SCHEME OF INSTRUCTION AND DETAILED SYLLABI FOR

B. TECH.

IN

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Effective from 2018 Batch onwards

Dr. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

JALANDHAR – 144011

Phone: 0181-2690301, 02 (Ext. 2502) **Fax: 0181-2690932** **Website: www.nitj.ac.in**

DR B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR

INSTITUTE VISION AND MISSION STATEMENTS

VISION

To build a rich intellectual potential embedded with interdisciplinary knowledge, human values and professional ethics among the youth, aspirant of becoming engineers and technologists, so that they contribute to society and create a niche for a successful career.

MISSION

To become a leading and unique institution of higher learning, offering state-of-the art education, research and training in engineering and technology to students who are able and eager to become change agents for the industrial and economic progress of the nation. To nurture and sustain an academic ambience conducive to the development and growth of committed professionals for sustained development of the nation and to accomplish its integration into the global Economy.

DR B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

JALANDHAR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROGRAMME OBJECTIVES

After completion of the B. Tech Programme, student will develop:

1. An ability to apply acquired knowledge of mathematics, science and computer science and engineering to solve engineering problems.
2. An ability to identify, formulate and analyze engineering problems.
3. An ability to design and implement a system, process, component or program to meet desired needs, within realistic constraints such as culture, society, environment, health and safety.
4. An ability to conduct investigation of complex problems to reach valid conclusions and to research the contemporary issues.
5. An ability to use appropriate skills, model tools and techniques necessary for computing and engineering practices.
6. An ability to demonstrate professional responsibilities pertaining to computer science and engineering by the analysis of societal, health, safety, legal and cultural issues.
7. An ability to produce engineering solution in global and societal context and demonstrate the needs for sustainable development.
8. Apply ethical principle, professional ethics and norms of computer engineering practices.
9. An ability to function effectively as an individual and multi- disciplinary teams.
10. An ability to prepare technical reports and make presentations for the effective delivery of technical information.
11. Recognition of the need for an ability to engage in lifelong learning.
12. An ability to incorporate appropriate economics and business practices for project, risk and change management.

Annexure-I

III SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-201	Digital Circuits and Logic Design	3	0	0	3	PC
2.	CSPC-203	Object Oriented Programming	3	0	0	3	PC
3.	CSPC-205	Data Structures and Algorithms	3	1	0	4	PC
4.	CSPC-207	Computer Networks	3	0	0	3	PC
5.	CSPC-209	Discrete Structures	3	0	0	3	PC
6.	MACI-203	Numerical Methods	3	1	0	4	CIC
7.	CSPC-221	Digital Circuits and Logic Design Laboratory	0	0	2	1	PC
8.	CSPC-223	Object Oriented Programming Laboratory	0	0	2	1	PC
9.	CSPC-225	Data Structures and Algorithm Laboratory	0	0	2	1	PC
10.	CSPC-227	Computer Networks Laboratory	0	0	2	1	PC
TOTAL			**18**	**2**	**8**	**24**	

IV SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-202	Database Management System	3	0	0	3	PC
2.	CSPC-204	Machine Learning	3	0	0	3	PC
3.	CSPC-206	Design and Analysis of Algorithms	3	1	0	4	PC
4.	CSPC-208	Computer Organization and Architecture	3	1	0	4	PC
5.	ECPC-252	Microprocessor and Microcontroller	3	0	0	3	PC
6.	HMCI-201	Economics for Engineering	3	0	0	3	CIC
7.	CSPC-222	Database Management System Laboratory	0	0	2	1	PC
8.	CSPC-224	Machine Learning Laboratory	0	0	2	1	PC
9.	CSPC-226	Design and Analysis of Algorithms Laboratory	0	0	2	1	PC
10.	ECPC-272	Microprocessor and Microcontroller Laboratory	0	0	2	1	PC
TOTAL			**18**	**2**	**8**	**24**	

Subject to be offered to ECE Department

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1	CSPC-212	Database Management Systems	2	0	0	2	PC
2	CSPC-214	Operating Systems	2	0	0	2	PC
3	CSPC-232	Operating Systems Lab	0	0	2	1	PC

Subject to be offered to BT Department

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1	CSPC-203	Object Oriented Programming	3	0	0	3	PC
2	CSPC-223	Object Oriented Programming Lab	0	0	2	1	PC

Subject to be offered to ICE Department

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-213	Data Structures and Algorithms	3	0	0	3	PC

V SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-301	Theory of Computation	3	1	0	4	PC
2.	CSPC-303	Operating Systems	3	0	0	3	PC
3.	CSPC-305	Software Engineering	3	0	0	3	PC
4.	CSPC-307	Information Security System	3	0	0	3	PC
5.	CSPC-309	Probability Theory for Data Analytics	3	1	0	4	PC
6.	CSPE-3XX	DE-I	3	0	0	3	PE
7.	CSPC-323	Operating Systems Laboratory	0	0	2	1	PC
8.	CSPC-325	Software Engineering Laboratory	0	0	2	1	PC
9.	CSPC-327	Information Security Laboratory	0	0	2	1	PC
10.	CSPE-3XX	DE-I Laboratory	0	0	2	1	PE
11.	CSCI-301	Minor Project, Phase-I	0	0	2	0*	CIC
TOTAL			**18**	**2**	**10**	**24**	

DEPARTMENTAL ELECTIVE (DE)-I

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-331	Advanced Programming Concepts using Java	3	0	0	3
2.	CSPE-333	Natural Language Processing	3	0	0	3
3.	CSPE-335	Web Technologies	3	0	0	3
4.	CSPE-351	Advanced Programming Concepts using Java Laboratory	0	0	2	1
5.	CSPE-353	Natural Language Processing Laboratory	0	0	2	1
6.	CSPE-355	Web Technologies Laboratory	0	0	2	1

VI SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-302	Data Analytics	3	1	0	4	PC
2.	CSPC-304	Cloud Computing	3	0	0	3	PC
3.	CSPC-306	Network Security and Cyber Forensics	3	0	0	3	PC
4.	CSPC-308	Data Mining and Data Warehousing	3	0	0	3	PC
5.	CSPE-3XX	DE-II	3	0	0	3	PE
6.	CSOE-XXX	OE-I	3	0	0	3	OE
7.	CSPC-322	Data Analytics Laboratory	0	0	2	1	PC
8.	CSPC-326	Network Security and Cyber Forensics Laboratory	0	0	2	1	PC
9.	CSPC-328	Data Mining and Data Warehousing Laboratory	0	0	2	1	PC
10.	CSPE-3XX	DE-II Laboratory	0	0	2	1	PE
11.	CSCI-301	Minor Project, Phase-II	0	0	2	2*	CIC
TOTAL			**18**	**1**	**10**	**25**	

*** Minor Project will be allotted in 5th Semester, will be evaluated after 6th Semester**

DEPARTMENTAL ELECTIVE (DE)-II

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-332	Advanced Computer Networks	3	0	0	3
2.	CSPE-334	Android Programming and Mobile Applications Development	3	0	0	3
3.	CSPE-336	Internet of Things	3	0	0	3
4.	CSPE-352	Advanced Computer Networks Laboratory	0	0	2	1
5.	CSPE-354	Android Programming and Mobile Applications Development Laboratory	0	0	2	1
6.	CSPE-356	Internet of Things Laboratory	0	0	2	1

OPEN ELECTIVE (OE) –I

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	IDOE-001	Introduction to Databases	3	0	0	3
2.	IDOE-002	Introduction to Embedded Systems	3	0	0	3

VII SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-401	Software Project Management	3	0	0	3	PC
2.	CSPC-403	Computer Graphics and Image Processing	3	0	0	3	PC
3.	CSPC-405	Artificial Intelligence	3	0	0	3	PC
4.	CSPE-4XX	DE-III	3	0	0	3	PE
5.	CSOE-XXX	OE-II	3	0	0	3	OE
6.	CSPC-423	Computer Graphics and Image Processing Laboratory	0	0	2	1	PC
7.	CSCI-300	Industrial Practical Training	0	0	8	2	CIC
8.	CSCI-400	Project (Phase-I)	0	0	4	0*	CIC
TOTAL			**15**	**0**	**14**	**18**	

DEPARTMENTAL ELECTIVE (DE)-III

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-431	Big Data Analytics	3	0	0	3
2.	CSPE-433	Block Chain Architecture & Use Cases	3	0	0	3
3.	CSPE-435	Distributed System	3	0	0	3

OPEN ELECTIVES (OE)-II

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	IDOE-003	Software Engineering Concepts	3	0	0	3
2.	IDOE-004	Artificial Intelligence and Applications	3	0	0	3

VIII SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-402	System Programming and Compiler Design	3	0	0	3	PC
2.	CSPE-4XX	DE-IV	3	0	0	3	PE
3.	CSPE-4XX	DE-V	3	0	0	3	PE
4.	CSOE-XXX	OE-III	3	0	0	3	OE
5.	CSPC-422	System Programming and Compiler Design Laboratory	0	0	2	1	PC
6.	CSCI-400	Project (Phase-II)	0	0	8	4	CIC
7.	CSCI-424	Industrial Lecture	0	0	2	1	CIC
TOTAL			**12**	**0**	**12**	**18**	

*** Major Project will be allotted in 7th Semester, will be evaluated in 8th Semester**

DEPARTMENTAL ELECTIVE (DE)-IV

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-442	High Performance Computing	3	0	0	3
2.	CSPE-444	Soft Computing	3	0	0	3
3.	CSPE-446	Wireless Networks	3	0	0	3

DEPARTMENTAL ELECTIVE (DE)-V

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-448	Social Network Analysis	3	0	0	3
2.	CSPE-450	Human Computer Interaction	3	0	0	3
3.	CSPE-452	Principles of Deep Learning	3	0	0	3

OPEN ELECTIVES (OE)-III

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	IDOE-005	Bio Informatics	3	0	0	3
2.	IDOE-006	Soft Computing Techniques	3	0	0	3

Summary Sheet of Credits

Semester	Course Category	Number	Credits	Total Credits
III	CIC	01	03	**24**
	PC	09	21	
	PE	-	-	
	OE	-	-	
IV	CIC	01	03	**24**
	PC	09	21	
	PE	-	-	
	OE	-	-	
V	CIC	1	0	**24**
	PC	08	20	
	PE	2	4	
	OE	-	-	
VI	CIC	1	2	**25**
	PC	7	16	
	PE	2	4	
	OE	1	3	
VII	CIC	2	2	**18**
	PC	4	10	
	PE	1	3	
	OE	1	3	
VIII	CIC	2	5	**18**
	PC	2	4	
	PE	2	6	
	OE	1	3	
Total Credits (III to VIII Sem)				**133**
Total No of PC		40	91	
Total No of PE		07	17	
Total No of OE		03	09	
Total Credits of CIC		08	16	
Credits from 1st year			47	**47**
Total Credits			**180**	**180**

Annexure-I

III SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-201	Digital Circuits and Logic Design	3	0	0	3	PC
2.	CSPC-203	Object Oriented Programming	3	0	0	3	PC
3.	CSPC-205	Data Structures and Algorithms	3	1	0	4	PC
4.	CSPC-207	Computer Networks	3	0	0	3	PC
5.	CSPC-209	Discrete Structures	3	0	0	3	PC
6.	MACI-203	Numerical Methods	3	1	0	4	CIC
7.	CSPC-221	Digital Circuits and Logic Design Laboratory	0	0	2	1	PC
8.	CSPC-223	Object Oriented Programming Laboratory	0	0	2	1	PC
9.	CSPC-225	Data Structures and Algorithm Laboratory	0	0	2	1	PC
10.	CSPC-227	Computer Networks Laboratory	0	0	2	1	PC
TOTAL			**18**	**2**	**8**	**24**	

DETAILED SYLLABI OF ALL SUBJECTS (3rd Semester onwards)

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-201
COURSE TITLE: DIGITAL CIRCUITS AND LOGIC DESIGN
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After completion of the course, students will be able to:

1. Demonstrate knowledge of binary number, boolean algebra and binary codes.
2. Design, simulate, built and debug complex combinational and sequential circuits based on an abstract functional specification.
3. Analyze combinational systems using standard gates and minimization methods such as karnaugh maps.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-201												
CO 1.	H	M	L									
CO 2.	M	M	H	L								
CO 3.	M	H	M	L								

TOPICS COVERED

Basics of Number Systems, Boolean Algebra and Logic Gates

Gate – Level Minimization: The map method, Four-variable map, Five-Variable map, product of sums simplification Don't-care conditions, NAND and NOR implementation other Two-level implementations, Exclusive – Or function, Hardware Description language (HDL).

Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

Synchronous Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, HDL for sequential circuits, State Reduction and Assignment, Design Procedure**.**

Registers and Counters: Registers, shift Registers, Ripple counters synchronous counters, other counters, HDL for Registers and counters.

Memory, CPLDs, and FPGAs: Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.

Asynchronous Sequential Logic: Introduction, Analysis Procedure, Circuits with Latches, Design Procedure, Reduction of state and Flow Tables, Race-Free state Assignment Hazards, Design Example.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. DIGITAL DESIGN – Third Edition, M.Morris Mano, Pearson Education/PHI.
2. Digital Principles and Design – Donald D.Givone, Tata McGraw Hill, Edition.
3. John F Wakerly, "Digital Design Principles and Practices 3/e", Pearson Education 2001.
4. J P. Hayes, "Introduction to Digital Logic Design", Addison-Wesley Publishing Co
5. Charles H. Roth, Jr. Fundamentals of logic design, Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-203
COURSE TITLE: OBJECT ORIENTED PROGRAMMING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand fundamentals of programming like variables, conditional and iterative execution, methods, etc.
2. Understand fundamentals of object-oriented programming, including defining classes, invoking methods, using class libraries, etc.
3. Have the ability to write a computer program to solve specified problems.
4. Be able to use the OOP environment to create, debug and run simple C++ programs.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-203												
CO 1.	M	M	H	L	L	L						
CO 2.	M	M	H	L	L							
CO 3.	L	M	H	M					M			
CO 4.		L	H	L				M	M			

TOPICS COVERED

Object oriented thinking: Need for OOP Paradigm, Procedural programming vs object oriented programming, object oriented concepts.

Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, function overloading, Math library functions.

Class: Difference between C structure and class, specifying a class, Defining member functions: inside and outside class, scope resolution operator, Array within a class, array of objects, Static data members and member functions, Object as function arguments, returning objects, Friend function, memory allocation for objects ,pointer to members, pointer to object, this pointer local classes.

Constructor and destructor: Constructor, types of constructors: default, parameterized and copy constructor, constructor overloading, constructor with default parameter, dynamic initialization of objects, destructor
Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator,Data Conversion: Basic to User Defined , User defined to basic, Conversion from one user-defined to other.

Inheritance and polymorphism: Base class, derived class, visibility modes, derivation and friendship, Types of inheritance, Containership, virtual function binding, pure virtual functions, Abstract class, pointer to derived class.

Console IO operations: C++ stream classes, Unformatted IO operations, formatted IO operations, managing output with manipulators.
Working with files: Classes for file stream operations, opening and closing files, File opening modes, file Pointers, Error handling during file operations, command line arguments. Templates: Class template, class template with parameter, function template, function template with parameter.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. BjraneStroustrup, "C++ Programming language",3rd edition, Pearson education Asia(1997)
2. LaforeR."Object oriented Programming in C++", 4th Ed. Techmedia,New Delhi(2002).
3. YashwantKenetkar,"Let us C++",1stEd.,Oxford University Press(2006)
4. B.A. Forouzan and R.F. Gilberg,CompilerScience,"A structured approach using C++" Cengage Learning, New Delhi.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-205
COURSE TITLE: DATA STRUCTURES AND ALGORITHMS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand the concepts of data structure, data type and array data structure.
2. Analyze algorithms and determine their time complexity.
3. Implement linked list data structure to solve various problems.
4. Understand and apply various data structure such as stacks, queues, trees and graphs to solve various computing problems using C-programming language.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-205												
CO 1.	M	L	H	L								
CO 2.		H	M	M	L		M	M				
CO 3.		M	M									
CO 4.		M	H	L	L							

TOPICS COVERED

Introduction: Basic Terminology, Elementary Data Organization, Structure operations, Algorithm Complexity and Time-Space trade-off

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C++, Character string operation, Array as Parameters, Ordered List, Sparse Matrices and Vectors.

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, and Application of stack: Conversion of Infix to prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Recursion: Recursive definition and processes, recursion in C, example of recursion, Tower of Hanoi Problem, simulating recursion, Backtracking, recursive algorithms, principles of recursion, tail recursion, removal of recursion.

Queues: Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues**.**

Linked list: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list, Garbage Collection and Compaction.

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm**.**

Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-trees.

Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting**.**

Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

File Structures: Physical Storage Media File Organization, Organization of records into Blocks, Sequential Files, Indexing and Hashing, Primary indices, Secondary indices, B+ Tree index Files, B Tree index Files, Indexing and Hashing Comparisons.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., New Delhi.
2. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education Asia, Delhi-2002
3. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., New Delhi.
4. Bruno R Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", Jhon Wiley & Sons, Inc.
5. GilbergForozan , "Data Structure – A pseudo code approach with C++", Cengage Learning, New Delhi.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-207
COURSE TITLE: COMPUTER NETWORKS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand basic computer network technology, data communications System and its components.
2. Identify the different types of network topologies and protocols, to enumerate the layers of the OSI model and TCP/IP
3. Identify the different types of network devices and their functions within a network.
4. Understand and build the skills of subnetting and be familiar with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-207												
CO 1.			M		M							
CO 2.	M	H					M					
CO 3.			M		M							
CO 4.			H		M	M						

TOPICS COVERED

Introduction: Goals and Applications of Networks, Network structure and architecture, OSI reference model, TCP/IP Protocol suite, Layering principles, Network Topology Design, connecting devices, Physical Layer Transmission Media, Line coding scheme, Basic idea of modulation and multiplexing, Switching methods.

Local Area Networks: LAN topologies: Bus topology, Ring topology, Token passing rings, FDDI, Star topologies, Asynchronous transfer mode, Ethernet, IEEE standards 802.3, 802.5. Wireless LANs: IEEE 802.11 and Bluetooth.

Reliable Data Delivery: Error detection and Correction, Error control (retransmission techniques, timers), Flow control (Acknowledgements, sliding window), Medium Access sub layer - Channel Allocations, LAN protocols - ALOHA protocols, CSMA, CSMA/CD, Overview of IEEE standards.

End-to-end Delivery, routing and Forwarding: Routing versus forwarding, Static and dynamic routing, Unicast and Multicast Routing. Distance-Vector, Link-State, Shortest path computation, Dijkstra's algorithm, Network Layer Protocols (IP, ICMP), IP addressing, sub netting and super netting (CIDR), IPV4 and IPV6, Address mapping-ARP, RARP, BOOTS, and DHCP.

Process-to-process Delivery: design issues, UDP and TCP segment formats, connection establishment and termination, Principles of congestion control, Approaches to Congestion control, Quality of service, Flow characteristics, Techniques to improve QoS.
Brief introduction to session and presentation layer.

