymorphism Part II: Object Oriented Methodology and Design In this part, students will be given experiments to design various UML diagrams such as USE case, Class Diagram, State Diagram, Activity Diagram, Sequence diagram etc., based on the given case study.	30

References	
1.	Deitel and Deitel, C++ How to Program, Third Edition, Pearson Publication.
2.	Robert Lafore, Object Oriented Programming in C++, Fourth Edition, SAMS publications.
3	Craig Larman, ―Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development‖, Third Edition, Pearson Education, 2005.
4	Ali Bahrami – Object Oriented Systems Development – McGraw Hill International Edition – 1999

<table>
<tr><th colspan="6">Technical Writing Lab</th></tr>
<tr><td colspan="2">Prerequisite:</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 28</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4">• Documentation using Google Docs and its usage for creating PPTs.
• Introduction to Documentation using Doxygen and Graph plotting software (e.g., matplotlib in Python)
• Using Drawing software (e.g. draw.io, open-office)
• Documentation using Latex/ Overleaf
o Presentation using Beamer: Introduction to creating slides, adding frames, dividing the slide into multiple columns, adding different blocks, etc.</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Introduction: LaTeX and its installation. The learner creates the first document using LaTeX, organizes content into sections using article and book class of LaTeX.
Styling Pages: Reviewing different paper sizes, examines packages, formats the page by setting margins, customizing header and footer, dividing the document into multiple columns.</td><td>8</td></tr>
<tr><td>Unit 3</td><td colspan="4">Formatting Content: formatting text (styles, size, alignment), the process of writing complex mathematics.
Tables and Images: creating basic tables, adding simple and dashed borders, merging rows and columns.</td><td>6</td></tr>
<tr><td></td><td colspan="4">Referencing and Indexing: the learner learns to add cross-referencing, add bibliography (references).</td><td>6</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Latex - A document preparation system, 2/e, by Leslie Lamport, Addison-Wesley, 1994</td></tr>
<tr><td>2.</td><td colspan="5">https://www.doxygen.nl/</td></tr>
<tr><td>3.</td><td colspan="5">Other online resources</td></tr>
</table>

Scheme and Syllabus of 4th Semester

Fourth Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credits**	**Type**
	22CPT211	Computer Networks	3-0-0	3	PC
	22CPT212	Computer Organization and Architecture	3-1-0	4	PC
	22CPT215	Theory of Computation	3-1-0	4	PC
	22CPT213	Database Information Systems	3-0-0	3	PC
	22CPT214	Machine Learning	3-0-0	3	PC
	22BMT201	Basics of Managements	3-0-0	3	MM
	22CPP216	Computer Networks Lab	0-0-4	2	PC
	22CPP217	Database Information Systems Lab	0-0-4	2	PC
	22CPP218	Machine Learning Lab	0-0-2	1	PC
				25	

Computer Networks				
Prerequisite: Data communication.	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit - I	**Introduction:** Internet – nuts and bolts, network service, network protocols, network edge, network core, performance metrics- delay, throughput, etc. protocols and service models.	4
Unit - II	**End-to-End protocols and Applications-I:** Application layer: principles of application layers, Domain Name System (DNS), HTTP, FTP, E-mail, www and etc. Pear to pear systems, video streaming, Socket programming. Flow control – window/credit schemes, rate control schemes, Congestion control Transport layer and TCP/IP. Introduction to ATM networks and Network Management And Interoperability.	8
Unit - III	**End-to-End protocols and Applications-II :** Introduction to transport layer, multiplexing and de-multiplexing, connection oriented and connection less end to end protocols, principles reliable data transfer, and congestion control.	11
Unit - IV	**Data Plane :** Introduction to network layer, layer 3 devices and inside, addressing – IPv4, IPv6, etc. NAT, **Control Plane :** Retransmission algorithms. Stability of queuing systems.. High speed switches scheduling, BroaPCast routing and spanning trees. Shortest path routing. Distributed routing algorithms, optimal routing, and traffic engineering. ICMP, SNMP,etc	11
Unit - V	**Future/Advanced Internet:** Internet of Things (IoT) and applications, Software Defined Networks (SDN) : Control plane, data-plane, and issues, Information centric networks (ICN), Content distribution networks (CDN) and Future Internet.(5 Classes)	6

References	
1.	Data Networks: Bertsekas and Gallagher, PHI
2.	Computer Networks: L. Peterson and Davie, Elsevier
3.	Computer Networking A top down Approach: J.F.Kurose, Pearson
4.	Computer Networks : Andrew S. Tanenbaum, Pearson

Computer Organization and Architecture				
Prerequisite: Nil	L	T	P	C
Total hours: 40	3	1	0	4

Course Content		Hrs
Unit 1	Organization of Computer Systems: CPU, Memory and I/O organization, Instruction encoding and addressing modes. Von-neumann versus Harvard Architecture, RISC and CISC architectures. Flynn Classification, Stack machines, subroutine calls, allocation and evaluation of data in stack machines. SIMD, SPMD and MIMD	8
Unit 2	CPU Organization: Addressing techniques, Instruction formats, Instruction set design, Instruction types: example for zero address, one address, two address and three address machines, Stack, accumulator and general purpose register organization.	8
Unit 3	Register Transfer Language: arithmetic, logic and shift micro operations and their hardware implementations as a simple ALU. Control Unit, Hardwired and Micro programmed control unit design.	8
Unit 4	Memory Organization: device characteristics, RAM organization, 1D and 2D organization, Virtual memory - Paging and Segmentation, High speed memories, Associative and Cache memory. Input-Output Design: IO interface, Bus structure, Modes of data transfer, Interrupts, Input Output Processor, Serial Communication Pipelining: Pipeline structure, Pipeline types - Instruction and Arithmetic pipelines. Interleaved memory organization, instruction prefetch, data buffers, pipeline performance measures.	8
Unit 5	Array processors: Routing mechanisms, Static v/s dynamic network. Multiprocessor systems, data flow concepts. Parallel processing languages.	8

References

1.	William Stallings, “Computer Organization and Architecture – Designing for Performance”, Pearson Education, Seventh Edition, 2006.
2.	David A. Patterson and John L. Hennessy, “Computer Architecture-A Quantitative Approach”, Elsevier, a division of reed India Private Limited, Fifth edition, 2012
3.	John P. Hayes, “Computer Architecture and Organization”, Tata McGraw Hill, Third Edition
4.	Carl Hamacher, Computer Organization, 5th Edition, Mc Graw Hill Publishers, 2002.

Theory of Computation				
Prerequisite: Nil	L	T	P	C
Total hours: 40	3	1	0	4

	Course Content	Hrs
Unit 1	**Basic Foundation**: Review Of SET Theory, Automata Theory, Alphabet, Power Of Alphabet, Kleen Closure, Positive Closure, String, Empty String, Concatenation, Language. **Finite Automata (FA):** Introduction, Deterministic Finite Automata (DFA) - Formal Definition, Simpler Notations (State Transition Diagram, Transition Table), Language of A DFA. Nondeterministic Finite Automata (NFA)- Definition, Language of an NFA, Equivalence Of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon Transitions, Minimization Of Deterministic Finite Automata, FA with Output (Moore and Mealy Machines) and Inter Conversion.	8
Unit 2	**REGULAR EXPRESSIONS (RE):** Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA’s to Regular Expressions, Converting Regular Expressions to Automata, Minimization of Finite Automata, Applications of Regular Expressions. **REGULAR GRAMMARS:** Chomsky Classification of Languages, Regular Grammars and FA, FA for Regular Grammar, Regular Grammar for FA. Proving Languages to be Non-Regular -Pumping Lemma, Applications, Closure Properties of Regular Languages.	8
Unit 3	**CONTEXT FREE GRAMMER (CFG):** Derivation Trees, Sentential Forms, Rightmost and Leftmost Derivations of Strings. Ambiguity in CFG’s, Minimization of CFG’s, Normal Forms (CNF, GNF), Pumping Lemma for CFL’s	8
Unit 4	**Pushdown Automata Theory:** Push Down Automata, Deterministic and Nondeterministic PDA, PDA And Languages, Construction; Acceptance of CFL, Acceptance by Final State and by Empty Stack and its Equivalence, Equivalence of CFG and PDA. **Turing Machines (TM):** Formal Definition and Behaviour, Languages of a TM, TM as Accepters, TM as a Computer of Integer Functions, TM with Storage in its State, TM as Subroutine, Minskey’s Theorem, Types of TMs, Multitrack, Mutitape, Nondeterministic TM, Encoding of TM, Computability and Acceptability.	8
Unit 5	**Recursive And Recursively Enumerable Languages (REL):** Properties of Recursive and Recursively Enumerable Languages. **Undecibility And Undecidable Problems:** Post's Correspondence Problem (PCP), Universal Turing Machine, The Halting Problem, Undecidable Problems about TMs. Context Sensitive Language and Linear Bounded Automata (LBA), Chomsky Hierarchy, Decidability	8

References	
1.	John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages and Computation, Pearson Education, India.
2.	Cohen, Introduction to Computer Theory, Addison Wesley.
3.	Martin, Introduction to Languages and Theory of Computation, TMH.
4.	Papadimitriou, Introduction to Theory of Computing, Prentice Hall.
5	K. L. P Mishra, N. Chandrashekaran, Theory of Computer Science-Automata Languages and Computation, Prentice Hall of India, India.

Database Information Systems				
Prerequisite: :NiL	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to Database System Database approach and Information systems , Database System Architecture, current advances in database technology, Database Systems Development Life Cycle- Prototyping methodology three-schema architecture, three- tiered architecture Hierarchical model, Network model, Relational model, Object oriented model, Multidimensional model	6
Unit 2	Database Models: ER-model notation, entity & entity type, relationship & relationship type, Degree, Cardinality & modality, Supertype/Subtype relationship Relational model concepts, Converting ER to Relational model	6
Unit 3	Introduction to SQL-DDL,DML and PCL, Advanced topics of SQL, PL/SQL language: Functions, Procedures & triggers, Views, Cursors etc. Formal query languages Relational Algebra and Relational Calculus Overview, Query processing and optimization	10
Unit 4	Relational schema, Functional dependencies, Inference axioms, Keys, closures, redundant FD’s , Decompositions, Join Dependencies Normalization, normal forms:1NF, 2NF, 3NF, BCNF, 4NF, 5NF, Best Database Design criterion Transactions, concurrency control, Crash Recovery, Physical DB design, file organizations, Indexing Structures, File indexing, hashing	14
Unit 5	Client/Server database architecture Application Development, Database Security, Overview of Distributed database, Data Warehousing and Data mining, Data Analytics	4

References	
1.	**Database System Concepts** ,Silberschatz A, Korth H F, and Sudarshan S, , McGraw Hill,,6th Ed.
2.	**Modern Database Management systems**, Hoffer J A, Prescott M B, and Topi H. Pearson Education Inc.,13th Edition
3.	**Fundamentals of Database Systems**, Elmasri R, Navathe S B, Pearson Education, 7th Edition..
4.	**Database Management System**, Raghurama krishnan & Johannes Gehrke, McGraw-Hill 3rd edition
5	**Commercial Application development using ORACLE Developer 2000 Forms 5.0**, Ivan Bayross, BPB Publications.

Machine Learning				
Prerequisite: Basic understanding of probability and statistics, linear algebra and calculus. A basic knowledge of programming (preferably Python) is essential.	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs
Unit 1	The learning problem – learning versus design, types of learning- supervised, unsupervised, reinforcement and other views of learning.	2
Unit 2	Training versus Testing: theory of generalization, interpreting the generalization bound. Generalization and over fitting: when does over fitting occur? Regularization, validation, cross validation. Bias-variance tradeoff. The Linear model: Linear classification, perceptron learning, linear regression, gradient descent, batch and stachastic gradient descent, convex functions, logistic regression, non linear transformation.	12
Unit 3	Generative vs discriminative models Supervised learning – Probability review, Bayes classifier, Naive Bayesian, MAP, MLE. K- nearest neighbor, measuring similarity using distance metrics, data normalization. Decision trees, constructing decision trees, ID3, C4.5. Random forest, Ensemble methods – bagging, boosting. Neural networks, going forward, biases, going backwards: back propagation of errors, MLP in practice, deriving back propagation network output and error, requirements of activation functions, learning rate, acceleration, decay, Loss functions - Sigmoid, Relu. SVM (Linear), optimal separation, kernels.	16
Unit 4	Unsupervised learning – the general problem, hierarchical and partitional clustering, K-means clustering, density based clustering, DBSCAN, autoencoders	8
Unit 5	Assessing classification performance – accuracy, sensitivity, specificity, the area under the ROC curve, confusion matrices, FAR, TPR, TNR, FRR, precision and recall	4

References	
1.	A first course in Machine learning, Simon Rogers and mark Girolami, CRC Press
2.	Learning from Data, Yaser S Abu-Mostafa, AML books
3.	Machine learning, Marsland, CRC press
4.	An Introduction to Machine Learning, Kubat Miroslav, Springer

Basics of Management				
Department: Department of Management Studies	L	T	P	C
Prerequisite: None	3	0	0	3

Course Learning Objectives

By the end of this course student will be able to:
1. Demonstrate the roles, skills and functions of managers.
2. Develop the understanding and cognizance of the importance of management principles.
3. Make effective application of acquired knowledge to diagnose and solve organizational problems and develop optimal managerial decisions.
4. Understand seven Ps of marketing and digital marketing strategies
5. Get to know about key people management processes.
6. Understand the decisions and processes in operations management.
7. Gain knowledge of financial systems, institutions, regulators and instruments.
8. Diagnose and communicate the complexities associated with management of various issues in the organizations and integrate the learning in handling these complexities

Course Content

- General Management Processes and Principles: - Concept, Functions and Principles of Management, Roles and skills of Managers;
- Functions of Management: Planning, Decision Making; Organizing: Organizational Design & Organizational Structures; Leading, Motivation, Communication and Controlling;
- Introduction to Human Resource and Marketing Management: Trends and Practices in People Management; Marketing Management Process and decisions, Marketing Mix;
- Introduction to Finance and Operations Management: Overview of Financial Systems, Financial Institutions, Markets and Instruments; Decisions & processes in Operations Management.

References

1.	Robbins, Stephen P. and Coulter, Mary (2019) 'Management', 14th edition, Prentice Hall of India
2.	Dessler, G. & Varkkey, B. (2018). Human Resource Management, 15e, Pearson.
3.	Laasch, O. (2021). Principles of Management-Practicing Ethics, Responsibility, Sustainablity, 2nd Edition, Sage Publications.
4.	Hill, Charles W L and McShane, Steven L. (2017) Principles of Management, Special Indian Edition, McGraw Hill Education
5.	Khan, M. Y. and Jain P. K. (Latest edition). Financial Management, Text, Problems & Cases. Tata McGraw Hill Company, New Delhi.
6.	Philip Kotler. (Latest edition). Marketing Management: Analysis, Planning, Implementation & Control. Prentice Hall of India.
7.	Koontz, Harold and Weihrich, Heinz & Ramachandra Aryasri A. (2016). Principles of Management, Latest edition, McGraw Hill Education

Computer Networks Lab				
Prerequisite: : The programming lab in C++, which means you need to be very comfortable with C++ and using standard debugging tools.	L	T	P	C
Total hours: 48	0	0	4	2

	Course Content	Hrs
	The laboratory experiments conducted on various tools Lab 1-3: Introduction networking (wireshark,, TCP dump, CISCO packet tracer) Lab 3-4: Introduction to socket programming Lab 5-9: Experiments on NS2 and NS3 Lab 10-12 : Experiments Mininet	48

References	
1.	Data Networks: Bertsekas and Gallagher, PHI
2.	Computer Networks: L. Peterson and Davie, Elsevier
3.	Computer Networking A top down Approach: J.F.Kurose, Pearson
4.	Computer Networks : Andrew S. Tanenbaum, Pearson

Database Information Systems Lab

Prerequisite: :Nil	L	T	P	C
Total hours: 48	0	0	4	2

	Course Content	Hrs
I	Design exercises and various Tools of designing the ER diagram and its mapping to relational model	8
II	Programming exercises on SQL –Detailed DDL commands and queries to create databses.	8
III	Programming exercises on SQL –Detailed DML commands	10
IV	Programming exercises on SQL –Detailed PCL commands	6
V	Programming Exercise on advanced topics of SQL, PL/SQL language : Functions, Procedures, triggers, Views, Cursors etc.	8
VI	There will be as semester Mini-Group Project on theme of Database Information system	8

References

1.	Database System Concepts ,Silberschatz A, Korth H F, and Sudarshan S, , McGraw Hill,,6th Ed.
2.	Modern Database Management systems, Hoffer J A, Prescott M B, and Topi H.,Pearson Education Inc.,13th Edition
3.	Fundamentals of Database Systems, Elmasri R, Navathe S B, Pearson Education, 7th Edition..
4.	Database Management System, Raghuramakrishnan & Johannes Gehrke, McGraw-Hill 3rd edition
5	Commercial Application development using ORACLE Developer 2000 Forms 5.0, Ivan Bayross, BPB Publications.

<table>
<tr><th colspan="6">Machine Learning Lab</th></tr>
<tr><td colspan="2">Prerequisite: Python Programming</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><th colspan="5">Course Content</th><th>Hrs</th></tr>
<tr><td>1</td><td colspan="4"><b>Perceptron Learning Algorithm:</b> 1. Generate a linearly separable data (random) set of size 20. Plot the examples $\{(x_n , y_n)\}$ as well as the target function f on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.
2. Run the perceptron learning algorithm on the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(x_n , y_n)\}$, the target function f , and the final hypothesis g in the same figure. Comment on whether f is close to g. Repeat everything in (2) with another randomly generated data set of size 100. Compare your results with (2)</td><td>3</td></tr>
<tr><td>2</td><td colspan="4"><b>Linear Regression:</b> Write a python script that can find w0 and w1 for an arbitrary dataset of number of hours studied versus rank of a students as $\{(x_n, y_n)\}$ pairs. Find the linear model, y = w T x, that minimizes the squared loss. Derive the optimal <b>w</b> for the total training loss:s MSE/RSS L = Σ(yn − wT xn)^2 . Using the model predict the rank for the number of hours studied. Load the data stored in the file syntheticdata.mat. Fit a 4th order polynomial function f (x; w) = w 0 + w 1 x + w 2 x ^{2} + w 3 x ^{3} + w 4 x ^{4} to this data. What do you notice about w2 and w4?) Fit a function f (x; w) = w 0 + w 1 x + w2 sin((x−a)'/ b, assuming a and b are fixed in some sensible range. Show a least square fit using this model. What do you notice about w1 and w2. Comment about generalization and overfitting.</td><td>3</td></tr>
<tr><td>3</td><td colspan="4"><b>Logistic Regression:</b> Handwritten Digits Data: You should download the two data files with handwritten digits data: training data (ZipDigits.train) and test data (ZipDigits.test). Each row is a data example. The first entry is the digit, and the next 256 are grayscale values between −1 and 1. The 256 pixels correspond to a 16 × 16 image. For this problem, we will only use the 1 and 4 digits, so remove the other digits from your training and test examples. Please submit your Python code implementing the logistic regression for classification using gradient descent. Familiarize yourself with the data by giving a plot of two of the digit images. Develop two features to measure properties of the image that would be useful in distinguishing between 1 and 4. You may use symmetry and average intensity (as discussed in class). As in the text, give a 2-D scatter plot of your features: for each data example, plot the two features with a red redx if it is a 4 and a blue blue if it is Classifying Handwritten Digits: 1 vs. 4. Implement logistic regression for classification using gradient descent to find the best separator you can using the training data only (use your 2 features from the above question as the inputs). The output is +1 if the example is a 1 and −1 for a 4. Give separate plots of the training and test data, together with the separators. Compute E in on your training data and E test , the test error on the test data after 1000 iterations. Now repeat the above using a 3rd order polynomial transform. As your final deliverable to a customer, would you use the linear model with or without the 3rd order polynomial transform? Explain.
<b>Regularization:</b> Logistic regression can also be augmented with the l_2 -norm regularization: min E(w) + λ||W || $^2_2$, where E(w) is the logistic loss. Please change your gradient descent algorithm accordingly and use cross-validation to determine the best regularization parameter. Plot the training and testing performance curves.Indicate in the plot the best regularization parameter you obtained (using cross validation).</td><td>6</td></tr>
</table>

4	**Neural Networks:** In this problem you will implement forward and backward propagation methods for a multi-layer neural network with K hidden layers. Assume that K is a user input less than 10. Implement the networks separately with the following activation functions: Sigmoid: Derive the gradient of the activation function. Confirm with numerical differentiation. Tanh: Derive the gradient of the activation function. Confirm with numerical differentiation. Assume that the last layer has a linear activation function and the loss function is $l(y, \hat{y}) = \|y - \hat{y}\|^{2}_{2}$. Submit your code (along with any instructions necessary to # run it), the forward pass outputs at each layer and the gradients of the parameters (W_{ij}^{k} , b^{k}_{i}). The input, output and the parameters of the network can be found in the MAT file associated with this problem. In this problem you will train a multi-layer neural network to recognize handwritten digits. Use the multi-layer neural network (with ReLU activation) that you implemented in the previous homework. Use 32 nodes in each layer and initialize the weights randomly. The data is also provided to you in a MAT file. • Report the training and validation accuracy as a function of iterations (with 5 hidden layers). Report the convergence speed of the training procedure (with 5 hidden layers) for the Stochastic Gradient Descent optimization algorithm. • Determine the number of hidden layers required via cross-validation. Report the training and validation accuracy for cross-validation. • Finally, report the best test error that you can achieve.	6
5	**Evaluation Metrics:** Consider a theoretical biometric matcher that generates distance scores in the range [−∞,∞]. Assume that the genuine and impostor score distributions due to this matcher can be approximately modeled as N(30, 10) and N(60, 15), respectively. Here, N(μ, σ2) denotes normal distribution with mean, μ, and variance, σ2 Suppose the following decision rule is employed: s is classified as a genuine score if s ≤ η; else it is classified as an impostor score. Here, η ∈ [0, 100]. • Plot the genuine and impostor distributions in a single graph. The distributions should be contained in the range [0, 100]. • If η = 50, what is the FMR (i.e., FAR) and FNMR (i.e., FRR) of the biometric matcher? • Given s is classified as a genuine score if s ≤ η; else it is classified as an impostor score. If η = 75, what is the FMR (i.e., FAR) and FNMR (i.e., FRR) of the biometric matcher? • Plot the DET curve of this matcher. • Plot the ROC curve and AUC of this matcher.	4
6	**SVM:** Classify the digits data as given for exercise 4 using a Support Vector Machine. Compute the values of W and an offset b, also draw the hyperplane.	8
7	**Decision Trees and Random Forest:** Generate three tables: Table one with attributes: Id, Exercise, Family history, Heart Attack Risk. Table two with attributes: Id, Smoker, Obese, Heart Attack Risk, Table three: Id, Obese, Falimy history and Heart Attack Risk. Generate s100 samples randomly for the three tables. List three bootstrap samples, using these bootstrap samples create decision trees that will be in the random forest model using entropy based information gain as the feature selection criteria. Assuming the random forest uses majority voting, what prediction will it return for the query: EXERCISE = rarely, SMOKER = false, OBESE = true, FAMILY = yes.	6
8	**Clustering:** A bank wants to detect fradulent credit card transactions. Using random function generate data for lots of transactions (each transaction is an amount of money, a shop, and the time and date) and some information about which credit cards were stolen, and the transactions that were performed on the stolen card. Generate random data files for the above description of atleast 200 transactions. Implement Agglomerative, Hierarchichal and Density based clustering techniques to cluster people's transactions together to identify patterns, so that stolen cards can be detected as changes in pattern. How well do you think this will work? There is much more data of transactions when cards are not stolen, compared to stolen transactions. How does it affect the learning, and what can you do about it.	6

References

1.	A first course in Machine learning, Simon Rogers and mark Girolami, CRC Press
2.	Learning from Data, Yaser S Abu-Mostafa, AML books
3.	Machine learning, Marsland, CRC press
4.	An Introduction to Machine Learning, Kubat Miroslav, Springer

Scheme and Syllabus of 5th Semester

Fifth Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credits**	**Type**
	22CPT301	Compiler Design	3-0-0	3	PC
	22CPT302	Cryptography	3-0-0	3	PC
	22CPT303	Operating System	3-0-0	3	PC
	22CPT304	Software Engineering	3-0-0	3	PC
	22CPT305	Emerging Technologies for CS	3-0-0	3	PC
	22CPTxxx	Program Elective-1	3-0-0	3	PE
	22CPP306	Compiler Design Lab	0-0-2	1	PC
	22CPP307	Cryptography Lab	0-0-2	1	PC
	22CPP308	Operating System Lab	0-0-2	1	PC
				21	

Honors					
	CSTxxx	Advance Data Structures and Algorithms		3	
	CSTxxx	Honors Elective-1*		3	
				6	

Minor CSE					
	22CPT104	Data Structures		3	PC
	22CPT303	Operating System		3	PC
				6	

Compiler Design				
Prerequisite: : Theory of Computation	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Language Translators: Compilers and Interpreters, Hybrid Compiler, Structure of a Compiler, Self Compiler and Cross Compiler. Lexical Analysis: Design and implementation of Lexical Analyzers, Finite automata and Regular expressions, Lex tool – the Lexical Analyzer Generator.	8
Unit 2	Syntax Analysis: Context Free Grammars, Derivation and Parse trees, Ambiguity of grammars. Bottom-up and Top-down Parsing - Shift Reduce Parser, Operator Precedence Parser, First and Follow functions, Left recursion, LL Parsers, Canonical collection of items, LR parsers, Conflict Resolution in LR parsers.	14
Unit 3	Syntax-Directed Translation: Syntax-directed definitions and translation schemes, Attributes and Translation Rules, Implementation of S-attributed and L-attributed definitions. Intermediate Code Generation: Intermediate codes, Three address codes, Translation of Expressions and Type Checking.	8
Unit 4	Code Optimization and Code Generation : Basic blocks, Flow graphs, DAG, Global data flow analysis, ud-chaining, Available expressions, Loop optimization, Compilation of Expression and Control structures. Error Detection and Recovery.	12

References	
1.	Aho, Lam, Sethi and Ullman: Compilers – Principles, Techniques and Tools, Pearson Education
2.	Tremblay and Sorenson: The Theory and Practice of Compiler Writing, BS Publications.
3.	Allen Holub : Compiler Design in C, Prentice Hall India.

