

As per NEP 2020

## **Four Years B.Sc. (Computer Science)**

Exit option

(after One year with Undergraduate Certificate in Computer Science/Two years with Undergraduate Diploma in Computer Science/ Three years with B.Sc.(Honours) Computer Science/ Four years B.Sc. .(Honours with Research) in Computer Science )

## **CURRICULUM**

(w.e.f. 2022-23)

**DEPARTMENT OF COMPUTER SCIENCE  
DOON UNIVERSITY, DEHRADUN**

## Course Structure

SEMESTER I						
Course Type	Course Code	Course Title	L	T	P	C
DSC	CSC101	Programming in C	3		1	4
DSC*	CSC102	Digital System Design	3		1	4
DSC**	CSC103	Computer Fundamentals	3	1		4
AECC		From University AECC Pool				2
VAC		From University VAC Pool				2
SEC	CSS101	Introduction to IoT	1		1	2
	CSS102	Open Office Tools			2	2
Generic	CSG101	Programming in C	3		1	4
	CSG102	Digital System Design	3	1		4
	CSG103	Computer Fundamentals	3	1		4
*This course can be taken from other department of Science stream. **This course can be taken from any department of any stream.					<b>Total Credits</b>	<b>22</b>
SEMESTER II						
Course Type	Course Code	Course Title	L	T	P	C
DSC	CSC151	Computer Architecture	3		1	4
DSC*	CSC152	Discrete Mathematics	3	1		4
DSC**	CSC153	Programming using Python	3		1	4
AECC		From University AECC Pool				2
VAC		From University VAC Pool				2
SEC	CSS151	Computer Hardware and Networking			2	2
	CSS152	Unix System Administration & Shell Programming			2	2
Generic	CSG151	Computer Architecture	3	1		4
	CSG152	Discrete Mathematics	3	1		4
	CSG153	Programming using Python	3		1	4
*Course can be taken from other departments of science stream. **Course can be taken from any department of any stream. <b>Exit option after one year with 44 credits to get Undergraduate Certificate in Computer Science</b>					<b>Total Credits</b>	<b>22</b>
SEMESTER III						
Course Type	Course Code	Course Title	L	T	P	C
DSC	CSC201	Data Structures	3		1	4
DSC*	CSC202	Programming using C++	3		1	4
DSC**	CSC203	Software Engineering	3	1		4
AECC		From University AECC Pool				2
VAC		From University VAC Pool				2

SEC	CSS200	Client side Web Technologies			2	2
Generic/ Elective	CSG201	Data Structures	3		1	4
	CSG202	Programming using C++	3		1	4
	CSG203	Software Engineering	3	1		4
	Elective	From the List of Electives of Computer Science				4
*Course can be taken from other department of science stream. **Course can be taken from any department of any stream.					<b>Total Credits</b>	<b>22</b>
<b>SEMESTER IV</b>						
<b>Course Type</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
DSC	CSC251	Database Management System	3		1	4
DSC*	CSC252	Numerical and Statistical Computing	3		1	4
DSC**	CSC253	Design and Analysis of Algorithms	3		1	4
AECC		From University AECC Pool				2
VAC		From University VAC Pool				2
SEC	CSS250	Server-side Web Technologies			2	2
Generic/ Elective	CSG251	Database Management System	3		1	4
	CSG252	Numerical and Statistical Computing	3	1		4
	CSG253	Design and Analysis of Algorithms	3		1	4
	Elective	From List of Electives Computer Science				4
*Course can be taken from other department of science stream. **Course can be taken from any department of any stream. <b>Exit option after Two years with 88 credits to get Undergraduate Diploma in Computer Science</b>					<b>Total Credits</b>	<b>22</b>
<b>SEMESTER V</b>						
<b>Course Type</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
DSC	CSC301	Operating Systems	3		1	4
DSC	CSC302	Theory of Computation	3	1		4
DSC*	CSC303	Optimization Techniques	3		1	4
DSE		From List of Electives Computer Science				4
GE		Elective from other Department				4
DSP	CSP301	Project 1			2	2
*Course can be taken from other department of science stream.					<b>Total Credits</b>	<b>22</b>
<b>SEMESTER VI</b>						
<b>Course Type</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
DSC	CSC351	Computer Networks	3		1	4
DSC	CSC352	Compiler Design	3		1	4
DSC*	CSC352	Programming in Java	3		4	4
DSE		From List of Electives Computer Science				4
GE		Elective from other Department				4

DSCP	CSCP350	Project 2			2	2
*Course can be taken from other department of science stream. <b>Exit option after Three years with 132 credits to award the degree of B.Sc. (Honours) in Computer Science if he/she earned 80 (from 18 DSC's and 2 DSE's) credits in Computer science discipline</b>			<b>Total Credits</b>		<b>22</b>	
<b>SEMESTER VII</b>						
<b>Course Type</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
DSC	CSC401	Computer Graphics	3		1	4
DSE/GE		From List of Electives of Computer science/Others				4
DSE/GE		From List of Electives of Computer science/Others				4
DSE/GE		From List of Electives of Computer science/Others				4
DSCP	DSP401	UG Dissertation Part 1			6	6
<b>Total Credits</b>					<b>22</b>	
<b>SEMESTER VIII</b>						
<b>Course Type</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
DSC	CSC451	Artificial Intelligence	3		1	4
DSE/GE		From List of Electives of Computer science/Others				4
DSE/GE		From List of Electives of Computer science/Others				4
DSE/GE		From List of Electives of Computer science/Others				4
DSCP	DSP450	UG Dissertation Part 2			6	6
<b>Exit option after Four years with 176 credits to award the degree of B.Sc.(Honors with Research) in Computer Science Major(Computer Science) and Minor(Other Discipline)</b>			<b>Total Credits</b>		<b>22</b>	

Electives can be taken from list of Electives or MOOC courses approved by authority.

### List of Electives:

Course Code	Course Title	L	T	P	C
CSE101	Big Data Analytics	3		1	4
CSE102	Business Intelligence	3		1	4
CSE103	Introduction to IOT	3		1	4
CSE104	Modeling and Simulation	3		1	4
CSE105	Operation Research	3	1		4
CSE106	Biometrics	3	1		4
CSE107	Computer Vision and Pattern Recognition	3		1	4
CSE108	Digital Image Processing	3		1	4
CSE109	Introduction to Cloud Computing	3		1	4

CSE110	Natural Language Processing	3		1	4
CSE111	Introduction to Data Science	3		1	4
CSE112	Digital Marketing	3		1	4
CSE113	Fuzzy Logic	3		1	4
CSE114	Data Mining and Warehousing	3		1	4
CSE115	Digital Signal Processing	3		1	4
CSE116	Probability and Statistical Inference	3	1		4
CSE117	Cryptography and Network Security	3		1	4
CSE118	Advanced Algorithms	3		1	4
CSE119	Information Theory and Coding	3	1		4
CSE120	Machine Learning	3		1	4
CSE121	Neural Networks	3		1	4
CSE122	Mobile Adhoc Networks	3		1	4
CSE123	Cloud Architecture	3		1	4
CSE124	Parallel and Distributed Computing	3		1	4
CSE125	Advanced Graph Theory	3	1		4
CSE126	Cyber Forensics	3	1		4
CSE127	Software Defined Networks	3	1		4
CSE128	Evolutionary Algorithms	3		1	4
CSE129	Block chain Technology	3		1	4
CSE130	Quantum Computing	3	1		4
CSE131	Research Methodology	4			4
CSE132	Cloud Security	3	1		4

This list may be appended from time to time.