Network applications and security: Naming and address schemes (domain name system, IP addresses, Uniform Resource Identifiers, etc.), HTTP as an application layer protocol, Remote login, Electronic mail, SMTP, FTP Commands and Replies, WWW and SNMP, Security in computer networks: principles of cryptography, symmetric key, public key, authentication protocols, digital signatures, firewalls. Security in different layers: secure E-mail, SSL, IP security.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Andrew S. Tanenbaum, "Computer Networks" Ed Pearson Education 4th Edition, 2010.
2. James F. Kurose and Keith W. Ross, "Computer Networking" Pearson Education, 2012.
3. William Stalling, "Data and Computer Communication", Pearson Education, 7th Edition, 2nd Indian Reprint 2010.
4. Behrouz A. Fourouzan,Data Communications and Networking, 2/e Tat McGrawhill,2006.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-209
COURSE TITLE: DISCRETE STRUCTURES
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests, which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Perform operations on various discrete structures such as sets, functions, relations, and sequences.
2. Solve problems using Counting techniques, Permutation and Combination, Recursion and generating functions.
3. Apply algorithms and use of graphs and trees as tools to visualize and simplify Problems.
4. Understand the various properties of algebraic systems like Rings, Monoids and Groups

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-209												
CO 1.	H	L										
CO 2.	L	M	H	M								
CO 3.	L	M	M	H	M	H						
CO 4.	H	M	L									

TOPICS COVERED

Set Theory: Basics of set theory, Cartesian Product of Sets, Partition of Sets, Concept of Relation & Properties of Relations, Different types of Relations, Tabular and Matrix Representation of Relations, Relations and Diagraphs, Composition of Relations, Functions and their different mappings, Composition of Function, Recursion and Recurrence Relations.

Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded I and complemented lattices.

Boolean Algebra: Partial Ordering, Totally ordered Sets, Dual Order, Hasse Diagram Lexicographic Ordering, Cover of an Element, Least and Greatest Elements, Minimal and Maximal Elements ,Upper and Lower Bound , Well-Order Set, Binary and n-Ary Operations, Lattices, Atoms of a Boolean Algebra, Boolean Expressions, Applications of Boolean Algebra to Switching Theory.

Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Universal and existential quantifiers.

Combinatorics & Graphs: Recurrence Relation, Generating function, Bipartite, Regular, Planar and connected graphs, connected components in a graph, Euler graphs, Hamiltonian path and circuits, Graph coloring, chromatic number, isomorphism and Homomorphism of graphs.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Liptschutz, Seymour, “Discrete Mathematics”, McGraw Hill.
2. Trembley, J.P & R. Manohar, “Discrete Mathematical Structure with Application to Computer Science”, McGraw Hill.
3. Kenneth H. Rosen, “Discrete Mathematics and its applications”, McGraw Hill.
4. Deo, Narsingh, “Graph Theory With application to Engineering and Computer.Science.”, PHI.
5. Krishnamurthy, V., “Combinatorics Theory & Application”, East-West Press Pvt. Ltd., New Delhi.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: MACI-203
COURSE TITLE: NUMERICAL METHODS
COURSE DESIGNATION: REQUIRED

PRE-REQUISITES: NONE

CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)

COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Understand the different numerical methods to solve the algebraic equations and to solve system of linear and non linear equations.
2. Understand the different numerical methods for interpolation, differentiation, integration and solving set of ordinary differential equations.
3. Understand how numerical methods afford a mean to generate solutions in a manner that can be implemented on digital computers.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
MACI-203												
CO 1.	H	M										
CO 2.		M	L	M								
CO 3.	M			M								

TOPICS COVERED

Roots of algebraic and transcendental equations, Bisection method, Regula – Falsi method, Newton –Raphson method, Bairstow's method and Graeffe's root squaring method.

Solution of simultaneous algebraic equations, matrix inversion and eigen-value problems, triangularisation method, Jacobi's and Gauss-Siedel iteration method, partition method for matrix inversion, power method for largest eigen-value and Jacobi's method for finding all eigen-values.

Finite differences, interpolation and numerical differentiation, forward, backward and central differences, Newton's forward, backward and divided difference interpolation formulas, Lagrange's interpolation formula, Stirling's and Bessel's central difference interpolation formulas, numerical differentiation using Newton's forward and backward difference formulas and numerical differentiation using Stirling's and Bessel's central difference interpolation formulas.

Numerical integration, Trapezoidal rule, Simpson's one-third rule and numerical double integration using Trapezoidal rule and Simpson's one-third rule.

Taylor's series method, Euler's and modified Euler's methods, Runge-Kutta fourth order methods for ordinary differential equations, simultaneous first order differential equations and second order differential equations.

Boundary value problems, finite difference methods for boundary value problems.

Partial differential equations, finite difference methods for elliptic, parabolic and hyperbolic equations.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. S SSastry, Introductionary Methods of Numerical Analysis, 3rd Edition, Prentice Hall of India Pvt.Ltd., New India -1999
2. S C Chapra and R P Canale, Numerical Methods for Engineers, 2nd Edition, McGraw Hill Book Company, Singapore 1990.
3. Kalavathy S., "Numerical Methods", Cengage Publishers, New Delhi.
4. Burden Richard L. ,Faires J. Douglas, "Numerical Anlaysis" , Cengage Learning , New Delhi.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-221
COURSE TITLE: DIGITAL AND ANALOG CIRCUITS LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES:
After the completion of the course, the students will be able to:

1. Analyze and design digital combinational circuits like decoders, encoders, multiplexers, and de-multiplexers including arithmetic circuits (half adder, full adder, and multiplier).
2. Analyze sequential digital circuits like flip-flops, registers, counters.
3. Gain knowledge of the nomenclature and technology in the area of memory devices: ROM, RAM, PROM, EPROM, etc.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-221												
CO 1.	H		M						M			
CO 2.		H		M					H			
CO 3.	M											

LIST OF PRACTICALS

1. Implementation of Boolean Function - Adders / Subtractors - Decoders Encoders - Multiplexer - Demultiplexers - Comparators - Parity Checker/Generator.
2. Register Counters - Shift Registers - General-purpose shift registers - Data transmission.
3. Project - A mini project involving clocked sequential networks design.
4. To implement BCD-to-7 Segment decoder and to verify the truth table.
5. To construct different types of flip-flops and verify their truth tables. Flip-flops like J-K flip-flops. S-R flip-flop. And D-flip-flops etc.
6. To construct and verify a Master-Slave flip-flop.
7. Construction and study of Modulo-N counter using IC's 7490 decade counter, 7493 binary counter.
8. EPROM Programming
9. Study & working of DMA controller.
10. Designing of Traffic control system.

This is only the suggested list of Practical's. Instructor may frame additional Practical's relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-223
COURSE TITLE: OBJECT ORIENTED PROGRAMMING LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-1-2)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Gain understanding about the object oriented principles in construction of robust and maintainable programs.
2. Have a competence to design, write, compile, test and execute programs using high level language.
3. Have an awareness of the need for a professional approach to design and the importance of good documentation to finish.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-223												
CO 1.	H		M	M								
CO 2.	L	H	H	M								
CO 3.				H		M		H	M	H		M

LIST OF PRACTICALS

1. Write a program to read a matrix of size m x n form the keyboard and display the same using function.
2. Write a Program to make the use of inline function.
3. Write a function power () which raise a number m to a power n. The function takes double value of m and integer value of n and returns the result. Use a default value of n is 2 to make the function to calculate squares when this argument is omitted.
4. Program to show that the effect of default arguments can be alternatively achieved by overloading.
5. Write a class ACCOUNT that represents your bank account and then use it.
6. The class should allow you to deposit money, withdraw money, calculate interest, send you a message if you have insufficient balance.
7. Write a class STRING that can be used to store strings, add strings, equate string, output strings.
8. Create the class TIME to store time in hours and minutes. Write a friend function to add two TIME objects.
9. Create two classes DM and DB. DM stores the distance in meter and centimeters and DB stores the distance in feet and inches. Write a program two add object of DM with the object of DB class.
10. Write a program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes inherits the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
11. Write Programs to demonstrate the concept of Default constructor, Parameterized constructor, Copy constructor, and Constructor overloading
12. Program to demonstrate the concept of destructor, multiple inheritance, multilevel inheritance, hybrid inheritance, and concept of containership.
13. Program to overload unary operator and overload binary operator
14. Program to show the concept of run time polymorphism using virtual function.
15. Program to work with formatted and unformatted IO operations.
16. Program to read the name and roll numbers of students from keyboard and write them into a file and then display it.
17. Program to copy one file onto the end of another, adding line numbers
18. Write a function template for finding the minimum value contained in an array.
19. Write a class template to represent generic vector (a series of float values). Include member function to perform following tasks.
    a. Create vector
    b. Modify the value of a given element
    c. To multiply by a scalar value
    d. To display vector in the form of (10, 20, 30,……)

This is only the suggested list of Practical's. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-225
COURSE TITLE: DATA STRUCTURES AND ALGORITHM LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand the importance of data structures and abstract data type, and their basic usability in different applications through different programming languages.
2. Analyze and differentiate different algorithms based on their time complexity.
3. Understand various data structure such as stacks, queues, trees, graphs, etc. to solve various computing problems.
4. Implement various kinds of searching and sorting techniques, and know when to choose which technique.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-225												
CO 1.	H	M	M	M	M							
CO 2.		H	M	M	M	M	M		L			
CO 3.	L	M	M	H	M							
CO 4.		H	H	L								

LIST OF PRACTICALS

Write Program in C or C++ for following.

1. Write a C+ program to implement push, pop, and display operations on stack of integers. The program should print appropriate messages for stack overflow, stack Underflow & stack empty.
2. Write a C++ program to convert & print a given valid parenthesized in fix Arithmetic expression to postfix expression. The expression consists of single character operands &+ , - ,*,/ operators .
3. Write a c program to evaluate a valid suffix / postfix expression using a Stack, assume that the suffix / postfix expression is read as a single line consisting of non negative single digit operands & binary arithmetic operands. The arithmetic operators are + (ADD), - (subtract), *(multiply) & / (divide).
4. Write a C++ program to simulate the working a queue of integers using an array. Provide the a) insert b) delete c) display
5. Write a C++ program to simulate the working of a circular queue of integers using an array. Provide the following operations: a) insert b) delete c)Display
6. Write a program to design a priority queue which is maintained as a set of queue (assume a maximum of 3 queues). The elements are inserted based upon the given priority. The deletion of an element is to be done starting from the 1st queue, if it is not empty .If it is empty, the elements from the 2nd queue will be deleted & so on.
7. Write a C++ program using dynamic variable & pointers to construct a singly linked list consisting of the following information in each node. Student id (integer), student name (character string) & semester (integer).
8. Write a C++ program using dynamic variables & pointers to construct an ordered(ascending) singly linked list based on the rank of the student, where each node consists of the following information student id(integer)student name(character), rank(integer)
9. Write a C++ program using dynamic variables & pointers to construct a singly linked list to perform the operations of a stack of integers
10. The program should print appropriate message for stack overflow & stack empty
11. Write a C++ program to show operations on a doubly linked where each node consists of integers.
12. Write C++ program to construct a binary search tree of integers and traverse the tree using all the methods i.e. inorder, preorder & postorder to display the elements in the tree
13. Write C++ program for the various searching techniques over a list of integers.
14. Write a C++ program to sort a list of N integers using the quick sort algorithm.
15. Write a C++ program to sort a list of N strings using the insertion sort algorithm.
16. Write a C++ program to sort a list of N integers using Heap sort algorithm.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-227
COURSE TITLE: COMPUTER NETWORKS LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student learning outcomes include:

1. Understand fundamental underlying principles of computer networking
2. Understand details and functionality of layered network architecture.
3. Illustrate design and implementation of data link, network and transport layer protocols within a simulated/real networking environment and practice packet/file transmission between nodes.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-227												
CO 1.		H	M		M							
CO 2.				M	L	M			L			
CO 3.	M			M	M							

LIST OF PRACTICALS

1. Network Physical Components Hands-on (Networks Cabling)
2. Brief introduction to Cisco Packet Tracer covering network devices, cables and end-devices. Connect one PC to another PC using the cable and also design a network of few computers using hub/switch and assign IP address and subnet mask to them. Implement Star topologies in Packet Tracer.
3. Implement Bus, Ring, Mesh, and Hybrid topology. Assign IP address and subnet mask to each computer and run the ping command to check the reachability of the systems. Send message between source and destination and observe the flow of the messages.
4. Design two separate network and connect them using Router
5. Implementation of various variants of Sliding Window protocols
6. Design Static route configuration, Dynamic route configuration, and Default route configuration
7. Design NAT and PAT
8. VLAN Design
9. Experiencing Real-world Network Infrastructure
10. Experiencing Real-world network Devices using Switches and Routers
11. Network Programming – IP handling
12. Network Programming – Sending Packets
13. Network Programming – UDP/TCP Sockets
14. Implementing Client/Server Applications
15. Packet Capturing and Analysis using the monitoring tool
16. Application test using the network monitoring tool

This is only the suggested list of practicals. Instructor may frame additional practicals relevant to the course contents.

IV SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-202	Database Management System	3	0	0	3	PC
2.	CSPC-204	Machine Learning	3	0	0	3	PC
3.	CSPC-206	Design and Analysis of Algorithms	3	1	0	4	PC
4.	CSPC-208	Computer Organization and Architecture	3	1	0	4	PC
5.	ECPC-252	Microprocessor and Microcontroller	3	0	0	3	PC
6.	HMCI-201	Economics for Engineering	3	0	0	3	CIC
7.	CSPC-222	Database Management System Laboratory	0	0	2	1	PC
8.	CSPC-224	Machine Learning Laboratory	0	0	2	1	PC
9.	CSPC-226	Design and Analysis of Algorithms Laboratory	0	0	2	1	PC
10.	ECPC-272	Microprocessor and Microcontroller Laboratory	0	0	2	1	PC
TOTAL			**18**	**2**	**8**	**24**	

Subject to be offered to ECE Department

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-212	Database Management Systems	2	0	0	2	PC
2.	CSPC-214	Operating Systems	2	0	0	2	PC
3.	CSPC-232	Operating Systems Lab	0	0	2	1	PC

Subject to be offered to BT Department

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-203	Object Oriented Programming	3	0	0	3	PC
2.	CSPC-223	Object Oriented Programming Lab	0	0	2	1	PC

Subject to be offered to ICE Department

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
2.	CSPC-213	Data Structures and Algorithms	3	0	0	3	PC

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-202
COURSE TITLE: DATABASE MANAGEMENT SYSTEM
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests, which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES

1. To understand the different issues involved in the design and implementation of a database system. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
2. To understand and use data manipulation language to query, update, and manage a database
3. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
4. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-202												
CO 1.	M		H		M	L						
CO 2.			H	L	M							
CO 3.			M	M	M	H	M	M	M			
CO 4.		M	H	L	H	L		H	M			M

TOPICS COVERED

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model:
ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL. Advantage of SQL.SQL data types and literals.Types of SQL commands.SQL operators and their procedure.Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

Crash Recovery: Failure classification, recovery concepts based on deferred update, recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates.

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Client/Server Databases: Client/Server concepts, approach, Client/Server environments, characterization of Client/Server computing, application partitioning, the two-layer, and the Three layer architecture, Client/Server communication, APIs in Client/Server computing, middleware technology, application developments, design concepts, Client application development tools, and database servers.

Integrity, Security and Repositories: Needs for database integrity, integrity constraints, non-procedural integrity constraints, integrity constraints specifications in SQL, introduction to database security mechanism, security specification in SQL, system catalogues

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Date C J, "An Introduction To Database System", Addision Wesley
2. Korth, Silbertz, Sudarshan, "Database Concepts", McGraw Hill
3. Elmasri, Navathe, "Fundamentals Of Database Systems", Addision Wesley
4. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication
5. Rob and Coronel, "Database Systems 5th Edition",Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-204
COURSE TITLE: MACHINE LEARNING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Demonstrate in-depth knowledge of methods and theories in the field of machine learning and provide an introduction to the basic principles, techniques, and applications of machine learning, classification tasks, decision tree learning.
2. Apply decision tree learning, bayesian learning and artificial neural network in real world problems.
3. Demonstrate the use of genetic algorithms and genetic programming.
4. Apply inductive and analytical learning with perfect domain theories.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-204												
CO 1.				M	H		H					
CO 2.						H	M	H				
CO 3.						M		M				
CO 4.	M											

TOPICS COVERED

Introduction: Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning (Classification and Regression Trees, Support vector machines), Unsupervised learning (Clustering), Instance-based learning (K-nearest Neighbor, Locally weighted regression, Radial Basis Function), Reinforcement learning (Learning Task, Q-learning, Value function approximation, Temporal difference learning).

Decision Tree Learning: Decision tree representation, appropriate problems for decision tree learning, Univariate Trees (Classification and Regression), Multivariate Trees, Basic Decision Tree Learning algorithms, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning.
Bayesian Learning: Bayes theorem and concept learning, Bayes optimal classifier, Gibbs algorithms, Naive Bayes Classifier, Bayesian belief networks, The EM algorithm.

Artificial Neural Network: Neural network representation, Neural Networks as a paradigm for parallel processing, Linear discrimination, Pairwise separation, Gradient Descent, Logistic discrimination, Perceptron, Training a perceptron, Multilayer perceptron, Back propagation Algorithm. Recurrent Networks, Dynamically modifying network structure.

Genetic Algorithms: Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.

Inductive and Analytical Learning: Learning rule sets, Comparison between inductive and analytical learning, Analytical learning with perfect domain theories: Prolog-EBG. Inductive Analytical approaches to learning, Using prior knowledge to initialize hypothesis (KBANN Algorithm), to alter search objective (Tangent Prop and EBNN Algorithm), to augment search operators (FOCL Algorithm).

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Mitchell T.M., Machine Learning, McGraw Hill (1997) 2nd ed.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2010) 2nd ed.
3. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag (2006) 2nd ed.
4. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press (2009) 1st ed.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-206
COURSE TITLE: DESIGN AND ANALYSIS OF ALGORITHMS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.
COURSE OUTCOMES
After the course completion, the student will be able to

1. Have basic knowledge of graph and matching algorithms.
2. Understand and design algorithms using greedy strategy, divide and conquer approach, dynamic programming, and max flow - min cut theory.
3. Analyze asymptotic runtime complexity of algorithms including formulating recurrence relations.
4. Have basic knowledge of computational complexity, approximation and randomized algorithms.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-206												
CO 1.	H	L										
CO 2.		H	H	L	M			M	M			
CO 3.	M	H	M	M	H	M	M				M	
CO 4.	L	M	M	H								

TOPICS COVERED

Basic Principles of Algorithm Design and Analysis

The Divide and Conquer Method: Overall technique, mergesort, quicksort, quickselect, FFT, etc.

The Greedy Method: Overall technique, the knapsack problem, optimal merge pattern, Huffman coding, minimum spanning tree, single-sourse shortest paths problem, etc.

Dynamic Programming: Overall technique, matrix chain problem, all-pairs shortest path problem, optimal binary search trees, etc.

Graph Traversal Techniques: Tree traversal and applications, depth-dirst search, bread-first search, connectivity algorithms, biconnectivity algorithms, etc.

Backtracking: Overall technique, generation of combinatorial objects such as graphs, sets, permutations, graph colorings, cliques, Hamiltonian cycles, etc.

Branch and Bound method: Overall method, the 0/1 knapsack problem, the job assignment problem, the traveling salesman problem, etc.

Lower bound theory: Techniques for determining complexity lower bounds of problems, algorithm modeling, application to lower bound on sorting, searching, and merging.

Introduction to the Theory of NP-completeness: Nondeterministic algorithms, complexity classes, NP-completeness, problem reduction, Specific NP-complete problems.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Introduction to Algorithms by Cormen, leiserson and Rivest, McGraw-Hill, Latest Edition.
2. Design and Analysis of Algorithms by SartajSahni and Ellis Horwitz, Galgotia Publications.
3. Design and Analysis of Algorithms by Ullman and Hopcroft, Pearson Education.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC -208
COURSE TITLE: COMPUTER ORGANIZATION & ARCHITECTURE
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design.
2. Identify where, when and how enhancements of computer performance can be accomplished.
3. Learn the sufficient background necessary to read more advance texts as well as journal articles on the field.
4. See how to use concepts of computer organization in real-life settings using various PC performance improvements.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC -208												
CO 1.	M	L	M	H			M					
CO 2.	M	L	L	M	M	M	H					
CO 3.	H			M		L					H	
CO 4.	L	M				H					M	

TOPICS COVERED

Introduction: Historical overview, economic trends, underlying technologies, Data Representation- Data Types, Complements. Fixed-Point Representation, Floating-Point Representation. Error Detection and Correction, Addition, Subtraction, Multiplication and Division algorithms and hardware.