Cryptography				
Prerequisite: Nil	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to Number Theory: Divisibility theory in integers. Extended Euclid’s algorithm. Modular Arithmetic – exponentiation and inversion. Fermat’s Little Theorem, Euler’s Theorem. Solution to congruence's, Chinese Remainder Theorem. Review of abstract algebra – Study of Ring Zn, multiplicative group Zn* and finite field Zp – Gauss Theorem (cyclicity of Zp*) - Quadratic Reciprocity. Primality Testing – Fermat test, Carmichael numbers, SolovayStrassen Test, Miller Rabin Test – analysis	8
Unit 2	Conventional Encryption , Classical Techniques- substitution and transposition ciphers, study of basic cryptanalysis possible on classical ciphers, Modern Techniques- block and stream ciphers and RC4	8
Unit 3	Shannon’s principles of diffusion and confusion, Design Principles of Block cipher: SPN and Fiestel Structure, The Data Encryption Standard (DES) , AES	8
Unit 4	Asymmetric cryptography: Public Key Encryption, Diffie – Hellman Key Exchange algorithm, RSA algorithm and its limitations	8
Unit 5	Cryptographic hash functions, secure hash algorithm, Message authentication, digital signature, RSA digital signature	8

References	
1.	W. Stallings,”Cryptography and Network Security Principles and practice”, 5/e, Pearson Education Asia, 2013
2.	Behrouz A. Forouzan and Debdeep Mukhopadhyay, “Cryptography and Network Security”, 2nd edition, Tata McGraw Hill, 2013
3.	N. Koblitz, Number Theory and Cryptography, Springer, 2001
4.	J. Katz and Y. Lindell, Introduction to Modern Cryptography, Third edition CRC press, 2020

Operating System					
		L	T	P	C
Total Hours: **42**		3	0	0	3
Prerequisite: Computer Organization and Architecture, Data structures and algorithms, Problem solving using C					
Course Content					Hrs
Unit 1	***Introduction:*** What is an operating system, Types of operating systems and differences among them, OS as a virtual machine; User and Operating-System Interface, System Calls, System Services, Linkers and Loaders, Booting, OS as a resource manager, Interrupts and traps, System calls, Limited direct execution, user versus kernel mode. ***CPU Scheduling:*** Process, Process v/s program, context switch, Process state diagram, CPU scheduling – FCFS, SJF, SRTF, Priority, Pre-emptive priority, Round Robin, MLFQ, Lottery, CFS, Multi-Processor Scheduling, Real-Time CPU Scheduling, Thread v/s process, Process and Thread APIs				10
Unit 2	***Synchronization:*** Inter-process communication and Processes: IPC in Shared-Memory Systems and Message-Passing Systems, Race condition, mutual exclusion, The Critical-Section Problem (CSP), Algorithmic solutions to CSP – Dekker’s, Peterson’s, Lamport Bakery Solution; Hardware Support for Synchronization – Test and Set, Compare and Swap; OS support for synchronization - Mutex Locks, Semaphores, Monitors; Condition Variables; Classic Problems of Synchronization – Producer Consumer, Sleeping Barber; Dining Philosopher’s Problem, Deadlock – Prevention, avoidance, detection and recovery, Safe state, Banker’s algorithm. Livelock.				10
Unit 3	***Memory Management:*** working set model, hardware support; Contiguous allocation- partitioned memory allocation – fixed and variable partitioning, memory management with bit maps – swapping – relocation- protection and sharing. Non contiguous allocation – Paging – principles , page allocation, segmentation. Virtual memory concepts, address translation, management of virtual memory, page replacement policies, protection and sharing, Thrashing; Caching principles and quantitative estimation of cache behavior				8
Unit 4	***I/O Management:*** Overview of Mass-Storage Structure, HDD Scheduling, NVM Scheduling, Error Detection and Correction, Storage Device Management, Swap-Space Management, SSD (Solid State Disks); I/O Systems -Overview; I/O Hardware; Kernel I/O Subsystem, Transforming I/O Requests to Hardware Operations ***File management:*** File Concept, Access Methods, Directory Structure, Protection, File-System Interface, Shared files. File-System Implementation: Structure and Operations; Directory Implementation; Allocation Methods; Free-Space Management; Case study: EXT, NTFS, HFS				8
Unit 5	***Security and Protection:*** Program Threats – stack overflow, return to libc, RoP, heap spraying, integer overflow, format string attacks; System and Network Threats; User Authentication; Principles of Protection - Protection Rings, Domains; Access Matrix, Implementation of the Access Matrix – Access Control Lists, capabilities; Revocation of Access Rights, Role-Based Access Control, Mandatory Access Control, Capability-Based Systems				6
References					
1.	Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, *Operating Systems:Three Easy Pieces* [online http://pages.cs.wisc.edu/~remzi/OSTEP/]				

2.	Abraham Silberschatz, Peter B. Galvin, Greg Gagne, *Operating System Concepts*. 9th edition. Wiley.
3.	Andrew Tanenbaum & Albert Woodhull, *Operating Systems: Design and Implementation*. Prentice-Hall.
4.	Maurice J Bach, *Design of Unix Operating System*. AT&T Bell Labs.
5.	Andrew Tanenbaum, *Modern Operating Systems*, Prentice Hall.
6.	William Stallings, *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson.
7.	Crowley: *Operating System A Design Approach*, TMH.

Software Engineering

Prerequisite: :Nil	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	**Introduction to Software Engineering**: The evolving Role of Software Engineering, The Changing Nature of Software, Legacy software, Software Evolution and Software Myths. Industrial Engineering Tools for Software Engineering.	8
Unit 2	**Process Models**: Software Process Models: The Waterfall Model, The Incremental Model, the RAD model, Evolution Process Model: Prototyping, The Spiral model, Concurrent Development Model. Agile Process Models: Extreme Programming (XP)	6
Unit 3	**Software Project Management:** Management Activities, Project Planning, Project scheduling, Risk management. Requirements Engineering. Feasibility study, requirement analysis, cost benefit analysis, planning systems, analysis tools and techniques.	6
Unit 4	**System Design:** design fundamentals, modular design, data and procedural design, object oriented design and UML. **System Development**: Code documentation, program design paradigms.	6
Unit 5	**Software Testing**: Test Strategies for Conventional Software, Test Strategies for Object – Oriented Software, Verification and Validation Testing, System Testing, Debugging. Black-Box and White-Box Testing, Basis Path Testing, Control Structure Testing, Regression Testing, Mutation Testing, Dataflow Testing.	8
Unit 6	**Software Maintenance**: Maintenance Characteristics, Maintainability, Maintenance Tasks and side effects	8

References	
1.	Pressman Roger S, Software Engineering A Practitioner’s Approach, TATA McGraw-Hill Publications, 6th Edition, 2005, ISBN No. 007-301933X
2.	Ian Sommerville, Software Engineering, Pearson Education, 7th Edition, 2008, ISBN: 978-81-7758-530-8.
3.	Ghezzi C. Jazayeri M and Mandrioli: Fundamentals of Software Engg. , PHI.
4.	Rajib Mall, Fundamentals of software engineering. PHI Learning Pvt. Ltd..
5.	Unified Modeling Language Reference manual”, Grady Booch, James Rambaugh, Ivar Jacobson, Pearson India, ISBN – 9788177581614 R5.

Emerging Technologies for CS					
Prerequisite: Operating system, computer network, etc.		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit - I	**Introduction:** emerging areas in CS, Internet of things (IoT) : introductions to IoT, sensors and its features, architectures and challenges, Applications – smart city, smart grid, Industrial IoT, etc.				8
Unit - II	**Computing and Applications:** introduction to Cloud computing, various cloud architectures and its applications, mobile edge computing, MEC architectures, design principles and applications, MEC integration with disruptive technologies.				8
Unit - III	**Blockchain:** Introduction to bockchain, principles and technologies, cryptocurrencies, smart contracts, Major applications and issues. **Drones –** introduction, drone design principles, smarm optimization, Theory of drones, applications, etc.				8
Unit - IV	**Decentralized learning:** Introduction to decentralized learnings, types of learnings, aggregation and communication challenges, privacy and security. **3D printing :** introduction to 3D printing and it's applications **Quantum technologies :** introduction, requirements, challenges, Q-bit principles, quantum computing, quantum cryptography, etc.				8
Unit - V	**Future/Advanced trends:** introduction to future social applications, augmented reality (AR), Virtual reality (VR), mixed reality (MR), extended reality (XR) and metaverse, design principles, major challenges and applications. 5G communication and its use cases, 5G and beyond technologies.				8
References					
1.	The course materials are mainly from the lecturing slides. Research papers from top conferences like SIGCOMM, MOBICOM, NSDI, MobiSys etc.				

Compiler Design Lab				
Prerequisite: : Theory of Computation	L	T	P	C
	0	0	2	2
Course Content				
1. Design and implement a lexical analyzer for a given language. 2. Design and implement a lexical analyzer using the Lex/Flex tool. 3. Implement an Operator precedence parser for a given operator grammar. 4. Implement First and Follow functions. 5. Develop an LL(1) parser for a given input grammar. 6. Develop an LR(1) parser for a given input grammar. 7. Intermediate Code generation for a given source code. 8. Control Flow graph generation from a given intermediate code. 9. DAG construction and performing local optimization. 10. Implementation of Constant Folding, Redundant-subexpression elimination, and other optimizations.				

References	
1.	Aho, Lam, Sethi and Ullman: Compilers – Principles, Techniques and Tools, Pearson Education
2.	Tremblay and Sorenson: The Theory and Practice of Compiler Writing, BS Publications.
3.	Allen Holub : Compiler Design in C, Prentice Hall India.
4.	John R. Levine, Tony Mason, Doug Brown : Lex & Yacc, O'Reilly.

Cryptography Lab				
Prerequisite: :Nil	L	T	P	C
Total hours: 40	0	0	2	2

	Course Content	Hrs
Unit 1	a. Euclidean and Extended Euclidean algorithm for finding the Greatest Common Divisor of two large integers. Computing the Multiplicative inverses in Zn. b. Repeated square and multiply algorithm for modular exponentiation in Zn. c. Determining the order of a group element. Finding a generator of a cyclic group. d. Chinese remainder theorem. e. Computation of Legendre symbol and Jacobi symbol f. Modular polynomial arithmetic g. RSA public key algorithm h. ElGamal Cryptosystem i. Rabin cryptosystem j. Diffie-Hellman Key exchange protocol.	
Unit 2	a. Fermat's factorization method b. Congruence of squares. Finding a congruence of squares modulo n to factor n. c. Construction of Finite Field of characteristic 2. d. Computations in elliptic curve over a finite field.	
Unit 3	a. Sieve of Eratosthenes b. Fermat primality test c. Solovay-Strassen probabilistic primality test d. Miller-Rabin probabilistic primality test	

References	
1.	Menezes, P.C. van Oorschot, S.A. Vanstone: Handbook of Applied Cryptography: CRC Press, 1996
2.	Abhijit Das and C.E.VeniMadhavan, Public-key Cryptography: Theory and Practice, Pearson, 2009.
3.	Darrel Hankerson, Alfred Menezes, Scott Vanstone,Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004
Instructions:	
	C/C++ Programming Language under Linux Operating System
	gmp-man-6.1.2.pdf (Refer GMP library manual)
	Code should be well modularised and documented, Code should be well modularised and documented

Operating System Lab				
	L	T	P	C
Number of Weeks: 14	0	0	2	2
Pre-requisite: C Programming, Linux basics, Python				

Course Content

1) Write a C/Python program to simulate CPU scheduling. Following CPU scheduling mechanisms need to be implemented:
   a) SJFS, FCFS
   b) Priority (pre-emptive & non-pre-emptive)
   c) Round Robin
   d) MLFQ
   e) Lottery
2) Given a list of process IDs, write a program to develop a tree depicting ancestor/parent/child relationship. This shall be a dynamic scenario, and the tree should be updated every second (as new child processes may be created and some may be killed or terminated normally/abnormally).
3) Given two processes P1 and P2 (created as parent/child process through fork/ two threads within same process or two independent processes through two different programs) both of which increment a shared variable, implement Dekker's & Peterson's solutions.
4) Implementation of Lamport-Bakery solution for (N>= 5) processes. Each process shall increment a shared counter by one.
5) Modify solution to producer-consumer problem so that it works wherein producer produces one item but consumer consumes two items. If buffer has only one item, consumer relinquishes critical section and waits till there are two or more items. The solution should be
   a) threads based
   b) independent process based
6) Write a program to check if there is a deadlock in the resource-allocation graph. If not, how can the process be allocated resources with no deadlock ever occurring.
7) Implement Sleeping Barber and dining Philosophers problem using semaphores.
8) Write a program in C that reads a file from the file system and displays its contents on the screen. Implement error handling and permission checking.
9) Write a program in C that implements a simple memory allocation algorithm such as first-fit or best-fit, and tests its performance using a benchmark program.
10) You are given a file named "input.txt" that contains parameters related to a disk in the first six lines - number of cylinders (track), number of sectors, bytes per sector, RPM, average seek time, initial head position. These parameters are in different lines of the same file. Track 0 is the outermost one.
    The seventh line of the file should contain a sequence of requests for track (cylinder) numbers.
    Write a program to output
    a) Average Rotational delay
    b) Total Seek Time to service all the requests for
       - SSTF (Shortest Seek time first)
       - LOOK
11) Create a virtual machine using Virtual Box or VMware, install an operating system on it, and configure it to run a web server. Test the web server using a web browser and network analysis tools
12) Implement buffer overflow attack using stack smashing.
13) Write a shell script that performs the following tasks:
    a) File manipulation: Create, delete, copy, and move files and directories.

- b) Text processing: Search for specific patterns in files and perform text transformations.
- c) System monitoring: Retrieve system information like CPU usage, memory utilization, and disk space.
- d) Automation: Automate a repetitive task on your Linux system using a shell script.

14) Implement a program in Linux that demonstrates the following process management concepts:
    - a) Process creation: Create child processes using the fork() system call.
    - b) Process termination: Terminate processes using the exit() system call.
    - c) Process synchronization: Synchronize processes using semaphores, mutexes, or other synchronization mechanisms.
    - d) Signal handling: Handle signals like SIGINT or SIGTERM in your program.

15) Develop a program that interacts with the Linux file system. Your program should enable users to:
    - a) Create files and directories.
    - b) Navigate through directories and display their contents.
    - c) Copy or move files and directories.
    - d) Change file permissions and ownership.

16) Write a simple Linux device driver that interacts with a custom hardware device or simulates a virtual device. Your device driver should:
    - a) Implement read and write operations to interact with the device.
    - b) Handle interrupts or other device-specific functionalities.
    - c) Test the device driver by accessing the device and performing read/write operations.

References	
1.	Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, *Operating Systems:Three Easy Pieces* [online http://pages.cs.wisc.edu/~remzi/OSTEP/]
2.	Abraham Silberschatz, Peter B. Galvin, Greg Gagne, *Operating System Concepts*. 9th edition. Wiley.
3.	Andrew Tanenbaum & Albert Woodhull, *Operating Systems: Design and Implementation*. Prentice-Hall.
4.	Maurice J Bach, *Design of Unix Operating System*. AT&T Bell Labs.
5.	Andrew Tanenbaum, *Modern Operating Systems*, Prentice Hall.
6.	William Stallings, *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson.
7.	Crowley: *Operating System A Design Approach*, TMH.

Advanced Data Structures and Algorithms (Honors)

Prerequisite: Data Structures, Design and Analysis of Algorithms	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	RAM model – Notations, Recurrence analysis - Master's theorem and its proof - Amortized analysis, Recurrence equations.	8
Unit 2	Advanced Data Structures: B-Trees, Binomial Heaps, Fibonacci Heaps, AVL trees, Red-black trees, B-trees, Splay trees, Interval trees; Disjoint set – union and path compression, Amortized analysis Greedy Algorithms: shortest distance, minimum spanning tree, interval scheduling, interval partitioning; Divide and Conquer: sorting, integer and polynomial multiplication.	10
Unit 3	Dynamic programming: Longest common subsequence. Chain of matrix multiplication, sequence alignment, Bellman Ford Convex hull and Voronoi diagrams, line segments, Optimal polygon triangulation; Primality testing, Integer factorization.	10
Unit 4	Graph algorithms: Matching and Flows; Parallel algorithms: Basic techniques for sorting, searching, merging. Intractability: Independent Set, Vertex Cover, Randomized algorithms, Probabilistic algorithms.	8
Unit 5	Approximate Algorithms: Vertex-cover, set-covering problems, Travelling Salesman problem. Complexity classes - NP-Hard and NP-complete Problems - Cook's theorem NP completeness reductions, undecidability	6

References:

1.	Cormen, Leiserson, Rivest: Introduction to Algorithms, Prentice Hall of India.
2.	Aho A.V , J.D Ulman: Design and analysis of Algorithms, Addison Wesley
3.	Brassard : Fundamental of Algorithmics, PHI.
4.	Sara Baase: Computer Algorithms: Introduction to Design and Analysis, Pearson Education.
5.	Papadimitriou, Steiglitz: Combinatorial Optimization: Algorithms and Complexity, PHI
6.	Motwani and Raghavan: Randomized Algorithms, Cambridge University Press

Scheme and Syllabus of 6th Semester

Sixth Semester					
S. No	**Code**	**Subject**	**L-T-P**	**Credits**	**Type**
	22CPT309	Artificial Intelligence	3-0-0	3	PC
	22CPT310	Computer and Network Security	3-0-0	3	PC
	22CPT311	Digital Image Processing	3-0-0	3	PC
	22CPT312	Parallel and Distributed Computing	3-0-0	3	PC
	22EET313	Smart Grid	3-0-0	3	PLEAS
	22CPTxxx	Program Elective-2	3-0-0	3	PE
	22CPP313	Computer and Network Security Lab	0-0-2	1	PC
	22CPP314	Digital Image Processing Lab	0-0-2	1	PC
	22CPP315	Parallel and Distributed Computing Lab	0-0-2	1	PC
				21	

Honors					
	CSTxxx	Honors Elective-1*		3	
	CSTxxx	Honors Elective-2*		3	
				6	

Minor CSE					
	22CPT211	Computer Networks		3	PC
	22CPT213	Database Information Systems		3	PC
				6	

Artificial Intelligence					
Prerequisite: Nil		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Overview of AI, Problems, Shift in focus of AI towards providing smarter solutions, Change in application domains of AI, State-of-the-art technologies in AI. Problem space and searching techniques, Types of production system, Control strategies, Heuristic search Techniques. Defining AI problems as a State Space Search: example, Production Systems, Types of production systems, Search and Control Strategies, Problem Characteristics.				6
Unit 2	Heuristic search techniques- Generate-and-test, Hill Climbing, Best First Search, A* , Problem Reduction, AO*, Constraint Satisfaction with inferencing, backtracking and local search, Mean-Ends Analysis. Knowledge representation, Representation, mappings, approaches and issues.				8
Unit 3	Propositional Logic and theorem proving, First order Predicate logic: syntax and semantics, Propositional v/s First Order Predicate Logic, Satisfiabilit problems, model finding, Inference algorithms: Backward and forward chaining, Resolution (proof by contradiction). Representing Simple facts in Logic, Representing Instances and Isa relationships, Computable Functions and Predicates, Using First Order Logic, Inferencing process and resolution, Unification algorithm. Knowledge Representation : Ontologies, objects, events, PEAS, Forward v/s backward reasoning, Matching and control knowledge, Levels of knowledge representation, entailment, implication, contradiction, contingency, model checking, Modus ponens inference rule, CNF clauses, Horn clauses. SAT Solvers: DPLL Weak Structures: Semantic Nets, Frames, Strong Structures: Conceptual Dependencies, Scripts. Expert Systems and applications : Representing and using domain knowledge, Expert system shells, Knowledge Acquisition.				10
Unit 4	Game Playing : Minimax Search Procedures, Adding alpha-beta cutoffs , State-of-the-Art Game Programs and modern examples, Watson and how it solved Jeopardy. Information Retrieval - Google's page rank algorithm, Introduction to natural language processing.				8
Unit 5	Uncertain knowledge and reasoning Quantifying uncertainity, Probabilistic reasoning,, Graphical Models, Bayesian networks, Bayesian inference, forward and backward inference, inference by enumeration, and variable elimination algorithm, Probabilistic reasoning overtime, Inference in temporal models. Sampling: prior sampling, rejection sampling, likelihood weighting. Hidden markov models, the forward algorithm, the HMM Viterbi algorithm. Concepts in Machine learning: Introduction , Foundations of AI v/s ML, When to use ML and when not to use, Framework for AI/ML application to a problem, Taxonomy of Computational Intelligence, Classification under Machine Learning.				10

References	
1.	Artificial Intelligence: A Modern Approach by Russel and Norvig, Third Edition, Pearson, 2015.
2.	Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-Graw Hill.
3.	Introduction to AI & Expert System: Dan W. Patterson, PHI.

Computer and Network Security					
Prerequisite: : Cryptography, Computer networks, etc.		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit - I	**Introduction:** Introduction (a) Security(b) Malware(c) OWASP top ten and other major security issues in the world(d) CVE and other information (e) Introduce various types of security areas				5
Unit - II	**Software and OS Security:** OS Security: Common Bugs, Buffer Overflow, Runtime Defenses against memory safety vulnerabilities, program verification and other vulnerabilities, Principles in OS Security; Mechanisms for confining bad code, Mechanisms for confining bad code: isolation, sandboxing, SFI and Virtualization, Trusted Computing				8
Unit - III	**Web Security:** Secure web site design (SQL injection, XSS, etc.), Browser Security,				7
Unit - IV	**Network Security:** TCP/IP, DDoS Attacks, Network worms and botnets: attacks and defenses, DNS and BGP security, Network defense tools – Firewall and Intrusion Detection**.**				11
Unit - V	**Future/Advanced Security:** Introduction - The Security in Existing wireless Networks, Upcoming wireless networks and challenges, Thwarting and malicious behavior – Naming and addressing, security association and secure neighbor discovery, secure routing in multichip wireless networks and privacy protection. Mobile OS Security and Privacy: Android, IOS security challenges, processor security, privacy, anonymity and censorship and other security issues according to the current situations and future requirements				9
References					
1.	Security in Computing (3rd edition)				
2.	Cryptography and Networks 7 edition				
3.	The course materials are mainly from the lecturing slides I?ve made and research papers from top conferences like NDSS, USENIX, SIGCOMM, MOBICOM, NSDI, MobiSys etc.				