## Learning Outcomes

### 1. Program Outcomes

#### Knowledge outcomes:

After completing B.Sc. . (Honours) Computer Science Program students will be able to:

**PO1:** To develop problem solving abilities using a computer.;

**PO2:** To prepare necessary knowledge base for research and development in Computer Science.

#### Skill outcomes:

After completing B.Sc. Computer Science Program students will be able to:

**PO3:** To build the necessary skill set and analytical abilities for developing computer based solutions for real life problems.

**PO4:** communicate scientific information in a clear and concise manner both orally and in writing.

**PO5:** To train students in professional skills related to Software Industry.

#### Generic outcomes:

Students will

**PO6:** Have developed their critical reasoning, logic judgment and communication skills.

**PO7:** Augment the recent developments in the field of IT and relevant fields of Research and Development.

**PO8:** Enhance the scientific temper among the students so that to develop a research culture and Implementation the policies to tackle the burning issues at global and local level. Students will

## 2. Program Specific Outcomes

**PSO1:** Students get knowledge and training of technical subjects so that they will be technical professional by learning C programming, Relational Database Management, Data Structure, Software Engineering, Graphics, Java, PHP, Networking, Theoretical Computer Science, System programming, Object Oriented Software Engineering.

**PSO2:** Students understand the concepts of software application and projects.

**PSO3:** Students understand the computer subjects with demonstration of all programming and theoretical concepts with the use of ICT.

**PSO4:** Development of in-house applications in terms of projects

**PSO5:** Students will build up programming, analytical and logical thinking abilities.

**PSO6:** Aware them to publish their work in reputed journals

**PSO7:** To make them employable according to current demand of IT Industry and responsible citizen.

**Syllabus**  
**(Semester I – Semester VIII)**

# SEMESTER I

## CSC101: Programming in C

L | T | P (3 | 0 | 1)

**Prerequisites:** Nil

**Course Outcome:** By the end of the course, students should be able to:

- Understand basic concepts of C Programming
- Attempt algorithmic solutions to problems
- Design and code moderate sized programs, and
- Read, understand and modify code written by others.

**Course Outline:**

**C Language Fundamentals:** Computer program, Programming Languages, compilation process, source code, object code, executable code. gcc/g++ compilers. Structure of a C Program, character set, keywords, identifiers, data types, constants, variables, expression, statement, comments. Input-Output in C, Operators & Expressions– relational, logical, assignment, increment/decrement, sizeof, bitwise, precedence and associativity, type conversion, typecast.

**Functions:** Modular programming Library functions, C header files– stdio.h, conio.h, time.h, stdlib.h, process.h. User-defined functions– function declaration, definition and call, formal and actual arguments, scope and lifetime of variables, recursion, iterative vs. recursive way of problem solving, macros without arguments, macros with arguments, macros vs. functions.

**Arrays, Pointers and Strings:** Arrays– one-dimensional and two-dimensional arrays, array as function argument. Pointers– address operator, dereference operator, pointer arithmetic, pointers and arrays, pointer to an array, array of pointers, pointers as function arguments, returning multiple values from a function, function returning pointers, pointers to functions, dynamic memory allocation, creation of one-dimensional and two-dimensional dynamic arrays, function pointers, memory leaks. Strings– string constant, string variable, reading/writing strings, format specifier for string, string manipulation using built-in and user-defined functions, array of strings- 2D array of characters, array of pointers to character.

**Structure and Union:** User-defined data type, structure, union, declaration and initialization, dot operator, copying and comparing structures, size of structure, array within structure, array of structures, nested structure, pointer within structure, pointer to structure, arrow operator, structure as function argument, function returning structure, self-referential structures.

**File management:** Opening a file in various modes, concept of buffer, FILE structure, pointer to FILE, read/write operations on file, closing a file, random access to file, command line arguments.

**Textbooks:**

1. B. W. Kernighan, D. M. Ritchie, “The C Programming Language”, Prentice Hall, 1990.
2. Herbert Schildt, “C: The Complete Reference”, McGraw Hill Education, 4<sup>th</sup>ed., 2000.
3. Stephen Prata, “C Primer Plus”, Sams Publishing, 5<sup>th</sup>ed.
4. Yashavant Kanetkar, “Let Us C”, BPB Publications, 13<sup>th</sup>ed., 2013.



**Prerequisites:** Nil

**Course Outcome:** By the end of the course, students should be able to:

- Understand the Boolean expressions and their realizations.
- Design combinational and sequential building blocks.
- Use these building blocks to design digital circuits.
- Learn Verilog to design/model digital system.

**Course Outline:**

**Number Systems:** Digital Computer, Number Systems– Number Representation, Binary, Octal, Hexadecimal, Unsigned and Signed Numbers, Arithmetic Operations, Fixed point and Floating Point representations, Use of different number systems in digital design, Binary Codes– BCD, EBCDIC, ASCII, Unicode, Gray codes, Excess-3, Error Detection and Correction codes.

**Boolean Algebra and Digital Logic:** Boolean Algebra, Truth Tables, Logic Gates– AND, OR, NOT, NAND, NOR, XOR, Digital Circuit Characterization– Fan-in/Fan-out, Switching Functions, Boolean Functions– Sum of Product and Product of Sum, Karnaugh Maps, Minimization of Boolean Functions, K-Maps with Don't Care, Multiple Output Functions.

**Combinational Logic & Circuit Design:** Combinational Circuits– Analysis and Design Procedures, Circuits for Arithmetic Operations– Code Conversion, Binary Adder, Binary Subtractor, Decimal Adder, Magnitude Comparator, Decoders and Encoders, Multiplexers and Demultiplexers, Introduction to HDL– HDL Models of Combinational circuits, Introduction to VHDL and Basic VHDL Modelling.

**Sequential Logic & Circuit Design:** Sequential Elements– Latches and Flip Flops– Analysis and Design Procedures, Application of Flip Flops– Clock Generation, Counters, Registers, Shift Registers, State Machine Concepts– State Diagram, State Table, State Assignment and State Reduction/Minimization, HDL for Sequential Logic Circuits. Asynchronous Sequential Logic– Analysis and Design of Asynchronous Sequential Circuits, Reduction of State and Flow Tables, Race-free State Assignment, Hazards.

**Memory & Programmable Logic Devices:** Memory hierarchy, Memory technologies– Cache memory, Virtual memory, TLBs, Design of memory– ROM and RAM, Programmable Logic Array (PLA), Programmable Array Logic (PAL). Different Logic families– TTL, ECL, MOS, CMOS– operation, design and specification.

**Input-Output Organization:** Peripheral Devices, I/O Modules, Isolated vs. Memory-Mapped I/O, Asynchronous Data Transfer, Modes of Transfer– Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA) controller, I/O Processors (IOP).

**Textbooks:**

1. M. Morris Mano, Digital Logic and Computer Design, Pearson Education, 1<sup>st</sup>ed., 2004.
2. M. Morris Mano, Michael D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, Pearson Education, 5<sup>th</sup>ed., 2014.

**References:**

1. David A. Patterson, John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 5<sup>th</sup>ed., 2016.
2. M. Morris Mano, Computer System Architecture, Pearson Education, 3<sup>rd</sup>ed., 2008.
3. John F. Wakerly, Digital Design Principles and Practices, Pearson Education, 4<sup>th</sup>ed., 2007.
4. Charles H. Roth Jr, Fundamentals of Logic Design, Jaico Publishing House, 5<sup>th</sup>ed., 2003.
5. Donald D. Givone, Digital Principles and Design, Tata McGraw Hill, 2003.