Register Transfer and Micro operations: Register transfer language, Inter-Register Transfer, Arithmetic Micro-operations, Logic and Shift micro-operations Language, Control functions.

Arithmetic Logic Unit: Arithmetic, logic and shift micro operations. Constructing an arithmetic logic shift unit.

Basic Computer Architecture and Design: Computer registers, Computer Instructions-Instruction Set Completeness. Classifying Instruction Set Architecture. Basic steps of Instruction Execution, Hardwired Control, Micro programmed Control. Horizontal and Vertical Microprogramming. Interrupts.

Central Processing Unit: General Register Organization. Stack Organized CPU. Instruction Formats, Addressing Modes. Data Transfer and Manipulation. RISCVs CISC.

Pipelining: Parallel and pipeline Processing, Pipeline Control, Pipeline Implementations, Conflicts Resolution, and Pipeline Hazards. Vector Processing, and Array Processors.

Memory Organization: Memory Systems: principle of locality, principles of memory hierarchy Caches, associative memory, main memory, Virtual memory, Paging and Segmentation, Memory Interleaving.

Input Output Organization: I/O performance measures, types and characteristics of I/O devices, I/O Modes-Programmed I/O, Interrupt Initiated I/O and DMA. Buses: connecting I/O devices to processor and memory, interfacing I/O devices to memory, processor, and operating system.

Parallel Computers: Classification, SIMD, MIMD Organizations, Connection Networks, Data Flow Machines, and Multithreaded Architectures.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. M Moris Mano, "Computer System Architecture", Pearson Education, 3rd Edition 1993.
2. David A. Patterson and John L. Hennessy, "Computer Organization & Design-The Hardware/Software Interface", Morgan Kaufmann, 2nd Edition 1997.
3. William Stallings, "Computer Organisation and Architecture, Designing for Performance", Pearson Education Asia, 6th Edition 2003.
4. Harry F. Jordan and Gita Alaghband, "Fundamentals of Parallel Processing", Pearson Education, 1st Edition 2003.
5. Barry Wilkinson Michael Allen, "Parallel Programming", prentice hall, 1999.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: ECPC-252
COURSE TITLE: MICROPROCESSOR AND MICROCONTROLLER
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand the architecture and organization of microprocessor along with instruction coding formats.
2. Understand, write structured and well-commented programs in assembly language with an ability to test and debug them in the laboratory.
3. Understand software/ hardware interrupts and further writes programs to perform I/O using handshaking and interrupts.
4. Understanding of digital interfacing and system connections.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ECPC-252												
CO 1.	H		H	M								
CO 2.	M	M	H	M				H				
CO 3.	L		M									
CO 4.			M	L								

TOPICS COVERED

INTEL 8086 Microprocessor: Pin Functions, Architecture, Characteristics and Basic Features of Family, Segmented Memory, Addressing Modes, Instruction Set, Data Transfer Instructions, Arithmetic, Logical, Shift and Rotate Instructions, String Instructions, Flag Control Instructions, Transfer of Control Instructions, Processor Control Instructions, Programming Examples, Interrupt Structures, Multitasking and Multiprogramming, MIN/MAX Modes of 8086,Co-processors 8087 and 8089.

Interrupts: 8086 Interrupts and Types, 8254 Software-Programmable Timer/Counter, 8259A Priority Interrupt Controller, Software Interrupt Applications.

Digital and Analog Interfacing: Programmable Parallel Ports and Handshake Input/output, Interfacing Keyboards and Alphanumeric Displays, Interfacing Microcomputer Ports to Devices, Developing the Prototype of a Microcomputer Based Instrument

Memories, Coprocessors, and EDA Tools: 8086 Maximum Mode and DMA Data Transfer, Interfacing and Refreshing Dynamic RAMs, The 8087 Math Coprocessor, Computer Based Design and development Tools

Introduction to 8051 Microcontroller : 8051-architecture and pin diagram, Registers, Timers Counters, Flags, Special Function Registers, Addressing Modes, Data types, instructions and programming, Single –bit operations, Timer and Counter programming, Interrupts programming, Serial communication, Memory accessing and their simple programming applications.

Hardware interfacing: I/O Port programming, Bit manipulation, interfacing to a LED, LCD, Keyboard, ADC, DAC, Stepper Motors and sensors.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Hall Douglas V, “Microprocessors and Interfacing”, Tata McGraw-Hill 1989.
2. Berry B Brey ,“The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386 And 80486, Pentium and Pentium ProProcessor Architecture, Programming and Interfacing”, Pearson Education 2003.
3. MathurAditya P, “Introduction to Microprocessors” Tata McGraw-Hill 1989.
4. Ray A Kbhurchandi, K M, “Advanced microprocessors and peripherals“,Tata McGraw Hill 2000.
5. James L Antonakos, “An Introduction to the Intel Family of Microprocessors: A Hands-On Approach Utilizing the 80x86 Microprocessor Family”, First Edition**.** Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: HMCI-201
COURSE TITLE: ECONOMICS FOR ENGINEERs
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

Course Objectives

1. To gain knowledge about different micro and macro aspects of economics.
2. To understand the complex relationships and intricacies of different economic variables.

3. To understand the micro and macro implications of economic policies and decisions.

Course Outcomes

The students will able to understand different terms and concepts of economics. The students will gain proficiency in understanding the changes in economic environment and their impact both at micro and macro levels.

Course Contents

Basic Economic concepts, Decision making under risk and uncertainty. Concept of utility, demand and supply, elasticity of demand and supply, Demand forecasting. Production function in short and long run: law of diminishing marginal returns, isoquant-isocost approach. Economies of scale. Shapes of different cost curves in short and long run. Price-output determination in perfect competition, monopoly, monopolistic competition and oligopoly. Macroeconomics: national income, business cycle, fiscal policy, monetary policy, price indices, inflation, theories of international trade.

Reference Books

1. Carl E Case, Ray C Fair and Sharon E Oster (2017), Principles of Economics, Pearson
2. John Sloman, Dean Garratt and Alison Wride (2014), Economics, 9th edition, Pearson.
3. Christopher R Thomas, S Charles Maurice and Sumit Sarkar (2010), Managerial Economics, 9th edition, McGraw Hill Publication.
4. H L Ahuja (2017), Managerial Economics, 9th edition, S Chand Publishing.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-222
COURSE TITLE: DATABASE MANAGEMENT SYSTEM LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student learning outcomes include:

1. Create database with different types of integrity constraints and use the SQL commands such as DDL, DML, DCL, TCL to access data from database objects.
2. Use database security & authorization in order to access database for the different kinds of the user.
3. Access and manipulate data using PL/SQL blocks.
4. Connect database to front end using JDBC and ODBC driver

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-222												
CO 1.	M	M	H	M	L							
CO 2.					M	H						
CO 3.	L	M		M								
CO 4.			L	M	H		M		M			

LIST OF PRACTICALS

1. Write SQL queries for Data Definition and Data Manipulation language.
2. Write SQL queries using Logical operators (=,<,>,etc.).
3. Write SQL queries using SQL operators (Between…. AND, IN(List), Like, ISNULL and also with negating expressions).
4. Write SQL query using character, number, date and group functions.
5. Write SQL queries for Relational Algebra (UNION, INTERSECT, and MINUS, etc.).
6. Write SQL queries for extracting data from more than one table (Equi-Join, Non-Equi-Join , Outer Join)
7. Write SQL queries for sub queries, nested queries.
8. Implement VIEWS, CURSORS, and TRIGGRS & write ASSERTIONS.
9. Implement FORMS and REPORTS.
10. Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
11. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.
12. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
13. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
14. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.
15. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
16. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
17. Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
18. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
19. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

* Students are advised to use **Developer 2000/Oracle-10i** orhigherversion or other latest version for above listed experiments. However depending upon the availability of software's, students may use **Power Builder /SQL SERVER**. Mini Project may also be planned & carried out throughout the semester to understand the important various concepts of Database.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-224
COURSE TITLE: MACHINE LEARNING LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the students will be able to:

1. Hands on knowledge of methods and theories in the field of machine learning for learning principles, techniques, and applications of machine learning.
2. Implement decision tree learning, bayesian learning and artificial neural network for real world problems.
3. Design and implement various classification techniques.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-224												
CO 1.			H	H								
CO 2.					H		M					
CO 3.			H	M	M				M			

LIST OF PRACTICALS

1. Study and Implement the Naive Bayes learner using WEKA. (The datasets taken can be: Breast Cancer data file or Reuters data set).
2. Study and Implement the Decision Tree learners using WEKA. (The datasets taken can be: Breast Cancer data file or Reuter¨s data set).
3. Estimate the accuracy of decision classifier on breast cancer dataset using 5-fold cross-validation. (You need to choose the appropriate options for missing values).
4. Estimate the precision, recall, accuracy, and F-measure of the decision tree classifier on the text classification task for each of the 10 categories using 10-fold cross-validation.
5. Develop a machine learning method to classifying your incoming mail.
6. Develop a machine learning method to Predict stock prices based on past price variation.
7. Develop a machine learning method to predict how people would rate movies, books, etc.
8. Develop a machine learning method to Cluster gene expression data, how to modify existing methods to solve the problem better
9. Select two datasets. Each dataset should contain examples from multiple classes. For training purposes assume that the class label of each example is unknown (if it is known, ignore it). Implement the Kmeans algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameter k.
10. Implement the EM algorithm assuming a Gaussian mixture. Apply the algorithm to your datasets and report the parameters you obtain. Evaluate performance by measuring the sum of Mahalanobis distance of each example from its class center. Test performance as a function of the number of clusters.
11. Suggest and test a method for automatically determining the number of clusters. Using a dataset with known class labels compare the labeling error of the K-means and EM algorithms. Measure the error by assigning a class label to each example. Assume that the number of clusters is known.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-226
COURSE TITLE: DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment**.**

COURSE OUTCOMES
After the course completion, the student will be able to:

1. Learn how to analyze algorithms and estimate their worst case and average case behavior.
2. Analyze fundamental data structures with the manner in which these data structures can be implemented.
3. Applying algorithm design strategies, sorting and searching techniques and solution of relevant recurrence relations to real life applications.
4. Design and implement optimization algorithms in specific applications.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-226												
CO 1.	H	L										
CO 2.		H	H	L	M			M	M			
CO 3.	M	H	M	M	H	M	M				M	
CO 4.	L	M	M	H								

LIST OF PRACTICALS

1. Program to implement Quick sort using the Divide and Conquer technique and analyze its Time Complexity.
2. Program to implement Merge sort using the Divide and Conquer technique and analyze its Time Complexity.
3. Program to perform Binary Search using the Divide and Conquer technique and analyze its Time Complexity.
4. Program to implement Starssen's Matrix Multiplication Algorithm and analyze its Time Complexity.
5. Program to find the Minimum Spanning tree using prim's algorithm.
6. Program to find the Minimum Spanning tree using kruskal's algorithm.
7. Program to solve the knapsack problem using greedy method.
8. Program to find the shortest path of the multistage graph using dynamic programming.
9. Program to solve the Traveling salesman problem using the dynamic programming approach.
10. Program to solve the Optimal Binary Search Tee problem using the dynamic programming approach.
11. Program to find the solution to the N queen's problem using backtracking.
12. Program to find the shortest path using Floyd's algorithm.
13. Program to solve Graph Coloring problem.
14. Program to solve Hamiltonian Cycle Problem.
15. Program to implement Knuth Morris Pratt algorithm and analyze its time complexity

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: ECPC-272
COURSE TITLE: MICROPROCESSOR AND MICROCONTROLLER LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After the course completion, the students will be able to:

1. Understand and gain the knowledge of computer hardware.
2. Configure computer systems and do various port programming.
3. Do assembly language programming for 8085, 8086 microprocessor.
4. Implement TSR programming with interrupt.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ECPC-272												
CO 4.	M	L										
CO 5.		L	M		M							
CO 6.			M	M	M				M			
CO 7.			M	H	M							

LIST OF PRACTICALS

I. Microprocessor 8086:

1. Arithmetic operation – Multi byte Addition and Subtraction, Multiplication and Division – Signed and unsigned Arithmetic operation, ASCII – arithmetic operation.
2. Logic operations – Shift and rotate – Converting packed BCD to unpacked BCD, BCD to ASCII conversion.
3. By using string operation and Instruction prefix: Move Block, Reverse string, Sorting, Inserting, Deleting, Length of the string, String comparison.
4. DOS/BIOS programming: Reading keyboard (Buffered with and without echo) – Display characters, Strings.

II. Interfacing:

1. 8259 – Interrupt Controller: Generate an interrupt using 8259 timer.
2. 8279 – Keyboard Display: Write a small program to display a string of characters.
3. 8255 – PPI: Write ALP to generate sinusoidal wave using PPI.
4. 8251 – USART: Write a program in ALP to establish Communication between two processors.

III. Microcontroller 8051

1. Reading and Writing on a parallel port.
2. Timer in different modes.
3. Serial communication implementation.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Subject to be offered to ECE Department

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-212
COURSE TITLE: DATABASE MANAGEMENT SYSTEM
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 2-0-0-2)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests, which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to:

1. To understand the different issues involved in the design and implementation of a database system. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
2. To understand and use data manipulation language to query, update, and manage a database
3. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-212												
CO 1.	M		H		M	L						
CO 2.			H	L	M							
CO 3.		M	H	L	H	L		H	M			M

TOPICS COVERED

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model:
ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL. Advantage of SQL.SQL data types and literals.Types of SQL commands.SQL operators and their procedure.Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Crash Recovery: Failure classification, recovery concepts based on deferred update, recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates.

Integrity, Security and Repositories: Needs for database integrity, integrity constraints, non-procedural integrity constraints, integrity constraints specifications in SQL, introduction to database security mechanism, security specification in SQL, system catalogues

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Date C J, "An Introduction To Database System", Addision Wesley
2. Korth, Silbertz, Sudarshan, "Database Concepts", McGraw Hill
3. Elmasri, Navathe, "Fundamentals Of Database Systems", Addision Wesley
4. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication
5. Rob and Coronel, "Database Systems 5th Edition",Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-214
COURSE TITLE: OPERATING SYSTEMS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 2-0-0-2)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the students would be able to:

1. Analyze the working of an operating system and its components.
2. Identify the working methodology of various subcomponents of the operating system.
3. Compare and analyze different approaches for memory management techniques.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-214												
CO 1	L	H	L	M								
CO 2		M		H	M							
CO 3		M		M	H							M

TOPICS COVERED

Operating systems objectives, services and functions: Characteristics of Modern Operating Systems, Characteristics of Batch and multiprogramming operating systems. Comparisons between real time systems and time-sharing systems, Operating system services and kernel features.

I/O management, I/O devices: Introduction to I/O management, I/O devices, Concepts of threading, Organization of I/O functions, polling, various modes of data transfer, Hardware/Software interface, I/O buffering.

Processes and Process scheduling policies: Introduction to processes management, operating system views of processes, various process transition states, Introduction to Processor scheduling, Introduction to various types of schedulers, Performance criteria in scheduling algorithms, Concept of FCFS scheduling algorithm, Concept of priority scheduling algorithm like SJF, Concept of non-preemptive and preemptive algorithms, Concept of round-robin scheduling algorithm, , Concept of multi-level queues, feedback queues.

Memory Management: Need of Memory management and its requirements, paging, segmentation, concept of fragmentation. Characteristics of contiguous & non-contiguous allocation techniques, Detail study of fragmentation, Virtual memory management, introduction to page-replacement, Need of various page-replacement policies, Concept of FIFO and optimal page-replacement algorithms, Concept of LRU approximation and its page-replacement algorithm, Concept of allocation algorithms.

File management System: Need of file management, its requirements, User's and operating system's view of file system, Concept of file directories and file sharing, secondary memory management**.**

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Peterson and Silberschatz, "Operating System Concepts", Addison-Wesley 6th Edition 2006.
2. Milenkoviac, "Operating Systems Concepts and Design", Tata McGraw-Hill 2005.
3. Andrews S. Tanenbaum, "Modern Operating Systems", Pearson Education, 3nd edition 2016.
4. W Stallings, " Opertating Systems" Pearson, 6th Edition 2012
5. Dhamdhere, D M, "Introduction to Systems Software", Tata Mc-Graw Hill 2015
6. N Chauhan, "Principles of Operating Systems" 1st ed. 2018

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-232
COURSE TITLE: OPERATING SYSTEMS LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE

CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)

COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After the course completion, the student would be able to:

1. Analyse the working of an operating system and its components
2. Compare and analyse different file systems being used in different operating systems.
3. Design and implement system-level applications for open-source operating systems

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-232												
CO 1	M	H		M				M	M			
CO 2		M		H	M							
CO 3		M		M	H	M	M					

.

LIST OF PRACTICALS

1. Simulation of the CPU scheduling algorithms such as Round Robin, SJF, FCFS, Priority
2. Simulation of page Replacement Algorithms such as FIFO, LRU, and Optimal
3. Simulation of paging techniques of memory management.
4. Simulation of file allocation Strategies like Sequential, Indexed, Linked
5. Simulation of file organization techniques such as Single Level Directory, Two Level, Hierarchical, DAG
6. To automate the allocation of IP addresses i.e. to set and configure the DHCP server and DHCP client.
7. To share files and directories between RedHat Linux operating systems i.e. to set and configure the NFS server and NFS clients.
8. To share files and directories between Red Hat Linux and Windows operating systems i.e. to set and configure the samba server.
9. To set and configure the DNS (Domain Name Server).
10. To set and configure the print server and to share printers between Windows and Red Hat Linux operating systems.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Subject to be offered to BT Department

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-203
COURSE TITLE: OBJECT ORIENTED PROGRAMMING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES

After the completion of the course, the students will be able to:

1. Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
2. Understand fundamentals of object-oriented programming, including defining classes, invoking methods, using class libraries, etc.
3. Have the ability to write a computer program to solve specified problems.
4. Be able to use the OOP environment to create, debug and run simple C++ programs.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-203												
CO 1.	M	M	H	L	L	L						
CO 2.	M	M	H	L	L							
CO 3.	L	M	H	M					M			
CO 4.		L	H	L				M	M			

TOPICS COVERED

Object oriented thinking: Need for OOP Paradigm, Procedural programming vs object oriented programming, object oriented concepts.

Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, function overloading, Math library functions.

Class: Difference between C structure and class, specifying a class, Defining member functions: inside and outside class, scope resolution operator, Array within a class, array of objects, Static data members and member functions, Object as function arguments, returning objects, Friend function, memory allocation for objects ,pointer to members, pointer to object, this pointer local classes.

Constructor and destructor: Constructor, types of constructors: default, parameterized and copy constructor, constructor overloading, constructor with default parameter, dynamic initialization of objects, destructor
Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator,Data Conversion: Basic to User Defined , User defined to basic, Conversion from one user-defined to other.

Inheritance and polymorphism: Base class, derived class, visibility modes, derivation and friendship, Types of inheritance, Containership, virtual function binding, pure virtual functions, Abstract class, pointer to derived class.