Digital Image Processing					
Prerequisite: Fundamental knowledge on signals and systems, basics of linear algebra and calculus, and programming skills		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Introduction to Digital Image Processing: Digital Image Representation, Fundamental Steps in DIP, Elements of Visual Perception, Image Sensing and Acquisition, Image Model, Sampling, Quantization, Basic Relationship Between the Pixels				6
Unit 2	Image Transforms: Discrete Fourier Transform (DFT), Properties of 2D DFT, Fast Fourier Transform, Inverse FFT, Discrete Cosine Transform and KL Transform, Discrete wavelet Transform, Convolution and Correlation				8
Unit 3	Image Enhancement: Spatial Domain- Basic Gray Level Transformations, Histogram processing, Smoothing and Sharpening Spatial Filters Frequency Domain- Smoothing and Sharpening frequency domain filters, Homomorphic filtering				8
Unit 4	Image Restoration: Overview of Degradation models, Unconstrained and constrained restorations, Inverse Filtering, Wiener Filter				6
Unit 5	Image Segmentation: Detection of discontinuities, edge linking and boundary detection, thresholding, region oriented segmentation Image Compression: Need for data compression, image compression models, loss-less and lossy compression				8
Unit 6	Representation and Description: Representation schemes, boundary descriptors, regional descriptors. Morphology: Dilation, erosion, opening, closing, Hit-or-Miss Transform, some basic morphological algorithms				6
References					
1.	Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson , 3rd Edition, 2008				
2.	Castleman. Digital Image Processing. Prentice Hall.				
3.	Anil K. Jain, Fundamentals of Digital Image Processing, Pearson , 2002				

Parallel and Distributed Computing				
Prerequisite: Programming in C, Data Structures, Operating Systems, Computer Architecture and Organization	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs
Unit 1	Parallel Computing, Sequential programs, Parallel Programs, Performance Metrics for Parallel Systems, Effect of Granularity on Performance, Scalability of Parallel Systems, Parallel Programming Platforms, Implicit Parallelism, SIMD & MIMD systems, Clusters, Single-Core and Multi-Core Processors, Physical Organization of Parallel Platforms, Cache Coherence, Posix-Threads, problem-Solving using P-threads.	8
Unit 2	Programming Using the Message-Passing Paradigm - MPI Principles of Message Passing Programming; Building blocks (Sending and Receiving Operations); Communication Library calls; Collective communication and Computation library calls, Programming Shared Address Space Platforms – OpenMP, Directive Parallel Programming; The OpenMP programming Model (Concurrent Tasks, Synchronization Constructs, Data Handling); Open libraries; OpenMP-Environment Variables;	10
Unit 3	Parallel Programs, Matrix Computations, Matrix-Vector Multiplication, Matrix-Matrix Multiplication, Solving system of Linear Equations; Parallel Implementation of Sparse Matrix Computations with Vector; Sorting algorithms, Issues in Sorting on Parallel Computers, Bubble Sort and its Variants, Quicksort; Parallelizing Quicksort; Sequential and Parallel Implementation of all-pairs of Shortest Paths Algorithms; Sequential & Parallel Search Algorithms; Depth-First Search Algorithms; Best-First Search Algorithms	8
Unit 4	Programming on Multi-Core Systems with GPU accelerators, An Overview of Brief History of GPUs; An Overview of GPU Programming; An Overview of GPU Memory Hierarchy Features; An Overview of CUDA enabled NVIDIA GPUs, Introduction to CUDA C, Parallel Programming using OpenACC, CUDA APIs, CUDA Libraries for Numerical and Non-Numerical Computations; The OpenCL – Heterogeneous Programming; OpenCL Libraries, The OpenCL Memory Model, Execution Model; Platform and Devices; An Overview of OpenCL API;	6
Unit 5	An Overview of MapReduce, An Overview of MapReduce Programming, An Overview of Hadoop Architecture /Execution (Master/slave, Namenode/Datanode); Hadoop Distributed File System (HDFS), An Overview of Hadoop Components, Hadoop – Control Flow and Data Flow; An overview of Hive (Distributed Data Warehouse); Hbase (Distributed Column based database, PIG –(Data Flow Language), Introduction to Spark, Spark RDD, Machine Learning Using Spark.	8

References	
1.	Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: Introduction to Parallel Computing, Second Edition Pearson Education – 2007
2.	Peter Pacheco, An Introduction to Parallel Programming, Morgan Kaufman Publishers, Elsevier (2011)
3.	Jason Sanders, Edward Kandrot, CUDA By Example – An Introduction to General-Purpose GPU Programming, Addison Wesley (2011)
4.	Rohit Chnadra, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, Ramesh Menon, Parallel Programming in OpenMP, Academic Press (2001)

5.	Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry Dana Schaa, (2011), Heterogeneous Computing with OpenCL McGraw-Hill, Inc. Newyork
6.	Michael J. Quinn, Parallel Programming in C with MPI and OpenMP McGraw-Hill International Ed (2003)
7.	Aru C Murthy, Vinod Kumar Vavilapalli, Doug Eadline, Joseph Niemiec, and Jeff Markham, Apache Hadoop YARN Moving beyond MapReduce and Batch Processing with Apache Hadoop 2, Addison Wesley, 2014

Smart Grid					
Prerequisite: Data communication, computer networks, data science, etc.		L	T	P	C
Total hours: 40		3	0	0	3
Course Content					Hrs
Unit - I	**Introduction to Power systems and Smart grid :** power systems: Load and Generation, Power Flow Analysis, Economic Dispatch and Unit Commitment Problems, Smart Gird: Definition, Applications, Government and Industry, Standardization				4
Unit - II	**Renewable Generation:** Carbon Footprint, Renewable Resources: Wind and Solar, Microgrid Architecture, Tackling Intermittency, Stochastic Models and Forecasting				8
Unit - III	**Smart Grid Communications:** Two-way Digital Communications Paradigm, Network Architectures, IP-based Systems, Power Line Communications, Advanced Metering Infrastructure **Wide Area Measurement:** Sensor Networks, Phasor Measurement Units, Communications Infrastructure, Fault Detection and Self-Healing Systems, Applications and Challenge				11
Unit - IV	**Renewable Generation:** Carbon Footprint, Renewable Resources: Wind and Solar, Microgrid Architecture, Tackling Intermittency, Stochastic Models and Forecasting **Intelligent Demand Response**: Definition, Applications, and State-of-the Art, Pricing and Energy Consumption Scheduling, Controllable Load Models, Dynamics, and Challenges, Electric Vehicles and Vehicle-to-Grid Systems, Demand Side Ancillary Services				11
Unit - V	**Data science for economics and market operations:** Energy and Reserve Markets, Market Power, Detection of market power using data science methods, Analysis of pricing patterns and market behaviour and Assessment of market concentration and its impact on competition and consumer welfare, Generation Firms- Optimization of scheduling and dispatch of generation units and Predictive maintenance using machine learning algorithms, improved operational efficiency through data-driven models., Locational Marginal Prices - Prediction of local marginal prices using data science techniques, Analysis of historical data to identify correlations and patterns, Accurate price forecasting and risk management strategies. Financial Transmission Rights - Estimation of financial value of transmission rights, Optimization of allocation and trading of transmission rights, Predictive models for assessing financial risks and returns. **Security and Privacy:** Cyber Security Challenges in Smart Grid, Load Altering Attacks, False Data Injection Attacks, Defence Mechanisms, Privacy Challenges				6
References					
1.	This course does not have any official textbook. The main source of learning for the students is the set of handouts provided by the instructor. The students will also need to read several recent papers in the field of smart grid, e.g., in the IEEE Transactions on Smart Grid, the IEEE Innovative Smart Grid Technologies Conference, and the IEEE Conference on Smart Grid Communications.				
2.	The course materials are mainly from the lecturing slides made and research papers from top conferences like SIGCOMM, MOBICOM, NSDI, MobiSys etc.				

Computer and Network Security Lab					
Prerequisite: : The programming lab in C++, which means you need to be very comfortable with C++ and using standard debugging tools.		L	T	P	C
Total hours: 28		0	0	2	2
Course Content					Hrs
	The laboratory experiments conducted on various tools Lab 1-3: Experiments on Network traffic scanning tools – Wireshark, Nmap, Nessus, etc tools Lab 3-4: Experiments on control hijacking attacks, and OS security Lab 5-9: Experiments on Network security (OpenSSL, OpenVAS, Snort, Metasploit, Firebug, etc.) Lab 10-12 : Experiments on Web security (DVWA, snort, etc)				28
References					
1.	Security in Computing (3rd edition)				
2.	Cryptography and Networks 7 edition				
3.	The course materials are mainly from the lecturing slides I’ve made and research papers from top conferences like NDSS, USENIX, SIGCOMM, MOBICOM, NSDI, MobiSys etc.				

Digital Image Processing Lab				
Prerequisite: Fundamental knowledge on image processing and programming skills	L	T	P	C
	0	0	2	1

Course Content

1. Familiarization with various image processing tools
2. Basic operations on images
3. Basic grey-level transformations
4. Image Negative
5. Logarithmic transformation
6. Power-law transformation
7. Perform the following over a given image
8. Grey level slicing
9. Zooming (Nearest neighbour interpolation, bilinear interpolation)
10. Bit-plane slicing
11. Histogram equalization and specification
12. Implementation of different image transforms (DFT, PCT, DWT, etc.)
13. Spatial filtering in presence of various noise
14. Filtering in frequency domain
15. Implementation of image deblurring techniques
16. Image segmentation (edge detection, line detection, point detection)
17. Implementation of region based image segmentation
18. Implementation of different morphological operations
19. Analysis of images using color models
20. Mini project

References	
1.	Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson , 3rd Edition, 2008
3.	Anil K. Jain, Fundamentals of Digital Image Processing, Pearson , 2002

Parallel and Distributed Computing Lab				
Prerequisite: C Programming, DSA	L	T	P	C
	0	0	2	1
Course Content				
	1. Implementation of pthreads, problem-solving using pThreads. 2. Problem-solving using openMP 3. Matrix multiplication using task. 4. Problem-solving using MPI, Sending and Receiving Operations 5. Parallel Programs, Matrix Computations, Matrix-Vector Multiplication, Matrix-Matrix Multiplication using MPI. 6. Parallel Implementation of Sparse Matrix Computations with Vector; Sorting Algorithms, Issues in Sorting on Parallel Computers; Bubble Sort and its Variants using GPU Resources. 7. Quicksort; Parallelizing Quicksort; Sequential and Parallel Implementation of all-pairs of Shortest Paths Algorithms; Sequential & Parallel Search Algorithms. 8. Depth-First Search Algorithms; Best-First Search Algorithms. Control Flow graph generation from a given intermediate code. 9. Implementation of MapReduce programs for large scale data handling. 10. Programming on Multi-Core Systems with GPU accelerators.			
References				
1.	Aru C Murthy, Vinod Kumar Vavilapalli, Doug Eadline, Joseph Niemiec, and Jeff Markham, Apache Hadoop YARN Moving beyond MapReduce and Batch Processing with Apache Hadoop 2, Addison Wesley, 2014			
2.	Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry Dana Schaa, (2011), Heterogeneous Computing with OpenCL McGraw-Hill, Inc. Newyork			
3.	Jason Sanders, Edward Kandrot, CUDA By Example – An Introduction to General-Purpose GPU Programming, Addison Wesley (2011) .			

Pool 1: Electives: 3-0-0 (Credits 3, Semesters 5/6/7)

1.	5G and Beyond
2.	Advanced Algorithms
3.	Blockchain Technologies
4.	Cyber Physical Systems
5.	Deep Learning
6.	Evolving Architectures
7.	Malware Analysis
8.	Artificial Neural Networks
9.	Nature Inspired Algorithm
10.	Parallelizing Compilers
11.	VLSI Algorithms
12.	Wireless Security
13.	Wireless Networks

<table>
<tr><th colspan="6">5G and Beyond</th></tr>
<tr><td colspan="2">Prerequisite: :Data communication, computer networks, etc.</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 40</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit - I</td><td colspan="4"><b>Introduction to mobile networks :</b> Mobile Networks (wireless communication), Need of Generations, Evolution of generation, Challenges of generations, 4G Network and architecture, From 4G to 5G, Network Architecture of 5G/6G, Challenges of 5G.</td><td>4</td></tr>
<tr><td>Unit - II</td><td colspan="4"><b>5G apps:</b> 5G apps (pull from user demands) - app overview and 5G objectives, 5G apps in the automotive sector and D2D technologies, IoT applications and technologies: Internet of Things in 5G Era, 3GPP Standards for the Internet-of-Things, Other 5G apps and technologies : Media and Entertainment Sector, eHealth sector</td><td>8</td></tr>
<tr><td>Unit - III</td><td colspan="4"><b>5G Technologies:</b> Overview and radio - 5G (core network): Network slicing, C-RAN, NFV, SDN, 5G technologies (mobile edge computing, management and orchestration, discussion).</td><td>11</td></tr>
<tr><td>Unit - IV</td><td colspan="4"><b>Overview of Active Research:</b> 6G, vision and thoughts towards 6G, network intelligence for 5G and beyond</td><td>11</td></tr>
<tr><td>Unit - V</td><td colspan="4"><b>Security for 5G and beyond :</b> Overview of security for 5G and beyond, enhanced system level security design, blockchain for 5G and beyond.</td><td>6</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Erik Dahlman, Stefan Parkvall, and Johan Sköld: 5G NR: The Next Generation Wireless Access Technology. Academic Press, 2018, ISBN: 9780128143230.</td></tr>
<tr><td>2.</td><td colspan="5">The course materials are mainly from the lecturing slides I have made and research papers from top conferences like SIGCOMM, MOBICOM, NSDI, MobiSys etc.</td></tr>
</table>

Advanced Algorithms				
Total Hours	L	T	P	C
42	3	0	0	3

Prerequisite: *Data Structures, Design and Analysis of Algorithms, C programming*

	Course Content	Hrs
Unit 1	***Review of Data Structures*** - Height balanced trees, AVL, Red-black trees, splay trees, Binomial and Fibonacci heaps, treaps, suffix tree, Range minimum query, Aho-Corasick automata, Hash tables, Tries, van Emde boes tree.	8
Unit 2	***String Algorithms:*** Exact String Matching: Rabin-Karp, KMP, Boyer Moore; Inexact string matching: Edit distance, Levenshtein distance computing algorithm. ***Computational Geometry:*** Convex Hull. Line-segment Intersection. Sweep Lines. Voronoi Diagrams. Range Trees. Optimal polygon triangulation.	8
Unit 3	***Linear Programming:*** Formulation of Problems as Linear Programs. Duality. Simplex, Interior Point, and Ellipsoid Algorithms. ***Online Algorithms:*** Ski Rental. River Search Problem. Paging. The k-Server Problem. List Ordering and Move-to-Front. ***Parallel Algorithms:*** PRAM. Pointer Jumping and Parallel Prefix. Bitonic sorting, Odd-even sorting, Maximal Independent Set.	10
Unit 4	***Approximation Algorithms:*** Greedy Approximation Algorithms. Dynamic Programming and Weakly Polynomial-Time Algorithms. Linear Programming Relaxations. Randomized Rounding. Vertex Cover, Wiring, and TSP. Fixed-Parameter Algorithms - Parameterized Complexity. Kernelization. Vertex Cover. ***Probabilistic algorithms:*** Primality testing, Integer factorization, ***Randomized algorithms:*** Monte Carlo – mincut, Las Vegas – quicksort	8
Unit 5	***Complexity classes*** - NP-Hard and NP-complete Problems - Cook's theorem NP completeness reductions – SAT, 3SAT, vertex cover, Independent Set, Hamiltonian cycle, travelling salesman.	8

References	
1.	Cormen, Leiserson, Rivest: Introduction To Algorithms, Prentice Hall Of India.
2.	Aho A.V , J.D Ulman: *Design And Analysis Of Algorithms*, Addison Wesley
3.	Jon Kleinberg And Éva Tardos: Algorithm Design, Pearson.
4.	Motwani And Raghavan: *Randomized Algorithms,* Cambridge University Press
5.	Vaizirani: *Approximation Algorithms,* Springer Verlag
6.	Papadimitriou, Steiglitz: *Combinatorial Optimization: Algorithms And Complexity*, Phi.

Blockchain Technologies

Prerequisite: : Nil	L	T	P	C
Total hours: 35	3	0	0	3

	Course Content	Hrs
Unit 1	Introduction to blockchain- Distributed Ledger Technology, Decentralization, Problems in Traditional Money transfer system, Digital Crypto currency, Bitcoin nuts and bolts, Generic elements of Blockchain, Bitcoin Network and Architecture, Block and transactions in a Blockchain, Advantages over Traditional Databases, Mining Mechanism, Types of Blockchain: Public, Private, Consortium,Hybrid	6
Unit 2	Cryptography: Elliptic Curve Cryptography, Hash Functions, Merkle Tree, Merkle Patricia Trie, Digital Signature, Wallets and Keys, User Addresses and Privacy CRYPTO CURRENCY History, Distributed ledger, Creation of Coins, Double spending,	3
Unit 3	Mechanics of Bitcoin, Bitcoin protocols, Transaction in Bitcoin Network, AltCoins, Ethereum ,Transactions in Ethereum, EVM, Accounts, Transactions, Gas, Fees, Smart Contracts, Wallets managing and protecting crypto assets, Types of Wallets, different ways of storing Bitcoin keys, security measures, Tokenizing, Risk and challenges,	8
Unit 4	Bitcoin Mining and consensus –definition, working of Consensus Mechanism, Byzantine Generals Problem, Nakamoto consensus,Properties of consensus mechanism , incentives in consensus,Types of Consensus Algorithms, Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Proof of Importance (PoI), Proof of Capacity (PoC) ,The Proof of Elapsed Time (PoET), Hybrid Proof of Activity (PoA), Proof of Authority (PoA), Proof of Burn (PoB) Byzantine Fault Tolerance (BFT), and other flavours of consensus mechanisms , Pros and Cons of Consensus Mechanisms, :, sybil resistance, Security analysis of various Consensus Mechanisms	9
Unit 5	Ethereum Syntax &, Structure ,Decentralized Apps (dApp), EVM, and the Ethereum blockchain, Eth 2.0, Sharding Chains ,Smart Contract, , MetaMask, Blockchain-based IoT Applications, Hyperledger, Components of Ethereum Ecosystem Smart contract on ethereum, Setting up Ethereum Node using Geth Client, Smart Contracts and DApps, Programming in solidity Truffle, Ganache CLI, Metamask, Remix, Solidity, Writing and Deploying Smart Contracts in Solidity, Connection to Web3.js Library, Vulnerabilities in Smart Contracts, Attacks, Prevention of Attacks, Decentralized Autonomous Organization (DAO), Building an Initial Coin Offering (ICO). Privacy and Scaling the blockchain, blockchain interoperability, future of blockchains	6
Unit 6	Use Cases and applications in Cryptocurrency and Other Sectors like Finance, Voting System, and Healthcare, Networks , Bitcoin: A Peer-to-Peer Electronic Cash System, Supply Chain Management (SCM) etc	3

References

1.	Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfede, “Bitcoin and Cryptocurrency Technologies”, Princeton University Press, 2016.
2.	Lantz, Lorne, and Daniel Cawrey, “Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications” O’Reilly Media,
3.	Imran Bashir, Mastering Blockchain: Distributed Ledger Technology, decentralization, and smart contracts explained, Packt Publishing Ltd, March 2018

Cyber-Physical Systems				
Prerequisite: :Data communication, computer networks, etc.	L	T	P	C
Total hours: 40	3	0	0	3

	Course Content	Hrs
Unit - I	**Introduction to CPS :** Characteristics of Cyber-Physical Systems (CPS), Cyber-Physical Systems (CPS) in the real world, Basic principles of design and validation of,	4
Unit - II	**CPS Hardware:** Industry 4.0, AutoSAR, IIOT implications, Building Automation, Medical CPS -CPS physical systems modeling and formalisms: CPS - Platform components - CPS HW platforms - Processors, Sensors, Actuators,	8
Unit - III	**CPS Network and systems:** , CPS Network - WirelessHart, CAN, Automotive Ethernet, Scheduling Real Time CPS tasks Principles of Dynmical Systems - Dynamical Systems and Stability, Controller Design Techniques and Performance under Packet drop and Noise,	11
Unit - IV	**CPS Implementations and Intelligence:** CPS implementation issues - From features to automotive software components, Mapping software components to ECUs, CPS Performance Analysis - effect of scheduling, bus latency, sense and actuation faults on control performance, network congestion, and building real-time networks for CPS, CPS Intelligent CPS	11
Unit - V	**Applications and Security for CPS :** Safe Reinforcement Learning, Robot motion control, Autonomous Vehicle control, Gaussian Process Learning, Smart Grid Demand Response, Building Automation, Secure Deployment of CPS, Secure Task mapping and Partitioning, State estimation for attack detection, Automotive Case study : Vehicle ABS hacking Power Distribution Case study : Attacks on Smart Grids	6

References	
1.	"Introduction to Embedded Systems – A Cyber–Physical Systems Approach " - E. A. Lee, Sanjit Seshia
2.	"Principles of Cyber-Physical Systems" - Rajeev Alur
3.	The course materials are mainly from the lecturing slides I?ve made and research papers from top conferences like SIGCOMM, MOBICOM, NSDI, MobiSys etc.