**Prerequisites:** Nil

**Course Outcome:** By the end of the course, students should be able to:

- Understand different components of a computer.
- Differentiate between hardware and software.
- Learn the basic concepts of operating systems, networking, internet.
- Understand various advanced and emerging technologies.

**Course Outline:**

**Introduction to Computer:** Definition, Computer Hardware & Computer Software Components: Hardware – Introduction, Input devices, Output devices, Central Processing Unit, Memory- Primary and Secondary. Software Introduction, Types – System and Application. Computer Languages: Introduction, Concept of Compiler, Interpreter & Assembler Problem solving concept: Algorithms – Introduction, Definition, Characteristics, Limitations, Conditions in pseudo-code, Loops in pseudocode.

**Operating system:** Definition, Functions, Types, Classification, Elements of command based and GUI based operating system. Computer Network: Overview, Types (LAN, WAN and MAN), Data communication, topologies.

**Internet:** Overview, Architecture, Functioning, Basic services like WWW, FTP, Telnet, Gopher etc., Search engines, E-mail, Web Browsers.

**Internet of Things (IoT):** Definition, Sensors, their types and features, Smart Cities, Industrial Internet of Things.

**Block chain:** Introduction, overview, features, limitations and application areas fundamentals of Block Chain.

**Crypto currencies:** Introduction, Applications and use cases

**Cloud Computing:** It nature and benefits, AWS, Google, Microsoft & IBM Services

**Emerging Technologies:** Introduction, overview, features, limitations and application areas of Augmented Reality, Virtual Reality, Grid computing, Green computing, Big data analytics, Quantum Computing and Brain Computer Interface

**Textbooks:**

1. Rajaraman V., “Fundamentals of Computers”, Prentice-Hall of India.
2. Norton P., “Introduction to Computers”, McGraw Hill Education.
3. Goel A., “Computer Fundamentals”, Pearson.
4. Balagurusamy E., “ Fundamentals of Computers”, McGraw Hill
5. Thareja R., “Fundamentals of Computers”, Oxford University Press.
6. Bindra J., “The Tech Whisperer- on Digital Transformation and the Technologies that Enable it ”, Penguin

**Prerequisites:** Basic programming skills.

**Course Outcome:** By the end of the course, students should be able to:

- Understand the basics of networking and communication protocols.
- Integrate sensors and actuators with Arduino.
- Implement IoT with Raspberry Pi.

**Course Outline:**

**Internet of Things (High level understanding):** Definition, ideas, vision and examples, IoT devices vs. computers, Concepts and Technologies behind the IoT, Technological trends leading to IoT, Societal Impact of IoT– Benefits, Risks, Privacy and Security.

**Internet of Things (Low level details):** Embedded Systems– Meaning, Structure and Components, Interfacing and Interaction with the physical world– Sensors and Actuators.

**Design and Implementation:** Hardware and Software components of IoT devices– Integrated Circuits, Microcontrollers, IoT Operating System.

**Networking and the Internet:** Basics of networking, Internet Structure, Internet Protocols, MANETs.

**Textbooks:**

1. Arshdeep Bahga, Vijay Madisetti, “Internet of Things: A Hands-On Approach”, 1<sup>st</sup> ed., 2015.
2. Adrian McEwen, Hakim Cassimally, “Designing the Internet of Things”, Wiley, 2015.
3. Simon Monk, “Programming Arduino: Getting Started with Sketches”, McGraw Hill Education, 2nd ed., 2016.
4. Peter Waher, “Learning Internet of Things”, Packt Publishing, 2015.

**Prerequisites:** Nil

**Course Outcome:** By the end of the course, students should be able to:

- Effectively use word processors and spreadsheets.
- Understand and work with databases to generate reports.
- Create presentations.

**Course Outline:**

**MS Word:** Working with documents, Opening & Saving files, Documents editing, Page Formatting – Margins, Layout, Border & Shading, Columns, Header & Footer, Footnotes & end notes. Paragraph setting, Alignments, Indents, Bullets & Numbering. Page break, Column break and line break, Creating sections & frames, Anchoring & Wrapping, Setting Document styles, Table of Contents, Index, Page Numbering, Date & Time, Author etc. Importing & Exporting documents, File sharing, Ruler, special symbols, Creating Master Documents, Creating Tables, Table settings, Borders, Alignments, Drawing – Inserting ClipArt, Pictures/Files etc., Tools – Word Completion, Spell Checks, Mail merge, Templates, Using Wizards, Tracking Changes, Security, Digital Signature, Printing Documents, Shortcut keys.  
**Open Office Writer:** Introduction to Open Office Suite, Selecting the application package, Write, edit and review documents, Control page layout, Use templates and styles, Fields, Work with graphics.

**MS Excel:** Spread Sheet & Applications, Working with Spreadsheets, Referring Cells & Selecting Cells – Shortcut Keys. Entering & Deleting Data, Filling Continuous rows, columns, Highlighting values, Find, Search & replace, Data from external files, Frames, Clipart, Pictures, Files, Functions, Formula. Hiding/ Locking cells, Anchoring objects, Sorting, Filtering, Creating Charts, Pivot Tables, Security.  
**Open Office Calc:** Introduction, Creating Worksheet & Workbooks, Charts & Graphs, Using styles & templates in calc, Using graphics in calc, Printing, exporting & e-mailing, Formulas & Functions, Data Analysis, Linking calc data, Sharing & reviewing documents, Calc macros, Keyboard shortcuts.

**MS Access:** Introduction to a Database, Starting Access, Access Screen, Creating a New Database, Creating Tables, Working with Forms, Creating queries, Finding Information in Databases, Creating Reports, Types of Reports, Importing data from other databases.  
**OpenOffice Base:** Introduction, Database Concepts – Creating Database, Tables, Views, Accessing other data sources, Creating Forms and Subforms, Creating Queries using Wizard and Design View, Creating Static and Dynamic Reports.

**MS Power point:** Introduction to presentation, templates, Setting backgrounds, presentation layouts. Creating and Formatting Presentation – Adding text, style, color, gradient fills, Arranging objects, Adding Header & Footer, Slide Background, Slide layout. Adding Graphics – pictures, movies, tables, animation & transition effect.  
**OpenOffice Impress:** Introduction, Creating Presentation, Formatting, Slide Masters, styles and Templates, Text Formatting, Creating and Formatting Graphic objects, Inserting spreadsheets and charts; Slides, Notes and Handouts, Printing, e-mailing and Exporting a presentation, Customizing Impress.

**Textbooks:**

1. R. Gabriel Gurley, “A Conceptual Guide to OpenOffice.org 3”, Createspace Independent Publishing, 2008.
2. John Walkenbach, Herb Tyson, Faithe Wempen, “Microsoft Office 2007 Bible”, 2007.

**References:**

1. OpenOffice.org Writer Guide ([link](#)).
2. OpenOffice.org Calc Guide ([link](#)).
3. OpenOffice.org Base Guide ([link](#)).
4. OpenOffice.org Impress Guide ([link](#)).

## **SEMESTER II**

### **CSC151: Computer Architecture**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basic Knowledge of Computers, Digital Electronics

#### **Course Outcome:**

1. To develop understanding of Computer Models and its usage.
2. To develop understanding of ALU Design.
3. To conceptualize the understanding of Control Unit design, Memory, IPC, Control Design.
4. To develop understanding of Memory & Input/output organization Overview.

**Register Transfer and Microoperations:** Components of a computer system, Von Neumann architecture, Computer System Interconnection, Register Transfer Language, Register Transfer, Bus and Memory Transfers, Microoperations– Arithmetic, Logic and Shift.