Console IO operations: C++ stream classes, Unformatted IO operations, formatted IO operations, managing output with manipulators.
Working with files: Classes for file stream operations, opening and closing files, File opening modes, file Pointers, Error handling during file operations, command line arguments. Templates: Class template, class template with parameter, function template, function template with parameter.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. BjraneStroustrup, “C++ Programming language”,3rd edition, Pearson education Asia(1997)
2. LaforeR.”Object oriented Programming in C++”, 4th Ed. Techmedia,New Delhi(2002).
3. YashwantKenetkar,”Let us C++”,1stEd.,Oxford University Press(2006)
4. B.A. Forouzan and R.F. Gilberg,CompilerScience,”A structured approach using C++” Cengage Learning, New Delhi.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-223
COURSE TITLE: OBJECT ORIENTED PROGRAMMING LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-1-2)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Gain understanding about the object oriented principles in construction of robust and maintainable programs.
2. Have a competence to design, write, compile, test and execute programs using high level language.
3. Have an awareness of the need for a professional approach to design and the importance of good documentation to finish.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-223												
CO 1.	H		M	M								
CO 2.	L	H	H	M								
CO 3.				H		M		H	M	H		M

LIST OF PRACTICALS

1. Write a program to read a matrix of size m x n form the keyboard and display the same using function.
2. Write a Program to make the use of inline function.
3. Write a function power () which raise a number m to a power n. The function takes double value of m and integer value of n and returns the result. Use a default value of n is 2 to make the function to calculate squares when this argument is omitted.
4. Program to show that the effect of default arguments can be alternatively achieved by overloading.
5. Write a class ACCOUNT that represents your bank account and then use it.
6. The class should allow you to deposit money, withdraw money, calculate interest, send you a message if you have insufficient balance.
7. Write a class STRING that can be used to store strings, add strings, equate string, output strings.
8. Create the class TIME to store time in hours and minutes. Write a friend function to add two TIME objects.
9. Create two classes DM and DB. DM stores the distance in meter and centimeters and DB stores the distance in feet and inches. Write a program two add object of DM with the object of DB class.
10. Write a program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes inherits the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
11. Write Programs to demonstrate the concept of Default constructor, Parameterized constructor, Copy constructor, and Constructor overloading
12. Program to demonstrate the concept of destructor, multiple inheritance, multilevel inheritance, hybrid inheritance, and concept of containership.
13. Program to overload unary operator and overload binary operator
14. Program to show the concept of run time polymorphism using virtual function.
15. Program to work with formatted and unformatted IO operations.
16. Program to read the name and roll numbers of students from keyboard and write them into a file and then display it.
17. Program to copy one file onto the end of another, adding line numbers
18. Write a function template for finding the minimum value contained in an array.
19. Write a class template to represent generic vector (a series of float values). Include member function to perform following tasks.
    e. Create vector
    f. Modify the value of a given element
    g. To multiply by a scalar value
    h. To display vector in the form of (10, 20, 30,……)

This is only the suggested list of Practical's. Instructor may frame additional Practicals relevant to the course contents.

Subject to be offered to ICE Department

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-213
COURSE TITLE: DATA STRUCTURES AND ALGORITHMS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand the concepts of data structure, data type and array data structure.
2. Implement linked list data structure to solve various problems.
3. Understand and apply various data structure such as stacks, queues, trees and graphs to solve various computing problems using C-programming language.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-213												
CO 1.	M	L	H	L								
CO 2.		M	M									
CO 3.		M	H	L	L							

TOPICS COVERED

Introduction: Basic Terminology, Elementary Data Organization, Structure operations.

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C++, Character string operation, Array as Parameters, Ordered List, Sparse Matrices and Vectors.

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, and Application of stack: Conversion of Infix to prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Recursion: Recursive definition and processes, recursion in C, example of recursion, Tower of Hanoi Problem, simulating recursion, Backtracking, recursive algorithms, principles of recursion, tail recursion, removal of recursion.

Queues: Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues**.**

Linked list: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list, Garbage Collection and Compaction.

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees**.**

Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting**.**

Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., New Delhi.
2. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education Asia, Delhi-2002
3. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., New Delhi.
4. Bruno R Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", Jhon Wiley & Sons, Inc.
5. GilbergForozan , "Data Structure – A pseudo code approach with C++", Cengage Learning, New Delhi.

V SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-301	Theory of Computation	3	1	0	4	PC
2.	CSPC-303	Operating Systems	3	0	0	3	PC
3.	CSPC-305	Software Engineering	3	0	0	3	PC
4.	CSPC-307	Information Security System	3	0	0	3	PC
5.	CSPC-309	Probability Theory for Data Analytics	3	1	0	4	PC
6.	CSPE-3XX	DE-I	3	0	0	3	PE
7.	CSPC-323	Operating Systems Laboratory	0	0	2	1	PC
8.	CSPC-325	Software Engineering Laboratory	0	0	2	1	PC
9.	CSPC-327	Information Security Laboratory	0	0	2	1	PC
10.	CSPE-3XX	DE-I Laboratory	0	0	2	1	PE
11.	CSCI-301	Minor Project, Phase-I	0	0	2	0*	CIC
TOTAL			**18**	**2**	**10**	**24**	

DEPARTMENTAL ELECTIVE (DE)-I

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-331	Advanced Programming Concepts using Java	3	0	0	3
2.	CSPE-333	Natural Language Processing	3	0	0	3
3.	CSPE-335	Web Technologies	3	0	0	3
4.	CSPE-351	Advanced Programming Concepts using Java Laboratory	0	0	2	1
5.	CSPE-353	Natural Language Processing Laboratory	0	0	2	1
6.	CSPE-355	Web Technologies Laboratory	0	0	2	1

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-301
COURSE TITLE: THEORY OF COMPUTATION
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand theoretical foundations of computer science.
2. Master regular languages, finite automata, pushdown automata, Turing recognizable Languages.
3. Employ finite stated machines to solve problems in computing.
4. Think analytically and intuitively for problem-solving situations in related areas of theory in computer science.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-301												
CO 1	H		M									
CO 2	M			H			M					
CO 3	M		H	H	M		M		M			
CO 4	M		M	H			M		M			

TOPICS COVERED

Basics in Theory of Computations: Basic concepts of strings, alphabets, languages, Principles of Mathematical Induction.

Languages and Grammars: Construct of a language, Grammar, Chomsky Classification of Formal Languages.

Finite Automata: Automata and Applications of Automata Theory, Deterministic and Non-Deterministic FA, Comparison and Equivalence of DFA and NFA.

Regular Expressions: Regular Expression, Equivalence of Regular Expression and Finite Automata, Equivalence of Regular Grammar and Finite Automata, Regular and Non- Regular Languages, Pumping Lemma for Regular Sets

Finite State Machines: Moore and Mealy Machines, Equivalence of Moore and Mealy Machines.

Context Free Language: Context Free Grammar, Derivation trees, Context Free Grammar Simplification, Chomsky & Greibach Normal forms, Ambiguities.

Pushdown Automata: Definition, Equivalence of PDA by Empty Store and PDA by Final State. Construction of PDA for CFLs.

Turing Machines: Introduction and Turing Machine Model, Computable functions and languages. Techniques for construction of Turing machines, Church's Hypothesis.

Undecidability: Recursive and recursively enumerable languages, Rice theorem, Post's correspondence problem.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. J E HopcroftAnd J D Ullman, "Introduction to Automata Theory, Languages and Computation", Narosa Publishers 2002.
2. K L P Mishra and N Chandrasekaran, "Theory of Computer Science", Prentice Hall Inc, .2002
3. Harry R Lewis and Chritos H Papadimitriou, "Elements of the Theory of Computation", Pearson Education 2001.
4. Peter Linz, "An Introduction to Formal Languages and Automata", Narosa Publishers 2002.
5. Michael Sipser, "Introduction to the theory of computation ", Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-303
COURSE TITLE: OPERATING SYSTEMS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)

COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the students would be able to:

1. Analyze the working of an operating system and its components.
2. Define and analyze the synchronization in processes.
3. Identify the working methodology of various subcomponents of the operating system.
4. Compare and analyze different approaches for memory management techniques.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-303												
CO 1	L	H	L	M								
CO 2		M	L	L								
CO 3		M		H	M							
CO 4		M		M	H							M

TOPICS COVERED

Operating systems objectives, services and functions: Characteristics of Modern Operating Systems, Characteristics of Batch and multiprogramming operating systems. Comparisons between real time systems and time-sharing systems, Operating system services and kernel features.

I/O management, I/O devices: Introduction to I/O management, I/O devices, Concepts of threading, Organization of I/O functions, polling, various modes of data transfer, Hardware/Software interface, I/O buffering.

Processes and Process scheduling policies: Introduction to processes management, operating system views of processes, various process transition states, Introduction to Processor scheduling, Introduction to various types of schedulers, Performance criteria in scheduling algorithms, Concept of FCFS scheduling algorithm, Concept of priority scheduling algorithm like SJF, Concept of non-preemptive and preemptive algorithms, Concept of round-robin scheduling algorithm, , Concept of multi-level queues, feedback queues.

Concurrency control schemes: Various approaches to concurrency control schemes, Concept of prouder/consumer problem, Mutual Exclusion, Concept of mutual exclusion first and second algorithm, Concept of mutual exclusion third algorithm including introduction and characteristics of semaphores, Introduction to Mutual exclusion with semaphores, Introduction to Interprocess Communication and Synchronization, Critical regions and Conditional critical regions in a Semaphore. Introduction to monitors, various modes of monitors.

Dead Locks: Concept of Deadlocks, issues related to its prevention, avoidance and detection / recovery, Concept of deadlock prevention and its avoidance, Concept of deadlock detection and recovery.

Memory Management: Need of Memory management and its requirements, paging, segmentation, concept of fragmentation. Characteristics of contiguous & non-contiguous allocation techniques, Detail study of fragmentation, Virtual memory management, introduction to page-replacement, Need of various page-replacement policies, Concept of FIFO and optimal page-replacement algorithms, Concept of LRU approximation and its page-replacement algorithm, Concept of allocation algorithms.

File management System: Need of file management, its requirements, User's and operating system's view of file system, Concept of file directories and file sharing, secondary memory management**.**

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Peterson and Silberschatz, "Operating System Concepts", Addison-Wesley 6th Edition 2006.
2. Milenkoviac, "Operating Systems Concepts and Design", Tata McGraw-Hill 2005.
3. Andrews S. Tanenbaum, "Modern Operating Systems", Pearson Education, 3nd edition 2016.
4. W Stallings, " Opertating Systems" Pearson, 6th Edition 2012
5. Dhamdhere, D M, "Introduction to Systems Software", Tata Mc-Graw Hill 2015
6. N Chauhan, "Principles of Operating Systems" 1st ed. 2018

. **DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING**
COURSE CODE: CSPC-305
COURSE TITLE: SOFTWARE ENGINEERING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests, which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, students will be able to:

1. Analysis of different software development process models.
2. Extract and analyze software requirements specifications for different projects.
3. Develop some basic level of software architecture/design and apply standard coding practices.
4. Apply different testing and debugging techniques and analyze their effectiveness.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-305												
CO 1.	M	H				L						
CO 2.		H			L					H		H
CO 3.	H		H		M							
CO 4.			L	H	M				M			H

TOPICS COVERED

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, the unified process.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

Design Engineering: Design process and Design quality, Design concepts, the design model.

Creating an architectural design: Software architecture, Data design, Architectural styles and patterns,

Architectural Design.

Object-Oriented Design: Objects and object classes, An Object-Oriented design process, Design evolution.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modelling Language (UML), Hands on practice of CASE tools.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Software Engineering- K.K. Agarwal&Yogesh Singh, New Age International Publishers
2. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiely.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
4. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition. McGrawHill International Edition.
5. Software Engineering- Sommerville, 7th edition, Pearson education.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-307
COURSE TITLE: INFORMATION SECURITY SYSTEM
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. To be familiar with information security awareness and a clear understanding of its importance.
2. To be familiar with how threats to an organization are discovered, analyzed, and dealt with.
3. To be familiar with network security threats and countermeasures and network security designs using available secure solutions (such As PGP, SSL, IPSec, etc).
4. To be exposed to original research in network security.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-307												
CO 1	M		M			H		M		M		
CO 2	M	H	M	M		H		H				
CO 3			H	M	H	M		H	H			
CO 4				M	M				M		M	

TOPICS COVERED

Overview: Services, Mechanisms, and Attacks, the OSI Security Architecture, A Model for Network, Security.

Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Stegnography.

Block Ciphers And The Data Encryption Standard: Simplified DES, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles, Block Cipher Modes of Operation.

Introduction To Finite Fields: Groups, Rings, and Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields of the Form GF(p), Polynomial Arithmetic, Finite Fields of the Form GF(2n).

Advanced Encryption Standard: Evaluation Criteria for AES, The AES Cipher.

Contemporary Symmetric Ciphers: Triple DES, Blowfish, RC5, Characteristics of Advanced Symmetric Block Ciphers, RC4 Stream Cipher.

Confidentiality Using Symmetric Encryption: Placement of Encryption Function, Traffic Confidentiality, Key Distribution, Random Number Generation.

Public-Key Encryption and Hash Functions: Introduction to Number Theory: Prime Numbers, Format's and Euler's Theorems, Testing for Primality, The Chinese Remainder Theorem, Discrete Logarithms.

Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, the RSA Algorithm, Recommended Reading and Web Site, Key Terms, Review Questions, and Problems.

Key Management and Other Public-Key Cryptosystems: Key Management, Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.

Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and MACs.

Hash Algorithms: MD5 Message Digest Algorithm, Secure Hash Algorithm, RIPEMD-160, and HMAC.

Digital Signatures and Authentication Protocols: Digital Signatures, Authentication Protocols, Digital Signature Standard.

System Security: Intruders: Intruders, Intrusion Detection, Password Management, Malicious Software: Viruses and Related Threats, Virus Countermeasures, Firewalls: Firewall Design Principles, Trusted Systems.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Prakash C. Gupta, Cryptography and Network Security, PHI, 1st ed., 2015
2. B. A. Forouzan, D Mukhopadhyay, Cryptography and Network Security, TMH, 3rd 2015
3. William Stallings, "Cryptography and network Security", Pearson Education 2015.
4. Atul Kahate, Cryptography and Network Security, TMH, 2nd ed., 2008

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC -309
COURSE TITLE: PROBABILITY THEORY FOR DATA ANALYTICS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Ability to conceptualize the necessity of randomness concept in practical situation
2. Ability to approximate the real problems using stochastic process and deduce results
3. Ability to deduce useful results and interpret them based on the analysis of queuing theory

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC -309												
CO 1.	H	M			M		M					
CO 2.	M	H		M	M	M			H			
CO 3.	H			M		L						

TOPICS COVERED

Probability: The concept of probability, The axioms of probability, Some important theorems on Probability, Assignment of Probabilities, Conditional Probability, Theorems on conditional probability, Independent Event's, Bayes' Theorem.

Random Variables and Probability Distributions: Random variables, Discrete probability distributions, Distribution functions for discrete random variables, Continuous probability distribution, Distributions for continuous random variables, joint distributions, Independent random variables.

Mathematical Expectation: Definition, Functions of random variables, some theorems on Expectation, The variance and Standard Deviation, Moments, Moment Generating Functions, Covariance, Correlation Coefficient.

Special Probability Distributions: The Binomial Distribution, The Normal Distribution, The Poisson Distribution, Relations between different distributions, Central limit theorem, Uniform distribution, Chi-square Distribution, Exponential distribution.

Sampling Theory: Population and Sample, Sampling with and without replacement, the sample mean, Sampling distribution of means, proportions, differences and sums, the sample variance, the sample distribution of variances.

Tests of Hypotheses and Significance: Statistical Decisions, Statistical hypotheses, Null Hypotheses, Tests of hypotheses and significance, Type I and Type II errors, level of significance, Tests involving the Normal distribution, One-Tailed and Two-tailed tests, Special tests of significance for large and small samples, The Chi-square test for goodness of fit.

Curve Fitting Regression and Correlation: Curve Fitting, The method of least squares, the least squares line, multiple regression, the linear correlation coefficient, Rank correlation, Probability interpretation of regression and correlation.

Discrete-Parameter Markov Chains: Introduction, Computation of n-step Transition Probabilities, State Classification and Limiting Distributions, Distribution of times between state changes, Irreducible finite chains with aperiode states, The M/G/1 Queuing System, Discrete-parameter, Birth-Death processes, Finite Markov chains with absorbing states.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Murray R. Spiegel, "Probability and Statistics", McGrawHll, Schaum's Outline Series
2. Kishor S Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Applications", Prentics Hall of India, 2000
3. A. Papoulis and S. UnnikrishnaPillai, "Probability, Random Variables and Stochastic Processes, McGraw Hill, 4th Edition
4. Richard A Johnson, Probability and Statistics for Engineers. Prenticshall, India, 2002.
5. Mondenhall, "Introduction to probability and statistics", Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-323
COURSE TITLE: OPERATING SYSTEMS LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE

CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)

COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Analyse the working of an operating system and its components and Define and analyse the synchronization process.
2. Identify the working methodology of multithreaded applications.
3. Compare and analyse different file systems being used in different operating systems.
4. Design and implement system-level applications for open-source operating systems

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-323												
CO 1	M	H		M				M	M			
CO 2		M	H	L								
CO 3		M		H	M							
CO 4		M		M	H	M	M					

.

LIST OF PRACTICALS

1. Installation and working with GCC on Linux
2. Simulation of the CPU scheduling algorithms such as Round Robin, SJF, FCFS, Priority
3. Simulation of MUTEX and SEMAPHORES.
4. Simulation of Bankers Deadlock Avoidance and Prevention algorithms.
5. Implementation of Process Synchronization (Reader-Writer, Sleeping Barber and Dining Philosopher's Problem)
6. Simulation of page Replacement Algorithms such as FIFO, LRU, and Optimal
7. Simulation of paging techniques of memory management.
8. Simulation of file allocation Strategies like Sequential, Indexed, Linked
9. Simulation of file organization techniques such as Single Level Directory, Two Level, Hierarchical, DAG
10. To automate the allocation of IP addresses i.e. to set and configure the DHCP server and DHCP client.
11. To share files and directories between RedHat Linux operating systems i.e. to set and configure the NFS server and NFS clients.
12. To share files and directories between Red Hat Linux and Windows operating systems i.e. to set and configure the samba server.
13. To set and configure the DNS (Domain Name Server).
14. To set and configure the print server and to share printers between Windows and Red Hat Linux operating systems.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-325
COURSE TITLE: SOFTWARE ENGINEERING LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Able to prepare SRS, design, test cases, software configuration management and risk management related document.
2. Develop function oriented and object oriented software design using tools like rational rose.
3. Implement unit testing and integration testing.
4. Able to track the progress of a project using Openproj or equivalent tool

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-325												
CO 1		H			L					H		H
CO 2	H		H		M							
CO 3			L	H	M				M			H
CO 4		M	M		L							

LIST OF PRACTICALS

1. Introduction and project definition.
2. Introduction to Project Management Software.
3. Software process overview.
4. Project planning.
5. Software Requirements Specification (SRS).
6. Introduction to UML and use case diagrams.
7. Flow of events and activity diagram.
8. OO analysis: discovering classes and class diagrams.
9. Interaction diagrams: sequence and collaboration diagrams.
10. Software Design: software architecture and object-oriented design.
11. Effort and cost estimation using COCOMO
12. State Transition Diagram.
13. Component and deployment diagrams.
14. Issue Tracking Systems like GitHub, BitBucket.
15. Software Configuration Management tools like CVS or SVN.
16. Software testing.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-327
COURSE TITLE: INFORMATION SECURITY LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end, which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understanding of various information security threats.
2. Understanding and implementation of security measures to encounter threats.
3. Implementation and understanding of encryption techniques.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-327												
CO 1.	M	M	H	M	L							
CO 2.					M	H						
CO 3.	L	M		M								

LIST OF PRACTICALS

1. Implementation of various transposition ciphers
2. Implementation of various substitution ciphers
3. Implementation of S-DES algorithm for data encryption
4. Implementation of S-AES algorithm for data encryption
5. Implementation of Symmetric Encryption Scheme – RC4.
6. Implementation of Chinese remainder theorem
7. Implementation of RSA cryptosystem.
8. Implementation of Rabin cryptosystem.
9. Implementation of ElGamal cryptosystem.
10. Implementation of various Secure Hash algorithm
11. Implementation of MAC algorithm.
12. Implementation of RSA digital signatures.
13. Implementation of Rabin digital signatures.
14. Implementation of ElGamal digital signatures.
15. Implementation of Diffie-Hallman key exchange algorithms

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENTAL ELECTIVE (DE) –I

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-331
COURSE TITLE: ADVANCED PROGRAMMING CONCEPTS USING JAVA
COURSE DESIGNATION: ELECTIVE

PRE-REQUISITES: NONE

CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)

COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Understand the concepts of advanced programming and practice on reusing components.
2. Learn principles and concepts of Object Orientation such as Abstraction, Data Hiding, Polymorphism.
3. Develop programs by using inbuilt libraries and importing Packages.
4. To be familiar with create and handle threads, interfaces and applets.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-331												
CO 1	H	M	M									
CO 2	M	H	H	M								
CO 3			H	H	H							

TOPICS COVERED

Overview of Basic OOP Concepts: Need for object-oriented paradigm: Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, datatypes, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, classes and objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling, inheritance, super keyword, polymorphism- method overriding, abstract classes.