<table>
<tr><th colspan="5"><b>Deep Learning</b></th></tr>
<tr><td colspan="2">Prerequisite: : Probability, Statistics, Algebra, Basic Computer Programming, Data Structures</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4">Course Overview: Introduction to Deep Learning and its Applications.
Introduction to Statistical Learning: Multi-Layer Perceptron, Back Propagation, Linear Regression, Loss Functions and Optimization: Optimization, stochastic gradient descent, dropout, batch normalization, etc.</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Convolutional Neural Networks: Convolution, pooling, Activation Functions, Back propagation of CNN, Weights as templates, Translation invariance, Training with shared parameters.
CNN Architecture Design and Discussion: AlexNet, VGG, GoogLeNet, ResNet, Capsule Net, etc.
Visualization and Understanding: Visualizing intermediate features and outputs, Saliency maps, Visualizing neurons, Cam-Grad, etc.</td><td>8</td></tr>
<tr><td>Unit 3</td><td colspan="4">Sequential Modelling: Recurrent and Recursive Nets, RNN, LSTM, GRU, Image captioning, visual question answering, etc.</td><td>6</td></tr>
<tr><td>Unit 4</td><td colspan="4">Generative Models: Encoder, Decoders, Variational Autoencoders, Generative Adversarial Networks like pix2pix, CycleGAN, etc.
Transformers based Models:.</td><td>8</td></tr>
<tr><td>Unit 5</td><td colspan="4">Deep Learning Applications:
Object Detection: RCNN, Fast RCNN, Faster RCNN, YOLO and variants, Retina Net, etc., Adversarial Attacks on CNN
Deep learning for NLP</td><td>8</td></tr>
<tr><td>Unit 6</td><td colspan="4">Deep learning Libraries and Frameworks: Keras, TensorFlow, PyTorch, AutoML, etc</td><td>4</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Ian Goodfellow and Yoshua Bengio and Aaron Courville, “Deep Learning,” MIT Press.</td></tr>
<tr><td>2.</td><td colspan="5">Michael A. Nielsen, “Neural Networks and Deep Learning,” Determination Press, 2015.</td></tr>
</table>

Evolving Architectures					
Prerequisite: Operating Systems, Computer Networks, DBMS, Algorithms		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Special, emerging and advanced topics in different areas of Computer Science and Engineering will be covered under this course. • Understand Taxonomy of new Architectures • Understand the Building Blocks of each architecture. • Install the Open-Source Tools • Study the State of the Art • Listen to an Expert (Academia / Industry) • Discuss Survey / Research Papers (Last 5-7 years) • Case Studies of Tool or Simulator • Build some components for a Simple Model as assignment.				8
Unit 2					8
Unit 3					8
Unit 4					9
Unit 5					9
References					
1.	Research Papers from Journals and Conferences				
2.	Technical and Research Reports from Consortiums / Committees				
3.	Red Books, White Papers, Request For Comments (RFCs)				
4.	Manuals, Guides, Blogs				

Malware Analysis				
Total Hours	L	T	P	C
42	3	0	0	3

Prerequisite: Fundamentals of Cryptography, Operating Systems, Computer Organization and Architecture, Data structures and algorithms, Programming

	Course Content	Hrs
Unit 1	***Introduction:*** Introduction to malware, OS security concepts, malware threats, evolution of malware, malware types- viruses, worms, rootkits, trojans, bots, spyware, adwares, logic bombs, malware analysis, static malware analysis, dynamic malware analysis.	8
Unit 2	***Advanced Static Analysis:*** x86 Architecture, Analyzing Windows programs, Portable executable file format, disassembling malicious executable programs. Anti-static analysis techniques- obfuscation, packing, metamorphism, polymorphism.	8
Unit 3	***Advanced Dynamic Analysis:*** Debugging malware - ollydbg, windbg, setting virtual environments- sandboxes, emulators, hypervisors, virtual machines, live malware analysis, dead malware analysis, analyzing traces of malware- system-calls, api-calls, registries, network activities. Anti-dynamic analysis techniques-anti-vm, runtime-evasion techniques.	9
Unit 4	***Malware Functionality:*** Downloaders, Backdoors, Credential Stealers, Persistence Mechanisms, Privilege Escalation, Covert malware launching-Launchers, Process Injection, Process Replacement, Hook Injection, Detours, APC Injection.	8
Unit 5	***Malware Detection Techniques:*** Signature-based techniques: malware signatures, packed malware signature, metamorphic and polymorphic malware signature. Non-signature based techniques: similarity-based techniques, machine-learning methods, invariant-inferences.	9

References	
1.	Thabet, A., Kleymenov, A.: Mastering Malware Analysis: A Malware Analyst's Practical Guide to Combating Malicious Software, APT, Cybercrime, and IoT Attacks.
2.	Bruce Dang, Alexandre Gazet, and Elias Bachaalany: Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation
3.	Peter Szor: The Art of Computer Virus Research and Defense, Addison Wesley Professional.
4.	Eric Filiol: *Computer Viruses: from theory to applications*, Springer.
5.	Michael Sikorski and Andrew Honig: Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, No Starch Press
6.	Christopher Elisan: Advanced Malware Analysis, McGraw-Hill Osborne Media.
7.	Michael Hale Ligh, Andrew Case: The Art of Memory Forensics: Detecting Malware, Wiley.
8.	Published articles from reputed Journals and Conferences.

Artificial Neural Networks

Prerequisite: :	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Overview of Biological Neurons, Structure of biological neurons relevant to Artificial Neural Networks(ANNs), Fundamental Concepts of Artificial Neural Networks Models of ANNs; Feedforward & feedback networks; learning rules; Hebbian learning rule, perception learning rule, delta learning rule, Widrow-Hoff learning rule, correction learning rule, Winner –lake all elarning rule, etc.	8
Unit 2	Single layer Perception Classifier: Classification model, Features & Decision regions; training & classification using discrete perceptron, algorithm, single layer continuous perceptron networks for linearly seperable classifications.	9
Unit 3	Multi-layer Feed forward Networks: Linearly non-seperable pattern classification, Delta learning rule for multi-perceptron layer, Generalized delta learning rule, Error back-propagation training, learning factors, Examples.	10
Unit 4	Single layer feedback Networks: Basic Concepts, Hopfield networks, Training & Examples., Associative memories: Linear Association, Basic Concepts of recurrent Auto associative memory: rentrieval algorithm, storage algorithm; By directional associative memory, Architecture, Association encoding & decoding, Stability.	9
Unit 5	Self organizing networks: Unsupervised learning of clusters, winner-take-all learning, recall mode, Initialisation of weights, seperability limitations, Learning Vector Quantization (LVQ). Applications of Artificial Neural Network in various domains	6

References

1.	S. Haykin, “Neural Networks and Learning Machine”s , 3rd Edition , Prentice-Hall , 2008 , ISBN No. 0131471392
2.	Jacek M. Zurada, “Introduction to Artificial Neural Systems , Jaico Publishing House; First edition.
3.	B Yegnanarayana, “Artificial neural networks”, 1st ed., Prentice Hall of India P Ltd, 2005.

Nature Inspired Algorithms

Prerequisite: **Programming in C**	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to Algorithms, Optimization, and Search for optimality, computational intelligence, Nature Inspired solutions and characteristic, Nature inspired Meta-heuristics and its brief history.	8
Unit 2	Analysis of Optimization Algorithms, Nature Inspired Algorithms, parameter Tuning and control Constrained and unconstrained optimizations, Random Walks and Optimizations, evolutionary strategies and Evolutionary Algorithms (EA), Simulated Annealing (SA) Algorithm and its behaviour, Genetic Algorithms(GA)- genetic operator, parameters, fitness functions, genetic programming and convergence analysis, GA variants	10
Unit 3	Swarm Intelligence optimization, Particle Swarm Optimization(PSO) Algorithm, Ant Colony Optimization (ACO) Algorithms, Artificial Bee Colony ACO) optimization algorithms, Cuckoo Search (CS) Algorithms, Intelligent Water Drop Algorithm (IWD), Bat Algorithms(BA), Firefly Algorithms(FA)	8
Unit 4	Applications of nature-inspired algorithm, machine learning using nature inspired algorithm, data clustering using NIA.	6
Unit 5	Parallel processing of NIA using Hadoop, Parallel data clustering using NIA. Multi-objective optimization and applications.	8

Parallelizing Compilers					
Prerequisite: Compiler Design		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Introduction – Compilation for parallel machines and automatic detection of parallelism, structure of a parallelizing compiler.				8
Unit 2	Dependence Theory and Practice - Types of dependences, data and control dependencies, dependence analysis, direction vectors, loop carried and loop independent dependences, tests for data dependence and their applicability, construction of data dependence and control dependence graphs.				18
Unit 3	Parallel Code Generation - Automatic extraction of parallelism, representation of iteration spaces of nested loops, loop based transformations such as loop distribution, loop coalescing, loop interchange and cycle shrinking transformation.				8
Unit 4	Interprocedural Analysis and Optimization - aliasing information, summary data flow analysis, interprocedural constant propagation, interprocedural data dependence analysis and parallelization of call statements.				8

References	
1.	Randy Allen, Ken Kennedy: Optimizing compilers for modern architectures. Morgan Kaufmann.
2.	Steven Muchnick : Advanced Compiler Design & Implementation, Morgan Kaufmann.
3.	Hector, Ullman, Widom : Database System Implementation, Pearson.

VLSI Algorithms				
Prerequisite: Compiler Design	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs
Unit 1	Introduction of VLSI Technology, VLSI design cycle, design styles, basic Layout rules and circuit abstraction, introduction to standard Cell, Gate array, FPGA	8
Unit 2	Overview of basic graph algorithms, Graph algorithms for physical Design Partitioning: Classification of partitioning algorithms, Karnighan-Lin Algorithm, FM Algorithm, Ratio cut algorithm	18
Unit 3	Floor-planning: Rectangular dual graph approach of floor-planning, hierarchical tree based approach, Integer programming based floor planning. Placement: placement by simulated annealing and force directed method	8
Unit 4	Routing: classification of routing algorithms, Global routing: Maze routing algorithms, line probe algorithms, Steiner tree based algorithms, Detailed, Single layer and two layer routing algorithms, routing in FPGAs	8

References	
1.	Naveed Shervawani, “ Algorithms for VLSI physical Design Automation “ III Ed Springer
2.	Sarrafzadeh and Wong “ An introduction to VLSI Physical design “ MGH
3.	Sze: VLSI Technology
4	Weste and Eshranghan, “ Introduction toVLSI Design”. Pearson Edu.
5	Sadiq M. Sait, Habib Youssef, "VLSI Physical Design Automation: Theory and Practice", World Scientific Publishing Company;
6	Cormen Leiserson, Rivest, “ Introduction to Algorithms”, Pearson Edu.

Wireless Security					
Total Hours		L	T	P	C
42		3	0	0	3
Prerequisite: Fundamentals of Computer Networks, Wireless Networks, Cryptography					
Course Content					Hrs
Unit 1	Introduction to Wireless Security, Overview of wireless networks, Wireless network architecture, Wireless network security goals and objectives,				6
Unit 2	Wireless Network Vulnerabilities and Threats, Wireless network security threats, Active and passive attacks, Wireless network vulnerabilities, Common attacks on wireless networks				8
Unit 3	Wireless Security Protocols: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), Wi-Fi Protected Access II (WPA2) 802.11i				10
Unit 4	Wireless Authentication and Encryption protocols: Password-based authentication, Certificate-based authentication, Secure Wireless Network Design, Secure wireless network design principles, Secure network configuration				10
Unit 5	Placement of access points and antennas, Site survey and signal analysis, Wireless Security Standards and Policies, Wireless network security standards, Wireless security policies and guidelines regulatory compliance, Wireless Security Management and Monitoring, Wireless network management, Wireless network monitoring and auditing, Incident response				8
References					
1.	""Hacking Exposed Wireless: Wireless Security Secrets & Solutions" by Johnny Cache, Vincent Liu, and Billy Rios (2nd Edition, 2010)				
2.	"802.11 Wireless Networks: Security and Analysis" by Alan Holt and Chi-Yu Huang (1st Edition, 2010)				
3.	""Wireless Network Security: A Beginner's Guide" by Tyler Wrightson (1st Edition, 2011)				
4.	""Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework" by Jessey Bullock and Jeff T. Parker (1st Edition, 2017)				
5.	“Wireless Network Security: A Practical Approach to Securing Your Wi-Fi Network".				

Wireless Networks				
Prerequisite: : Computer Networks	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to Wireless Networking ,History of wireless networks, Difference between Wireless and Fixed Telephone Networks, Development of Wireless Networks, Wireless Network Architecture, Benefits of Wireless Networks, Wireless Networking Applications Radio propagation models, Narrowband digital modulation and Coding under wireless fading environments.	8
Unit 2	Medium Access and Resource allocation Techniques: Basics of CDMA and OFDM, Randomized medium access- Unslotted and Slotted Aloha, IEEE 802.11 CSMA protocol, channel allocation in (TDMA/FDMA/CDMA)-based wireless networks under the protocol model. Wireless LANs: Technology,IEEE 802.11 Wireless LAN Standard, Radio based Wireless LANs, Wi-Fi, Wimax	8
Unit 3	Routing Layer: Introduction, Routing protocols- Routing, Dynamic source routing, Destination sequence distance vector, Overview ad-hoc routing protocols , Application- RFID, Bluetooth, Zigbee, NFC	8
Unit 4	TCP enhancements for wireless protocols - Traditional TCP: Congestion control, fast retransmit/fast recovery, Implications of mobility - Classical TCP improvements: Indirect TCP, Snooping TCP, Mobile TCP, Time out freezing, Selective retransmission, Transaction oriented TCP - TCP over 3G wireless networks.	8
Unit 5	Emerging industry standards such as 4G Cellular – 4G features and challenges - Applications of 4G – 4G Technologies: Multicarrier Modulation, 5G, IEEE 802.11p, Cognitive Radio	8

References	
1.	William Stallings , Wireless Communications and Networks, Pearson Education, 2009
2.	Jon W. Mark and W. Zhuang , Wireless Communications and Networking, Pearson Ed, 2009
3.	Upena Dalal, Wireless Communication and Networks, Oxford publications, 2015

Pool 2: Electives: (3-0-0 (Theory) + 0-0-2 (Practical)) (Credits 4, Semesters 7/8)

1	Advanced Computer Networks
2	Advanced Computer Networks Lab
3	Advanced Database Systems
4	Advanced Database Systems Lab
5	Biometrics
6	Biometrics Lab
7	Computer Vision
8	Computer Vision Lab
9	Data Analytics
10	Data Analytics Lab
11	Data Mining
12	Data Mining Lab
13	Decentralized Learning
14	Decentralized Learning Lab
15	Embedded System Design
16	Embedded System Design Lab
17	Full Stack Development
18	Full Stack Development Lab
19	Information Retrieval
20	Information Retrieval Lab
21	Internet of Things (IoT)
22	Internet of Things (IoT) Lab
23	IoT based Robotics
24	IoT based Robotics Lab
25	Natural Language Processing
26	Natural Language Processing Lab
27	Program Analysis
28	Program Analysis Lab
29	Social Network Analysis
30	Social Network Analysis Lab
31	Software Testing
32	Software Testing Lab
33	Topics in Computing
34	Topics in Computing Lab
35	Topics in Operating System
36	Topics in Operating System Lab

<table>
<tr><td colspan="6"><b>Advanced Computer Networks</b></td></tr>
<tr><td colspan="2">Prerequisite: : Data communication, computer networks, etc.</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 40</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit - I</td><td colspan="4"><b>Software Defined Network-1:</b> Logically-Centralized Control, SDN Software Stack, Data-Plane Verification, Forwarding Table Verification, Debugging-Diagnosis, Measurement – Overview, APIs.</td><td>8</td></tr>
<tr><td>Unit - II</td><td colspan="4"><b>Software Defined Networks - II:</b> Resource Management, Device Heterogeneity, Scalability: Data-Plane, Control-lane Extending Data-Plane: OpenFlow++, SDN Applications: Data-center & Cloud & Wide-Area-Networks.</td><td>8</td></tr>
<tr><td>Unit - III</td><td colspan="4"><b>Advances in Wireless Networks:</b> Wireless networking: Bluetooth, 802.11 standards, Information theory, bandwidth, multiple access, Wireless Terahertz Networks, 5G and 6G communication, Intelligent Transportation Systems.</td><td>8</td></tr>
<tr><td>Unit - IV</td><td colspan="4"><b>Emerging networking technologies -I:</b> Host configuration and service discovery principles, Future routing architectures, IPv6 deployment scenarios and challenges, IPv6 transition/integration, Advanced IP multicast, including IPv6 multicast and SSM.</td><td>8</td></tr>
<tr><td>Unit - V</td><td colspan="4"><b>Emerging networking technologies - II :</b> Data centre networks<b>,</b> Delay-tolerant networking Future home network, architectures, IP network management and monitoring, Social Networks</td><td>8</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Software Defined Networking by Thomas D Nadeau and Ken Gray.</td></tr>
<tr><td>2.</td><td colspan="5">Hagen S, (2006). IPv6 Essentials.</td></tr>
<tr><td>3.</td><td colspan="5">The course materials are mainly from the lecturing slides made and research papers from top conferences like SIGCOMM, MOBICOM, NSDI, MobiSys etc.</td></tr>
</table>

Advanced Computer Networks Lab				
Prerequisite: : The programming lab in C++, which means you need to be very comfortable with C++ and using standard debugging tools.	L	T	P	C
Total hours: 36	0	0	2	2

	Course Content	Hrs
	The laboratory experiments conducted on various tools Lab 1-3: Experiments on SDN using mininet, NS,etc Lab 3-4: Experiments on Internet of things using NS3, Netsim, etc Lab 5-9: Experiments on Wireless networks using NS3, Netsim, etc Lab 10-12 : Experiments on 5G and beyond using NS3, Netsim, etc	36

References	
1.	Software Defined Networking by Thomas D Nadeau and Ken Gray.
2.	Hagen S, (2006). IPv6 Essentials.
3.	The course materials are mainly from the lecturing slides and research papers from top conferences like SIGCOMM, MOBICOM, NSDI, MobiSys etc.

Advanced Database Systems

Prerequisite: : Database Information Systems	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs
Unit 1	Query Processing and Optimization – Implementation of Database operations, External Sorting, Size Estimations, Equivalence Rules, Heuristic-based Optimization, Materialized Views, Incremental View Maintenance.	14
Unit 2	Transaction Processing and Implementation - Concurrency Control Protocols, Two-phase Lock Protocol and its variants, Deadlock Prevention and Detection schemes and implementation, Timestamp-based Ordering Protocol, Log-based Recovery Management.	12
Unit 3	Modern Database Systems - Database System Architectures, Distributed Database Systems, Parallel Databases, Times in Databases, Multimedia Databases.	8
Unit 4	Distributed Databases - Data Storage, Global Catalog, Distributed Transaction Processing, Two-Phase Commit Protocol, Distributed Query Processing.	8

References	
1.	Silberschatz , Korth, Sudarshan : Database System Concepts, McGrall Hill.
2.	Elmasri and Navathe : Fundamentals of Database Systems, 3rd Edition, Addison Wesley.
3.	Hector, Ullman, Widom : Database System Implementation, Pearson.
4.	Ceri and Pelagatti : Distributed Databases – Principles and Systems, McGraw Hill.

Advanced Database Systems Lab				
Prerequisite: : Database Information Systems	L	T	P	C
	0	0	2	1
Course Content				
1. Programming exercises on Query Processing and Implementation of Database operations. 2. Programming exercises on Query Optimization – Cost-based and Heuristic-based Optimization. 3. Programming exercises on Transaction Processing. 4. Programming exercises on Concurrency Control Protocols. 5. Programming exercises on Log-based Recovery Management. 6. Programming exercises on Distributed Transaction Processing, and Distributed Query Processing.				

References	
1.	Silberschatz ,Korth, Sudarshan : Database System Concepts, McGrall Hill.
2.	Elmasri and Navathe: Fundamentals of Database Systems, 3rd Edition, Addison Wesley.
3.	Hector, Ullman, Widom: Database System Implementation, Pearson.
4.	Ceri and Pelagatti: Distributed Databases – Principles and Systems, McGraw Hill.

Biometrics					
Prerequisite: :A basic knowledge of statistics, linear algebra, and programming is expected.		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Introduction: Person recognition, Biometric systems, Biometric functionalities, biometrics system errors, the design cycle of biometric systems				6
Unit 2	Fingerprint recognition: friction ridge patterns, finger print acquisition, feature extraction and matching, palm prints				8
Unit 3	Face recognition: image acquisitions, face detection, feature extraction and matching, handling pose, illumination and expression variations				8
Unit 4	Iris recognition: image acquisition, Iris segmentation, Iris normalization, Iris encoding and matching, Iris quality assessment techniques				6
Unit 5	Additional Biometric Traits: Ear, Gait, Hand geometry, Soft biometrics Multimiometrics: sources of multiple evidence, fusion levels: sensor, feature, score, rank and decision level fusion				8
Unit 6	Security of biometric systems: adversary attacks, attacks at user interface, attacks on biometric processing, attacks on template database				6
References					
1.	Introduction to Biometrics, Anil K Jain Arun Ross, Springer				
2.	The Science of Biometrics, Ravindra Das, Springer				
3.	Practical Biometrics, Julian Ashbourn, Springer				
4.	Introduction to Biometrics, Anil K Jain Arun Ross, Springer				

Biometrics Lab				
Prerequisite: A basic knowledge of statistics, linear algebra, and programming is expected.	L	T	P	C
	0	0	2	1

Course Content		
	Familiarization with image processing toolbox, implementation of fingerprint recognition algorithms and systems, feature extraction and matching algorithms, design of face recognition systems, face detection, implementation of iris recognition systems, design of multimodal biometric system using fingerprint, face, speech, etc., fusion strategies, design of biometric system using other biometric traits (ear, gait, Hand geometry, etc.), Security of biometric systems, Mini project	

References	
1.	Introduction to Biometrics, Anil K Jain Arun Ross, Springer
2.	The Science of Biometrics, Ravindra Das, Springer
3.	Practical Biometrics, Julian Ashbourn, Springer

<table>
<tr><th colspan="6">Computer Vision</th></tr>
<tr><td colspan="2">Prerequisite: Fundamental knowledge on image processing and machine learning, basics of linear algebra and calculus, and programming skills</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><th colspan="5">Course Content</th><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4">Introduction to Computer Vision: Applications of Computer Vision, Basic concepts of Digital Image Formation, Capture and Representation, Pixel Relationships, Linear Filtering, Correlation, Convolution, Image in Frequency Domain, Fourier Transform</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Visual Features and Representation: Edge, Corner Detection, SIFT, SURF, HoG, LBP, GLCM, etc.
Feature Matching, Bag-of-words, VLAD, RANSAC, Hough Transform, Image Pyramids, 2D Transformations</td><td>12</td></tr>
<tr><td>Unit 3</td><td colspan="4">Machine Learning for Computer Vision:
Basic stages in Machine Learning, Image classification, Object Detection, Semantic Segmentation
Overview of Machine Learning Algorithms: Neural Networks, Support Vector Machine (SVM), Dimensionality Reduction Techniques</td><td>6</td></tr>
<tr><td>Unit 4</td><td colspan="4">Convolutional Neural Networks (CNNs): Introduction to CNNs, Evolution of CNN Architectures, Visualization and Understanding CNN
CNNs for Different Computer Vision Tasks: Recognition, Detection, Segmentation, and Activity Recognition</td><td>8</td></tr>
<tr><td>Unit 5</td><td colspan="4">Deep Generative Models in Vision: GANs, VAEs, etc.
Modern Approaches: Attention Models in Vision, Vision Transformer (ViT)</td><td>8</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010</td></tr>
<tr><td>2.</td><td colspan="5">Simon Prince, Computer Vision: Models, Learning, and Inference, 2012</td></tr>
<tr><td>3.</td><td colspan="5">Bishop, Christopher M, Pattern Recognition and Machine Learning, Springer, 2006</td></tr>
<tr><td>4.</td><td colspan="5">Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016</td></tr>
</table>

Computer Vision Lab					
Prerequisite: Fundamental knowledge on image processing, machine learning, and programming skills		L	T	P	C
Total hours: 28		0	0	2	2
Course Content					Hrs
Unit 1	1. Familiarization with various computer vision tools 2. Basic operations on images and videos 3. Linear filtering and convolution 4. Implementation of different image transforms				8
Unit 2	1. Implementation of various feature descriptors (SIFT, SURF, HoG, LBP, GLCM, etc.) Edge Detection, Line Detection and Corner Detection 2. Edge detection, line detection and corner detection 3. Implementation of feature matching algorithms				10
Unit 3	1. Multi-layer Perceptrons, Backpropagation and its applications 2. Implementation of CNN architectures for various tasks such as classification, segmentation, object detection, etc., and transfer learning 3. Implementation of GAN and ViT models 4. Mini project				10
References:					
1.	Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010				
2.	Bishop, Christopher M, Pattern Recognition and Machine Learning, Springer, 2006				
3.	Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016				

<table>
<tr><td colspan="7"><b>Data Analytics</b></td></tr>
<tr><td colspan="3">Prerequisite: Basic understanding of probability and statistics, linear algebra and calculus. A basic knowledge of programming (preferably Python) is essential.</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="3">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="6"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td colspan="2">Unit 1</td><td colspan="4">Data Understanding and Preparation Introduction, Reading data from various sources, Data visualization, Distributions and summary statistics, Relationships among variables, Extent of Missing Data. Segmentation, Outlier detection, Automated Data Preparation, Combining data files, Aggregate Data, Duplicate Removal, Sampling DATA, Data Caching, Partitioning data, Missing Values Data : Gather, extract, analyse, and manipulate data to draw conclusions or insights. With algorithms and coding with dataset available</td><td>10</td></tr>
<tr><td colspan="2">Unit 2</td><td colspan="4">Introduction to Data Mining: Classification- Naïve Bayes, Clustering- K means , Model development & techniques Data Partitioning, Model selection, Model Development Techniques,</td><td>10</td></tr>
<tr><td colspan="2">Unit 3</td><td colspan="4">Neural networks, Decision trees, Logistic regression, Discriminant analysis, Support vector machine, Bayesian Networks, Linear Regression, Logistic Regression, Association rules..</td><td>10</td></tr>
<tr><td colspan="2">Unit 4</td><td colspan="4">Model Evaluation and Deployment Introduction, Model Validation, Rule Induction Using CHAID, Automating Models for Categorical and Continuous targets, Comparing and Combining Models, Evaluation Charts for Model Comparison, Meta Level Modelling, Deploying Model, Assessing Model Performance, Updating a Model. Visualisation</td><td>10</td></tr>
<tr><td colspan="7">References</td></tr>
<tr><td>1.</td><td colspan="6">Daniel T. Larose and Chantal D. Larose, Discovering Knowledge in Data: An Introduction to Data Mining, 2nd Edition, Wiley, 2014. ISBN: 978-0-470-90874-7</td></tr>
<tr><td>2.</td><td colspan="6">Recommended Reading: Foster Provost and Tom Fawcett, Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking, O’Reilly, 2013. ISBN: 978-1-449-36132-7</td></tr>
</table>

Data Analytics Lab					
Prerequisite: Python programming basics		L	T	P	C
Total hours: 28		0	0	2	2
Course Content					Hrs
Unit 1	Visualization: a. Find the data distributions using box and scatter plot. b. Find the outliers using plot. c. Plot the histogram, bar chart and pie chart on sample data				4
Unit 2	Descriptive statistics in R a. Write an R script to find basic descriptive statistics using summary b. Write an R script to find subset of dataset by using subset Reading and writing different types of datasets a. Reading different types of data sets (.txt, .csv) from web and disk and writing in file in specific disk location. b. Reading Excel data sheet in R. c. Reading XML dataset in R.				6
Unit 3	Descriptive statistics in R a. Write an R script to find basic descriptive statistics using summary b. Write an R script to find subset of dataset by using subset Apply multiple regressions, if data have a continuous independent variable. Apply on above dataset.				6
Unit 4	a. Install relevant package for classification. b. Choose classifier for classification problem. c. Evaluate the performance of classifier.				6
Unit 5	Installing Hadoop, PIG, Hive, Visualizing Big data sets, Applying Parallel machine learning models to handle large scale data.				6
References:					
1.	Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media, 2017.				
2.	Joshua N. Milligan, Learning Tableau: Create effective data visualizations, build interactive visual analytics and transform your organization, Packt Publishing Limited, 2020.				
3.	Nathan Marz, James Warren: Big Data: Principles and best practices of scalable realtime data systems, 2020.				