**Central Processing Unit:** Computer Arithmetic– ALU, Integer Representation and Arithmetic, Floating-Point Representation and Arithmetic, Decimal Arithmetic, CU Implementation– Hardwired and Multi Programmed, Multiplier Control Unit, CPU Control Unit, Instruction Set Architecture– Addressing Modes and Design, CISC and RISC paradigm, Basic MIPS implementation– Building data path– Control Implementation scheme.

**The 8086 microprocessor:** Introduction to 8086 – Microprocessor architecture – Addressing modes, Instruction set and assembler directives- 8086 signals – Basic configurations – System bus timing –System design using 8086- System Bus Structure- Memory Interfacing, I/O interfacing, Parallel communication interface – Serial communication interface – D/A and A/D Interface.

**Parallel Processing concepts:**Instruction level parallelism, Parallel processing challenges, Flynn’s classification, Pipelining, Vector Processing, Superscalar processors, Multi-core Processors– Multithreading, Multicore processor Architecture, Multiprocessor configurations – Coprocessor, closely coupled and loosely coupled configurations,Cache Coherence Protocols, Synchronization, Memory Consistency.

#### **Textbooks:**

1. M. Morris Mano, “Computer System Architecture”, Pearson, 3<sup>rd</sup>ed., 2007.
2. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 5<sup>th</sup>ed., 2002.
3. Yu-Cheng Liu, Glenn A. Gibson, “Microcomputer Systems: The 8086 / 8088 Family: Architecture, Programming and Design”, Prentice Hall of India, 2<sup>nd</sup>ed., 2007.

#### **References:**

1. David A. Patterson, John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Morgan Kaufmann, 3<sup>rd</sup>ed., 2007.
2. John L. Hennessy, David A. Patterson, “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann, 5<sup>th</sup>ed., 2012.

3. John P. Hayes, “Computer Architecture and Organization”, McGraw-Hill Education, 2<sup>nd</sup>ed., 1998.
4. William Stallings, “Computer Organization and Architecture – Designing for Performance”, Pearson Education, 6<sup>th</sup>ed., 2003.
5. Douglas V. Hall, “Microprocessors and Interfacing, Programming and Hardware”, TMH, 2012.

### **Computer Architecture Lab**

**Introduction to Assembly Language Programming:** The Need and Use of the Assembly Language, Assembly Program Execution, An Assembly Program and its Components, The Program Annotation, Directives, Input Output in Assembly Program, Interrupts, DOS Function Calls (Using INT 21H), The Types of Assembly Programs, COM Programs, EXE Programs, How to Write Good Assembly Programs.

**Assembly Language Programming (Part-I):** Simple Assembly Programs, Data Transfer, Simple Arithmetic Application, Application Using Shift Operations, Larger of the Two Numbers, Programming with Loops and Comparisons, Simple Program Loops, Find the Largest and the Smallest Array Values, Character Coded Data, Code Conversion, Programming for Arithmetic and String Operations, String Processing, Some More Arithmetic Problems.

**Assembly Language Programming (Part-II):** Use of Arrays in Assembly, Modular Programming, the stack, FAR and NEAR Procedures, Parameter Passing in Procedures, External Procedures, Interfacing Assembly Language Routines to High Level Language, Programs, Simple Interfacing, Interfacing Subroutines with Parameter Passing, Interrupts, Device Drivers in Assembly.



**Pre-requisites:** Basic Knowledge of Mathematics

**Course outcomes:**

1. Understand and interpret the fundamental mathematical structures like Set theory, Relation and Functions
2. Write recursive definitions of sequences and collections of objects
3. Understand the concepts and applications of vector algebra
4. Understand and interpret the basic concepts of Graph Theory
5. Apply the use of graph theory concepts solving various Computer Science and Engineering problems.

**Logic and Proofs:** Propositional Logic– Binary logic and propositions, Propositional Variables, Truth table, Logical connectives– Negation, Conjunction, Disjunction, Conditional, Biconditional, Universal connectives, Well-formed Formulas, Tautology, Contradiction and Contingency, Propositional Equivalences, Duality, Predicate Logic– Predicates, Quantifiers– Existential and Universal quantifier, Predicate formulas, Equivalence of formulas involving quantifiers, Normal forms– CNF/DNF, PCNF/PDNF, Normal forms for First Order Logic– Prenex Normal Form, Rules of Inference.

Proof Techniques– Introduction to Proof, Definitions– Theorem, Lemma, Corollary and Conjecture, Methods of Proof– Direct Proofs, Indirect Proofs– Proof by Contraposition, Proof by Contradiction, Proof by Cases, Mathematical Induction.

**Counting and Recurrence:** Set Theory, Countable and Countably Infinite Sets, Pigeonhole Principle, Permutation and Combination, Principle of Inclusion and Exclusion, Generating functions– Definition, Generating Permutations and Combinations. Recurrence– Recurrence Relations, Linear Recurrence Relations with constant coefficients and their solution, Solving Linear Recurrence Relations using Generating Functions.

**Binary and Ordered Relations:** Binary Relation, Properties of Binary Relations– Reflexive, Symmetric and Transitive Relation, Equivalence Relation, Closure of Relations– Reflexive, Symmetric and Transitive Closure, Warshall’s algorithm, Ordered Relation– Partial Order and Posets, Hasse diagram of Poset, Maximal, Minimal, Maximum and Minimum of poset, glb and lub, Isomorphic ordered set, Well ordered set, Lattice, Properties of lattice, Distributed and Complemented lattice, Applications of Lattice, Topological Sort.

**Graph Theory:** Definition of Graph, Types– Directed and Undirected Graph, Complete Graph, Bipartite Graph, Multigraph, Weighted Graph, Graph Representation– Adjacency matrix and Adjacency list, Graph Isomorphism, Connectivity and Path, Euler and Hamiltonian Paths and Circuits, Shortest path– Dijkstra’s algorithm, Planar Graph, Euler’s theorem for Planar Graphs, Graph Coloring.

Trees– Basic terminology and properties, Tree Traversal– Inorder, Preorder and Postorder, Expression Trees– Infix, Prefix and Postfix notations, Spanning Trees– Kruskal’s and Prim’s algorithms for Minimum Spanning Trees (MST).

**Algebra:** Definition and elementary properties of Semigroups, Monoids, Groups, Subgroups, Generators and Cyclic group, Permutation group, Cosets, Lagrange's Theorem, Rings, Integral Domains and Fields.

**Textbooks:**

1. Kenneth H. Rosen, "Discrete Mathematics and Its Applications", Tata McGraw Hill, 7<sup>th</sup>ed., 2012.
2. C. L. Liu, "Elements of Discrete Mathematics", McGraw Hill, 2<sup>nd</sup>ed., 1986.
3. Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, "Discrete Mathematical Structures", Pearson Education, 6<sup>th</sup>ed., 2008.
4. J. P. Tremblay, R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill, 1<sup>st</sup>ed., 2001.
5. Susanna S. Epp, "Discrete Mathematics with Applications", 4<sup>th</sup>ed., 2010.