Packages and Interfaces: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring packages – Java.io, Java.util.

Exception handling and multithreading: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes.The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – boarder, grid, flow, card and grib bag.

Applets: Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Swing: Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing-JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Networking: Basics of network programming, addresses, ports, sockets, simple client server program, multiple clients, Java .net package Packages – java.util,

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley& sons.
2. An Introduction to OOP, second edition, T. Budd, pearson education.
3. An introduction to Java programming and object oriented application development, R.A. Johnson-Thomson.
4. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, seventh Edition, Pearson Education.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-333
COURSE TITLE: NATURAL LANGUAGE PROCESSING
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES:
After the course completion, the student will be able to

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

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-333												
CO 1			M			H		M		M		
CO 2	M	H		M		H						
CO 3			H		H			H	H			
CO 4				M	M				M		M	

TOPICS COVERED

Introduction to Natural Language Understanding: The study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems, Linguistic Background: An outline of English syntax. Introduction to semantics and knowledge representation, Some applications like machine translation, database interface.

Grammars and Parsing: Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Transition Network Grammars, Top-Down Chart Parsing. Feature Systems and Augmented Grammars: Basic Feature system for English, Morphological Analysis and the Lexicon, Parsing with Features, Augmented Transition Networks.

Grammars for Natural Language: Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language, Handling questions in Context-Free Grammars. Human preferences in Parsing, Encoding uncertainty, Deterministic Parser.

Computational morphology: Lemmatization, Part-of-Speech Tagging, Finite-State Analysis.

Ambiguity Resolution: Statistical Methods, Probabilistic Language Processing, Estimating Probabilities, Part-of-Speech tagging, Obtaining Lexical Probabilities, Probabilistic Context-Free Grammars, Best First Parsing. Semantics and Logical Form, Word senses and Ambiguity, Encoding Ambiguity in Logical Form.

Application of NLP: Intelligent Work Processors: Machine Translation; User Interfaces; Man-Machine Interfaces: Natural language Querying Tutoring and Authoring Systems. Speech Recognition Commercial use of NLP.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. AksharBharti, VineetChaitanya and Rajeev Sangal, NLP: A Paninian Perspective, Prentice Hall, New Delhi
2. James Allen, Natural Language Understanding, 2/e, Pearson Education, 2003
3. D. Jurafsky, J. H. Martin, Speech and Language Processing, Pearson Education, 2002
4. L.M. Ivansca, S. C. Shapiro, Natural Language Processing and Language Representation
5. T. Winograd, Language as a Cognitive Process, Addison-Wesley

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-335
COURSE TITLE: WEB TECHNOLOGY
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand the HTML, XML DOM, JavaScript, CSS and applet codes
2. Design dynamic and interactive web pages by embedding Java Script code in HTML
3. Use Java Script to validate user input and create good, effective and customized websites

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-335												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						

TOPICS COVERED

Introduction to HTML: HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets.

Introduction to JavaScript: Scripts, Objects in Java Script, Dynamic HTML with Java Script
XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's

Web Servers and Servlets: Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, and Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues.

Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

JSP Application Development: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations

Database Access : Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Web Programming, building internet applications, Chris Bates 2nd edition, WILEY Dreamtech
2. The complete Reference Java 2 Fifth Edition by Patrick Naughton and Herbert Schildt. TMH (Chapters: 25)
3. Java Server Pages –Hans Bergsten, SPD O'Reilly.
4. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia
5. JoclSklar, "Web Warrier guide to web design technologies", Cengage Learning, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-351
COURSE TITLE: ADVANCED PROGRAMMING CONCEPTS USING JAVA LABORATORY
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Make use of members of classes using java API.
2. Demonstrate the ability to employ various types of selection constructs in a Java program.
3. Be able to employ a hierarchy of Java classes to provide a solution to a given set of requirements.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-351												
CO 1		H			L					H		H
CO 2	H		H			M						
CO 3		H	L		M				M			H

LIST OF PRACTICALS

1. a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
   b) Write a Java program to multiply two given matrices.
2. a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
   b) Write a Java program for sorting a given list of names in ascending order.
3. Write a java program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
4. a) Develop an applet that displays a simple message.
   b) Develop an applet that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
5. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.
6. Write a Java program for handling mouse events.
7. (a) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds. (b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
8. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the

Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

9. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)
10. a) Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No light is on when the program starts.
    b) Write a Java program that allows the user to draw lines, rectangles and ovals.
11. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are seperated by commas. Write a java program to display the table using JTable component.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-353
COURSE TITLE: NATURAL LANGUAGE PROCESSING LABRATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Implement compose key NLP elements to develop higher level processing chains
2. Implement appropriate solutions for solving typical NLP sub problems (tokenizing, tagging, parsing)
3. Implement the typical problems and processing layers in NLP
4. Implement NLP problems to decompose them in adequate independent components

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-353												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						
CO 4		H		H		H		H		M		M

LIST OF PRACTICALS

1. Cricket Game Prediction.
2. Machine Translation from English-Hindi.
3. Query Expansion for Information Retrieval.
4. Emotion detection for texts
5. Large Hindi Corpora
6. Crisp Query relevant Summary of each retrieved webpage
7. Semantic Query-Webpage Relevance
8. Visual Portrayal for Each Web Page
9. Provision of Hindi Key Board
10. Scalability to any Language

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-355
COURSE TITLE: WEB TECHNOLOGY LABRATORY
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Apply markup languages to design effective web pages.
2. Design the DB applications using JDBC, JSP and servlets.
3. Design dynamic and interactive web pages by embedding Java Script code in HTML.
4. Use Java Script to validate user input and create good, effective and customized websites

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-355												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						
CO 4		H		H		H		H		M		M

LIST OF PRACTICALS

The students are advised to get exposed to web technologies like HTML, XML and their variants as well as Java Programming

1 Write HTML/Java scripts to display your CV in Web Browser.
2 Creation and annotation of static web pages using any HTML editor.
3 Create a web page which includes a map and display the related information when a hot spot is clicked in the map.
4 Create the several Frames using HTML and display to the web browser.
5 Create Scientific Calculator using JavaScript.
6 To design web page to create a real time clock with a timing event using java script event handling mechanism.
7 Create a web page that displays college information using Style Sheet.
8 Create a Client Side Scripts for Validating Web Form Controls using DHTML.
9 Write a program to use XML and JavaScript for creation of your homepage.
10 Write a program in XML for creation of DTD which specifies a particular set of rules.
11 Create a Stylesheet in CSS/XSL and display the document in Web Browser.
12 Write a Java Servelet for HTTP Proxy Server.
13 Use JSP pages for sharing session and application data of HTTP Server.
14 Write a program to use JDBC connectivity program for maintaining database by sending queries.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

VI SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-302	Data Analytics	3	1	0	4	PC
2.	CSPC-304	Cloud Computing	3	0	0	3	PC
3.	CSPC-306	Network Security and Cyber Forensics	3	0	0	3	PC
4.	CSPC-308	Data Mining and Data Warehousing	3	0	0	3	PC
5.	CSPE-3XX	DE-II	3	0	0	3	PE
6.	CSOE-XXX	OE-I	3	0	0	3	OE
7.	CSPC-322	Data Analytics Laboratory	0	0	2	1	PC
8.	CSPC-326	Network Security and Cyber Forensics Laboratory	0	0	2	1	PC
9.	CSPC-328	Data Mining and Data Warehousing Laboratory	0	0	2	1	PC
10.	CSPE-3XX	DE-II Laboratory	0	0	2	1	PE
11.	CSCI-301	Minor Project, Phase-II	0	0	2	2*	CIC
TOTAL			**18**	**1**	**10**	**25**	

* Minor Project will be allotted in 5th Semester, will be evaluated after 6th Semester

DEPARTMENTAL ELECTIVE (DE)-II

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-332	Advanced Computer Networks	3	0	0	3
2.	CSPE-334	Android Programming and Mobile Applications Development	3	0	0	3
3.	CSPE-336	Internet of Things	3	0	0	3
4.	CSPE-352	Advanced Computer Networks Laboratory	0	0	2	1
5.	CSPE-354	Android Programming and Mobile Applications Development Laboratory	0	0	2	1
6.	CSPE-356	Internet of Things Laboratory	0	0	2	1

OPEN ELECTIVE (OE) –I

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	IDOE-001	Introduction to Databases	3	0	0	3
2.	IDOE-002	Introduction to Embedded Systems	3	0	0	3

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-302
COURSE TITLE: DATA ANALYTICS
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-1-0-4)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:

1. Understand what Big Data is and why classical data analysis techniques are no longer adequate.
2. Understand the benefits that Big Data can offer to businesses and organizations.
3. Learn conceptually how Big Data is stored.
4. See how Big Data can be analyzed to extract knowledge.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-302												
CO 1		M	H		M				H		H	H
CO 2		H	H	M	M				H	H	M	
CO 3					H	M			H		M	
CO 4		H		H		M		H			H	H

TOPICS COVERED

Mathematical concepts in data analytics: Descriptive Statistics, Probability Distributions, Inferential Statistics, Inferential Statistics through hypothesis tests, Regression & ANOVA, Regression, ANOVA (Analysis of Variance)

Differentiating algorithmic and model based frameworks Regression: Ordinary Least Squares, Ridge Regression, Lasso Regression, K Nearest Neighbors Regression & Classification

Supervised Learning with Regression and Classification techniques -1, Bias-Variance Dichotomy, Model Validation Approaches, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Regression and Classification Trees, Support Vector Machines

Supervised Learning with Regression and Classification techniques -2, Ensemble Methods: Random Forest Neural Networks, Deep learning

Unsupervised Learning and Challenges for Big Data Analytics, Clustering, Associative Rule Mining, Challenges for big data analytics

Prescriptive analytics, creating data for analytics through designed experiments, creating data for analytics through Active learning, creating data for analytics through Reinforcement learning

Hadoop Overview: Introduction to Hadoop, RDBMS vs Hadoop, key aspects of hadoop, hadoop components, hadoop conceptual layer, high level architecture of hadoop.

Hadoop Architecture: Hadoop architecture, Hadoop ecosystem components, Hadoop storage: HDFS, Hadoop processing, Map Reduce Framework, Hadoop server roles.

Hadoop big data technology landscape: NoSQL, Types of NoSQL database, Advantages, New SQL, Comparison of SQL, NoSQL and NewSQL.

Theory and Methods: Measures and evaluation, Supervised Learning, Linear and Logistic Regression, Decision trees, Unstructured data analytics.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

2. Montgomery, Douglas C., and George C. Runger. Applied statistics and probability for engineers. John Wiley & Sons, 2010
3. Big data and Analytics by Seema Acharya and Subhashini Chellappan.
4. Hadoop: The Definitive Guide by Tom White.
5. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL andGraph by David Loshin.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-304
COURSE TITLE: CLOUD COMPUTING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand cloud computing models and architecture.
2. Understand security implications in cloud computing.
3. Analyse the operation, implementation and performance of cloud computing systems, and the relative merits and suitability of each for complex data-intensive applications.
4. Analyse the migration risks and cost in Cloud Computing.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-304												
CO 1	H		H	M							M	
CO 2			M	M		H		H				
CO 3							H		H			H
CO 4	M			H						H		

TOPICS COVERED

Introduction: Cloud-definition, benefits, usage scenarios, History of Cloud Computing – Cloud Architecture – Types of Clouds – Business models around Clouds – Major Players in Cloud Computing – issues in Clouds – Eucalyptus – Nimbus – Open Nebula, CloudSim, Risks Involved in Cloud Computing.

Cloud Services: Types of Cloud services: Software as a service – Platform as a Service – Infrastructure as a Service – database as a Service – Monitoring as a Service – Communication as services. Service providers – Google, Amazon, Microsoft Azure, IBM, Salesforce.

Collaborating Using Cloud Services: Email Communication over the Cloud – CRM Management – Project Management – Event Management – Task Management – Calendar – Schedules – Word Processing – Presentation – Spreadsheet – Databases – Desktop – Social Networks and Groupware, Work Loan Management in Cloud.

Virtualization for Cloud: Need for Virtualization – Pros and cons of Virtualization – Types of Virtualization – System Vm, Process VM, Virtual Machine monitor – Virtual machine properties – Interpretation and binary translation, HLL VM – Hypervisors – Xen, KVM, VMWare, Virtual Box, Hyper-V.

Other Ways to Collaborate Online: Collaborating via Web - Based Communication Tools - Evaluating Web Mail Services – Evaluating Web Conference Tools – Collaborating via Social Networks and Groupware - Collaborating via Blogs and Wikis.

Security, Standards and Applications: Security in Cloud: Cloud security challenges – Software as a Service Security, Common Standards: The Open Cloud Consortium – The Distributed Management Task Force – Standards for application Developer – Standards for Messaging – Standards for Security, End user access to cloud computing, Mobile Internet devices and the cloud.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. K Jayaswal, J Kallakurchi, DJ Houde, and D Shah, "Cloud Computing Black Book" 1st edition, 2018.
2. Z Mahmood and R puttini, "Cloud Computing Concepts Technology & Architecture", 1st edition, 2014
3. John Rittinghouse and James Ransome, Cloud Computing, Implementation, Management and Strategy, CRC Press, 2010.
4. David E. Y. Sarna Implementing and Developing Cloud Application, CRC press 2011.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-306
COURSE TITLE: NETWORK SECURITY AND CYBER FORENSICS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Analyze the security issues related to network layer, transport layer and application layer.
2. Study of Cyber laws and forensics.
3. Evaluation of various forensics tools.
4. Analyze and validate forensics data.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-306												
CO 1.	H		M									
CO 2.	M	H			M							
CO 3.		M	M		H							
CO 4.		M	H	M	M	M						

TOPICS COVERED

Network Layer Security and Transport Layer Security
IPSec Protocol – IP Authentication Header – IP ESP – Key Management Protocol for IPsec. Transport layer Security: SSL protocol, Cryptographic Computations – TLS Protocol.

E-mail Security & Firewalls
PGP – S/MIME – Internet Firewalls for Trusted System: Roles of Firewalls – Firewall related terminology- Types of Firewalls – Firewall designs – SET for E-Commerce Transactions.

Introduction to Forensics
Introduction to Traditional Computer Crime, Traditional problems associated with Computer Crime. Introduction to Identity Theft & Identity Fraud. Types of CF techniques – Incident and incident response methodology – Forensic duplication and investigation. Preparation for IR: Creating response tool kit and IR team. – Forensics Technology and Systems – Understanding Computer Investigation – Data Acquisition.

Evidence Collection and Forensics Tools
Processing Crime and Incident Scenes – Working with Windows and DOS Systems. Current Computer Forensics Tools: Software/ Hardware Tools.

Computer Forensic Analysis and Validation
Validating Forensics Data – Data Hiding Techniques – Performing Remote Acquisition – Network Forensics – Email Investigations – Cell Phone and Mobile Devices Forensics.

Introduction to Wireless LAN Security
Wireless LAN Configuration, IEEE 802.11 WLAN Security, IEEE 802.11i RSN Operation

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Man Young Rhee, "Internet Security: Cryptographic Principles", "Algorithms and Protocols", Wiley Publications, 2003.
2. Nelson, Phillips, Enfinger, Steuart, "Computer Forensics and Investigations", Cengage Learning, India Edition, 2008.
3. John R.Vacca, "Computer Forensics", Cengage Learning, 2005
4. Richard E.Smith, "Internet Cryptography", 3rd Edition Pearson Education, 2008.
5. MarjieT.Britz, "Computer Forensics and Cyber Crime": An Introduction", 3rd Edition, Prentice Hall, 2013.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-308
COURSE TITLE: DATA MINING AND DATA WAREHOUSING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Design a data mart or data warehouse for any organization
2. Extract knowledge using data mining techniques
3. Explore recent trends in data mining such as web mining, spatial-temporal mining

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-308												
CO 1	H		H		L	L						
CO 2		M	M	H	L				H			
CO 3		H		M	H	H	M				M	

TOPICS COVERED

Introduction: Data Mining Concepts, Input, Instances, Attributes and Output, Knowledge Representation & Review of Graph Theory, Lattices, Probability & Statistics

Machine learning concepts and approaches: Supervised Learning Framework, concepts & hypothesis, Training & Learning, Boolean functions and formulae, Monomials, Disjunctive Normal Form & Conjunctive Normal Form, A learning algorithm for monomials

Data Preparation: Data Cleaning, Data Integration & Transformation, Data Reduction

Mining Association Rules: Associations, Maximal Frequent & Closed Frequent item sets, Covering Algorithms & Association Rules, Linear Models & Instance-Based Learning, Mining Association Rules from Transactional databases, Mining Association Rules from Relational databases & Warehouses, Correlation analysis & Constraint-based Association Mining.

Classification and Prediction: Issues regarding Classification & Prediction, Classification by Decision Tree induction, Bayesian classification, Classification by Back Propagation, k-Nearest Neighbor Classifiers, Genetic algorithms, Rough Set & Fuzzy Set approaches

Cluster Analysis: Types of data in Clustering Analysis, Categorization of Major Clustering methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based Clustering methods

Mining Complex Types of Data: Multidimensional analysis & Descriptive mining of Complex data objects, Mining Spatial Databases, Mining Multimedia Databases, Mining Time-series & Sequence data, Mining Text databases, Mining World -Wide Web

Data Mining Applications and Trends in Data Mining: Massive Datasets/Text mining, Agent-Based Mining

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Jiawei Han and MichelineKamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 2000 (ISBN: 1-55860-489-8).
2. Ian H. Witten and Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java implementations", Morgan Kaufmann Publishers, San Fransisco, CA (2000).
3. Dorian Pyle, "Data Preparation for Data Mining", Morgan Kaufmann, (1999)
4. Korth, Silbertz, Sudarshan, "Database Concepts", McGraw Hill
5. Elmasri, Navathe, "Fundamentals Of Database Systems", Addision Wesley

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-322
COURSE TITLE: DATA ANALYTICS LABRATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Preparing for data summarization, query, and analysis.
2. Applying data modelling techniques to large data sets
3. Creating applications for Big Data analytics
4. Building a complete business data analytic solution

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-322												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						
CO 4		H		H		H		H		M		M

LIST OF PRACTICALS

1. Perform setting up and Installing Hadoop in its two operating modes:
2. Implement the following file management tasks in Hadoop:
3. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.
4. Implement Stop word elimination problem Using suitable example.
5. Implement a Map Reduce program that mines weather data.
6. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.
7. Write a Pig Latin scripts for finding TF-IDF value for book dataset (A corpus of eBooks available at: Project Gutenberg)
8. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes.
9. Install, Deploy & configure Apache Spark Cluster. Run apache spark applications using Scala.
10. Apply suitable data analytics techniques using Apache Spark on Amazon food dataset, find all the pairs of items frequently reviewed together.