Data Mining					
Prerequisite: :		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Overview of the Data Mining and Knowledge Discovery from Databases Process, Data Warehousing and OLAP, Data Preprocessing: Summary Data Structures, imensionality Reduction				6
Unit 2	Association Rule Mining: Frequent Item set Mining Methods, Rule Generation, Interestingness Measures				6
Unit 3	Classification: Decision Trees, Instance Based, Support Vector Machines, Computational Learning Theory, Associative Classification				10
Unit 4	Clustering: Partitional, Hierarchical, Density Based, Grid Based, Advanced Methods				7
Unit 5	Sequence Mining Complex Data Mining				6
Unit 6	Web Mining: Information Retrieval, Link Analysis, Search Engines, Usage Analysis Data Mining Applications				7
References					
1.	1. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann/Elsevier India, 3rd edition, 2011.				
2.	Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann, 2005, ISBN: 0-12-088407-0.				

Data Mining Lab

Prerequisite:	L	T	P	C
Total hours: 28	0	0	2	2

Course Content		Hrs
Unit 1	Implementation of Data Pre-processing	4
Unit 2	Implementation of Association Rule Mining	4
Unit 3	Implementation of Decision Trees and Support Vector Machines	6
Unit 4	Implementation of Various Clustering Algorithms	8
Unit 5	Implementation of Sequence Mining Algorithms	2
Unit 6	Implementation of Web Mining Algorithms	4

References	
1.	1. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann/Elsevier India, 3rd edition, 2011.
2.	Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann, 2005, ISBN: 0-12-088407-0.

Decentralized Learning				
Prerequisite: Machine learning, deep learning, etc.	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs
Unit - I	**Challenges in Big Data and Traditional AI:** Understanding nature of Big Data, Data privacy as a bottleneck, Impact of training Data and model bias, Model drift and performance degradation, FL as the main solution for a data problems.	4
Unit - II	**Introduction to Federated Learning:** Understanding the current state of ML, Distributed Learning, Understanding of FL, FL system considerations, FL system architecture, Understanding of FL system flow: from initialization to continuous operations, Basics of secure aggregation, Different designs of FL (HFL, VFL, FTL, RFL).	8
Unit - III	**Systems and Frameworks :** Statistical and Systems Heterogeneity, Statistical and Systems Heterogeneity, Variations of federated aggregations, Local Training and Scalability of Federated Learning Systems, Straggler Management, Systems Bias in Federated Learning, Frameworks of FL (FATE, Flower, Tensorflow Federated(TFF), OpenFL, PySyft)	11
Unit - IV	**Privacy and Security:** Data Leakage in Federated Learning, Differential Privacy within Federated Systems, Approach for protecting against Data Leakage, Private Parameter aggregation for Federated Learning, Security and Robustness to Federated Learning, Dealing with Byzantine threats to neural network.	11
Unit - V	Decentralized learning for communication systems and blockchain for DL	6

References	
1.	Federated Learning with Python: Design and Implement a Federated Learning System and Develop Applications Using Existing Frameworks, George Jeno, Kiyoshi Nakayama, Packt Publishing, Limited, 2022.
2.	Federated Learning: A Comprehensive Overview of Methods and Applications, Heiko Ludwig, Nathalie Baracaldo, Springer Nature, 2022.
3.	The course materials are mainly from the lecturing slides I?ve made and research papers from top conferences like NDISS, USNIX, SIGCOMM, MOBICOM, NSDI, MobiSys etc.

Decentralized learning Lab					
Prerequisite: : The programming lab in python and ML tools		L	T	P	C
Total hours: 36		0	0	2	2
Course Content					Hrs
	The laboratory experiments conducted on various tools Lab 1-3: Develop and FL setup using IOT devices and a sever setup to design FL average or other. Lab 3-4: Collaborative Learning experiments using FL Lab 5-9: Experiments on Data Model poisoning Lab 10-12 : Projects on Decentralized learning				36
References					
1.	Federated Learning with Python: Design and Implement a Federated Learning System and Develop Applications Using Existing Frameworks, George Jeno, Kiyoshi Nakayama, Packt Publishing, Limited, 2022.				
2.	Federated Learning: A Comprehensive Overview of Methods and Applications, Heiko Ludwig, Nathalie Baracaldo, Springer Nature, 2022.				
3.	The course materials are mainly from the lecturing slides I?ve made and research papers from top conferences like NDISS, USNIX, SIGCOMM, MOBICOM, NSDI, MobiSys etc.				

Embedded System Design				
Prerequisite: Compiler Design	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to embedded systems., design representations, level of abstractions, design methodologies. Models and architectures, Taxonomy of models and architectures	8
Unit 2	Brief descriptions of specification languages, Specification requirement for embedded systems, Spec Chart and Spec Chart Description. Design challenges & issues, hardware and software design, co-design of software and hardware, ASIC	18
Unit 3	Design quality estimation: Quality matrix, software and hardware estimation.	8
Unit 4	Microcontroller 8051: Architecture, programming, interfacing and use cases, Instruction addressing modes, Interrupts, Counters and timers. Introduction to the ARM processors.	8

References	
1.	Embedded Systems: Real-Time Operating Systems for Arm Cortex-M Microcontrollers" by Jonathan Valvano
2.	Embedded Systems: Design and Applications with the 68HC12 and HCS12" by Steven Barrett and Daniel Pack
3.	Embedded Systems: A Contemporary Design Tool" by James K. Peckol

Embedded System Design Lab

Prerequisite: Digital Logic Design	L	T	P	C
Total hours: 28	0	0	2	2

	Course Content	Hrs
Unit 1	**Proteus Introduction:** Introduce students to the basic circuit simulations and simulator user interface of Proteus. **LED Blinking:** Design a simple embedded system using a microcontroller to blink set of LEDs at the specific patterns. Vary the blinking pattern and observe the LEDs behaviour. **LCD Display:** Write programs in the C programming language to initialize the 16x2 LCD and display the message. The messages to displayed are defined in the code.	8
Unit 2	**Keypad Interfacing:** Design the hex keypad using push button then displayed the pressed key on the LCD. Next use the predefined he keypad module to implement simple calculator. **Seven Segment Display:** design the seven segment display using LEDs. Use predefined seven segment module and interface it with hex keypad to display the pressed digit on the display. **Digital Counter:** Design four digit increment and decrement digital counter using a microcontroller and display the count on a 7-segment display.	10
Unit 3	**Elevator Movement:** Design the program to simulate the movement of elevator and display the floor numbers when keys are pressed inside the elevators or it moves. **Quiz Buzzer:** Design a quiz buzzer system that will act as the central controller for the quiz system, managing participant buzzers and increment the counter for the team who press the buzzer button first. **Date & Time: D**esign an embedded system to display the current date and time on an LCD (Liquid Crystal Display) using a Real-Time Clock (RTC) module.	10

References:	
1.	Embedded Systems: Real-Time Operating Systems for Arm Cortex-M Microcontrollers" by Jonathan Valvano
2.	Embedded Systems: A Contemporary Design Tool" by James K. Peckol

Full Stack Development					
Prerequisite:		L	T	P	C
Total Hours: **42**		3	0	0	3
Course Content					Hrs
Unit 1	***Introduction to Full Stack Development:*** Overview of full stack development, roles and responsibilities of a full stack developer, Understanding the client-server architecture. ***Front-End Development Fundamentals:*** Introduction to HTML5, CSS3, JavaScript, jQuery, Bootstrap. Building web pages with styling, programming concepts, and syntax. DOM manipulation and event handling. Introduction to responsive design and mobile-first development.				10
Unit 2	***Front-End Frameworks:*** Introduction to popular front-end frameworks (e.g., React, Angular, Vue.js). Building dynamic and interactive web applications using a framework of choice Components, state management, and routing in front-end frameworks. Working with APIs to fetch and update data Handling, form input and validation.				10
Unit 3	***Back-End Development Fundamentals:*** Introduction to server-side programming languages (e.g., PHP, Node.js, Python), Working with HTTP protocols and RESTful APIs, Handling data persistence with databases (SQL or NoSQL- MongoDB, Cassandra), Creating server-side routes and handling requests, Implementing user authentication and authorization.				8
Unit 4	***Back-End Frameworks and APIs:*** Building server-side applications using a framework (e.g., Express, Django), Implementing APIs for data retrieval and manipulation, Database integration and querying, Securing APIs and implementing access control. ***Full Stack Frameworks:*** MEAN Stack (MongoDB, Express.js, Angular, Node.js), MERN Stack (MongoDB, Express.js, React, Node.js), LAMP Stack (Linux, Apache, MySQL, PHP).				8
Unit 5	***Deployment and DevOps:*** Introduction to cloud platforms and hosting providers, Deploying web applications to cloud platforms, Continuous integration and deployment (CI/CD), Version control systems (e.g., Git) and collaboration tools.				6
References					
1.	"Full Stack JavaScript Development with MEAN" by Adam Bretz, Colin J. Ihrig, and Patrick Mulder (Publisher: Manning Publications, 2016)				
2.	"Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" by Vasan Subramanian (Publisher: Apress, 2019)				
3.	"Full Stack Development for Beginners: Learn React, Node.js, MongoDB, and TypeScript" by Jonas Fehre (Publisher: Packt Publishing, 2021)				
4.	"Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins (Publisher: O'Reilly Media, 2018)				

5.	"MongoDB: The Definitive Guide: Powerful and Scalable Data Storage" by Kristina Chodorow (Publisher: O'Reilly Media, 2013)
6.	"JavaScript and jQuery: Interactive Front-End Web Development" by Jon Duckett (Publisher: Wiley, 2014)
7.	"PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide" by Larry Ullman (Publisher: Peachpit Press, 2017)
8.	"Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4" by Jacob Lett (Publisher: Packt Publishing, 2018)

Full Stack Development Lab				
Prerequisite:.	L	T	P	C
Hrs: 28	0	0	2	2

Course Content		Hrs
Unit 1	Creation of basic web pages/applications using HTML and CSS. Write JavaScript code to create interactive elements on a webpage, such as buttons, forms, and event handling. Use server-side languages like Node.js, Python (with Flask/Django), or Ruby (with Ruby on Rails) to create a basic web application. Integrate a database (e.g., MySQL, PostgreSQL, MongoDB) with your web application to store and retrieve data. Implement user registration and login functionality using authentication techniques such as JWT (JSON Web Tokens) or OAuth. Create a RESTful API that allows users to perform CRUD (Create, Read, Update, Delete) operations on resources.	10
Unit 2	Use popular front-end frameworks like React, Angular, or Vue.js to build more dynamic and interactive web applications. Build a single-page application that communicates with the backend through APIs and updates the UI without full page reloads. Implement state management in your front-end application using tools like Redux or Vuex. Ensure your web application looks and functions well on different devices and screen sizes. Write unit tests for your code and debug common issues in both the front-end and back-end.	10
Unit 3	Deployment and Hosting: Deploy your full-stack application to a cloud platform (e.g., AWS, Azure, Heroku) and make it publicly accessible. Version Control: Use Git for version control to manage changes to your codebase effectively. Real-time Features: Integrate real-time communication features using technologies like WebSockets or WebRTC. Security Considerations: Implement security best practices to protect your application from common web vulnerabilities (e.g., Cross-Site Scripting, SQL Injection).	8

References	
1.	"Full Stack JavaScript Development with MEAN" by Adam Bretz, Colin J. Ihrig, and Patrick Mulder (Publisher: Manning Publications, 2016)
2.	"Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" by Vasan Subramanian (Publisher: Apress, 2019)
3.	"Full Stack Development for Beginners: Learn React, Node.js, MongoDB, and TypeScript" by Jonas Fehre (Publisher: Packt Publishing, 2021)
4.	"Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins (Publisher: O'Reilly Media, 2018)

5.	"MongoDB: The Definitive Guide: Powerful and Scalable Data Storage" by Kristina Chodorow (Publisher: O'Reilly Media, 2013)
6.	"JavaScript and jQuery: Interactive Front-End Web Development" by Jon Duckett (Publisher: Wiley, 2014)
7.	"PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide" by Larry Ullman (Publisher: Peachpit Press, 2017)
8.	"Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4" by Jacob Lett (Publisher: Packt Publishing, 2018)

<table>
<tr><th colspan="6"><b>Information Retrieval</b></th></tr>
<tr><td colspan="2">Prerequisite: nil</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 40</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4">Introduction: Goals and history of IR. The impact of the web on IR. Basic IR Models: Boolean and vector-space retrieval models; ranked retrieval; text-similarity metrics; TF-IDF (term frequency/inverse document frequency) weighting; cosine similarity. Basic Tokenizing, Indexing, and Implementation of Vector-Space Retrieval: Simple tokenizing, stop-word removal, and stemming; inverted indices; efficient processing with sparse vectors;</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Performance metrics: recall, precision, F-measure, and NPCG; Evaluations on benchmark text collections. Query Operations: Relevance feedback; Query expansion.
Text Representation: Word statistics; Zipf's law; Porter stemmer; morphology; index term selection; using thesauri.</td><td>8</td></tr>
<tr><td>Unit 3</td><td colspan="4">Web Search: Search engines; spidering; meta-crawlers; directed spidering; link analysis , HITS, hubs and authorities, Google PageRank); Text Categorization: Categorization algorithms: Rocchio, nearest neighbor</td><td>8</td></tr>
<tr><td>Unit 4</td><td colspan="4">Text Classification :Language-Model Based Retrieval : Using naive Bayes text classification for ad hoc retrieval. Improved smoothing for document retrieval.
Text Clustering: Clustering algorithms: agglomerative clustering; k-means; expectation maximization (EM). Applications to web search and information organization.
Recommender Systems: Read this paper by Herlocker et al.
Collaborative filtering and content-based recommendation of documents and products.</td><td>8</td></tr>
<tr><td>Unit 5</td><td colspan="4">Recommender Systems: Collaborative filtering and content-based recommendation of documents and products. Ethical Issues in IR: Privacy, Fairness, Fake news and disinformation, Filter bubble, Viewpoint diversity, fostering extremism, Internet addiction.
Information Extraction and Integration: Extracting data from text; semantic web; collecting and integrating specialized information on the web. Question Answering: Semantic parsing. Question Answering from structured data and text.
Deep Learning for IR: Word embeddings. Neural language models.</td><td>8</td></tr>
<tr><td colspan="6">References</td></tr>
<tr><td>1.</td><td colspan="5">Modern Information Retrieval, Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Addison-Wesley, 2000. http://people.ischool.berkeley.edu/~hearst/irbook/</td></tr>
<tr><td>2.</td><td colspan="5">Information Retrieval: Implementing and Evaluating Search Engines by S. Buttcher, C. Clarke and G. Cormack, MIT Press, 2010.</td></tr>
<tr><td>3.</td><td colspan="5">Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data by B. Liu, Springer, Second Edition, 2011.</td></tr>
<tr><td>4.</td><td colspan="5">Cross-Language Information Retrieval by By Jian-Yun Nie Morgan & Claypool Publisher series 2010</td></tr>
<tr><td>5.</td><td colspan="5">Multimedia Information Retrieval by Stefan M. Rüger Morgan & Claypool Publisher series 2010.</td></tr>
<tr><td>6</td><td colspan="5">Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. (Eds.), Recommender Systems Handbook. 1st Edition., 2011, 845 p. 20 illus., HarPCover, ISBN: 978-0-387-85819-7, Relevant Research Papers</td></tr>
</table>

Information Retrieval Lab				
	L	T	P	C
Total hours: 28	0	0	2	2

Course Content		Hrs
Assignment 1	Pre-processing of a Text Document: Stop word removal, Stemmer, Lemmatizer, Cosine-Similarity Measure using TF-IDF	4
Assignment 2	Document similarity using Boolean Model / Extended Boolean Model / Vector Space Model	4
Assignment 3	Inverted Index and postings	2
Assignment 4	Simple web crawler	2
Assignment 5	Terrier search engine	4
Assignment 6	Parse XML text, generate Web graph and compute topic specific page rank - Page Rank Algorithm	4
Assignment 7	Mining Flipkart review and perform sentiment analysis based on both review and ratings.	4
Assignment 8	Perform movie recommendation system by scrapping real-time movie rating from imdb website.	4

References:	
1.	https://www.analyticsvidhya.com/blog/2021/09/essential-text-pre-processing-techniques-for-nlp/ https://www.kaggle.com/code/sudalairajkumar/getting-started-with-text-preprocessing
2.	https://minoli110.medium.com/use-of-vector-space-model-vsm-to-calculate-the-similarity-between-2-text-documents-3dfb31138fc2 https://github.com/manan-paneri-99/Vector-Space-based-Document-Retrieval-system https://www.geeksforgeeks.org/document-retrieval-using-boolean-model-and-vector-space-model/
3.	https://medium.com/@fro_g/writing-a-simple-inverted-index-in-python-3c8bcb52169a https://towardsdatascience.com/using-inverted-index-for-efficient-document-similarity-computation-a8d3fb8f0c12
4.	https://towardsdatascience.com/web-scraping-with-scrapy-8071fd627051 https://towardsdatascience.com/https-towardsdatascience-com-5-tips-to-create-a-more-reliable-web-crawler-3efb6878f8db
5.	http://terrier.org/docs/v2.2.1/index.html https://github.com/terrier-org/terrier-core
6.	https://www.geeksforgeeks.org/xml-parsing-python/ https://towardsdatascience.com/pagerank-3c568a7d2332
7.	https://www.analyticsvidhya.com/blog/2022/09/sentiment-analysis-on-flipkart-dataset/ https://machinelearningprojects.net/flipkart-reviews-extraction-and-sentiment-analysis/
8.	https://www.geeksforgeeks.org/scrape-imdb-movie-rating-and-details-using-python/ https://janineb.medium.com/movie-recommendation-system-d6aa8583cdb

Internet of Things

Prerequisite: Nil	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs
Unit 1	Introduction: Internet of Things and Connected Products, IoT paradigm, Smart objects, Goal orientation, Convergence of technologies; Business Aspects of the Internet of Things. IoT Architectures and Protocols: Importance, Communication models in IoT, Layers in IoT architecture, Role of protocols in IoT communication.	6
Unit 2	Wireless Technologies for IoT: Wi-Fi and 802.15.x family - features, range, power consumption; Zigbee: Network topology, mesh networking, Zigbee stack, LoRaWAN, SigFox, Cellular technologies for IoT (2G, 3G, 4G, NB-IoT, 5G). Latest developments in communication technologies.	8
Unit 3	IoT Network Topologies: Overview (star, mesh, hybrid, etc.), selection based on advantages and limitations. Network Protocols: overview of IoT network protocols. IPv6 and its significance in IoT addressing, 6LoWPAN- IPv6 over Low-power Wireless Personal Area Networks, Header Compression. RPL-overview and operation, Case studies of network protocols in IoT deployments.	10
Unit 4	IoT Application Protocols: Introduction (MQTT, CoAP, HTTP, etc.), Comparison of IoT protocols (features, performance, scalability), MQTT - Concepts, message structure, QoS levels, CoAP - Principles, RESTful architecture, resource discovery, HTTP in IoT- Web services, REST APIs, JSON/XML data formats. Data: OMA Lightweight M2M (LwM2M) protocol, OPC Unified Architecture (OPC UA), BACnet, Modbus. Data Protocols and Formats: IoT data formats (JSON, XML, CBOR, Protocol Buffers). IoT Standards and Interoperability	10
Unit 5	IoT Security Protocols: Security challenges and threats in IoT, Secure communication protocols for IoT (DTLS, TLS, IPsec), Authentication and access control in IoT, Security protocols for device management and firmware updates, Privacy protection and data encryption in IoT.	4
Unit 6	Emerging Trends: Blockchain, 5G and its impact, IoT and edge-cloud integration, latest advancements in IoT architectures and protocols.	4

References	
1.	The Internet of Things: Key Applications and Protocols, David Boswarthick, OlivierHersent, and Omar Elloumi, Wiley.
2.	Building the Internet of Things with IPv6 and MIPv6, Daniel Minoli, Wiley.
3.	Architecting the Internet of Things, Dieter Uckelmann, Mark Harrison, and Florian Michahelles, Springer.
4.	Latest research articles.

Internet of Things Lab					
Prerequisite:		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs
Unit 1	Setting up communication using XBEE and BLE. Data Exchange and interfacing Sensors				8
Unit 2	Programming on Cooja and CoAP Setting up CoAP on programmable boards.				10
Unit 3	Using CoAP to set up communication. Implementation of homogeneous and heterogeneous networks. Data processing on application layer.				10
References:					
1.	Building the Internet of Things with IPv6 and MIPv6, Daniel Minoli, Wiley.				
2.	Architecting the Internet of Things, Dieter Uckelmann, Mark Harrison, and Florian Michahelles, Springer.				
3.	Latest research articles.				

IoT based Robotics				
Prerequisite: Nil	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to IoT and Robotics: Overview of IoT and Robotics; Historical development of IoT and Robotics; Applications of IoT and Robotics; Types of IoT devices; Types of Robotics;	6
Unit 2	Introduction to the Internet of Things. Protocols and Architectures. IoT Hardware: IoT devices and sensors; IoT networks and communication protocols; IoT gateways and controllers; IoT platforms and services	10
Unit 3	IoT Software: Introduction to IoT protocols; IoT data management and analytics; IoT security and privacy; IoT programming and development;	8
Unit 4	Robotics Fundamentals: Robotics history and evolution; Robotics components and structure. Robotics Hardware: Types of robots and their applications; Robotics sensors and actuators; Robotics control systems; Robotics power systems. Robotics Software: Robotics programming and development; Robotics motion planning and control; Robotics perception and vision; Robotics intelligence and autonomy.	10
Unit 5	Robotics Applications: Industrial Robotics; Service Robotics; Medical Robotics	4
Unit 6	IoT and Robotics Integration: Use cases and examples; Challenges and opportunities; Future trends and directions	4

References	
1.	The Internet of Things: Key Applications and Protocols, David Boswarthick, Olivier Hersent, and Omar Elloumi, Wiley
2.	Building the Internet of Things with IPv6 and MIPv6, Daniel Minoli, Wiley
3.	Learn Robotics Programming, Danny Staple, Packt Publishing, 2nd ed.
4.	Robotics Simplified, Jisu Elsa Jacob and Manjunath N, BPB Publications.