**Pre-requisite:** Basics of Mathematics and computer knowledge

**Course outcomes:**

1. Demonstrate a clear understanding of the basic python programming concepts, data structures & regular expressions.
2. Implement file handling operations & understand OOPS concepts using Python
3. Understand modules & implement web development framework
4. Understanding role of python in IOT and analyze data with the help of numpy and pandas

Computers and Programs: The Universal Machine, Program Power, what is Computer Science? Hardware Basics, Programming Languages, Problem Solving, Methods of Describing a Program, Flow Charts for Structured Programming, Computer Model, Procedures and Environments, Executing Procedure Calls and Returns, Global and Local Variables. Interfacing Procedures – Introduction – Reference Parameters – Automatic Protection of Arguments – Expression as Arguments in a Procedure Call – Function Procedures – Name Parameters – Parameters that Stand for Procedures and Functions – Recursion. Algorithm Design – Problem Solving Aspect – Top Down Design, Implementation of Algorithms – Fundamental Algorithms (Discuss the Design of Algorithms only). Program, Characteristics of a good program, Modular Approach, Programming style, Documentation and Program Maintenance, Compilers and Interpreters, Running and Debugging Programs, Syntax Errors, Run-Time Errors, Logical Errors, Concept of Structured Programming.

Inside a Python Program: Names, Expressions, Outputs, Assignment, Loops and Example Programs. Computing with Numbers and Strings: Numeric Data Types, The String Data Types, Input-Output as String Manipulation, File Processing, Objects and Graphics: Simple Graphics Programming, Using Graphical Objects, Choosing Coordinates, Interactive Graphics.

Defining Functions: Parameters, Processing, Return Values and Program Structure, Decision Structures: Simple Decisions, Two-Way Decisions, Multi-Way Decisions, Exception Handling, Loop Structures and Booleans.

Simulation and Design: Pseudorandom Numbers, Top-Down Design, Bottom-Up Implementation, Other Design Techniques, Defining Classes: Data Processing with Class, Objects and Encapsulation.

Data Collections: Applying Lists, Lists of Objects, designing with Lists and Classes, Case Study of Python Calculator, Non-Sequential Collection.

**Textbooks:**

1. John M. Zelle, “Python Programming: An Introduction to Computer Science”, Franklin, Beedle & Associates Publishers, 3<sup>rd</sup>ed.
2. R.G. Dromey, “How to Solve It by Computer”, Pearson Education, 5<sup>th</sup>ed., 2007.

**References:**

1. Richard L. Halterman, “Learning to Program with Python”, Southern Adventist University Publisher, 2011.
2. C.H. Swaroop, “A Byte of Python”, Ebsshelf Inc., 2013.
3. Allen Downey, “Think Python: How to Think Like a Computer Scientist”, Green Tea Press, 2012.

**Prerequisites:** Nil

**Course Outcome:** By the end of the course, students should be able to Understand about:

- basic peripheral devices and configuration of those
- Networking devices and related configuration
- Computer /Laptops and related devices and configurations
- Printing devices and related configuration with other devices.

**Course Outline:**

**Microprocessor System :** Introduction of System overview, Introduction to Processors, Memory Interfacing, Interfacing I/O Devices, Interfacing Data Converters, Display Interface, Serial I/O and Data Communication, Higher level Processors

**Introduction to PC Architecture:** Study of PC-AT/ATX System, Pentium, Core, Core 2 Duo, Core 2 Duo, I3, I5, I7 Processor Basics of Processor and CPU, Block Diagram of Computer and Computer Generation, Motherboards, Chipset and Controllers, BIOS and the Boot Process, Computer Memory.

**Internal Components :** IDE and SATA Devices: Hard Disk Drive and CD/DVDs Drives, SCSI Devices, Floppy Disk, Zip Drive, Backup Drive, Expansion Cards- LAN Card, IDE Card , VGA and SVGA Cards, Sound Card, Interface Cards, I/O cards, Video Cards, USB Card, Fire-Wire Cards, Internal Ports, Cables and Connector Types.

**External Components :** Monitors:- CRT, LCD and LED Displays, Printers:- Dot-Matrix Printer, Inkjet Printer, Laser Printer, Scanner:- Photo Scanner, Documents Scanner, Bar Code Scanner, Keyboards, Mouse, External Modem, Ports and Connectors, Batteries, Power supply, Pen Drives, SCSI interface devices, Laptop Computers, Digital Advance storage technology.

**Network Components :** Introduction of Network Cable like UTP, STP, Fiber Optics, Hub, Unmanageable Switch, Manageable Switch, Router, Modem, Wi-Fi, Access Point, PCI Wireless Card, USB Wireless Device, Print Server, USB Network Sharer, Backup Device, Server Hardware etc.

**Textbooks:**

- (1) Microprocessor Architecture Programming and Application with the 8085 Ramesh Gaonkar Penram International Publication
- (2) Electronics and Radio Engineering M.L. Gupta Dhanpat rai & Sons, New Delhi
- (3) PC AND CLONES Hardware, Troubleshooting and Maintenance B. Govinda rajalu, Tata Mc-graw-Hill Publication

(4) PC Troubleshooting and Repair Stephen J. Bigelow Dream tech Press, New Delhi

**CSS-151: Unix System Administration and Shell Programming**

**L | T | P (0 | 0 | 2)**

**Pre-requisite:** Basic knowledge of Operating System concepts along with DOS (Disk Operating System) is essential.

**Course outcomes:**

1. Understand the UNIX operating system and its memory management, input/output processing, internal and external commands.
2. Learn the File Systems and Process Management of UNIX.
3. Learn and explore the use of operating system utilities such as text editors..
4. Understand Shell Scripting and Shell Programming.
5. Capability to name and state the function of unix commands

**Getting Started:** File Management, Directories, File Permission, Environment, Basic Utilities, Pipes & Filters, Processes, Communication, The vi Editor.

**Shell Programming:** What is Shell? What is shell script, Writing and executing the shell script, Shell variable (user defined and system variables), System calls, Using system calls, Pipes and Filters, Decision making in Shell Scripts (If else, switch), Loops in shell, Functions, Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep), Special Variables, Using Arrays, Basic Operators, Shell Substitutions, Quoting Mechanisms, IO Redirections, Shell Functions, Manpage Help, Simple filters, the process, TCP/IP Networking Tools, The X Window System, Filters using regular expressions.

Customizing the environment, Programming in C/C++ in UNIX environment, Shell programming, Perl, System and network administration.

**Advanced Unix:** Regular Expressions, File System Basics, User Administration, System Performance, System Logging, Signals and Traps.

**Textbooks:**

1. Sumitabha, Das, “Unix Concepts and Applications”, Tata McGraw-Hill Education, 2006.
2. Evi Nemeth, Garth Synder, Trent R. Hein, Ben Whaley, “Unix and Linux System Administration Handbook”, Pearson Education, 2<sup>nd</sup>ed., 2010.
3. Venkatesh Murthy, “Introduction to Unix & Shell Programming”, Pearson Education, 2006.
4. Stephan Prata, “Advanced Unix – A Programmers Guide”, BPB PUB.
5. Brian W. Kernighan, Rob Pike, “The Unix Programming Environment”, PHI.
6. W. Richard Stevens, Stephen A. Rago, “Advanced Programming in the UNIX Environment”, Pearson Education, 2<sup>nd</sup>ed., 2011.
7. The UNIX System Manuals.

## SEMESTER III

### CSC201: Data Structures

**L | T | P (3 | 0 | 1)**

**Prerequisites:** CSC 101\_Programming in C

**Course Outcome:** By the end of the course, students should be able to:

- Understand different problem solving techniques
- Differentiate between sequential lists and linked lists.
- Learn and implement searching and sorting techniques.
- Understand the fundamentals and application of trees.
- Describe the graph terminologies and applications.

**Introduction to problem solving approach:** Algorithmic solution, analysis of algorithms– space and time complexity, asymptotic analysis, step counting and time complexity analysis.