11. Write a single Spark application that:
    Transposes the original Amazon food dataset, obtaining a PairRDD of the type:
    <user_id> → <list of the product_ids reviewed by user_id>
    Counts the frequencies of all the pairs of products reviewed together;
12. Writes on the output folder all the pairs of products that appear more than once and their frequencies. The pairs of products must be sorted by frequency.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-326
COURSE TITLE: NETWORK SECURITY AND CYBER FORENSIC LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After completion of this course, the students would be able to

1. Basic Data Recovery and Analysis – National White Collar Crime Center
2. A working knowledge of the policies and procedures used in the forensic examination of computer evidence.
3. Working with computer forensic evidence under the immediate supervision of a Senior Forensic Examiner
4. Apply the knowledge of computer forensics to an actual case while being supervised.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-326												
CO 1	H	H	H	M							M	
CO 2			M	H			M	H	M			
CO 3	M			M								
CO 4	M			H								

LIST OF PRACTICALS

1. Hands on training on the case management system to input case data.
2. Working on evidence handling and security procedures.
3. Implement key fob and alarm code and understand how to set burglar alarm.
4. Working on verification procedure within the forensic laboratory.
5. Implement how to forensically wipe a target drive and how to ensure a drive has been wiped.
6. Implement various network security protocols studded in the theory classes.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-328
COURSE TITLE: DATA MINING AND DATA WAREHOUSING LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Synthesize the data mining fundamental concepts and techniques from multiple perspectives.
2. Develop skills and apply data mining tools for solving practical problems
3. Develop research skills by reading the data mining literature and develop advance relevant programming skills

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-328												
CO 1	H	M				H	H	M				
CO 2	L	H	H	H	H				H			
CO 3				H		H	H	H	H	H	H	H

LIST OF PRACTICALS

Students are required to perform practical's in Oracle/MS SQL Server and STATISTICA *Data Miner*

1. Building a Database Design using ER Modeling and Normalization Techniques
2. Implementation of functions ,Procedures, Triggers and Cursors
3. Load Data from heterogeneous sources including text files into a predefined warehouse schema.
4. Design a data mart for a bank to store the credit history of customers in a bank .Use this credit profiling to process future loan applications.
5. Feature Selection and Variable Filtering (for very large data sets)
6. Association Mining in large data sets
7. Interactive Drill-Down, Roll up, Slice and Dice operations
8. Generalized EM &*k*-Means Cluster Analysis
9. Generalized Additive Models (GAM)
10. General Classification and Regression Trees (GTrees)
11. General CHAID (Chi-square Automatic Interaction Detection) Models
12. Interactive Classification and Regression Trees
13. Goodness of Fit Computations

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENT ELECTIVE (DE)-II

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-332
COURSE TITLE: ADVANCED COMPUTER NETWORKS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Understand packet switching networks and routing in packet switching networks with different routing algorithms.
2. Describe traffic management at packet level, flow level and flow aggregate levels of packet switching networks.
3. Explain the architecture of TCP/IP and protocols associated with TCP/IP and to analyze the network applications, network management and security issues
4. Apply the knowledge about QoS, VPNs, and tunneling and overlay networks and to understand mobile networking and wireless sensor networking

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-332												
CO 1.	H		M		M							
CO 2.	M			H						L		
CO 3.				M		M	H					
CO 4.		M	H		M		H		M		M	

TOPICS COVERED

Reliable Protocol:
Transmission Control Protocol (TCP): Error Control, Flow Control, Congestion Control, Timers, And TCP Options: NOP, MSS, Window Scale Factor, Timestamp, SACK-Permitted And SACK Options

Stream Control Transmission Protocol (SCTP): Introduction, Services, Features, Packet Format, Association, State Transition Diagram, Flow Control, Error Control, Congestion Control

Congestion Control and Resource Allocation: Issues In Resource Allocation: Network Model, Taxonomy, Evaluation Criteria; Queuing Disciplines: FIFO, Fair Queuing; TCP Congestion Control: Additive Increase/Multiplicative Decrease, Slow Start, Fast Retransmit and Fast Recovery; Congestion-Avoidance Mechanisms: DECbit, Random Early Detection (RED), Source-Based Congestion Avoidance; Quality of Service: Application Requirements, Integrated Services (RSVP), Differentiated Services (EF 516, AF), Equation-Based Congestion Control

Next Generation Network: Unicast Routing Protocols: RIP, OSPF; Multicasting And Multicast Routing Protocols: Introduction, Multicast Addresses, IGMP, Multicast Routing, Routing Protocols, MBone

Internet Protocol Version 6: IPV6 Addressing: Introduction, Address Space Allocation, Global Unicast Addresses, Auto configuration, Renumbering; IPV6 Protocol: Packet Format, Transition from Ipv4 TO Ipv6; Generic Routing Encapsulation (GRE) For Tunnelling.

ICMPv6: Error Messages, Informational Messages, Neighbours-Discovery Messages, Group Membership Messages

Wireless LAN: Infrared vs. Radio Transmission, Infrastructure and Ad Hoc Networks. IEEE 802.11, System Architecture, Protocol Architecture, Physical Layer, Medium Access Control Layer, MAC Management, Future Development, HIPERLAN, Protocol Architecture, Physical Layer, Channel Access Control Sublayer, Medium Access Control Sublayer, Information Bases and Networking, Bluetooth, User Scenarios, Physical Layer, MAC Layer, Networking, Security, Link Management

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1) Behrouz A. Forouzan , "TCP/IP Protocol Suite", McGraw- Hill, 4/e, 2015.
2) Larry L. Peterson & Bruce S. Davie, "Computer Network: A System Approach", Morgan Kaufmann, 5/e, 2012.
3) Jochen Schiller, "Mobile Communications", Pearson Addison-Wesley, 2/e, 2010.
4) James F. Kurose, Keith W. Ross, "Computer Networking", Pearson, 2016.
5) Charles M. Kozierok, "The TCP/IP Guide", No starch press, 2018.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-334
COURSE TITLE: ANDROID PROGRAMMING AND MOBILE APPLICATION DEVELOPMENT
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Design and implement the user interfaces for mobile applications.
2. Design the mobile applications that is aware of the resource constraints of mobile devices.
3. Develop advanced mobile applications that accesses the databases and the web.
4. Develop useful mobile applications in the current scenario using Google Android and Eclipse simulator.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-334												
CO 1	H	H	M	M								
CO 2		M	L	L								
CO 3		M		H	M				M			
CO 4		M		M	H							

TOPICS COVERED

INTRODUCTION
Mobile Applications – Characteristics and Benefits – Application Model – Infrastructure and Managing Resources – Mobile Software Engineering – Frameworks and Tools – Mobile devices Profiles.

USER INTERFACE
Generic UI Development – VUIs and Mobile Applications – Text to Speech techniques – Designing the right UI – Multimodal and Multichannel UI – Gesture based UIs – Screen Elements and Layouts – Voice XML – Java API.

APPLICATION DESIGN
Memory Management – Design patterns for limited memory – Work flow for Application Development – Techniques for composing Applications – Dynamic Linking – Plug ins and rules ofthumb for using DLLs – Concurrency and Resource Management – Look and feel.

APPLICATION DEVELOPMENT
Intents and Services – Storing and Retrieving data – Communication via the Web – Notification and Alarms – Graphics and Multimedia – Telephony – Location based services – Packaging and Deployment – Security and Hacking.

TOOLS
Google Android Platform – Eclipse Simulator – Android Application Architecture – Event based programming – Apple iPhone Platform – UI tool kit interfaces – Event handling and Graphics services – Layer Animation.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. ZigurdMednieks, Laird Dornin, G, Blake Meike and Masumi Nakamura, "Programming Android", O"Reilly, 2011.
2. Reto Meier, Wrox Wiley, "Professional Android 2 Application Development", 2010.
3. Alasdair Allan, "iPhone Programming", O"Reilly, 2010.
4. Wei-Meng Lee, "Beginning iPhone SDK Programming with Objective-C", Wrox Wiley, 2010.
5. Stefan Poslad, "Ubiquitous Computing: Smart Devices, Environments and interactions", Wiley, 2009.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-336
COURSE TITLE: INTERNET OF THINGS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to
1. Analyze various protocols of IoT
2. Design portable IoT using appropriate boards
3. Develop schemes for the applications of IoT in real time scenarios
4. Design business Intelligence and Information Security for IoT

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-336												
CO 1	H		M	M								
CO 2		M		L								
CO 3			M	H	M			H	H			
CO 4		M		H	H							

TOPICS COVERED

Introduction: Definition – Foundations – Challenges and Issues - Identification - Security. Components in internet of things: Control Units – Sensors – Communication modules –Power Sources – Communication Technologies – RFID – Bluetooth – Zigbee – Wifi – Rflinks –Mobile Internet – Wired Communication-IoT Platform Overview-Raspberry pi-Arduino boards.

IoT Protocols: Protocol Standardization for IoT-M2M and WSN Protocols-SCADA and RFID Protocols-Issues with Iot Standardization-Protocols-IEEE 802.15.4-BACNet Protocol-Zigbee Architecture - Network layer – APS Layer – Security.

Resource Management in the Internet of Things: Clustering - Software Agents - Data Synchronization - Clustering Principles in an Internet of Things Architecture - The Role of Context - Design Guidelines -Software Agents for Object – Data Synchronization- Types of Network Architectures - Fundamental Concepts of Agility and Autonomy-Enabling Autonomy and Agility by the Internet of Things - The Evolution from the RFID-based EPC Network to an Agent based Internet of Things- Agents for the Behaviour of Objects.

Case Study and IoT Application Development: IoT applications in home- infrastructuressecurity-Industries- IoT electronic equipments. Use of Big Data and Visualization in IoTIndustry 4.0 concepts - Sensors and sensor Node – Interfacing using Raspberry Pi/Arduino- Web Enabled Constrained Devices.

Web of Things: Web of Things versus Internet of Things-Architecture Standardization for WoT-Platform Middleware for WoT- WoT Portals and Business Intelligence-Cloud of Things:Grid/SOA and Cloud Computing-Cloud Standards –Cloud of Things Architecture-Open Source e-Health sensor platform.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Honbo Zhou, "The Internet of Things in the Cloud:A Middleware Perspective" — CRC Press-2012.
2. Dieter Uckelmann, Mark Harrison, "Architecting the Internet of Things", Springer- 2011.
3. Arshdeep Bahga, Vijay Madisetti, "Internet of Things (A Hands-On-Approach)", VPT, 2014.
4. Olivier Hersent, David Boswarthick, Omar Elloumi, "The Internet of Things – Key applications and Protocols", Wiley, 2012.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-352
COURSE TITLE: ADVANCED COMPUTER NETWORKS LABORATORY
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Implement various networks environment and passing packets through them using different routing techniques.
2. Design various error control, flow control, and congestion control mechanism in TCP.
3. Implement networks environment for simulating various access techniques and Queuing algorithm.
4. Design scenarios for wireless networks using simulation tools.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-352												
CO 1	H		M		M							
CO 2		M		L								
CO 3			M		M				H		L	
CO 4		M		H	H							

LIST OF PRACTICALS

Note: Implementation should be done using C/C++, QualNet/NS2 and other similar tools.

1. Write a program to transfer a file from one system to another system using TCP and UDP sockets.
2. Write a program to demonstrate communication between different processes using IPC.
3. Write a Program to implement Routing Information Protocol (RIP) for a set of nodes.
4. Write a program to implement flow control and congestion control in TCP.
5. Write a program to implement queuing algorithm which will discard the staled packets.
6. Write a congestion control algorithm for routers which will inform the host nodes to stop sending when threshold is reached.
7. Create a network of multiple routers and hosts to simulate RED and Drop Tail Queuing algorithm.
8. Write a program to simulate Group Communication and implement Carrier sensing techniques.
9. Design scenarios for wireless networks using simulation tools.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. W. Richard Stevens, "UNIX network programming", Prentice Hall, 1990.
2. W. Richard Stevens, "UNIX Network Programming, Volume 1: The Sockets Networking API", Prentice Hall, 3/e, 2003.
3. W. Richard Stevens, "UNIX Network Programming, Volume 2: Interprocess Communications", Prentice Hall, 2/e, 2012

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-354
COURSE TITLE: ANDROID PROGRAMMING AND MOBILE APPLICATIONS DEVELOPMENT LABRATORY
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Design and implement the user interfaces for mobile applications.
2. Design the mobile applications that is aware of the resource constraints of mobile devices.
3. Develop advanced mobile applications that accesses the databases and the web.
4. Develop useful mobile applications in the current scenario using Google Android and Eclipse simulator.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-354												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						
CO 4		H		H		H		H		M		M

LIST OF PRACTICALS

1. Develop an application that uses GUI components, Font and Colors.
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Implement an application that implements Multi-threading.
8. Develop a native application that uses GPS location information.
9. Implement an application that writes data to the SD card.
10. Implement an application that creates an alert upon receiving a message.
11. Develop a mobile application that creates alarm clock.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-356
COURSE TITLE: INTERNET OF THINGS LABRATORY
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Implement various protocols of IoT
2. Implement portable IoT using appropriate boards
3. Implement schemes for the applications of IoT in real time scenarios
4. Implement business Intelligence and Information Security for IoT

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-356												
CO 1	H		M	M								
CO 2		M		L								
CO 3			M	H	M			H	H			
CO 4		M		H	H							

LIST OF PRACTICALS

1. Programming of Raspberry Pi3 and MSP430
2. Peripheral interfacing to Raspberry Pi3 and MSP430
3. Trouble shooting Raspberry Pi3 and MSP430 based systems
4. Linux /Android programming techniques
5. Evaluate networking technologies for application within IoT projects
6. Trace the relationship between IoT, cloud services and software agents
7. Apply effective techniques to create IoT based projects
8. Mini project

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

OPEN ELECTIVE (OE)-I

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: IDOE-001
COURSE TITLE: INTRODUCTION TO DATABASES
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Identify the importance of multimedia database and related concepts.
2. Design multimedia database based on the current environment and requirement.
3. Develop application which incorporate multimedia database as a group work.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IDOE-001												
CO 1		M				H		M				
CO 2	L		H		H				H			
CO 3				H		H			H	H		

TOPICS COVERED

Introduction: An introduction to Object-oriented Databases; Multidimensional Data Structures: k-d Trees, Point Quadtrees, The MX-Quadtree, R-Trees, comparison of Different Data Structures

Image Databases: Raw Images, Compressed Image Representations, Image Processing: Segmentation, Similarity-Based Retrieval, Alternative Image DB Paradigms, Representing Image DBs with Relations, Representing Image DBs with R-Trees, Retrieving Images By Spatial Layout, Implementations

Text/Document Databases: Precision and Recall, Stop Lists, Word Stems, and Frequency Tables, Latent Semantic Indexing, TV-Trees, Other Retrieval Techniques

Video Databases: Organizing Content of a Single Video, Querying Content of Video Libraries, Video Segmentation, video Standards

Audio Databases: A General Model of Audio Data, Capturing Audio Content through Discrete Transformation, Indexing Audio Data

Multimedia Databases: Design and Architecture of a Multimedia Database, Organizing Multimedia Data Based on The Principle of Uniformity, Media Abstractions, Query Languages for Retrieving Multimedia Data, Indexing SMDSs with Enhanced Inverted Indices, Query Relaxation/Expansion

Creating Distributed Multimedia Presentations: Objects in Multimedia Presentations, Specifying Multimedia Documents with Temporal Constraints, Efficient Solution of Temporal Presentation Constraints, Spatial Constraints.

Spatial Concepts and Data Models: Models of spatial information, Design extending the ER model with spatial concepts, Extending the ER model pictograms, Object oriented data model with UML.

Spatial Query Languages: Extending the SQL for spatial data, Examples of queries that emphasis spatial data, Object relational schema examples queries.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Principles of Multimedia Database Systems, V.S. Subramanian, Elseveir(Morgan Kauffman).
2. Spatial Databases, ShashiShekhar, SanjivChawla, Pearson Education.
3. Multimedia Databases: An object relational approach, Lynne Dunckley, Pearson Education.
4. Multimedia Database Systems, Prabhakaram, Springer.
5. Maheshwari Jain, "DBMS: Complete Practical Approach", Firewall Media, New Delhi

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: IDOE-002
COURSE TITLE: INTRODUCTION TO EMBEDDED SYSTEMS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand the general process of embedded system development
2. Understand General System Theory, how to apply embedded system and how to differentiate this from the traditional mechanistic theory.
3. Evaluate how architectural and implementation decisions influence performance and power dissipation.
4. Evaluate how architectural and implementation decisions influence performance and power dissipation.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IDOE-002					H							
CO 1	H		H	M							M	
CO 2			M					H				
CO 3				M								
CO 4	M			H								

TOPICS COVERED

Introduction: Introduction to Embedded Systems, Classification of Embedded System, Concept of Embedded System Design, and Design challenges: Processor technology, IC technology, Design technology and Trade-offs.

Hardware and Software Co-Design in Embedded System: Buffers and latches, Reset circuit, Chip, Timers and counters and watch dog timers, Universal asynchronous receiver, transmitter (UART), Pulse width modulators, LCD controllers. Development of fixed ROM image, Code generation tools: Emulator, Simulator and Debugger.

Embedded software development environments: Challenges and issues in embedded software development, Device drivers, System calls and Programming languages: assembly languages, high level languages like C/C++, Source Code Engineering tool for Embedded C/C++. Introduction to Embedded Java.

Processor and memory Organization: Custom Single Purpose Processor Hardware, General-Purpose Processor: Introduction, Basic Architecture, Application Specific Instruction Set Processors (ASIPS), Microcontrollers and Digital Signal Processors. Memory writes ability, Storage performance, Tradeoff s, Memory hierarchy and cache.

Software Engineering in Embedded System: Software Engineering practice in the embedded Software development process. Software models used in designing, Unified Modeling language, Software maintenance.

Embedded Operating System: Operating system services, Embedded Operating system, Real Time Operating system, Interrupt latency and Response time, Interrupts Routines in RTOS, Introduction to VxWorks and Micro OS-II.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. David E Simon, "An Embedded Software Primer", 1/e Pearson Education 1999.
2. Raj Kamal, "Embedded Systems", Tata McGraw-Hill 2004.
3. Bruce Powel Douglass, "Real-Time UML: Developing Efficient Objects for Embedded Systems", 2/E Addison Wesley 2004.
4. Muhammad Ali Mazidi, Janice GillispieMazidi, "The 8051 Micro controller & Embedded Systems", 1/e Pearson Education 2000.
5. Valvano, "Embedded Microcomputer Systems : A real time interfacing",Cengage Learning, New Delhi

VII SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-401	Software Project Management	3	0	0	3	PC
2.	CSPC-403	Computer Graphics and Image Processing	3	0	0	3	PC
3.	CSPC-405	Artificial Intelligence	3	0	0	3	PC
4.	CSPE-4XX	DE-III	3	0	0	3	PE
5.	CSOE-XXX	OE-II	3	0	0	3	OE
6.	CSPC-423	Computer Graphics and Image Processing Laboratory	0	0	2	1	PC
7.	CSCI-300	Industrial Practical Training	0	0	8	2	CIC
8.	CSCI-400	Project (Phase-I)	0	0	4	0*	CIC
TOTAL			**15**	**0**	**14**	**18**	

DEPARTMENTAL ELECTIVE (DE)-III

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
4.	CSPE-431	Big Data Analytics	3	0	0	3
5.	CSPE-433	Block Chain Architecture & Use Cases	3	0	0	3
6.	CSPE-435	Distributed System	3	0	0	3

OPEN ELECTIVES (OE)-II

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
3.	IDOE-003	Software Engineering Concepts	3	0	0	3
4.	IDOE-004	Artificial Intelligence and Applications	3	0	0	3

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-401
COURSE TITLE: SOFTWARE PROJECT MANAGEMENT
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the students would be able to:

1. Understood the basics of the project management techniques.
2. Learnt the feasible solution and optimum solution for the resource management. Learnt the time estimation and critical path for project.
3. Learnt about the application of probability techniques in the decision making.
4. Learnt the various inventory models and simulations in the resource planning and management.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-401												
CO 1	H	H	M	M								
CO 2		M	L	L								
CO 3		M		H	M				M		H	H
CO 4		M		M	H							H

TOPICS COVERED

Introduction: Project Management (PM) Fundamentals, People, Process, and Product, Technology Classic mistakes, PMI Processes, Software project phases, Organizational structures, Project charter Statement of Work (SOW)

Planning Phase: Development lifecycle models, Matching lifecycles to projects, Project plans Work Breakdown Structures (WBS)

Software Quality and Metrics: Attributes and metrics for different types of projects viz development project, reengineering, maintenance, testing projects; use of metrics in decision making

Project Monitoring and Control: Project control and management processes, effort data collection, monitoring and control, quantitative techniques in monitoring and control processes, tools and techniques used in quantitative project management.