IoT based Robotics Lab					
Prerequisite:		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs
Unit 1	Setting up communication using XBEE and BLE. Data Exchange and interfacing Sensors				8
Unit 2	Interfacing with Actuators. Programming Motion and automation. Controller based interfacing.				10
Unit 3	Visual interfacing and controlling motion. Analysis of robotic arm and conveyor belts.				10
References:					
1.	Learn Robotics Programming, Danny Staple, Packt Publishing, 2nd ed.				
2.	Robotics Simplified, Jisu Elsa Jacob and Manjunath N, BPB Publications.				

Natural Language Processing				
Prerequisite:	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Introduction to NLP - Various stages of NLP –The Ambiguity of Language: Why NLP Is DifficultParts of Speech: Nouns and Pronouns, Words: Determiners and adjectives, verbs, Phrase Structure. Statistics Essential Information Theory : Entropy, perplexity, The relation to language, Cross Entropy, Character Encoding, Word Segmentation, Sentence Segmentation, Introduction to Corpora, Corpora Analysis. Inflectional and Derivation Morphology, Morphological analysis and generation using Finite State Automata and Finite State transducer.	6
Unit 2	Language Modelling, Words: Collocations- Frequency-Mean and Variance –Hypothesis testing:The t test, Hypothesis testing of differences, Pearson's chi-square test, Likelihood ratios. Statistical Inference: n –gram Models over Sparse Data: Bins: Forming Equivalence Classes- N gram model – Statistical Estimators- Combining Estimators	6
Unit 3	Word Sense Disambiguation, Methodological Preliminaries, Supervised Disambiguation: Bayesian classification, An informationtheoretic approach, Dictionary-Based Disambiguation: Disambiguation based on sense, Thesaurusbased disambiguation, Disambiguation based on translations in a second-language corpus.	6
Unit 4	Markov Model: Hidden Markov model, Fundamentals, Probability of properties, Parameter estimation, Variants, Multiple input observation. The Information Sources in Tagging: Markov model taggers, Viterbi algorithm, Applying HMMs to POS tagging, Applications of Tagging	6
Unit 5	Parsing, The Probability of a String, Problems with the Inside-Outside Algorithm, Parsing for disambiguation, Treebanks, Parsing models vs. language models, Phrase structure grammars and dependency, Lexicalized models using derivational histories, Dependency-based models.	8
Unit 5	Shallow Parsing and Chunking, Shallow Parsing with Conditional Random Fields (CRF), Lexical Semantics, WordNet, Thematic Roles, Semantic Role Labelling with CRFs. Statistical Alignment and Machine Translation, Text alignment, Word alignment, Information extraction, Text mining, Information Retrieval, NL interfaces, Sentimental Analysis, Question Answering Systems, Social network analysis, Text Summarization.	10

References	
1.	D. Jurafsky, J.H. Martin, Speech and Language Processing, 3rd Online Edition (available at https://web.stanford.edu/~jurafsky/slp3/).
2.	J. Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.

Natural Language Processing Lab

Prerequisite:	L	T	P	C
Total hours: 28	0	0	2	2

	Course Content	Hrs
Unit 1	Implementation of Pre-processing of Text (Tokenization, Stop word Removal, Stemming and Lemmatization etc.) and Morphological Analysis	6
Unit 2	Implementation of N-gram Models	4
Unit 3	Implementation of Word Sense Disambiguation	2
Unit 4	Implementation of POS Tagging and Named Entity Recognition	6
Unit 5	Implementation of CKY Parsing and Mini Project	10

References	
1.	D. Jurafsky, J.H. Martin, Speech and Language Processing, 3rd Online Edition (available at https://web.stanford.edu/~jurafsky/slp3/).
2.	J. Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.

Program Analysis					
Prerequisite: Operating system, Computer Architecture and organization		L	T	P	C
Total Hours: **42**		3	0	0	3
Course Content					Hrs
Unit 1	Structure of a program, data flow and control flow, basic blocks, forward and backward flow analysis; Program Analysis Tools: debuggers, disassemblers, decompilers, emulators, virtualized environments				8
Unit 2	Dataflow Analyses and Transformations; Control Dependence Graph, Inter-procedural analysis, Program Analysis via Graph Reachability				8
Unit 3	Program Analysis Techniques: Static analysis, Dynamic analysis, Instrumentation, Hybrid analysis, Automatic Generation of High-Coverage Tests for Complex Systems Programs, Input fuzzing,				8
Unit 4	Control-flow analysis; Abstract interpretation; Symbolic execution and program testing, Mixing type checking and symbolic execution; Model checking				8
Unit 5	Case Study: C, JAVA, .apk, .elf for program analysis				10
References					
1.	Nielson, Nielson, and Hankin: Principles of Program Analysis, Springer.				
2.	Ravi Sethi, Alfred V. Aho, Monica S. Lam, D. Jeffrey Ulman: Compilers : Principles, Techniques, & Tools, 2nd Edition, Pearson				
3.	Muchnick: Advanced Compiler Design and Implementation, Morgan Kaufmann.				
4.	Published articles from reputed Journals and Conferences.				
5.	"Program Analysis and Compilation: Theory and Practice" by Thomas Reps, Mooly Sagiv, and J. Lim. (Year: 1997, Publisher: MIT Press)				
6.	"Program Analysis for Software Tools and Engineering" by Thomas Reps. (Year: 2007, Publisher: Springer)				
7.	"Data Flow Analysis: Theory and Practice" by Uday P. Khedker. (Year: 2015, Publisher: CRC Press)				
8.	"Formal Methods for Software Verification" by Grigore Rosu and Klaus Havelund. (Year: 2018, Publisher: CRC Press)				

Program Analysis Lab					
Prerequisite:		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs.
Unit 1	1) Analyze the program to identify any potential code optimization opportunities. a) Write a program that contains multiple nested conditional statements (if, else if, else). b) Use Egypt tool to create a flow graph of this program. Visualize the graph using Graphviz and SVG. 2) Design and implement a control flow analysis for detecting potential null pointer dereferences in each C/C++ codebase using the "Clang Static Analyzer" tool. a) Choose a C/C++ codebase (your project or open-source project) with multiple occurrences of pointer usages and potential null pointer dereferences. Use the "Clang Static Analyzer" to perform control flow analysis on the selected codebase. b) Identify expressions or code paths where the "Clang Static Analyzer" detects potential null pointer dereferences. c) Note the location of potential null pointer dereferences identified by the tool. 3) Implement dynamic program analysis techniques to look for the pointers and array during runtime. a) Generate a program flow during runtime using dynamic binary analysis tool. b) Note the location of pointers and array identified by the tool. 4) Implement program analysis using emulators. a) Write a simple C program that simulates a basic calculator. b) Choose a suitable emulator tool (e.g., QEMU, Bochs, VirtualBox, etc.) and install it on your machine. c) Compile the C program for the target architecture compatible with the emulator environment. d) Run the program within the emulator environment. e) Use the emulator's debugging and monitoring features to analyse the program's execution. f) Find out the entry point of the program. g) Trace the flow of the program execution, including function calls and return statements, using the emulator's tracing features. 5) Perform backward flow analysis using cflow to identify reaching definitions for each variable. Determine which assignments may influence the value of a variable at a particular point in the program. a) Write a C program that includes assignments to variables and multiple points where variables are used and modified. b) Execute the C program with cflow to perform backward flow analysis. c) Observe the output generated by cflow, which represents the reaching definitions for each variable at different points in the program. d) Identify and list the assignments that may influence the value of a variable at a particular point in the program. 6) Implement binary structure analysis on a compiled executable. a) Choose a compiled binary your project or open-source binary. b) Select the tool for examining ELF file. c) Use the selected tool to analyse the binary and identify the number of registers used by the program. d) Determine the entry point of the binary using the selected tool. The entry point is where the program's execution begins. e) Examine the various sections of the binary (code section, data section, symbol table).				8

	7) Implement a C program that takes input from users or external sources. Use fuzz testing tools to generate random or invalid inputs and analyse how the program handles such inputs. a) Write the C code for the program that takes input from users or external sources. b) Choose a fuzz testing tool (e.g., American Fuzzy Lop (AFL), libFuzzer) and install it on your system. c) Modify the C program to accept input from the fuzz testing tool rather than traditional user input. d) Execute the fuzz testing tool with the modified C program as the target. e) Monitor the C program's behavior during the fuzz testing process. Identify any crashes, exceptions, or unexpected outputs. f) Examine the logs or reports generated by the fuzz testing tool after the testing process 8) Write a C program with several functions and use cflow to generate the function call graph. Identify the functions that have the most dependencies (incoming and outgoing calls) and assess their importance in the program. Consider how reducing dependencies could improve code maintainability. a) Write a C program with several functions that demonstrate function call interactions. b) Use cflow to generate the function call graph for the C program. c) Visualize the call graph to analyse the relationships between functions. d) Find out the high control dependencies in the program.
References:	
1.	Nielson, Nielson, and Hankin: Principles of Program Analysis, Springer.
2.	Ravi Sethi, Alfred V. Aho, Monica S. Lam, D. Jeffrey Ulman: Compilers : Principles, Techniques, & Tools, 2nd Edition, Pearson
3.	Muchnick: Advanced Compiler Design and Implementation, Morgan Kaufmann.
4.	Published articles from reputed Journals and Conferences.
5.	"Program Analysis and Compilation: Theory and Practice" by Thomas Reps, Mooly Sagiv, and J. Lim. (Year: 1997, Publisher: MIT Press)
6.	"Program Analysis for Software Tools and Engineering" by Thomas Reps. (Year: 2007, Publisher: Springer)
7.	"Data Flow Analysis: Theory and Practice" by Uday P. Khedker. (Year: 2015, Publisher: CRC Press)
8.	"Formal Methods for Software Verification" by Grigore Rosu and Klaus Havelund. (Year: 2018, Publisher: CRC Press)

Social Network Analysis				
Prerequisite: Data Structures and Algorithms	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Basics of Graph Theory: Basic definitions of graphs and multigraphs; adjacency matrices, independent sets and cliques, vertex colouring, chromatic number, matching, vertex cover, edge cover, independent set, cut-set, spanning trees.	8
Unit 2	Network Models: Properties of Real-World Networks: Degree Distribution, Clustering Coefficient, Average Path Length. Random Graphs, Small-World Model, Preferential Attachment Model, Modeling of Real-World Networks using Random Graphs, Small-World Model and Preferential Attachment Model	8
Unit 3	Network Centrality: Degree Centrality, Eigenvector Centrality, PageRank, Centrality, Closeness Centrality.	4
Unit 4	Basics of Deep Learning: Basic terminologies, Artificial Neural Networks (ANN) and their limitations, Convolution Neural Networks (CNNs), Limitations of CNNs on graph data	6
Unit 5	Deep Learning on Graphs: Theory of Graph Neural Networks (GNNs) (Graph Convolution Network (GCN), Graph Attention Network (GAT) and GraphSAGE), Semi supervised Node classification using GNNs, Real life applications of GNNs (including Google Maps, Pinterest)	8
Unit 6	Applications of Graph Neural Networks for Network Analysis: Node classification, Link Prediction, Recommendations, Community Detection, MRI analysis, NLP related applications.	8

References	
1.	Social media mining: an introduction, Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu. Cambridge University Press, 2014.
2.	Networks, Crowds, and Markets: Reasoning About a Highly Connected World by David Easley and Jon Kleinberg, Cambridge University Press.
3.	Networks: An introduction by Mark Newman, Oxford university press, 2018.
4.	Graph Representation Learning by William L. Hamilton, 2020.

Social Network Analysis Lab				
Prerequisite: Data Structures and Algorithms	L	T	P	C
Total hours: 28	0	0	2	1

	Course Content	Hrs
Unit 1	Graph Representation and Analysis: • Constructing adjacency matrices for given graphs and multigraphs. • Identifying independent sets and cliques in a network. • Investigating the chromatic number and vertex coloring of a graph. • Finding matching, vertex cover, and edge cover in various graphs. • Analyzing cut-sets and spanning trees in a network.	4
Unit 2	Real-World Network Properties: • Analyzing degree distributions in real-world networks. • Calculating clustering coefficients and average path lengths of networks. • Generating random graphs using different algorithms (e.g., Erdős-Rényi model). • Simulating small-world networks and comparing them with real-world networks. • Building networks based on preferential attachment and studying their properties.	4
Unit 3	Network Centrality Analysis: • Calculating degree centrality for nodes in a given network. • Implementing eigenvector centrality and PageRank algorithms. • Comparing different centrality measures and their outcomes. • Identifying influential nodes in social networks using centrality measures. • Analyzing closeness centrality and its impact on information flow.	4
Unit 4	Deep Learning Basics: • Implementing a simple artificial neural network (ANN) for classification. • Implementing convolutional neural networks (CNNs) for some real life application(s).	4
Unit 5	Graph Neural Network Implementation: • Implementing a basic Graph Convolution Network (GCN) from scratch. • Building a Graph Attention Network (GAT) and understanding its attention mechanism. • Developing a GraphSAGE model for node representation learning. • Comparing the performance of GNNs with traditional methods for node classification.	6
Unit 6	Real-Life Applications of GNNs: • Performing node classification using GNNs on benchmark datasets. • Predicting missing links in a given network using link prediction techniques. • Building a recommendation system based on GNN embeddings. • Detecting communities in social networks using GNN-based algorithms.	6

	- Exploring how GNNs can be applied to analyze networks in MRI data and NLP tasks.	

References	
1.	Social media mining: an introduction, Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu. Cambridge University Press, 2014.
2.	Networks, Crowds, and Markets: Reasoning About a Highly Connected World by David Easley and Jon Kleinberg, Cambridge University Press.
3.	Networks: An introduction by Mark Newman, Oxford university press, 2018.
4.	Graph Representation Learning by William L. Hamilton, 2020.

Software Testing				
Prerequisite: Software Engineering	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs
Unit 1	**Introduction**: Software Testing, Importance of testing, Roles and Responsibilities, Testing Principles, Attributes of Good Test, V-Model, Test Case Generation, SDLC Vs STLC.	8
Unit 2	**Types of Testing:** Unit Testing, Integration Testing, System Testing, Regression Testing, Acceptance Testing, Functional/Non Functional Testing, Static and Dynamic Testing **Categorization of testing methods**: Manual Testing, Automation Testing and Automated Testing Vs. Manual Testing, Testing Tools.	6
Unit 3	**Non Functional Testing:** Performance Testing, Load Testing, Security Testing, Scalability Testing, Compatibility Testing, Stress Testing, Installation Testing.	6
Unit 4	**Software Testing Methodologies:** Validation & Verification, White Box Testing, Black Box Testing, Grey Box Testing. White/Glass Box Testing: Statement Coverage Testing, Branch Coverage Testing, Path Coverage Testing, Conditional Coverage Testing, Loop Coverage Testing, Mutation testing, Data Flow Testing. Black Box Testing: Boundary Value Analysis, Equivalence Class Partition, State Based Testing, Cause Effective Graph, Decision Table.	6
Unit 5	**Software Testing Life Cycle:** Requirements Analysis, Test Planning, Objective, Scope of Testing, Schedule, Approach, Roles & Responsibilities, Assumptions, Risks & Mitigations, Entry & Exit Criteria, Test Automation, Deliverables.	8
Unit 6	**Test Cases Design**: Write Test cases, Review Test cases, Test Cases Template, Types of Test Cases, Difference between Test Scenarios and Test Cases, Test Oracle, Test Environment setup, Understand the SRS, Hardware and software requirements, Test Data.	8

References	
1.	A.P. Mathur, " Foundations of Software Testing", Pearson publications.
2.	Naresh Chauhan , "Software Testing Principles and Practices " Oxford University Press, New Delhi.
3.	Srinivasan Desikan and Gopalaswamy Ramesh, "Software Testing – Principles and Practices", Pearson Education.

<table>
<tr><th colspan="6">Software Testing Lab</th></tr>
<tr><td colspan="2">Prerequisite: Software Engineering</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 28</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs</td></tr>
<tr><td>Unit 1</td><td colspan="4">Study of Software Testing Automation Tools: Unit Testing (e.g. JUnit, CPPunit), Test Case Generation and Test Coverage.</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Functional Testing using Automated Tools (e.g. selenium), Coverage Analysis, Test Sequence Generation and Validation.</td><td>10</td></tr>
<tr><td>Unit 3</td><td colspan="4">Performance Testing using Automated Tools (e.g JMeter), Load Testing, Acceptance Testing, Mutation Testing, Defect Management.</td><td>10</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Book-1 : Foundations of Software Testing, by Aditya P. Mathur (Pearson)</td></tr>
<tr><td>2.</td><td colspan="5">Book-2: Software Testing: Principles and Practices, by Srinivasan Desikan and Gopalaswamy Ramesh (Pearson)</td></tr>
<tr><td>3.</td><td colspan="5">Book-3: Software Testing, by Yogesh Singh (Cambridge)</td></tr>
</table>

Topics in Computing				
Prerequisite: Operating Systems, Computer Networks, DBMS, Algorithms	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs
Unit 1	Cloud Computing: Virtualization and Containerization, Cloud Computing, Docker, Hypervisors	7
Unit 2	High Performance Computing: Big Data and Analytics, Data Science	7
Unit 3	Modern Networking: Internet of Things, Mobile Edge Computing, SDN and NFV	7
Unit 4	Distributed Ledger Technology: Blockchain, Bitcoin, Ethereum, Mining, Proof of Work, Proof of Stake, Wallets, Atomic Swap	7
Unit 5	Security Computing: Darknets, Deep and Dark Web, The Onion Routing	7
Unit 6	Quantum Computing:	7

References	
1.	William Stalling, “Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud”, Addison-Wesley Professional, ISBN: 9780134175478
2.	Kai Hwang, Jack Dongarra, Geoffrey C. Fox, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things” · 2013, Elsevier Science, ISBN: 9780128002049
3.	Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, “Bitcoin and Cryptocurrency Technologies” Princeton University Press ISBN: 9781400884155
4.	Documentation from torproject.org

Topics in Computing Lab					
Prerequisite:		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs
Assignment 1	Virtualization				2
Assignment 2	Virtual Machine Introspection				2
Assignment 3	Cloud Software Setup (OpenStack)				4
Assignment 4	Hadoop Ecosystem Setup				4
Assignment 5	Software Defined Network Setup				4
Assignment 6	Internet of Things: Environmental Setup				4
Assignment 7	Blockchain Implementation				4
Assignment 8	Quantum Computing Handson				4
References:					
1.	https://help.ubuntu.com/community/Xen https://help.ubuntu.com/community/KVM/Installation http://www.ubuntu.com/download/desktop http://www.centos.org/download/				
2.	https://code.google.com/archive/p/insight-vmi/				
3.	http://docs.openstack.org/icehouse/install-guide/install/yum/content/				
4.	https://hadoopecosystemtable.github.io/ https://hadoop.apache.org/				
5.	http://networkstatic.net/how-to-build-an-sdn-lab-without-needing-openflow-hardware/ http://networkstatic.net/openflow-openvswitch-and-kvm-sdn-lab-installation-app/				
6.	http://www.nimbits.com/howto_install.jsp https://thingspeak.com/ https://www.paraimpu.com/ https://www.contiki-ng.org/				
7.	https://www.javatpoint.com/blockchain-java https://www.javatpoint.com/building-a-blockchain-using-python				
8.	https://qiskit.org/documentation/index.html				

Topics in Operating Systems				
Total Hours	L	T	P	C
42	3	0	0	3
Prerequisite: Fundamentals of Operating Systems and Computer Organization and Architecture				

	Course Content	Hrs
Unit 1	Review of Operating System, Virtual Machines: Overview, Types of VMs and Their Implementations, Virtualization and Operating-System Components; Hypervisors	8
Unit 2	Design issues of Distributed OS, Distributed v/s network operating system. process management, inter-process communication, scheduling, deadlocks Communication: Client Server, RPC; Distributed Concurrency, Transactions. Design and implementation of distributed file systems, DFS Naming and Transparency, Remote File Access distributed shared memory	10
Unit 3	Embedded Operating System: ARM architecture, interrupts and exceptions; Process management, process synchronization, Threads, Memory management - paging, I/O buffer management, File systems	10
Unit 4	Real time Operating System: Scheduling, design principles, Uniprocessor real-time OS, Multi-processor real-time OS	8
Unit 5	Mobile Operating System: Android as a case study	6

References	
1.	"Operating System Internals and Design Principles" by William Stallings (9th Edition, 2021)
2.	"Modern Operating Systems" by Andrew S. Tanenbaum and Herbert Bos (4th Edition, 2014)
3.	"Operating System Concepts" by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin (10th Edition, 2018)
4.	"The Design and Implementation of the FreeBSD Operating System" by Marshall Kirk McKusick, George V. Neville-Neil, and Robert N.M. Watson (2nd Edition, 2015)
5.	Published articles from reputed Journals and Conferences.

Topics in Operating Systems Lab

Total Hours	L	T	P	C
28	0	0	2	2

Prerequisite: Fundamentals of Operating Systems and Computer Organization and Architecture

Course Content		Hrs
Unit 1	- Install different operating systems (e.g., Windows, Linux distributions) on virtual machines to understand the installation process and gain familiarity with various OS environments. - Set up virtual networks with multiple VMs to learn about IP addressing, subnetting, DHCP, and DNS configurations. - Create vulnerable VMs and practice ethical hacking or penetration testing techniques to identify and fix security vulnerabilities. -Experiment with Docker to create, deploy, and manage containers for various applications and services. -Create VM clusters to simulate high availability and load-balancing scenarios for critical services. -Practice deploying virtual machines on cloud platforms like AWS, Azure, or Google Cloud to understand cloud infrastructure. -Explore container orchestration tools like Kubernetes and deploy multi-container applications on virtual machines. -Experiment with nested virtualization, where you run virtual machines inside virtual machines.	12
Unit 2	-Create a cluster of computers using a distributed OS like Linux-based ones (e.g., CentOS, Ubuntu) or specialized distributions like Beowulf. Learn how to configure the network and set up intercommunication between cluster nodes. -Implement load-balancing algorithms to distribute computational tasks evenly across nodes in the cluster, improving system performance. -Set up and experiment with distributed file systems like Hadoop Distributed File System (HDFS) or GlusterFS to understand data storage and retrieval across the cluster. - Run parallel processing tasks across the cluster using tools like Apache Spark or MPI (Message Passing Interface) to solve computationally intensive problems. - Research and test various distributed algorithms, such as leader election, consensus protocols (e.g., Paxos, Raft), and distributed data structures (e.g., distributed hash tables).	10
Unit 3	-Write a simple C program on your host computer that prints "Hello, Embedded World!" (or any other basic functionality) to the console. -Use the ARM cross-compiler toolchain to cross-compile the C program for the ARM architecture. This will generate an ARM executable binary. -Create multiple tasks with different priorities and execution times using RTOS APIs.	6

References

1.	"Operating System Internals and Design Principles" by William Stallings (9th Edition, 2021)
2.	"Modern Operating Systems" by Andrew S. Tanenbaum and Herbert Bos (4th Edition, 2014)
3.	"Operating System Concepts" by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin (10th Edition, 2018)

4.	"The Design and Implementation of the FreeBSD Operating System" by Marshall Kirk McKusick, George V. Neville-Neil, and Robert N.M. Watson (2nd Edition, 2015)
5.	Published articles from reputed Journals and Conferences.