**Sequential Lists and Linked Lists:** Sequential lists, arrays– single and multi-dimensional arrays, sparse matrix, algorithm to store sparse matrices, singly, doubly and circular linked lists, list traversal algorithms, stacks– array implementation and linked list implementation, applications of stack, queues– array implementation and linked list implementation, circular queue and dequeue.

**Searching and Sorting:** Searching algorithms– linear search, binary search, comparison of linear and binary search, constant time search using hashing, hash functions, collision resolution techniques– linear probing and chaining, Sorting algorithms– bubble sort, selection sort, insertion sort, merge sort, quick sort, radix sort, shell sort, bucket sort, comparison of sorting techniques, priority queues, binary heap, heapsort.

**Trees:** Basic terminology, tree traversals, expression trees, post/pre/infix notation, binary search tree, search, insertion and deletion operations in BST, balanced BST, AVL tree, insertion and deletion in AVL tree.

**Graphs:** Graph theory terminology, graph representation, graph traversal algorithms, Breadth First Search and connected components, Depth First Search and strongly connected components, applications of BFS and DFS.

#### **Textbooks:**

1. Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, “Data Structures using C and C++”, 2<sup>nd</sup>ed., Pearson Education, 2006.
2. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, “Fundamentals of Data Structures in C”, Universities press, 2<sup>nd</sup>ed., 2008.
3. Robert Sedgewick, Kevin Wayne, “Algorithms”, Pearson Education, 4<sup>th</sup>ed., 2011.
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, PHI, 3<sup>rd</sup>ed., 2010.

5. Seymour Lipschutz, "Data Structures", Schaum's outlines, McGraw Hill Education, 1<sup>st</sup>ed., 2014.
6. Donald E. Knuth, "The Art of Computer Programming", Vol. 1 and Vol. 3.



**Prerequisites:** CSC 101, **Programming in C**

**Course Outcome:** By the end of the course, students should be able to:

- Remember various OOPS terminologies and concepts
- Understand the dynamic modeling concepts and associated models and diagrams
- Implement various C++ concepts for solving a given problem.
- Understand the fundamental concepts of the standard template library.

**OOPS paradigm:** Object & Classes, Links and Associations, Generalization and Inheritance, Aggregation, Abstract Classes, A sample Object Model, Multiple Inheritance, Meta Data, Candidate Keys, Constraints.

**Dynamic Modeling:** Event and States, Operations and Methods, Nested State, Diagrams, Concurrency, Relation of Object and Dynamic Models, Advanced Dynamic Model Concepts Keys, Constraints. **Functional Modeling:** Functional Models, Data Flow Diagrams, Specifying Operations, Constraints, a Sample Functional Model.

**Programming in C++:** Classes and Objects in C++, Functions, Constructors, Inheritance, Function Overloading, Operator Overloading, I/O Operations, Real Life Applications, Extended Classes Pointer, Virtual Functions, Polymorphisms, Class Templates, Function Templates, Exception Handling, C++ I/O.

Translating Object Oriented Design into an Implementation, OMT Methodologies, Examples and Case Studies to Demonstrate Methodology, Comparison of Methodology, SA/SD and JSD.

**Introduction to Standard Template Library:** STL– containers, algorithms, iterators, other STL elements. Container classes– vector, list, deque, set, multiset, map, multimap, stack, queue, priority queue, bitset. Function objects– unary and binary function objects, built-in function objects, create a function object. The string class– introduction, manipulating string objects, relational operations, comparison.

**Textbooks:**

1. James R. Rumbaugh, “Object Oriented Design and Modeling”, PHI.
2. Booch Grady, “Object Oriented Analysis and Design with Application”, Pearson, 3<sup>rd</sup>ed.
3. Dillon and Lee, “Object Oriented Conceptual Modeling”, New Delhi PHI-1993.
4. Stephen R. Shah, “Introduction to Object Oriented Analysis and Design”, TMH.
5. Berzin Joseph, “Data Abstraction; The Object Oriented Approach Using C++”, McGraw Hill.
6. Herbert Schildt, “C++: The Complete Reference”, McGraw Hill, 4<sup>th</sup>ed., 2003.
7. Walter Savitch, “Absolute C++”, Pearson, 5<sup>th</sup>ed., 2012.
8. Lipman, Stanley B, Jonsce Lajole, “C++ Primer Reading”, AWL-1999
9. Bjarne Stroustrup, “The C++ Programming Language”, Pearson, 3<sup>rd</sup>ed., 2002.
10. E. Balagurusamy, “Object Oriented Programming with C++”, TMH, 6<sup>th</sup>ed., 2013.

**Prerequisites:** CSC 103 Computer Fundamentals

**Course Outcome:** By the end of the course, students should be able to:

- Understand the terminologies of software engineering paradigms
- Describe the software engineering requirements and metrics
- Analyze the software development life cycle..
- Explain the software maintenance and current trends in software engineering
- Demonstrate the Computer Aided Software Engineering (CASE) tools

**Software Engineering Paradigms:** Software Characteristics, Software myths, Software Applications, Software Engineering Definitions, Software Process Models, Waterfall, Prototyping, Spiral (including WIN-WIN Spiral), RAD, Process iteration, Process activities, Software Project Management: Management activities, Project planning, Project scheduling, Risk management and activities.

**Software Requirements Engineering:** Requirements Engineering Processes, Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. Software Requirements, Functional and non-functional requirements, User requirements, System requirements, Interface specification, Software Requirement Specification (SRS) document. Specification languages.

Software Metrics and Measures— Process Metrics, Project metrics, Software Project Estimation Models- Empirical, Putnam, COCOMO models.

**Software Design Process:** Principles of software design, Design Strategies, Levels of software design, Interface Design, Coding, Software Reuse. Software Testing, Software Reliability, Software Safety, Defect testing, Debugging Tools.

**Maintenance:** Types of Maintenance, Maintenance Cost, Software Configuration Management, Software Reuse, Software Evolution, Reverse Engineering, Introduction to legacy systems, Software Quality Assurance— plans & activities, Software Documentation. Role of documentation in maintenance and types of documentation.

**Current trends in Software Engineering:** Software Engineering for projects & products. Introduction to Web Engineering and Agile process.

**CASE Tools:** Computer Aided Software Engineering (CASE), Introduction to CASE tools, Types of CASE tools, Project Management Tools, Analysis tools, Design tools, Programming tools, Prototyping tools, Maintenance tools, Advantages and disadvantages of CASE tools.

**Textbooks:**

1. K.K. Aggarwal, Yogesh Singh, “Software Engineering”, New Age International, 2<sup>nd</sup>ed., 2005.
2. R.S. Pressman, “Software Engineering – A practitioner’s approach”, McGraw Hill, 5<sup>th</sup>ed., 2001.
3. Stephen R. Schach, “Classical & Object Oriented Software Engineering”, IRWIN, 1996.

4. James Peter, W. Pedrycz, “Software Engineering: An Engineering Approach”, John Wiley & Sons.
5. I. Sommerville, “Software Engineering”, Addison Wesley, 2002.

**Prerequisites:** Nil

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of HTML and CSS
- Differentiate between Javascript and CSS
- Explain the XML and associated terminologies
- Understand the AJAX framework and syntax
- Develop a client server application

**HTML and CSS:** Concept of WWW, Internet and WWW, HTTP Protocol: Request and Response, Web browser and Web servers, Features of Web 2.0, Basic HTML, formatting and fonts, commenting code, color, hyperlink, lists, tables, images, simple HTML forms, website structure.