Estimation and Budgeting: Estimation, Budgeting, Project selection, NPV, ROI, Payback models

Scheduling: Project network diagram fundamentals, PERT techniques, Gantt charts, Critical chain scheduling, resource leveling and remote smoothing techniques

Risk and Change Management: Risk management, Change control, MS-Project

Configuration Management: configuration management process and configuration management audit

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Kathy Schwalbe, “Information Technology Project Management”, Cengage Learning, 7/e, 2013.
2. M. Cottrell and B. Hughes, "Software Project Management", McGraw-Hill, 5/e, 2009.
3. Quantum PM, “Microsoft Office Project Server 2003 Unleashed”, Pearson Education India, 2005.
4. Robert T. Futrell, Donald F. Shafer and Linda Isabell Shafer , “Quality Software Project”, Pearson India, 2002.
5. D. J. Henry, “Software Project Management – A Real-World Guide to Success”, Addison-Wesley, 2003.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-403
COURSE TITLE: COMPUTER GRAPHICS AND IMAGE PROCESSING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
On completion of the course students will be able to

1. Demonstrate an understanding of contemporary graphics hardware and formation of images.
2. Create interactive graphics applications using different platforms.
3. Learn the signal processing algorithms and techniques in image enhancement and image restoration.
4. Application of various image processing techniques to solve real world problems

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-403												
CO 1	H	M	H	M							M	
CO 2			M	H				H	H			
CO 3	M			M								
CO 4	M			H							H	H

TOPICS COVERED

Introduction: Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms

2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems. 2-D viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland – Hodgeman polygon clipping algorithm.

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods, 3-D geometric transformations, rotation, scaling, reflection and shear transformations, composite transformations, 3-D **v**iewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.
Fundamentals: Introduction, Origin, Areas of Image Processing, steps in Digital Image Processing, Components of Image Processing System, Image Sensing, Sampling and Quantization, Neighboring of Pixels, Mathematical and perceptual preliminaries, human visual system model, image signal representation, imaging system specification building image quality, role of computers, image date formats.

Image Enhancement and Restoration: Enhancement: Spatial Filtering, Introduction to Fourier Transformation, Restoration: A model of the Image Degradation/ Restoration Process.

Image Compression: Image compression models, Error free compression, Lossy compression. Image segmentation: Line detection, Edge Detection, Edge linking and Boundary Detection, and Region-based segmentation

Object Recognition: Pattern and pattern classes, Recognition based on Decision Theoretic Methods, Structural Methods.

Applications of Image processing: Picture data archival, machine vision, medical image processing.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. "Computer Graphics", second Edition, Donald Hearn and M.Pauline Baker, PHI/Pearson Education.
2. "Computer Graphics Second edition", Zhigandxiang, Roy Plastock, Schaum's outlines, Tata Mc- Graw hill edition.
3. Pratt, W. K. Digital Image Processing, John Wiley, N. Y.
4. Jain, A.K. fundamentals of Digital Image Processing, Englewood Cliffs, Prentice Hall
5. Digital Image Processing by Rafael C. Gonzalez, Richard E. Woods, 3rd edition, pearson, 2014

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-405
COURSE TITLE: ARTIFICIAL INTELLIGENCE
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
2. Formalize a given problem in the language/framework of different AI methods
3. Describe basic AI algorithms (e.g., standard search algorithms or resolution).
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-405												
CO 1	H	M		H		H	H		M		M	
CO 2	M	H	M	M	H	L						
CO 3	M		H		M							
CO 4								M	H	H	M	M

TOPICS COVERED

Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas

Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic

Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation& Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Machine Learning: Concept of learning, Concept creation, learning automation, supervised and Unsupervised Learning, learning tasks & learning strategies, single layer & multiplayer Perceptions, Back propagation, learning by inductions, Competitive Learning, HebbianCoincidence Learning, Attractor Networks Samuel's checkers algorithm. Hopfield nets, Adaptive resonance theory

Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition. Case studies: MYCIN, RI.

TEXT BOOKS, AND/OR REFERENCE MATERIAL:

1. Rich and K. Knight," Artificial Intelligence", Tata McGraw Hill.
2. George F. Luger, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Pearson Education.
3. Russell & Norvig, "Artificial Intelligence 'a Modern Approach", Pearson Education.
4. Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", PHI.
5. E. Charnaik and D. McDermott," Introduction to Artificial Intelligence", Addison-Wesley Publishing Company.
6. Nils J. Nilson, "Principles of Artificial Intelligence", Narosa Publishing Co.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-423
COURSE TITLE: COMPUTER GRAPHICS AND IMAGE PROCESSING LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)
COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After completion of this course, the students would be able to

1. Develop interactive graphics applications using different platforms.
2. Implementation of various 2D and 3D image transformation techniques.
3. Implementation of various techniques of image enhancement, reconstruction, compression and segmentation

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-423												
CO 1	H	H	H	M							M	
CO 2			M	H			M	H	M			
CO 3	M			M								

TOPICS COVERED

1. Implementation of various line drawing algorithms.
2. Implementation of various circle drawing algorithms.
3. Implementation of various line clipping algorithms.
4. Implementation of various ellipse drawing algorithms.
5. Implementation of various translation, rotation and scaling techniques in 2D plane.
6. Implementation of various composite transformations techniques on an objects.
7. Implement of various histogram equalization and non-linear filtering algorithms.
8. Implement of various edge detection, 2-D DFT, DCT, and DWT techniques.
9. Implement of various segmentation algorithm using watershed transform.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENTAL ELECTIVE (DE)-III

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-431
COURSE TITLE: BIG DATA ANALYTICS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
On completion of the course students will be able to

1. Analyze tools and practices for working with big data
2. Understand various application of big data analytics to different domains
3. Integrate and correlate large amounts of information/data using Big Data techniques

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-431												
CO 1	M	H		H								
CO 2			H	H	H	L		H				
CO 3				M		H	H				M	

TOPICS COVERED

Introduction to Big Data: Analytics – Nuances of big data – Value – Issues – Case for Big data – Big data options Team challenge – Big data sources – Acquisition – Features of Big Data - Security, Compliance, auditing and protection - Evolution of Big data – Best Practices for Big data Analytics - Big data characteristics - Volume, Veracity, Velocity, Variety – Big Data Architecture – Big Data and Cloud.

Data Analysis: Evolution of analytic scalability – Convergence – parallel processing systems – Cloud computing – grid computing – Map reduce Basics – Map Reduce Algorithm Design - enterprise analytic sand box – analytic data sets – Analytic methods – analytic tools – Cognos – Microstrategy – Pentaho - Regression Modeling - Multivariate Analysis - Bayesian Modeling - Inference and Bayesian Networks - Support Vector and Kernel Methods.

Stream Computing : Introduction to Streams Concepts – Stream data model and architecture - Stream Computing, Sampling data in a stream – Filtering streams – Counting distinct elements in a stream – Estimating moments – Counting oneness in a window – Decaying window - Realtime Analytics Platform(RTAP) applications IBM Infosphere – Big data at rest – Infosphere streams – Data stage – Statistical analysis – Intelligent scheduler – Infosphere Streams.

Predictive Analytics and Visualization: Predictive Analytics – Supervised – Unsupervised learning – Neural networks – Kohonen models – Normal – Deviations from normal patterns – Normal behaviours – Expert options – Variable entry - Mining Frequent itemsets - Market based model – Apriori Algorithm – Handling large data sets in Main memory – Limited Pass algorithm – Counting frequent itemsets in a stream – Clustering Techniques – Hierarchical – K- Means – Clustering high dimensional data Visualizations - Visual data analysis techniques, interaction techniques; Systems and applications.

Frameworks and Applications: IBM for Big Data – Map Reduce Framework - Hadoop – Hive – Sharding - MongoDB – NoSQL Databases - S3 - Hadoop Distributed file systems – Hbase – Impala – Analyzing big data with Twitter and Facebook – Big data for E-commerce – Big data for blogs.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Frank J Ohlhorst, "Big Data Analytics: Turning Big Data into Big Money", Wiley and SAS Business Series, 2012.
2. AnandRajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.
3. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", Wiley and SAS Business Series, 2012.
4. Paul Zikopoulos, Chris Eaton, Paul Zikopoulos, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGraw Hill, 2011.
5. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch , James Giles, David Corrigan, "Harness the Power of Big data – The big data platform", McGraw Hill, 2012.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-433
COURSE TITLE: BLOCK CHAIN ARCHITECTURE & USE CASES
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
On completion of the course students will be able to

1. Analyze algorithms for public digital ledger to share information in a trustworthy and secure way
2. Design applications of Blockchain from cryptocurrencies to various other domains, including business process management, smart contracts, IoT and so on..
3. Evaluate the security aspects along with various use cases from different application domains.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-433												
CO 1	M	H		H								
CO 2			H		H	H		H				
CO 3				M		H	H		H			H

TOPICS COVERED

Introduction to Blockchain: History of Digital Money to Distributed Ledgers, Design Primitives: Protocols, Security, Consensus, Permissions, Privacy

Blockchain Architecture and Design: Basic crypto primitives: Hash, Signature; Hashchain to Blockchain; Basic consensus mechanisms

Consensus: Requirements for the consensus protocols, Proof of Work (PoW), Scalability aspects of Blockchain consensus protocols

Permissioned Blockchains: Design goals, Consensus protocols for Permissioned Blockchains

Hyperledger Fabric I: Decomposing the consensus process, Hyper ledger fabric components, Chaincode Design and Implementation

Hyperledger Fabric II: Beyond Chaincode, fabric SDK and Front End, Hyperledger composer tool

Use case I: Blockchain in Financial Software and Systems (FSS): Settlements, KYC, Capital markets, Insurance

Use case II: Blockchain in trade supply chain: Provenance of goods, visibility, trade supply chain finance, invoice management discounting, etc

Use case III: Blockchain for Government: Digital identity, land records and other kinds of record keeping between government entities, public distribution system social welfare systems

Blockchain Cryptography, Privacy and Security on Blockchain

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Mastering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas Antonopoulos, OReilly, 2018
2. Blockchain for new economy by Melanie Swa, OReilly
3. https://www.hyperledger.org/projects/fabric

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-435
COURSE TITLE: DISTRIBUTED SYSTEMS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
On completion of the course students will be able to

1. Analyze algorithms for coordination, communication, security and synchronization in distributed systems
2. Design and Implement distributed file systems and distributed algorithms for deadlocks.
3. Evaluate the effectiveness and shortcomings of their solutions

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-435												
CO 1	M	H		H								
CO 2			H	H	H	L		H				
CO 3				M		H	H				M	

TOPICS COVERED

Characterization of Distributed Systems: Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges.

System Models: Architectural models, Fundamental Models

Theoretical Foundation for Distributed System: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport's& vectors logical clocks, Causal ordering of messages, global state, and termination detection.

Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non token based algorithms, performance metric for distributed mutual exclusion algorithms.

Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms.

Agreement Protocols: Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem, Atomic Commit in Distributed Database system.

Distributed Objects and Remote Invocation: Communication between distributed objects, Remote procedure call, Events and notifications, Java RMI case study.

Security: Overview of security techniques, Cryptographic algorithms, Digital signatures Cryptography pragmatics, Case studies: Needham Schroeder, Kerberos, SSL& Millicent.

Distributed File Systems: File service architecture, Sun Network File System, The Andrew File System, Recent advances.

Transactions and Concurrency Control: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control.

Distributed Transactions: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault -tolerant services, highly available services, Transactions with replicated data.

Distributed Algorithms: Introduction to communication protocols, Balanced sliding window protocol, Routing algorithms, Destination based routing, APP problem, Deadlock free Packet switching, Introduction to wave & traversal algorithms, Election algorithm.
CORBA Case Study: CORBA RMI, CORBA services.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Singhal & Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill
2. Coulouris, Dollimore, Kindberg, "Distributed System: Concepts and Design", Pearson Ed.
3. Gerald Tel, "Distributed Algorithms", Cambridge University Press
4. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann.
5. Andrew S. Tanenbaum, Distributed Operating Systems, ACM Press.

OPEN ELECTIVE (OE)-II

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: IDOE-003
COURSE TITLE: SOFTWARE ENGINEERING CONCEPTS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
On completion of the course students will be able to

1. Describe the expectations, pressures and problems faced in developing software and the need for processes, tools, techniques and approaches;
2. Analyze, design, test and maintain software systems
3. Identify risks during software development.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IDOE-003												
CO 1		M	M	H								
CO 2			M		L							
CO 3			H		H	M		M			M	M

TOPICS COVERED

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

Design Engineering: Design process and Design quality, Design concepts, the design model.
Creating an architectural design: Software architecture, Data design, Architectural styles and patterns, Architectural Design.

Object-Oriented Design: Objects and object classes, An Object-Oriented design process, Design evolution.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modelling Language (UML)

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Software Engineering- K.K. Agarwal&Yogesh Singh, New Age International Publishers
2. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiely.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
4. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition. McGrawHill International Edition.
5. Software Engineering- Sommerville, 7th edition, Pearson education.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: IDOE-004
COURSE TITLE: ARTIFICIAL INTELLIGENCE AND APPLICATIONS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
2. Formalize a given problem in the language/framework of different AI methods
3. Describe basic AI algorithms (e.g., standard search algorithms or resolution).
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IDOE-004												
CO 1	H	M		H		H	H		M		M	
CO 2	M	H	M	M	H	L						
CO 3	M		H		M							
CO 4								M	H	H	M	M

TOPICS COVERED

Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas

Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic

Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation& Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Case Studies: Domain specific case study to be undertaken which will explore the application of AI to that domain.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Rich and K. Knight," Artificial Intelligence", Tata McGraw Hill.
2. George F. Luger, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Pearson Education.
3. Russell &Norvig, "Artificial Intelligence 'a Modern Approach", Pearson Education.
4. Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", PHI.
5. E. Charnaik and D. McDermott," Introduction to Artificial Intelligence", Addison-Wesley Publishing Company.
6. Nils J Nilson, "Principles of Artificial Intelligence", Narosa Publishing Co.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSCI-300
COURSE TITLE: INDUSTRIAL PRACTICAL TRAINING
COURSE DESIGNATION: REQUIRED
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-0-4)
COURSE ASSESSMENT METHODS: The assessment of the course involves evaluation of the daily dairy maintained by the students during their industrial training, Report, evaluation from the organization in which training was done, followed by seminar and viva-voce.

COURSE OUTCOMES
After undergoing two months industrial training, Students will be able to

1. Apply their technical knowledge and engineering methods to solve real-life situations
2. Learn modern tools and contemporary ideas by practicing self-learning
3. Learn work ethics by Interacting with engineers and other professional groups thereby, increasing technical, interpersonal and communication skills, both oral and written

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSCI-300												
CO 1		H	H	H					H	M	M	
CO 2			M	M	H	H	H		M			
CO 3					M	M		H		H	H	H

TOPICS COVERED
The fundamental objective of Industrial Training is to prepare students for future employment in their chosen engineering discipline. Industrial Training allows students to practice what they have learned and to develop key professional attributes. During the 8 week Industrial training the students are exposed to the discipline of working in a professional engineering organization and develop an understanding of the functioning.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSCI-400
COURSE TITLE: PROJECT (Phase-I)
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-4-2)
COURSE ASSESSMENT METHODS: One mid semester presentation and one end semester presentation is conducted. The evaluation committee comprising of members of Department evaluate the project based on Report, Power Point Presentation, Project work and viva-voce.

COURSE OUTCOMES
On completion of the course students will be able to

1. Able to learn modern tools and contemporary ideas by practicing self-learning
2. Apply the skills and practices ofdesign and analysis during system development
3. Contribute the acquired knowledge in their area ofinterest in the form of process / system / product / prototypes as end product
4. Able to interact with multi-disciplinary teams and as an individual
5. Prepare the project reports based on prescribed format in a complete manner

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSCI-400												
CO 1	M	H				H	H	M			M	
CO 2		M	H	H							H	
CO 3	H		H		H			H				H
CO 4		H		M				M	H	H		
CO 5	M								H	H		

TOPICS COVERED
This is project work (phase-I) to be done by the students in the seventh semester. The evaluation committee of the Department shall evaluate the project for 2 credits assigned for the project. A report of the project work carried out during the semester shall be submitted at the end of the semester approved by the project guide and HOD.

VIII SEMESTER

S. No	Course Code	Course Title	Teaching Load			Credit	Course Type
			L	T	P		
1.	CSPC-402	System Programming and Compiler Design	3	0	0	3	PC
2.	CSPE-4XX	DE-IV	3	0	0	3	PE
3.	CSPE-4XX	DE-V	3	0	0	3	PE
4.	CSOE-XXX	OE-III	3	0	0	3	OE
5.	CSPC-422	System Programming and Compiler Design Laboratory	0	0	2	1	PC
6.	CSCI-400	Project (Phase-II)	0	0	8	4	CIC
7.	CSCI-424	Industrial Lecture	0	0	2	1	CIC
TOTAL			**12**	**0**	**12**	**18**	

*** Major Project will be allotted in 7th Semester, will be evaluated in 8th Semester**

DEPARTMENTAL ELECTIVE (DE)-IV

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-442	High Performance Computing	3	0	0	3
2.	CSPE-444	Soft Computing	3	0	0	3
3.	CSPE-446	Wireless Networks	3	0	0	3

DEPARTMENTAL ELECTIVE (DE)-V

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	CSPE-448	Social Network Analysis	3	0	0	3
2.	CSPE-450	Human Computer Interaction	3	0	0	3
3.	CSPE-452	Principles of Deep Learning	3	0	0	3

OPEN ELECTIVES (OE)-III

S. No	Course Code	Course Title	Teaching Load			Credit
			L	T	P	
1.	IDOE-005	Bio Informatics	3	0	0	3
2.	IDOE-006	Soft Computing Techniques	3	0	0	3

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-402
COURSE TITLE: SYSTEM PROGRAMMING AND COMPILER DESIGN
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand the design aspects of various different language processors
2. Identify theory and practice of compilation in the lexical analysis, syntax, and semantic analysis, code generation and optimization phases of compilation.
3. Exemplify and compare various function of parser along with its types for design of compiler.
4. Understand a parser such as a bottom-up SLR parser without using compiler-generation tools.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-402												
CO 1	M	M		H								
CO 2		H		H	M							
CO 3			M		L							
CO 4			M		L						L	

TOPICS COVERED

Introduction to Language Processor: Overview, fundamentals of language processing and symbol tables

Assemblers and Macros: Pass structure of assemblers, design of two pass assemblers using any hypothetical computer, design of macro processors

Linker and Loaders: Relocation and linking operation, design of linker, absolute and relocating loaders

Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

Lexical analysis: interface with input, parser and symbol table, token, lexeme and patterns. Difficulties in lexical analysis, Error reporting. Implementation. Regular definition, Transition diagrams, LEX.

Syntax analysis: CFGs, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.

Syntax directed definitions: inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.

Type checking: type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

Run time system: storage organization, activation tree, activation record, parameter passing, symbol table, dynamic storage allocation.