Pool 3: Honors Electives: 3-0-0 (Credits 3)

Honors Electives		***Credit***	***L***	***T***	***P***
1.	Advances in Compiler Design	3	3	0	0
2.	Android Programming	3	3	0	0
3.	Big Data Analytics	3	3	0	0
4.	Cloud Security	3	3	0	0
5.	Cyber Security	3	3	0	0
6.	Data Compression	3	3	0	0
7.	Data Visualization	3	3	0	0
8.	Digital Forensic	3	3	0	0
9.	Distributed Systems	3	3	0	0
10.	E-commerce	3	3	0	0
11.	Embedded System Security	3	3	0	0
12.	Hardware Software Codesign	3	3	0	0
13.	Image Analysis	3	3	0	0
14.	Intrusion Detection	3	3	0	0
15.	Neural Network	3	3	0	0
16.	Network on Chip	3	3	0	0
17.	Network Performance Modeling	3	3	0	0
18.	Parallel Processing & Algorithms	3	3	0	0
19.	Parallelizing Compiler	3	3	0	0
20.	Pattern Recognition	3	3	0	0
21.	Public Key Infrastructure and Trust Management	3	3	0	0
22.	Quantum Computing	3	3	0	0
23.	Quantum Cryptography	3	3	0	0
24.	Real Time Systems	3	3	0	0
25.	Robotics and Control	3	3	0	0
26.	Security Analysis of Protocols	3	3	0	0
27.	Selected Topics in Cryptography	3	3	0	0
28.	Social Media Mining	3	3	0	0
29.	Software Project Management	3	3	0	0
30.	System on Chip	3	3	0	0
31.	Wireless Sensor Networks	3	3	0	0

Advances in Compiler Design				
Prerequisite: Basic course in Compiler Design	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Modern Compiler Design – Structure of Compilers for Modern Programming Languages, Cross Compiler, Just-In-Time (JIT) and Adaptive Compilation	8
Unit 2	Runtime System Architectures. Parser Development - LR Parsers and LR Grammars – Design and Implementation.	10
Unit 3	Parser and Ambiguity, Conflict Resolution, Lex and Yacc Tools. Optimizing Compiler - Control-flow Analysis, Control-flow Graphs, Basic Blocks.	10
Unit 4	Data-flow Analysis Methods, Dependence Analysis, Global Optimizations, Loop Optimizations.	8
Unit 5	Peephole Optimization and Optimal Code Generation, Data Dependence Analysis in Loops, Loop Scheduling.	6

References:	
1.	Aho, Lam, Sethi and Ullman: Compilers – Principles, Techniques and Tools, Pearson Education 2. 3. 4.
2.	Steven Muchnick : Advanced Compiler Design & Implementation, Morgan Kaufmann
3.	Holub: Compiler Design in C, Prentice Hall India.
4.	Keith Cooper and Linda Torczon : Engineering a Compiler, Morgan Kaufmann.

<table>
<tr><td colspan="6"><b>Android Programming</b></td></tr>
<tr><td colspan="2">Prerequisite: None</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">Basics: Review of Java Programming, Setting up and configuring Android Studio setup, Android Emulator Hello Android example, AndroidManifest.xml, R.java file, Activity, Fragment,</td><td>10</td></tr>
<tr><td>Unit 2</td><td colspan="4">Layout Manager - Relative Layout, Linear Layout, Table Layout, Grid Layout. Activity, Intent & Fragment: Activity Lifecycle, Activity Example, Intent – implicit and explicit, Intent filters, Fragment Lifecycle, Fragment Example</td><td>8</td></tr>
<tr><td>Unit 3</td><td colspan="4">UI Widgets – buttons (toggle, switch, image), check box; Android Menu: Option Menu, Context Menu, Popup Menu; View.</td><td>8</td></tr>
<tr><td>Unit 4</td><td colspan="4">Android Service: lifecycle, example, Data Storage, Shared Preference, SQLite, Content Provider, Android Notification Adding functionality: Multimedia API, Speech API, telephony API.</td><td>10</td></tr>
<tr><td>Unit 5</td><td colspan="4">Location API Sensors: Sensor API, Working with WiFi, Working with Camera, Motion Sensor, Position Sensor; Android Graphics App development project.</td><td>6</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Official Android Website</td></tr>
</table>

Big Data Analytics					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Overview of Database Management Systems, Introduction to Big Data, Introduction to distributed file system, Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics.				8
Unit 2	Apache Hadoop & Hadoop Eco-System, Moving Data in and out of Hadoop, Understanding inputs and outputs of MapReduce, Data Serialization. Hadoop Architecture.				10
Unit 3	Hadoop Storage: HDFS, Common Hadoop Shell commands, Anatomy of File Write and Read, Name-Node, Secondary Name-Node, and Data-Node, Hadoop MapReduce paradigm, Map and Reduce tasks, Job.				10
Unit 4	Task trackers - Cluster Setup, SSH & Hadoop Configuration– HDFS Administering, Monitoring & Maintenance. Pig, Pig Latin Language, Hive Introduction, Hive queries. Spark Introduction. Cassandra CQL				8
Unit 5	Query language and CQL data model: Key space, Table definition, Column, and Data Types. Mongo DB Cluster analysis, K-means algorithm, Naïve Bayes, Parallel k-means using Hadoop, parallel particle swarm algorithm using MapReduce, case studies on big data mining. Parallel swarm Intelligence.				6
References:					
1.	Dan Sulliva ,NoSQL for Mere Mortals 1st Edition., Pearson Publishers, 2014				
2.	Pramod J. Sadalage, Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence 1st Edition, Pearson Publishers,ISBN-13: 978-0321826626, 2017.				
3.	John D. Kelleher, Brian Mac Namee, Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies (The MIT Press)				
4.	John D. Kelleher, Brendan Tierney, Data Science (MIT Press Essential Knowledge series).				

Cloud Security					
Prerequisite: Computer Networks, Operating System		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Introduction of Cloud Computing: Taxonomy and related technologies, Essential Characteristics, Service and Deployment Models				8
Unit 2	Virtualization: Types of Virtualization and Hypervisors, Virtualization at Storage, Compute and Network, Hypervisors (Types and Case studies), Virtual Machine Provisioning, Virtual Machine Migration.				10
Unit 3	Architectures:Standards, Orchestration, Provisioning, Portability, Interoperability, Federated Cloud, Security: CIA Triad, Vulnerabilities in Cloud, Threats to Infrastructure, Data and Access Control				10
Unit 4	Identity Management; Multi Tenancy Issues; Attack taxonomy; Intrusion Detection, VM Specific attacks, VM Introspection, Management; Trusted Cloud Initiative of Cloud Security Alliance (CSA).				8
Unit 5	Forensics: NIST Forensics Reference Architecture, Forensic Science Challenges, Architectural Issues, Evidence Collection and Analysis, Anti-Forensics, Incident Response, Standards and Framework				6
References:					
1.	K. Hwang, G. C. Fox, and J. Dongarra, Distributed and Cloud Computing, 1st ed.: Morgan Kaufmann, 2011				
2.	R. Buyya, J. Broberg, and A. M. Goscinski, Cloud Computing: Principles and Paradigms: Wiley-Blackwell, 2011				
3.	S. Dinkar and G. Manjunath, Moving to the Cloud: Developing Apps in the New World of Cloud Computing Syngress Media, U.S., 2012.				
4.	W. Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, 1st ed.: Addison-Wesley Professional, 2015.				
5.	T. Erl, Z. Mahmood, and R. Puttini, Cloud Computing: Concepts, Technology & Architecture: Prentice Hall/PearsonPTR, 2014				
6.	R. L. Krutz and R. D. Vines, Cloud Security - A Comprehensive Guide to Secure Cloud Computing, Wiley Publishing, 2010				
7.	T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy - An Enterprise Perspective on Risks and Compliance, O Reilley Publishers, 2009.				
8.	V. (J. R.) Winkler, G. Speake, P. Foxhoven, Securing the Cloud: Cloud Computer Security Techniques and Tactics, Syngress, 2011.				

Cyber Security				
Prerequisite: None	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Overview of Cyber Security, Internet Governance – Challenges and Constraints, Cyber Threats, Need for a Comprehensive Cyber Security Policy. Cyber Security Safeguards (Overview): Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Security policy, Threat Management.	8
Unit 2	Network Security & Web Security: Security Issues in TCP/IP, which includes TCP, DNS, Routing (basic problems of security in TCP/IP, IPsec, BGP Security, DNS Cache poisoning, etc), Network Defense tools such as Firewalls, Filtering, DNSSec, NSec3, Distributed Firewalls.	10
Unit 3	Web Application Security: Cross-Site Scripting Attacks, Cross-Site Request Forgery, SQL Injection Attacks Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques, AntiMalware software, Network based Intrusion detection Systems, Network based Intrusion Prevention Systems, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis, System Integrity Validation	10
Unit 4	Cyber Forensics: Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation, Conducting diskbased analysis, Investigating Information-hiding, Scrutinizing E-mail, Validating Email header information, Tracing Internet access, Tracing memory in real-time.	8
Unit 5	Security in Mobile Platforms: Android vs. ioS security model, threat models, information tracking, rootkits, Threats in mobile applications, analyzer for mobile apps to discover security vulnerabilities, Viruses, Spywares, and keyloggers and malware detection. Cyberspace and the Law	6

References:	
1.	Latest research papers, journals and articles
2.	Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Nina Godbole and SunitBelapure.
3.	Cybersecurity Essentials By Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short · 2018
4.	Cybersecurity: Attack and Defense Strategies: Infrastructure Security with Red Team and Blue Team TacticsBook by ErdalOzkaya and Yuri Diogenes

<table>
<tr><th colspan="6">Data Compression</th></tr>
<tr><td colspan="2">Prerequisite: Object Oriented Analysis and Design</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">Introduction: Compression techniques, lossless compression, lossy compression, measures of performance, modeling and coding.</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Mathematical preliminaries - Overview, introduction to information theory, models, physical models, probability models, Markov models.</td><td>10</td></tr>
<tr><td>Unit 3</td><td colspan="4">Basic Coding Schemes: Statistical Methods - Shannon-Fano Algorithm, Huffman Algorithm, Adaptive Huffman Coding. Arithmetic Coding (Encoding, Decoding, Adaptive Coding). Dictionary Methods - LZ77, LZ78, LZW Algorithms. Case study of lossless compression standards.</td><td>10</td></tr>
<tr><td>Unit 4</td><td colspan="4">Lossless Compression standards: zip, gzip, bzip, unix compress, GIF, JBIG. Image and Video Compression: Discrete Cosine Transform, JPEG. Wavelet Methods - Discrete Wavelet Transform, JPEG 2000</td><td>8</td></tr>
<tr><td>Unit 5</td><td colspan="4">Motion Compensation, Temporal and Spatial Prediction. MPEG and H.264. Audio Compression: Digital Audio, WAVE, FLAC, MPEG-1/2 Audio Layers.</td><td>6</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Khalid Sayood. 2012. Introduction to Data Compression (4th ed.). Elsevier</td></tr>
<tr><td>2.</td><td colspan="5">David Salomon, Giovanni Motta. 2010. Handbook of Data Compression. Springer, London</td></tr>
</table>

Data Visualization					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Modern Visualisation tools and techniques, Create multiple versions of digital visualizations using various software packages.				8
Unit 2	Identify appropriate data visualization techniques given particular requirements imposed by the data.				10
Unit 3	Apply appropriate design principles in the creation of presentations and visualizations; Analyse, critique, and revise data visualizations				8
Unit 4	Information overload and issues in decision making Design of visual encoding schemes to improve comprehension of data and their use in decision making				6
Unit 5	Use of Tableau - Data visualization tool for data analysts, scientists, statisticians, etc. to visualize the data and get a clear opinion based on the data analysis, Comparing classifiers- ROC curves, McNemar’s test, other statistical tests.				10
References:					
1.	A first course Sosulski, K. (2018). Data Visualization Made Simple: Insights into Becoming Visual. New York: Routledge				
2.	The Visual Display of Quantitative Information (2nd Edition). E. Tufte. Graphics Press, 2001.				

<table>
<tr><td colspan="6"><b>Digital Forensics</b></td></tr>
<tr><td colspan="2">Prerequisite: Operating Systems, Computer Networks & Security</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">File System Forensics: Duplicating hard disks for "dead analysis”, reading hidden data on a disk's Host Protected Area (HPA), Direct versus BIOS access, dead versus live acquisition</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Disk partitions - DOS, Apple, and GPT partitions, BSD disk labels, Sun Volume; multiple disk volumes - RAID and disk spanning.</td><td>10</td></tr>
<tr><td>Unit 3</td><td colspan="4">Analyzing FAT, NTFS, Ext2, Ext3, UFS1, and UFS2 file systems, Finding evidence: File metadata, recovery of deleted files, Using The Sleuth Kit (TSK), Autopsy Forensic Browser, and related open source tools</td><td>10</td></tr>
<tr><td>Unit 4</td><td colspan="4">Web Forensics: network-based evidence in Windows and Unix environments, Reconstructing Web browsing, email activity, Tracing domain name ownership and the source of e-mails</td><td>8</td></tr>
<tr><td>Unit 5</td><td colspan="4">System Forensics: Windows Registry changes, Duplicating and analyzing the contents of PDAs and flash memory devices Electronic document, computer image verification and authentication.</td><td>6</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Brian Carrier. File System Forensic Analysis, Addison Wesley</td></tr>
<tr><td>2.</td><td colspan="5">Chris Prosise, Kevin Mandia. Incident Response and Computer Forensics, McGraw Hill. Course Technology.</td></tr>
<tr><td>3.</td><td colspan="5">Linda Volonino, Reynaldo Anzaldua, and Jana Godwin. Computer Forensics: Principles and Practices, Prentice Hall.</td></tr>
<tr><td>4.</td><td colspan="5">Keith J. Jones, Richard Bejtlich, and Curtis W. Rose. Real Digital Forensics: Computer Security and Incident Response, Addison Wesley.</td></tr>
<tr><td>5.</td><td colspan="5">Vacca, John R., Computer Forensics Computer Crime Scene Investigation, Charles River Media.</td></tr>
<tr><td>6.</td><td colspan="5">Nelson, Phillips, Enfinger, Steuart. Guide to computer Forensics and Investigation</td></tr>
</table>

Distributed Systems					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Introduction to Distributed Systems, OS and Advanced OS, various distributed systems, Trends in Distributed System and challenges, Networking: network protocols, point-to-point communication. Introduction – Clocks, events and process states – Synchronizing physical clocksLogical time and logical clocks – Global states,Limitations, Lamport’s logical clock, vector clock, causal ordering, global state, Cuts. Distributed Mutual Exclusion: Lamport, Recart-agrawala, and Maekawa’s algorithms; Suzuki-kasami broadcast algorithm, and Raymond’s tree based algorithm , Elections algorithms				8
Unit 2	Transactions and Concurrency Control– Transactions -Nested transactions – Locks – Optimistic concurrency control – Timestamp ordering – Atomic Commit Distributed transactions: two phase commit, three-phase commit, ACID/BASE models Techniques of Inter process Communication: the API for internet protocols – External data representation and Multicast communication, Sun RPC: programming and implementation, Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation And Objects: Remote Invocation – Introduction – Request-reply protocols – Remote procedure call – Remote method invocation. Case study: Java RMI – Group communication – Publish-subscribe systems – Message queues – Shared memory approaches – Distributed objects.				10
Unit 3	Case study: Enterprise Java Beans -from objects to components. Distributed Deadlock Detection: Resource Vs. Communication deadlock, Replication, Strategies to handle deadlock, Ho-Ramamoorthy, Path-Pushing, Edge-Chasing, Diffusion Computation-based algorithms. Agreement Protocols: System model, Classification of agreement problems, Solutions to Byzantine Agreement (BA) problems.Distributed Scheduling: Issues in Load Distribution, Components of a load distribution algorithm, Load Distribution Algorithms, V-system, Sprite, and Condor.				10
Unit 4	Network file systems: design, NFS, AFS (scale), DFS & CIFS (cache control), CODA (redundancy) Google File System (GFS), Hadoop Distributed File System (HDFS)Distributed Shared Memory: Algorithms for implementing DSMs, Memory Coherence, and Coherence Protocols, IVY Process Management: Process Migration: Features, Mechanism – Threads: Models, Issues, Implementation.				8
Unit 5	Resource Management: IntroductionFeatures of Scheduling Algorithms –Task Assignment Approach – Load Balancing Approach – Load Sharing Approach Recovery: Classification of failures, Synchronous and Asynchronous Check pointing and Recovery. Fault Tolerance: Commit Protocols, Voting Protocols, Failure Resilient Processes. Protection and Security: Access Matrix Model, Implementation of access matrix, Unix, and Amoeba. Case study-Distributed systems.				6
References:					
1.	Andrew S. Tanenbaum, Maarten Van Steen, “Distributed Systems Principles and Paradigm,” 2nd Edition, Pearson				
2.	George Coulouris, Jean Dollinmore, Tim Kindberg, Gordon Blair “Distributed SystemsConcepts and Design,” 5th Edition, Pearson				

3.	M. Singhal& N. Shivaratri, “Advanced Concepts in Operating Systems: Distributed, Database and Multiprocessor Operating Systems”, Tata McGraw Hill, 2015
4.	John Bloomer, “Power Programming with RPC,” O’Reilly & Associates, Inc
5.	Advanced Programming in the Unix Environment by W. Richard Stevens, Addison-Wesley.
6.	Liu M.L., “Distributed Computing, Principles and Applications”, Pearson Education
7.	Distributed Systems - An Algorithmic approach by Sukumar Ghosh.

E-Commerce

Prerequisite: knowledge of Digital Market, Basics of Computer Network and security	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Introduction: Definition of Electronic Commerce, technology and prospects, incentives for engaging in electronic commerce, needs of E-Commerce, E-Commerce Infrastructure, advantages and disadvantages, Impact of E-commerce on business, E-Commerce Models.	8
Unit 2	Network Infrastructure for E- Commerce. Internet and Intranet based E-commerce: Issues, problems and prospects, Network Infrastructure, Network Access Equipments, Broadband telecommunication. Mobile Commerce: Introduction, Wireless Application Protocol, WAP technology, Mobile Information device	10
Unit 3	Web Security: Security Issues on web, Importance of Firewall, components of Firewall, Transaction security, Emerging client server, Security Threats, Network Security, Factors to consider in Firewall design, Limitation of Firewalls. Encryption: Encryption techniques, Symmetric Encryption: Keys and data encryption standard, Triple encryption, Secret key encryption.	10
Unit 4	Asymmetric encryption: public and private pair key encryption, Digital Signatures, Virtual Private Network. Customer Service Expectations of the E-commerce Experience.	6
Unit 5	Electronic Payments: Overview, The SET protocol, Payment: Smart card, credit card, magnetic strip card, E-Checks, Credit/Debit card based EPS, online Banking. EDI Application in business, E- Commerce Law, Forms of Agreement, Govt. policies and Agenda	8

References:

1.	Turban, “Electronic Commerce 2004: A Managerial Perspective”, Pearson Education
2.	Pete Lohsin , John Vacca “Electronic Commerce”, New Age International
3.	Bajaj and Nag, “E-Commerce the cutting edge of Business”, TMH 6
4.	Laudon, “E-Commerce: Business, Technology, Society”, Pearson Education

Embedded System Security					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Security Flaws and Attacks in Embedded systems: Code injection, Invasive and Non invasive physical and logical attacks				8
Unit 2	Defenses Against Code Injection Attacks: Methods using Address Obfuscation and Software Encryption, Anomaly Detection.				10
Unit 3	Safe Languages, Code Analyzers Compiler, Library, and Operating System Support for embedded systems				10
Unit 4	Security, Control Flow Checking, IP Protection: Encryption of IP Cores, additive and Constraint-Based watermarking.				8
Unit 5	Implementation of DES 3DES, AES, RC4, MD5, RSA algorithms				6
References:					
1.	Security in Embedded Hardware				

Hardware Software Codesign					
Prerequisite: Logic System Design/ Digital Logic Design		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Codesign overview, device Modeling and methodologies of system design				8
Unit 2	Hardware software partitioning and scheduling, Co simulation.				10
Unit 3	Synthesis and verifications, Architecture, Interface and reconfiguration.				10
Unit 4	System on chip, Application specific processors (DSP)				8
Unit 5	Codesign tools and case studies				6
References:					
1.	A Practical Introduction to Hardware/Software Codesign, Patrick Schaumont,Springer, 2009, ISBN 978-1-4419-5999-7				
2.	Specification and Design of Embedded Systems Daniel D. Gajski, Frank Vahid, S. Narayan, & J. Gong, Prentice Hall, 1994				
3.	Hardware / Software Co-Design: Principles and Practice,JStaunstrup and Wayne Wolf, Prentice Hall, 1994				

Image Analysis

Prerequisite: None	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs.
Unit 1	Image Preliminaries & Image Processing: Overview, Computer imaging systems, Human visual system, image model, etc. Geometric transformations: Translation, rotation, scaling and shearing.	8
Unit 2	Frequency transformation: Discrete Fourier transform (DFT), fast Fourier transform (FFT), shorttime Fourier transform (STFT), Multi-resolution Expansions: Wavelet Transforms in 1-D and 2- D. The Fast Wavelet Transform Wavelet Packets Transform	10
Unit 3	Feature Extraction and Dimension Reduction Color, Texture, Shape and structure Features in spatial and frequency domains, Corner Detection, Hough Transform, Principal Component Analysis, Linear Discriminant Analysis, Feature Reduction in Input and Feature Spaces. Image Segmentation. Gray-level thresholding, Supervised vs. Unsupervised thresholding, Binarization using Otsu's method, Locally adaptive thresholding.	10
Unit 4	Color-based segmentation, Region oriented segmentation, Use of motion in segmentation, Spatial techniques, Frequency domain techniques. Features Based Image Matching:Scale Space Image Processing.	8
Unit 5	Different Feature descriptors: Key Point Detection, SIFT descriptor SURF descriptor Bag of Visual Words approach, Geometric consistency check, Vocabulary tree Panoramic Imaging, Template Matching, Mono Panorama, Stereo Panorama.	6

References:

1.	J G Proakis and D G Manolakis, "Digital Signal Processing," Pearson, Fourth edition
2.	Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Prentice Hall, 3rd Edition, 2007.
3.	Bishop, Pattern Recognition and Machine Learning
4.	Duda, Pattern Classification.

Intrusion Detection					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Introduction- Intrusion Detection System (IDS), Intrusion Prevention System (IPS).				8
Unit 2	Unauthorized access – buffer overflow, packet fragmentation, out-of-spec packets Review of Network protocol – TCP/IP, Intrusion detection through tcpdump				10
Unit 3	IDS and IPS – Architecture and internals. Malicious and non-malicious traffic, IP headers, TCP, UDP and ICMP protocols and header formats.				10
Unit 4	Header information to detect intrusion, logs and their analysis.				6
Unit 5	IDS through reaction and response Intrusion analysis – data correlation, tools, SNORT- A case study				8
References:					
1.	Matt Fearnow, Stephen Northcutt, Karen Frederick, and Mark Cooper. Intrusion Signatures and Analysis, SAMS.				
2.	Carl Endorf, Gene Schultz, Jim Mellander, Intrusion Detection and Prevention, McGraw Hill				
3.	Paul E. Proctor. The Practical Intrusion Detection Handbook, Prentice Hall.				
4.	Stephen Northcutt and Judy Novak. Network Intrusion Detection, SAMS.				

Neural Networks

Prerequisite: Basic understanding of probability and statistics, linear algebra and calculus. A basic knowledge of programming (preferably Python) is essential.	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs.
Unit 1	Introduction to Neural Architecture, McCulloch-Pitts networks, Learning Rules, Perceptrons.	8
Unit 2	Regression and least mean square algorithm, Multilayer perceptrons.	8
Unit 3	Back propagation: generalized delta rule, limitations, modifications – momentum, variable learning rate, conjugate gradient, Radial-basis function networks.	10
Unit 4	Support vector Machines, Unsupervised learning and self-organization, Boltzmann machines and deep networks, Convolutional networks.	10
Unit 5	Recurrent networks, Associative Memories, Adaptive Resonance Theory, Applications of Neural Networks.	6

References:	
1.	Simon Haykin: Neural Networks: A Comprehensive Foundation, Pearson

Network on Chip

Prerequisite: Computer Architecture, Logic System Design	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs.
Unit 1	The Concept of route packet not wires for On-Chip Interconnection Networks, Topology and design architecture of Network-on-Chip, Area and power trade off NoC protocols.	8
Unit 2	Routing and Flow Control mechanism, Verification of Communications in Networks-on-Chips. Application Mapping on Network-on-Chip.	10
Unit 3	Resource Allocation for QoS On-Chip Communication, routing techniques in different 2D/ 3D NoC topology, performance evaluation in terms of throughput, latency, gitter.	10
Unit 4	Signal Integrity and Reliability of Network-on-Chip, Testing of Network-on- Chip Architectures, Test and Fault Tolerance for NoC Infrastructures.	8
Unit 5	Reconfigurable Network-on-Chip Design, Security in NoCs. Energy and Power estimation techniques Network-on-Chips.	6

References:

1.	Giovanni De Micheli, Luca Benini, DavideBertozzi, Networks on Chips:Technology and Tools, Morgan Kaufmann, 2006.
2.	Fayez Gebali, HaythamElmiligi, Mohamed Watheq El-Kharashi, Networkson- Chips: Theory and Practice, CRC Press, 2017.
3.	SudeepPasricha, NikilDutt, On-Chip Communication Architectures: System on Chip Interconnect, Morgan Kaufmann, 2010.