**CSS:** Need for CSS, introduction to CSS, basic syntax and structure, using CSS, Inline styles, embedded style sheets, conflicting styles, linking external style sheets, positioning elements, backgrounds, element dimensions, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS, CSS2, Overview and features of CSS3.

**Javascript:** Client side scripting with JavaScript, variables, functions, conditions, loops and repetition, Pop up boxes, Advance JavaScript: Javascript and objects, JavaScript own objects, the DOM and web browser environments, Manipulation using DOM, forms and validations, DHTML: Combining HTML, CSS and Javascript, Events and buttons.

**References:**

1. Chuckmusiano and Bill Kenndy, HTML The Definite Guide, O Reilly, 2000.
2. Joseph Schmuller, Dynamic HTML, BPB publications, 2000.
3. Jeffrey C Jackson, Web Technology– A Computer Science perspective, Pearson Education, 2007.
4. Dave Taylor, “Creating Cool Web Sites with HTML, XHTML, and CSS”, Wiley.
5. Virginia DeBolt, “Integrated HTML and CSS: A Smarter, Faster Way to Learn”, Wiley, 2008.
6. Patrick Carey, “New Perspectives on HTML, XHTML and XML”, 3<sup>rd</sup>ed., Course Tech.
7. Michael Young, “XML Step by Step”, 2<sup>nd</sup>ed., Microsoft Press, 2002.

## **SEMESTER IV**

### **CSC251: Database Systems**

**L | T | P (3 | 0 | 1)**

**Pre-requisite:** Basics of Computer Fundamental

#### **Course outcomes:**

1. Demonstrate a clear understanding of the basics of Database and its use.
2. Implement Relational Model for Industry as well for all organizations
3. Understanding Normalization for fast access of records as well transactions

**Introduction:** Overview of databases, Data models, DBMS architecture and data independence, History of Database Systems. Entity-Relationship Modeling: Basic concepts, constraints, keys, Design issues, weak entities, enhanced E-R, Sub Classes, Super classes, inheritance, specialization and generalization.

**Relational Data Model and Normalization:** Relational model concepts, relational constraints, relational algebra, relational calculus. SQL: basic queries, nested subqueries, aggregate functions, null values, complex queries, database modification commands, programming using SQL, embedded SQL, dynamic SQL. Database Design– Functional dependencies, Normalization, Normal form– 1NF, 2NF, 3NF, BCNF.

**Integrity and Security:** Domain Constraints, Referential Integrity Constraints, Assertions, Triggers, Security and Authorization– Authorization in SQL, Encryption and Authentication.

**File Organization:** Indexed sequential access files, implementation using B+ trees, hashing, hashing functions, collision resolution, extendible hashing, dynamic hashing approach implementation and performance.

**Transaction and Concurrency Control:** Transaction concept, Transaction state, ACID properties and their implementation. Concurrency Control– Lock Based Protocols, Timestamp Based Protocols, Validation Based Protocols, Multiple Granularity. Recovery System– Failure Classification, Storage Structure, Recovery and Atomicity, Log based recovery.

#### **Textbooks:**

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Pearson Education, 5<sup>th</sup>ed., 2008.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database Concepts”, McGraw-Hill, 6<sup>th</sup>ed., 2013.
3. R. Ramakrishanan, J. Gehrke, “Database Management Systems” McGraw-Hill, 3<sup>rd</sup>ed., 2002.
4. Peter Rob, Carlos Coronel, “Database Systems: Design, Implementation and Management”, 7<sup>th</sup> ed., 2006.

**Pre-requisite:** Basics of Mathematics, Number System, Float Values, and Integer Values

**Course outcomes:**

1. Demonstrate a clear understanding of the basics of Approximation of float data
2. Understanding the statistical computing and this will help in real time systems

**Computer Arithmetic and Errors:** Floating Point Arithmetic, Machine epsilon, Types of errors: Round off Error, Chopping Error, Truncation Error, Associative and Distributive Law in Floating Point arithmetic, Inherent Error, Error propagation, Numerical Instability, Error in the approximation of a function, Error in series approximation.

**Equations Solving Methods:** Solution of algebraic and transcendental equation using bisection method, Regula-falsi method, Newton-Raphson method, Solution of simultaneous linear equations using Gauss-elimination method, Jacobi's iterative method, Gauss-Seidel iterative method, LU decomposition methods.

**Interpolation:** Finite difference and operators, Newton forward, Newton backward, central differences, Stirling's interpolation, divided difference formula.

**Differentiation and Integration:** Numerical differentiation, formula for derivatives, maxima and minima of a tabulated function, Numerical integration: Newton-cotes formula, Trapezoidal rule, Simpson's rule, Weddle's rule.

**Solution of Ordinary Differential Equations** using Picard's method, Taylor's series method, Euler's method, modified Euler's method, Runge-Kutta method, predictor-corrector method.

**Statistical Computing:** Curve fitting, Cubic Spline and Approximation– Method of least squares, fitting of straight lines, polynomials, exponential curves, Frequency Chart– Different frequency chart like Histogram, Frequency curve, Pi-chart. Regression analysis– Linear and Non-linear regression, multiple regressions, Statistical Quality Control methods.

**Textbooks:**

1. S.S. Sastry, "Introductory Methods of Numerical Analysis", 5<sup>th</sup>ed., PHI, 2012.
2. M.K. Jain, S.R.K. Iyengar, R.K. Jain, "Numerical Methods for Scientific and Engineering Computation", New Age International, 2003.
3. R.S. Gupta, "Elements of Numerical Analysis", Cambridge University Press, 2015.

**Prerequisites:** Programming, Data Structures

**Course Outcome:** By the end of the course, students should be able to:

- Understand concept of asymptotic analysis and perform complexity analysis of iterative and recursive algorithms.
- Formulate and solve time complexity recurrence relations using various techniques.
- Solve computational problems using various algorithmic paradigms like divide-and-conquer, greedy, dynamic programming, backtracking, branch-and-bound.

**Course Outline:**

**Introduction:** Review of Asymptotic Notations, Mathematical analysis for Recursive and Nonrecursive algorithms, solving recurrence relations.

**Algorithm Design Techniques:** Brute Force, Exhaustive Search, Divide and conquer, Merge sort, Quick sort, Binary search, Multiplication of Large Integers, Strassen's Matrix Multiplication. Greedy strategy– General Approach and problems like Optimal Merge Patterns, Minimum Spanning Trees algorithms, Knapsack Problem, Huffman Code, Job sequencing with deadlines, single source shortest path. Dynamic Programming– General Approach, Memoization, Multistage Graph, Matrix-Chain Multiplication, Longest Common Subsequence, Knapsack Problem, Floyd Warshall algorithm, Optimal Binary Search Trees.

**Limitations of Algorithm Power:** Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP, NP-Hard and NP-Complete Problems, Intractability, Cook's Theorem, Reductions. Coping with the Limitations – Backtracking concept; Branch & Bound method, Approximation Algorithms.

**Textbooks:**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", Prentice Hall of India, 3<sup>rd</sup>ed., 2010.
2. R. C. T. Lee, S. S. Tseng, R. C. Chang, Y. T. Tsai, "Introduction to the Design and Analysis of Algorithms: A Strategic Approach" McGraw Hill, 2006.
3. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, 2007.
4. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", University Press, 2<sup>nd</sup>ed., 2008.
5. Kenneth A. Berman, Jerome Paul, "Algorithms: Sequential, Parallel and Distributed", Cengage Learning, 2004.
6. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
7. Michael T. Goodrich, Roberto Tamassia, *Algorithm Design*, Wiley, 2002.
8. S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Higher Education, 2006.