Intermediate code generation: intermediate representations, translation of declarations, assignments, control flow, boolean expressions and procedure calls. Implementation issues.

Code generation and instruction selection: issues, basic blocks and flow graphs, register allocation, code

generation, dag representation of programs, code generation from dags, peep hole optimization, code generator generators, specifications of machine.

TEXT BOOKS, AND/OR REFERENCE MATERIAL:

1. D M Dhamdhere, System Programming, TMH, 1st ed., 2012
2. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools , Addison-Wesley, 1988.
3. Fischer and R. LeBlanc. Crafting a Compiler , Benjamin Cummings, 1991..
4. C. Holub. Compiler Design in C , Prentice-Hall Inc., 1993.
5. Appel. Modern Compiler Implementation in C: Basic Design , Cambridge Press.
6. Fraser and Hanson. A Retargetable C Compiler: Design and Implementation , Addison-Wesley.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPC-422
COURSE TITLE: SYSTEM PROGRAMMING AND COMPILER DESIGN LABORATORY
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE

CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-2-1)

COURSE ASSESSMENT METHODS: Assignments for each topic to be evaluated in the lab, and final evaluation at the end which includes Viva Voce, Conduct of experiment.

COURSE OUTCOMES

After the course completion, the student would be able to:

1. Implement different language processors.
2. Implement various lexical analysis techniques in compiler design.
3. Implement various code optimization techniques in compiler design.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPC-422												
CO 1	M	H		M				M	M			
CO 2		M	H	L								

.

LIST OF PRACTICALS

1. Implementing of design of 2 pass assembler.
2. Implementing of design of macro processor.
3. Implementing of design of linker.
4. Implementation of Lexical Analyzer.
5. Implementation of Top Down Parser.
6. Implementation of Bottom Up Parser.
7. Design and Implementation of Two Pass Macro- Processor.
8. Use of LEX and YACC tools.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENTAL ELECTIVE (DE)-IV

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-442
COURSE TITLE: HIGH PERFORMANCE COMPUTING
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Recognize the feasibility of applying a soft computing methodology for a particular problem
2. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems
3. Apply genetic algorithms to combinatorial optimization problems
4. Effectively use existing software tools to solve real problems using a soft computing approach
5. Evaluate and compare solutions by various soft computing approaches for a given problem.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-442												
CO 1	M		M									
CO 2		H	M	H	H	M						
CO 3	M			M								
CO 4						H		M				
CO 5							H		H		H	

TOPICS COVERED

Parallel Processing Concepts: Levels of parallelism (instruction, transaction, task, thread, memory, function), Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc), Architectures: N-wide superscalar architectures, multi-core, multi-threaded

Parallel Programming with CUDA: Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: (Examples: IBM CELL BE, Nvidia Tesla GPU, Intel Larrabee Microarchitecture and Intel Nehalem microarchitecture), Memory hierarchy and transaction specific memory design, Thread Organization

Fundamental Design Issues in Parallel Computing: Synchronization, Scheduling, Job Allocation, Job Partitioning, Dependency Analysis, Mapping Parallel Algorithms onto Parallel Architectures, Performance Analysis of Parallel Algorithms

Fundamental Limitations Facing Parallel Computing: Bandwidth Limitations, Latency Limitations, Latency Hiding/Tolerating Techniques and their limitations,

Power-Aware Computing and Communication: Power-aware Processing Techniques, Power-aware Memory Design, Power-aware Interconnect Design, Software Power Management

Advanced Topics: Petascale Computing, Optics in Parallel Computing, Quantum Computers, Recent developments in Nanotechnology and its impact on HPC

TEXT BOOKS, AND/OR REFERENCE MATERIAL:

1. "Highly Parallel Computing", by George S. Almasi and Alan Gottlieb
2. "Advanced Computer Architecture: Parallelism, Scalability, Programmability", by Kai Hwang, McGraw Hill 1993
3. "Parallel Computer Architecture: A hardware/Software Approach", by David Culler Jaswinder Pal Singh, Morgan Kaufmann, 1999.
4. "Scalable Parallel Computing", by Kai Hwang, McGraw Hill 1998.
5. "Principles and Practices on Interconnection Networks", by William James Dally and Brian Towles, Morgan Kauffman 2004.
6. GPU Gems 3 --- by Hubert Nguyen
7. Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, © 2003.
8. Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series, © 2007.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-444
COURSE TITLE: SOFT COMPUTING
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand and describe soft computing techniques and their roles in building intelligent machines
2. Recognize the feasibility of applying a soft computing methodology for a particular problem
3. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems
4. Apply genetic algorithms to combinatorial optimization problems
5. Effectively use existing software tools to solve real problems using a soft computing approach

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-444												
CO 1	M		M									
CO 2		H	M	H	H	M						
CO 3	M			M								
CO 4						H		M				
CO 5							H		H		H	

TOPICS COVERED

Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen'sself organizing networks - Hopfield network.

Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition -Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees -Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls -Simulated annealing – Evolutionary computation.

Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction -Rank method - Rank space method.

SoftcomputingAnd Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998.
2. LaureneFausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
3. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, USA 1995.
4. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998.
5. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-446
COURSE TITLE: WIRELESS NETWORKS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand the wireless network basic concept with cell structure, modulation techniques, coding techniques and application
2. Understand research problems of wireless networks pose in disciplines such as signal processing, communications.
3. Apply MAC layer protocols and guided on the protocols developed.
4. Apply Routing protocols for wireless network.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-446												
CO 1	M			L								
CO 2		L		H	H	M						
CO 3	M	M		M								
CO 4						H		M				

TOPICS COVERED

Basics of wireless data transmission: frequencies & regulations, signal propagation, propagation models, modeling the propagation loss, multiplexing, spread spectrum, antennas, cellular systems.

Cellular system concept: Cellular Hierarchy, System Management, Cellular Reuse Pattern, Ways of increasing the system capacity, Channel assignment to the cells

Media Access Techniques: SDMA, FDMA, TDMA, CDMA, Aloha, CSMA, BTMA, DBTMA, FAMA, PUMA, DAMA, PRMA, C-PRMA, MACA, MACA-BI, MACAW, CARMA, CSMA/CA, polling.

Wireless LANs: IEEE 802.11 a/b/e/f/g/i, HIPERLAN, HomeRF, OpenAir.

Wireless PANs: Bluetooth: IEEE 802.15,. UWB PAN Technology

Wireless MAN (IEEE 802.16): IEEE 802.16-2004(802.16d) for fixed WiMAX and 802.16(802.16e) for mobile WiMAX

Wireless Telecommunication Systems: Basic architecture and working of followings: WLL, GSM, Handover process, GPRS, EDGE, UMTS, CDMA2000, 3G and 4G Systems,

Software defined Radio: The Software Radio concept, Minimum radio standard, Basic elements of Software Radio architecture

Emerging wireless technologies for mobile data networking.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Michel Daoud Yacoub, "Wireless Technology: Protocols, Standards, and Techniques", CRC Press, 2001.
2. K. Wesołowski, "Mobile Communication Systems", Wiley Publication, 2002.
3. J. Schiller, "Mobile Communications", Addison-Wesley, 2004.
4. J. Geier, "Wireless LAN", 2/e, SAMS, 2001.
5. G. Held, "Data Over Wireless Networks", McGraw-Hill, 2001.

DEPARTMENTAL ELECTIVE (DE)-V

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-448
COURSE TITLE: SOCIAL NETWORK ANALYSIS
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Understand common text mining and social network analytics techniques carried out in contemporary organizations.
2. Analyze the complexities of collecting, integrating, processing and managing text and network data from a wide range of internal and external sources.
3. Understand various analytical methods can be used to uncover the potential of text and network data to gain actionable insights and support marketing decisions.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-448												
CO 1	M		H	M			H					
CO 2		M				H	M	M				
CO 3		M					H		H		H	

TOPICS COVERED

Introduction
Introduction to Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web – Statistical Properties of Social Networks -Network analysis - Development of Social Network Analysis - Key concepts and measures in network analysis - Discussion networks - Blogs and online communities - Web-based networks.

Modeling and Visualization
Visualizing Online Social Networks - A Taxonomy of Visualizations - Graph Representation - Centrality-Clustering - Node-Edge Diagrams - Visualizing Social Networks with Matrix-Based Representations- Node-Link Diagrams - Hybrid Representations - Modelling and aggregating social network data – Random Walks and their Applications –Use of Hadoop and Map Reduce - Ontological representation of social individuals and relationships.

Mining Communities
Aggregating and reasoning with social network data- Advanced Representations - Extracting evolution of Web Community from a Series of Web Archive - Detecting Communities in Social Networks - Evaluating Communities – Core Methods for Community Detection & Mining - Applications of Community Mining Algorithms - Node Classification in Social Networks.

Text and Opinion Mining
Text Mining in Social Networks -Opinion extraction – Sentiment classification and clustering - Temporal sentiment analysis - Irony detection in opinion mining - Wish analysis - Product review mining – Review Classification – Tracking sentiments towards topics over time.

Tools for Social Network Analysis
UCINET – PAJEK – ETDRAW – StOCNET – Splus –R – NodeXL – SIENA and RSIENA – Real world Social Networks

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Charu C. Aggarwal, "Social Network Data Analytics", Springer; 2011.
2. Peter Mika, "Social Networks and the Semantic Web", 1st edition, Springer, 2007.
3. BorkoFurht, "Handbook of Social Network Technologies and Applications", 1st edition, Springer, 2010.
4. Guandong Xu, Yanchun Zhang and Lin Li, "Web Mining and Social Networking – Techniques and applications", 1st edition, Springer, 2011.
5. Giles, Mark Smith, John Yen, "Advances in Social Network Mining and Analysis", Springer, 2010.
6. Ajith Abraham, Aboul Ella Hassanien, VáclavSnáel, "Computational Social Network Analysis: Trends, Tools and Research Advances", Springer, 2009.
7. Toby Segaran, "Programming Collective Intelligence", O'Reilly, 2012.
8. Sule Gündüz-Ogüdücü, A. Şima Etaner-Uyar, "Social Networks: Analysis and Case Studies", Springer, 2014.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-450
COURSE TITLE: HUMAN COMPUTER INTERACTION
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the completion of the course, the students will be able to:
1. Gain knowledge on the interplay between humans, tasks, technology, and contexts
2. Gain knowledge on important human factors that affect HCI
3. Apply HCI principles, guidelines, methods, and techniques

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-450												
CO 1		M	H		M						H	
CO 2		H		M	M				H	H		
CO 3						M			H		M	

TOPICS COVERED

Introduction to Human-computer Interaction - Methodology for Designing User-computer Interfaces -Task analysis -Conceptual, semantic, syntactic, and lexical models.

Design of an interactive system - Interaction Styles -Question and answer -Form-based - Command language - Menus -Natural language -Direct manipulation -Virtual Reality - Augmented Reality -Other emerging interaction styles.

Design and Evaluation Process -Prototyping -Testing and evaluating interface designs - Guidelines and criteria for designing UI, UI Software and Specifications -Languages and tools for specifying and building interfaces -Dialogue independence –UIMSLanguages and software abstractions -Programming support tools -. Basic Interaction Tasks, Techniques, and Devices.

Human Performance -Scientific foundations for designing user interfaces -Visual presentation of information - Graphical design -Designing experiments - Introduction to Research in HumanComputer Interaction -Why do HCI research? -Research prototypes -Interdisciplinary nature of HCI research -Examples of HCI research.

New Interaction Techniques -New modes of human-computer communication -Voice Gesture - Eye movement - Tangible user interfaces -Brain-computer interfaces - Case Study.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Wilbert O Galitz, "The Essential Guide To User Interface Design", Wiley Dreamatech, 3 rd edition, 2007 Ben Shneidermann, "Designing The User Interface - Strategies for Effective HumanComputer Interaction", 4th Edition, Pearson Education Asia, 2004
2. Alan Dix, Janet Fincay, GreGoryd, Abowd, and Russell Bealg, "Human – Computer Interaction", 3rd edition, Pearson, 2003
3. Yvonne Rogers , Helen Sharp, and Jenny Preece, "Interaction Design: Beyond Human - Computer Interaction", 3rd edition, Wiley, 2011

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSPE-452
COURSE TITLE: PRINCIPELS OF DEEP LEARNING
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Analyze theory and approaches for learning with deep neural networks.
2. Acquire knowledge of different variants of deep learning and their typical applications.
3. Train and model with deep architectures as well as have hands-on experience in using deep learning frameworks for this purpose.
4. Implement basic versions of some of the core deep network algorithms.
5. Correlate it with machine learning approaches.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSPE-452												
CO 1	M		M									
CO 2		H	M		H	M						
CO 3	M			M								
CO 4					H			M				
CO 5							H		H		H	

Basics of Deep leaning

Deep learning architectures: Convolutional Neural Networks : Neurons in Human Vision-The Shortcomings of Feature Selection-Vanilla Deep Neural Networks Don't Scale-Filters and Feature Maps-Full Description of the

Convolutional Layer-Max Pooling-Full Architectural Description of Convolution Networks-Closing the Loop on MNIST with Convolutional Networks-Image Preprocessing Pipelines Enable More Robust Models-Accelerating Training with Batch Normalization-Building a Convolutional Network for CIFAR-10-Visualizing Learning in Convolutional Networks Leveraging Convolutional Filters to Replicate Artistic Styles-Learning Convolutional Filters for Other Problem Domains-Training algorithms.

Memory Augmented Neural Networks

Neural Turing Machines-Attention-Based Memory Access-NTM Memory Addressing Mechanisms-Differentiable Neural Computers-Interference-Free Writing in DNCs-DNC Memory Reuse-Temporal Linking of DNC Writes-Understanding the DNC Read Head-The DNC Controller Network Visualizing the DNC in Action-Implementing the DNC in TensorFlow-Teaching a DNC to Read and Comprehend.

Deep Reinforcement Learning

Deep Reinforcement Learning Masters Atari GamesWhat Is Reinforcement Learning?-Markov Decision Processes (MDP)-Explore Versus Exploit-Policy versus Value Learning-Pole-Cart with Policy Gradients-Q-Learning and Deep Q-Networks-Improving and Moving Beyond DQN.

Implementing Neural Networks in TensorFlow

Introduction to TensorFlow, working of TensorFlow, TensorFlow Alternatives, Installation, Creation and Manipulation of TensorFlow Variables-TensorFlow Operations-Placeholder Tensors-Sessions in TensorFlow-Navigating Variable Scopes and Sharing Variables-Managing Models over the CPU and GPU-Specifying the Logistic Regression Model in TensorFlow-Logging and Training the Logistic Regression Model-Leveraging TensorBoard to Visualize Computation Graphs and Learning-Building a Multilayer Model for MNIST in TensorFlow.

Applications

Deep learning for computer vision, Deep Learning Applications at the Enterprise Scale, Deep Learning Models for Healthcare Applications.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. Nikhil Buduma, Nicholas Locascio, "Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms", O'Reilly Media, 2017.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning, Adaptive Computation and Machine Learning series", MIT Press, 2017.

OPEN ELECTIVE (OE)-III

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: IDOE-005
COURSE TITLE: BIO INFORMATICS
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES
After the course completion, the student will be able to

1. Apply advanced skills to critically analyse and solve problems in biotechnology.
2. Use techniques, skills, and modern engineering tools necessary for engineering practice.
3. Analyze protein sequences, identify proteins, and retrieve protein structures from databases. View and interpret these structures. Understand homology modelling and computational drug design.
4. Design and conduct experiments, as well as to analyze and interpret data.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IDOE-005												
CO 1	M		M									
CO 2	L	H				M						L
CO 3	M			H	H							
CO 4					H	M			L			

TOPICS COVERED

Fundamentals of Bioinformatics and Information Technology: Introduction to bioinformatics, Experimental sources of biological data, publicly available databases, Operating systems - including Windows and UNIX, Networks - including the Intranets and the Internet

Analytical Science and Bioinformatics: High throughput sequencing, Experimental determination of protein structures, Gene expression monitoring, Proteomics, Metabolomics

Statistical Methods in Bioinformatics: Basic mathematics, Vectors and matrices, Multivariate statistics - particularly exploratory methods and pattern recognition

Bioinformatics Algorithms and Tools: Visualization of sequence data, Sequence alignment, Homology searching - including BLAST, Gene expression informatics, Introduction to gene finding

Applications and Commercial Aspects of Bioinformatics: Visualization of sequence data, Drug discovery, Genetic basis of disease, Personalized medicine and gene-based diagnostics, Legal, ethical and commercial ramifications of bioinformatics

Bioinformatics: The Business of Research: Research methodology (focusing on computer-based research), Case studies of areas of current bioinformatics research Routes to research funding (academic and commercial), Bioinformatics business models, Intellectual property rights

Software Engineering in Bioinformatics: Advanced programming using Java and BioJava, Advanced database work using SQL, Interfacings programs with databases. Data interoperability using XML

Principles of Programming and Databases using Java and SQL: Fundamental principles of programming, Object-oriented programming using Java, Introduction to databases using Oracle.

PERL programming: Data manipulation, File maintenance, Pipelining Packaging and interfacing system facilities

TEXT BOOKS, AND/OR REFERENCE MATERIAL:

1. Bioinformatics for Dummies, Jean-Michel Claverie, Cedric Notredame, 2003, John Wiley & Sons
2. Bioinformatics Computing, Bryan P. Bergeron, 2002, Prentice Hall
3. Introduction to Bioinformatics, Teresa Attwood, David Parry-Smith, 2001, Prentice Hall
4. Beginning Perl for Bioinformatics, James Tisdall,2001, O'reilly

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: IDOE-006
COURSE TITLE: SOFT COMPUTING TECHNIQUES
COURSE DESIGNATION: ELECTIVE
PRE-REQUISITES: NONE
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 3-0-0-3)
COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Apply soft computing based solutions to real world and engineering problems.
2. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems
3. Apply genetic algorithms to combinatorial optimization problems.
4. Apply neural networks to pattern classification and regression problems and compare solutions by various soft computing approaches for a given problem**.**

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IDOE-006												
CO 1.				M	H		H					
CO 2.		H		M			M	H				
CO 3.						M		M				
CO 4.	M					H			H			

TOPICS COVERED

Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen'sself organizing networks - Hopfield network.

Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition -Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees -Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls -Simulated annealing – Evolutionary computation.

Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction -Rank method - Rank space method.

SoftcomputingAnd Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

TEXT BOOKS, AND/OR REFERENCE MATERIAL:

1. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998.
2. LaureneFausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
3. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, USA 1995.
4. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998.
5. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: CSCI-400
COURSE TITLE: PROJECT (Phase-II)
COURSE DESIGNATION: REQUIRED
PRE-REQUISITES:
CONTACT HOURS/CREDIT SCHEME: (L-T-P-C: 0-0-8-4)
COURSE ASSESSMENT METHODS: One mid semester presentation and one end semester presentation is conducted. The evaluation committee comprising of members of Department and external experts evaluate the project on the basis of Report, Power Point Presentation, Project work and viva-voce.

COURSE OUTCOMES

1. Able to learn modern tools and contemporary ideas by practicing self-learning
2. Apply the skills and practices ofdesign and analysis during system development
3. Contribute the acquired knowledge in their area ofinterest in the form of process / system / product / prototypes as end product
4. Able to interact with multi-disciplinary teams and as an individual
5. Prepare the project reports based on prescribed format in a complete manner

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CSCI-400												
CO 1	M	H				H	H	M			M	
CO 2		M	H	H							H	
CO 3	H		H		H			H				H
CO 4		H		M				M	H	H		
CO 5	M								H	H		

TOPICS COVERED

This is project work (phase-II) to be done by the students in the eighth semester. The evaluation committee comprising of members of Department and external experts evaluate the project for 4 credits assigned for the project. A report of the project work carried out during the semester shall be submitted at the end of the semester approved by the project guide and HOD.