Network Performance Modelling				
Prerequisite: None	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs.
Unit 1	Introduction to Network Modeling: Network modeling, Computer Network as a discrete event system, Modeling and measurement tools, Network performance metrics – first order and second order metrics, Network capacity, Difference between throughput and capacity	8
Unit 2	Network Calculus: Models for data flows, arrival curves and service curves, Greedy shapers, Basic min-plus and maxplus calculus, min-plus and max-plus systems, Optimal smoothing, FIFO systems and aggregate scheduling, Time varying shapers, Systems with losses	6
Unit 3	Case studies – (1) Analyzing spanning tree based data forwarding using network calculus, (2) Bound on loss rate Stochastic Scheduling and Resource Allocation: Stochastic scheduling, dynamic resource allocation, Dynamic programming models for stochastic scheduling, Queuing networks – open loop and closed loop networks, Jackson networks, Network fairness – proportional and max-min fairness, Markov process and its application for analyzing network resource allocation and fairness, available bandwidth estimation	11
Unit 4	Case studies – (1) TCP/IP flow and congestion control, (2) Modeling dynamic routing and scheduling as a queuing network problem, (3) Analysis of IEEE 802.11 channel access using two dimensional Markov process. Network Games: Introduction to game theory, Zero sum games, Nash equilibrium, Pareto optimality, Cooperative and Non-cooperative games, General network games – resource sharing games, routing games, congestion games, Mechanism design. Case studies – (1) Selfish routing in networks and price of anarchy, (2) Oblivious routing, (3) Network resource allocation games.	11
Unit 5	Protocol Analysis: Modeling discrete event system using petri-nets, basics of petri nets, stochastic petri nets, queuing petri nets, properties of petri nets, structural analysis of petri nets, Petri net modeling tools – simQPN, Case studies – (1) Wireless channel model using stochastic petri net, (2) Data center network throughput analysis using queuing Petri Nets	6

References:	
1.	"Routing, Flow, and Capacity Design in Communication and Computer Networks", MichałPióro, DeepankarMedhi, ISBN: 0125571895, Publisher: Morgan Kaufmann
2.	The Network Calculus Book by Jean-Yves Le Boudec and Patrick Thiran is available for free download:http://ica1www.epfl.ch/PS_files/NetCal.html
3.	Anurag Kumar, D. Manjunath and Joy Kuri, "Communication Networking: An Analytical Approach" Morgan Kaufman Publishers
4.	Dimitri P. Bertsekas and Robert G. Gallager, "Data Networks": Materials are available at http://web.mit.edu/dimitrib/www/datanets.html
5.	"Network Optimization: Continuous and Discrete Models", D. Bertsekas
6.	Research Publications - will be discussed and distributed time to time

Parallel Processing & Algorithms				
Prerequisite: None	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Introduction to parallel computing. Parallel processing terminology, Pipelining Vs Data parallelism, Control parallelism, Scalability, Control parallel approach, Data parallel approach, Data parallel approach with I/O.	8
Unit 2	The PRAM Shared-Memory Model, Distributed-Memory or Graph Models, Circuit Model and Physical Realizations PRAM and Basic Algorithms, PRAM Submodels and Assumptions, Data Broadcasting, Semigroup or Fan-In Computation, Parallel reduction, Prefix sums, List ranking, Preorder tree traversal, Merging two sorted lists, Graph coloring, Reducing the number of processors, Problems defying fast solutions on PRAMS.	10
Unit 3	Thread and process level parallel architectures: MIMD, multi-threaded architectures. Distributed and shared memory MIMD architectures. Dynamic interconnection networks.Mapping and scheduling: Mapping data to processors on processor arrays and multicomputers, Dynamic Load Balancing on multicomputers, Static scheduling on UMA multiprocessors, Deadlock. Parallel programming and parallel algorithms: Programming models, parallel programming on multiprocessors and multicomputers.	10
Unit 4	Parallel algorithm structure, analyzing parallel algorithm. Elementary parallel algorithms, Matrix algorithms, sorting, Graph algorithms. Parallel Algorithm Complexity, Asymptotic Complexity, Algorithm Optimality and Efficiency, Complexity Classes, Parallelizable Tasks and the NC Class, Parallel Programming Paradigms, Solving Recurrences	8
Unit 5	Sorting and Selection Network: Design of Sorting Networks, Batcher Sorting Networks, Mesh-Base Architectures: Sorting on a 2D Mesh or Torus, Routing on a 2D Mesh or Torus, Numerical 2D Mesh Algorithms, Low-Diameter Architectures: Hypercubes and Their Algorithms, Sorting and Routing on Hypercubes	6

References:	
1.	J. Jaja, An Introduction to Parallel Algorithms, Addison Wesley, 1992
2.	F. T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Morgan Kaufmann Publishers, San Mateo,California,1992
3.	BehroozParhami, Introduction to Parallel Processing, Algorithms and Architecture, kluwer academic publishers,2002ed

<table>
<tr><td colspan="6"><b>Parallelizing Compiler</b></td></tr>
<tr><td colspan="2">Prerequisite: Basic course in Compiler Design</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">Introduction – Compilation for parallel machines and automatic detection of parallelism, structure of a parallelizing compiler.</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Dependence Theory and Practice - Types of dependences, data and control dependencies, dependence analysis.</td><td>10</td></tr>
<tr><td>Unit 3</td><td colspan="4">Direction vectors, loop carried and loop independent dependences, tests for data dependence and their applicability, construction of data dependence and control dependence graphs.</td><td>10</td></tr>
<tr><td>Unit 4</td><td colspan="4">Parallel Code Generation - Automatic extraction of parallelism, representation of iteration spaces of nested loops, loop based transformations such as loop distribution, loop coalescing, loop interchange and cycle shrinking transformation.</td><td>8</td></tr>
<tr><td>Unit 5</td><td colspan="4">Interprocedural Analysis and Optimization - aliasing information, summary data flow analysis, interprocedural constant propagation, interprocedural data dependence analysis and parallelization of call statements.</td><td>6</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Randy Allen, Ken Kennedy: Optimizing compilers for modern architectures. Morgan Kaufmann</td></tr>
<tr><td>2.</td><td colspan="5">Steven Muchnick : Advanced Compiler Design & Implementation, Morgan Kaufmann.</td></tr>
</table>

Pattern Recognition

Prerequisite: An undergraduate level understanding of probability, statistics and linear algebra is assumed. A basic knowledge of Python is essential.	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	The classification process: features, training and learning, approaches to classification Non metric methods: Information, Entropy and Impurity, decision tree classifier- ID3, C4.5. Discriminant functions: linear discriminant functions, piece-wise linear discriminant functions, generalized discriminant functions.	8
Unit 2	Statistical pattern recognition: measured data and measurement errors, probability theory, conditional probability and Bayes rule, Naive Bayes classifier, Continuous random variables, The multivariate Gaussian, Covariance matrix and MahalanobisdistanceParametric learning: Bayesian decision theory, discriminant functions and decision boundaries, MAP (Maximum A Posteriori Estimator)	10
Unit 3	Non Parametric learning: Histogram estimator and Parzen windows, k-NN classification, Artificial Neural Networks, Kernel Machines, SVM. Feature extraction and selection: reducing dimensionality, feature selection- Inter/Intra class distance.	10
Unit 4	Feature extraction: Principal component analysis, Linear discriminant analysis.Unsupervised learning: Clustering, K- Means clustering, Fuzzy c-Means clustering, (Agglomerative) Hierarchical clustering	8
Unit 5	Estimating and Comparing Classifiers: No free lunch, Bias and variance trade-off, cross-validation and resampling methods, Measuring classifier performance, Comparing classifiers- ROC curves, McNemar’s test, other statistical tests	6

References:	
1.	Pattern Classification, Duda Hart, Wiley
2.	Pattern Recognition and Classification, Geoff Dougherty, Springer
3.	Statistical Pattern Recognition, Andrew R Webb, Wiley
4.	Pattern Recognition and Machine Learning, Christopher Bishop, Springer
5.	Pattern Recognition and Image Analysis, Earl Gose, Johnsonbaugh, PHI

Public Key Infrastructure and Trust Management				
Prerequisite:	L	T	P	C
Total hours: 40	3	0	0	3

Course Content		Hrs.
Unit 1	Public key infrastructure - components and architecture. PKI interoperability, deployment and assessment PKI data structures – certificates, validation, revocation, authentication, cross certification.	10
Unit 2	Repository, Certification Authority (CA) and Registration Authority (RA), trusted third party, digital certificates PKI services – authentication, non-repudiation, privilege management, privacy, secure communication.	12
Unit 3	Key management – certificate revocation list, root CA, attacks on CA, key backup.	12
Unit 4	PKI standards – SSL, LDAP, IPSec, X.500, X.509, S/MIME Trust models – strict v/s loose hierarchy, four corners distribution. Certificate path processing – path construction and path validation.	6

References:	
1.	Ashutosh Saxena, Public Key Infrastructure, Tata McGraw Hill
2.	Carlisle Adams, Steve Lloyd. Understanding PKI: Concepts, Standards, and Deployment Considerations, Addison Wesley.
3.	John R. Vacca. Public Key Infrastructure: Building Trusted Applications and Web Services, AUERBACH.
4.	Messaoud Benantar, Introduction to the Public Key Infrastructure for the Internet, Pearson Education.

Quantum Computing					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Introduction to quantum computing				8
Unit 2	Relevant Linear algebra for quantum computing, Postulates of quantum mechanics,				10
Unit 3	Classical computing, Quantum circuits, Quantum Fourier Transform				10
Unit 4	Quantum search algorithms, Physical realization of quantum computers.				8
Unit 5	Quantum noise, Quantum operations, quantum information and quantum channel				6
References:					
1.	Pittenger A. O., An Introduction to Quantum Computing Algorithms				
2.	Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press.				
3.	Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific.				

Quantum Cryptography					
Prerequisite:		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Preliminaries: Quantum Information Theory, Quantum Information Theory, Unconditional Secure Authentication and Entropy.				8
Unit 2	Quantum Key Distribution: Quantum Channel, Public Channel, QKD Gain, Finite Resources, Adaptive Cascade: Introduction, Error Correction and the Cascade Protocol, Adaptive Initial Block-Size Selection, Fixed Initial Block-Size, Dynamic Initial BlockSize.				10
Unit 3	Attack Strategies on QKD Protocols: Attack Strategies in an Ideal Environment, Individual Attacks in an Realistic Environment. QKD Systems, Statistical Analysis of QKD Networks in Real-Life Environment: Statistical Methods, Results of the Experiments, Statistical Analysis.				10
Unit 4	QKD Networks Based on Q3P : QKD Networks, PPP, Q3P, Routing and Transport. QuantumCryptographic Networks from a Prototype to the Citizen.				8
Unit 5	The Ring of Trust Model, Model of the Point of Trust Architecture, Communication in the Point of Trust Model, Exemplified Communications, A Medical Information System Based on the Ring of Trust.				6
References:					
1.	Quantum Cryptography and Secret-Key Distillation, Gilles van Assche, Cambridge University Press, 2006.				
2.	Paul Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, Oxford University Press (2007).				
3.	Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press (2000).				

Real Time Systems				
Prerequisite: None	L	T	P	C
Total hours: 40	3	0	0	3

	Course Content	Hrs.
Unit 1	Introduction : Definition, Typical Real Time Applications; Digital Control, High Level Controls, Signal Processing etc., Release Times, Deadlines, and Timing Constraints, Hard Real Time Systems and Soft Real Time Systems, Reference Models for Real Time Systems: Processors and Resources, Temporal Parameters of Real Time Workload, Periodic Task Model, Precedence Constraints and Data Dependency	7
Unit 2	Real Time Scheduling: Common Approaches to Real Time Scheduling: Clock Driven Approach, Weighted Round Robin Approach, Priority Driven Approach, Dynamic Versus Static Systems, Optimality of Effective-Deadline-First (EDF) and Least-Slack-Time-First (LST) Algorithms, Offline Versus Online Scheduling, Scheduling Aperiodic and Sporadic jobs in Priority Driven and Clock Driven Systems	8
Unit 3	Resources Access Control: Effect of Resource Contention and Resource Access Control (RAC), Nonpreemptive Critical Sections, Basic Priority-Inheritance and Priority-Ceiling Protocols, Stack Based Priority-Ceiling Protocol, Use of Priority-Ceiling Protocol in Dynamic Priority Systems, Preemption Ceiling Protocol, Access Control in Multiple-Unit Resources, Controlling Concurrent Accesses to Data Objects	8
Unit 4	Multiprocessor System Environment :Multiprocessor and Distributed System Model, Multiprocessor Priority-Ceiling Protocol, Schedulability of FixedPriority End-to-End Periodic Tasks, Scheduling Algorithms for End-to-End Periodic Tasks, Endto-End Tasks in Heterogeneous Systems, Predictability and Validation of Dynamic Multiprocessor Systems, Scheduling of Tasks with Temporal Distance Constraints.	9
Unit 5	Real Time Communication : Model of Real Time Communication, Soft and Hard RTCommunication systems , Priority-Based Service and Weighted Round-Robin Service Disciplines for Switched Networks, Medium Access Control Protocols forBroadcast Networks, Internet and Resource Reservation Protocols, Real Time Protocols, Communication in Multicomputer System. An Overview of Real Time Operating Systems and Databases: Features of RTOS, UNIX as RTOS, POSIX Issues, Temporal Consistency, Concurrency Control.	8

References:	
1.	Real Time Systems: Theory and Practice – Mall Rajib, Pearson Education, 2009
2.	Real-Time Systems: Scheduling, Analysis, and Verification – Albert M. K. Cheng, Wiley, 2002.
3.	H. Kopetz, "Real time systems: Design Principles for distributed embedded applications", Springer Publications, 2011.
4.	Douglass, Real Time UML: Advances in the UML for Real-Time Systems, 3/e, AddisonWesley, 2004.
5.	Awad, Kuusela& Ziegler, Object-Oriented Technology for Real Time Systems: A Practical Approach Using OMT and Fusion, l/e, Pearson Education, 1996.
6.	Ward & Mellor, Structured Development for Real-Time Systems, Vol. III: Implementation Modeling Techniques, Prentice Hall, 1986.

Robotics and Control					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Introduction to robotics-origin of automation, Classification of robots, Rotations and translation of vectors.				8
Unit 2	Transformations and Euler angle representations, Homogeneous transformations, Problems, Trajectory planning.				10
Unit 3	Actuators, Velocity and position sensors. Range, proximity, touch sensors.				10
Unit 4	Control of Robot Manipulators: PD control, Nonlinear Control, Stability, Lyapunov's Direct Method.				8
Unit 5	Adaptive Control, Robot Vision, Image segmentation, Template matching, Polyhedral objects, Shape analysis, Grasping and industrial automation.				6
References:					
1.	M. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control Wiley (2006)				
2.	Mikell P Groover, Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, "Industrial Robotics, Technology programming and Applications",				
3.	Craig. J. J. "Introduction to Robotics- mechanics and control", Addison- Wesley, 1999				
4.	Nagrath Gopal "Control Systems Engineering -Principles and Design" New Age Publishers				
5.	K. Ogata, "Modern control engineering", Pearson 2002.				

<table>
<tr><th colspan="6">Security Analysis of Protocols</th></tr>
<tr><td colspan="2">Prerequisite:</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 40</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5"><b>Course Content</b></td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">Cryptographic background; Authentication, Key establishment and IP security;</td><td>8</td></tr>
<tr><td>Unit 2</td><td colspan="4">Denial of service; Anonymity and MIX networks; Fairness and contract signing, Privacy and protection of individual information; Wireless security (mobile phones, WiFi);</td><td>12</td></tr>
<tr><td>Unit 3</td><td colspan="4">Protocol analysis tools: Finite-state checking; Infinite-state symbolic analysis; Probabilistic model checking; Game-based verification; Process algebras (spi-calculus and applied pi calculus); Protocol logics (BAN, DDMP, Isabelle);</td><td>12</td></tr>
<tr><td>Unit 4</td><td colspan="4">Introduction to Probabilistic polynomial time calculus; Relating cryptographic and formal models.</td><td>8</td></tr>
<tr><td colspan="6">References:</td></tr>
<tr><td>1.</td><td colspan="5">Latest reputed conference and journal articles as chosen by the instructor.</td></tr>
<tr><td>2.</td><td colspan="5">Maximum Security, 2nd Edition, SAMS Books by Anonymous, 1998</td></tr>
<tr><td>3.</td><td colspan="5">Maximum Linux Security, SAMS Books by Anonymous, 2000, ISBN: 0-672- 31670-6.</td></tr>
<tr><td>4.</td><td colspan="5">10 Risks of PKI: What You're not Being Told about Public Key Infrastructure, by Ellison and Schneier</td></tr>
</table>

Selected Topics in Cryptography					
Prerequisite:		L	T	P	C
Total hours: 40		3	0	0	3
Course Content					Hrs.
Unit 1	Basic Concepts: Information theoretic vs. computational security. One way functions, Pseudo randomness generators and functions, Permutations, hash functions.				8
Unit 2	Private-key encryption using pseudo randomness. Private-key authentication. – Public key encryption (and number theory). Public key authentication.				12
Unit 3	Interactive protocols: Touch of complexity theory, Interactive proof systems; 0-knowledge proof systems,0-knowledge authentication, Electronic cash; non-interactive zero-knowledge.				12
Unit 4	Oblivious transfer: Definitions, constructions, and applications, Secure Multiparty computations, Database (differential) privacy. – Proofs of work – Block-chain consensus protocols.				8
References:					
1.	Introduction to Modern Cryptography: Principles and Protocols, by Jonathan Katz and Yehuda Lindell				
2.	A Graduate Course in Applied Cryptography by Dan Boneh and Victor Shoup				
3.	The Joy of Cryptography by Mike Rosulek.				
4.	OdedGoldreich: Foundations of Cryptography Vol 1 and Vol 2				

Social Media Mining

Prerequisite: None	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Online Social Networks (OSNs): Introduction - Types of social networks (e.g., Twitter, Facebook), Measurement and Collection of Social Network Data.	8
Unit 2	Social Networks - Basic Structure and Measures, Basics of Text Processing over Social Data, Entity linking and entity resolution for Social data.	10
Unit 3	Characteristics of OSNs: Information Diffusion, Experimental studies over OSNs, Sampling, Fundamentals of Social Data Analytics: Topic Models, Random Walks, Heterogeneous Information Networks	10
Unit 4	Applied Social Data Analytics: Recommendation Systems, Community identification and link prediction.	8
Unit 5	Advanced Topics: Online experiments for Computational Social Science, Big Data Sampling	6

References:	
1.	Matthew A. Russell. Mining the Social Web: Data Mining Facebook, Twitter, Linkedin, Google+, Github, and More, 2nd Edition, O'Reilly Media
2.	Jennifer Golbeck, Analyzing the social web, Morgan Kaufmann
3.	Charu Aggarwal (ed.), Social Network Data Analytics, Springer
4.	Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, Social Media Mining An Introduction, Cambridge University Press

Software Project Management

Prerequisite: Software Engineering, Computer Programming (C/Java/Python/C++), Microsoft Excel	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Software Project Concepts: Software Project Categorization, Stakeholders, Software project Activities, Practices & Standards, Selecting Process Models (Spiral, Incremental, Prototyping, RAD, Agile).	8
Unit 2	Estimation & Evaluation techniques,Cost Benefit Analysis, Risk Analysis for Project Evaluation, Program management, Project effort and cost estimation; Basis of estimation, Estimation method categorization, SLOC, Function Point Analysis, COCOMO, Putnam’s work. Estimation using FP.	10
Unit 3	Project Planning: Stepwise planning, Activity based approach (WBS), Sequencing and Scheduling of Activities, Critical Path Method. Risk Analysis and Management: Risk Identification, Projection, Risk Identification, Projection, Risk Refinement, Risk Monitoring and Management Schedule and Cost Monitoring: Collecting Data & Reporting, Graphical Visualization techniques, Cost Monitoring, Earned Value analysis, Requirements management, Change Control.	10
Unit 4	Contract Management: Types of Contracts, Stages in Contract Placement, Typical Terms of a Contract, Contract Management and Acceptance.	6
Unit 5	Software Configuration Management (SCM), SCM Tools, Project Reviews Testing and Software Reliability, Metrics, ISO and CMMI, Project Scheduling & Tracking, Software Quality Assurance, Software Configuration Management	8

References:	
1.	Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, 6th Edition, Tata McGraw Hill, 2017.
2.	Pankaj Jalote, Software Project Management in Practice.
3.	Roger S. Pressman, Software Engineering
4.	Royce, “Software Project Management”, Pearson Education, 1999.
5.	Robert K. Wysocki, Effective Software Project Management, Wiley, 2009.

System on Chip					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Transaction-Level Modeling& Electronic System-Level Languages,				8
Unit 2	Hardware Accelerators, Media Instructions, Co-processors				10
Unit 3	System-Level Design Methodology ,High-Level Synthesis (Cto-RTL),				10
Unit 4	Hardware Synthesis and Architecture Techniques Source-Level Optimizations.				8
Unit 5	Scheduling Resource, Binding and Sharing.				6
References:					
1.	De Micheli, editor Special Issue on Hardware/Software Co-design Proceedings of IEEE, Vol 85, No. 3, March 1997				
2.	D. D. Gajski, F. Vahid, S. Narayan, J. Gong :Specification and Design of Embedded Systems, Prentice Hall, Englewood Cliffs, NJ, 1994				
3.	J. Staunstrup and W. Wolf, editors: Hardware/Software Co-Design: Principles and Practice Kluwer Academic Publishers, 1997				
4.	G. DeMicheli, R. Ernst, and W. Wolf, editors, Readings in Hardware/Software Co-Design, Academic Press, 2002.				

Wireless Sensor Networks

Prerequisite: None	L	T	P	C
Total hours: 42	3	0	0	3

	Course Content	Hrs.
Unit 1	Introduction: Introduction to adhoc/sensor networks: Key definitions of adhoc/ sensor networks, unique constraints and challenges, advantages of adhoc/sensor network, driving applications, issues in adhoc wireless networks/sensor network, data dissemination and gathering, Historical Survey of Sensor Networks	8
Unit 2	Basic Architectural Framework:Traditional layered stack, Cross-layer designs, Sensor network architecture, Physical Layer, Basic Components, Hardware Platforms: Motes, Sensor Devices, Types of Sensors, Sensor's Specification	8
Unit 3	MAC Protocols : Fundamentals of MAC protocols - Low duty cycle protocols and wakeup concepts - Contention Based protocols - Schedule-based protocols - SMAC - BMAC - Traffic-adaptive medium access protocol (TRAMA) - The IEEE 802.15.4 MAC protocol. Routing Protocols: Issues in designing a routing protocol, classification of routing protocols, table-driven, on-demand, hybrid, flooding, hierarchical, and power aware routing protocols.	12
Unit 4	Sensor network security: Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Layer wise attacks in wireless sensor networks, possible solutions for jamming, tampering, black hole attack, flooding attack. Key Distribution and Management.	8
Unit 5	Secure Routing – SPINS, reliability requirements in sensor networks. Programming in WSNs: Challenges and limitations of programming WSNs, Introduction to TinyOS, -Programming in Tiny OS using NesC, Emulator TOSSIM, Open research issues	6

References:

1.	Feng Zhao, Leonidas Guibas, “ Wireless Sensor Network”, Elsevier, 1st Ed. 2004 (ISBN: 13-978-1-55860-914-3)
2.	Kazem, Sohraby, Daniel Minoli, TaiebZnati, “Wireless Sensor Network: Technology, Protocols and Application”, John Wiley and Sons 1st Ed., 2007 (ISBN: 978-0-471-74300-2).
3.	Raghavendra, Cauligi S, Sivalingam, Krishna M., ZantiTaieb, “Wireless Sensor Network”, Springer 1st Ed. 2004 (ISBN: 978-4020-7883-5).
4.	E. H. Callaway, Jr. E. H. Callaway, Wireless Sensor Networks Architecture and Protocols:, CRC Press , 2009