**Prerequisites:** Programming, Data Structures

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Scripting Languages.
- Understand the Web Development

**Unit 1 Introduction to PHP:** Basic Syntax, defining variables and constants, data types, operators and expression, decision making statements, loop making constructs, mixing decisions and looping with HTML

**Unit 2 String Handling:** Creating a string and accessing its content, searching and replacing content of a string, formatting a string, string built-in-functions

**Unit 3 Handling HTML Form with PHP:** Creating a form, submitting data to the server at the

backend using GET and POST methods, GET vs POST methods, create a student' registration form

**Unit 4 Database Connectivity with MySQL:** Connectivity with database, database creation, creating tables, insertion and retrieval of the data from the database, data manipulation.

**Textbooks: References**

1. Nixon, R., *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Reilly, 2018.
2. Murach J, Murach's, *PHP and MySQL*, 2nd Edition, Mike Murach & Associates, 2014.
3. Holzner S., *PHP: The Complete Reference*, McGraw Hill, 2017.



**Prerequisites:** Computer Fundamentals

**Course Outcome:** By the end of the course, students should be able to:

- Understand concept of different type of Operating Systems
- Understand the Program, Processes difference and used corresponding to devices.
- Understand the efficient memory utilization with file management

**Operating System Overview:** Operating Systems– objectives and functions, evolution– early Operating Systems, Parallel systems, Distributed Systems, Process Control & Real-time Systems. Modern Operating Systems, Virtual Machines, OS Design considerations for Multiprocessor and Multicore architectures. OS Organization– Processor and user modes, Kernel, System Calls, System Programs, System Boot, Overview and Booting process of various OS– Microsoft Windows, Modern UNIX, Linux, Android.

**Process Management:** System view of the process and resources, process abstraction, process hierarchy, process control, execution of the OS. Threads– concept, issues, libraries, thread programming using pthread, multicore processors and threads, multithreading models, process and thread management in Linux, Android and Windows. Process scheduling– Uniprocessor, Multiprocessor and Real-time scheduling algorithms, Traditional UNIX scheduling, Linux scheduling.

Concurrency– Process/Thread synchronization, Mutual Exclusion Principles of Concurrency, Critical Section Problem, Hardware support, OS support (semaphores, mutex), Programming Language support (monitors), Classical synchronization problems– Readers/Writers problem, Producer/Consumer problem.

Deadlocks– Deadlock characterization, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Dining Philosophers Problem. Linux and Android interprocess communication (IPC) and concurrency mechanisms.

**Memory Management:** Logical vs. Physical Address space, Memory Partitioning– Fixed and Dynamic Partitioning, Buddy System, Relocation, Paging, Segmentation. Virtual Memory– Demand Paging, Page Replacement, Frames allocation, Thrashing, Allocating Kernel Memory. Memory Management in Linux.

**I/O and File Management:** I/O Devices, Buffering, Disk Scheduling, Sector Queuing, Linux I/O. File– File Concept, File Organization, Access Methods, File Sharing and Protection, Logical and Physical File System, Directory Structure, Allocation Methods– Contiguous, Sequential and Indexed Allocation, Linux Virtual File System.

**Case Study:** Linux and Windows Operating Systems.

**Textbooks:**

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts”, John Wiley Publications, 8<sup>th</sup>ed., 2008.
2. William Stallings, “Operating Systems: Internals and Design Principles”, Pearson, 7<sup>th</sup>ed., 2013.
3. Robert Love, “Linux Kernel Development”, Pearson, 1<sup>st</sup>ed., 2010.

**References:**

1. Dhananjay M. Dhamdhere, “Systems Programming and Operating Systems”, Tata McGraw-Hill, 2<sup>nd</sup>ed., 1999.
2. Gary Nutt, “Operating Systems: A Modern Perspective”, Pearson, 3<sup>rd</sup>ed., 2009.
3. Maurice J. Bach, “The Design of the UNIX Operating System”, PHI.

## **CSC302: Theory of Computation**

**L | T | P (3 | 1 | 0)**

**Prerequisites:** Discrete Mathematics

**Course Outcome:** By the end of the course, students should be able to:

- Understand formal languages, grammars and Chomsky hierarchy.
- Design regular grammar, DFA, NFA, Mealy and Moore machine, PDA, Turing machines.
- Understand the notion of decidability and computability.

### **Course Outline:**

Finite automaton (FA), Transition system, Acceptance by a finite automaton, Deterministic and non-deterministic automaton (DFA and NFA), Equivalence of DFA and NFA, Minimization of states in a finite automaton, Mealy and Moore machines.

Formal languages and grammar, Chomsky's hierarchy, Regular grammar and regular expression, REs and FA, Closure Properties, Pumping lemma for regular sets.

Push down automaton (PDA), Acceptance by PDA using final state or empty stack, Context free language, Deterministic CFL, Deterministic PDA, Context free grammar and derivation trees, Leftmost and rightmost derivation, Ambiguity in context free grammar, Pumping lemma for context free languages.

Turing machine (TM), TM as computable functions and accepters, Non-deterministic TM, type-0 grammar, Halting problem of a TM, Linear bounded automaton (LBA) model, Context sensitive languages and grammars.

Unsolvability problems, Reduction techniques, Decidability– Post's correspondence problem (PCP), Rice's theorem, Decidability of membership, Emptiness and equivalence problems of languages, P, NP, NP-Completeness; Satisfiability and Cook's theorem.

### **Textbooks:**

1. KLP Mishra, N. Chandrasekaran, "Theory of Computer Science (Automata, Languages and Computation)", PHI, 3<sup>rd</sup>ed.
2. Peter Linz, Jones, Bartlett, "An Introduction to Formal Languages and Automata", 5<sup>th</sup>ed.
3. John E. Hopcroft, J.D. Ullman, Rajiv Motwani, "Introduction to Automata Theory, Languages and Computation", Pearson Education, 3<sup>rd</sup>ed.
4. Michael Sipser, "Introduction to the Theory of Computation", Cengage Learning, 3<sup>rd</sup>ed.

## **CSC303: Optimization Techniques**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basics of mathematics

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of optimization theory.
- Understand the optimize solution to real time problems.

**Introduction:** Optimization, Types of Problems and Algorithms, Review of Linear Algebra and Analysis, Convex Sets and Convex Functions.

**Unconstrained Optimization:** Basic properties of solutions and algorithms, Global convergence, Basic Descent Methods: Line Search Methods, Steepest Descent and Newton Methods, Modified Newton methods, Globally convergent Newton Method. Nonlinear Least Squares Problem and Algorithms, Conjugate Gradient Methods, Trust-Region Methods.

**Constrained Optimization:** First Order Necessary Conditions, Second Order Necessary Conditions, Duality, Constraint Qualification. Convex Programming Problem and Duality Linear Programming: The Simplex Method, Duality and Interior Point Methods, Karmarkar's algorithm, Transportation and Network flow problem. Quadratic Programming: Active set methods, Gradient Projection methods and sequential quadratic programming Dual Methods: Augmented Lagrangians and cutting-plane methods Penalty and Barrier Methods Interior Point Methods.

**Textbooks:**

1. David Luenberger and Yinyu Ye, "Linear and Nonlinear Programming", 3<sup>rd</sup>ed., Springer, 2008.
2. Fletcher R., "Practical Methods of Optimization", John Wiley, 2000.
3. S.S. Rao, "Engineering Optimization: Theory and Practice", New Age International Pvt. Ltd., New Delhi, 2000.
4. G. Hadley, "Linear Programming", Narosa Publishing House, New Delhi, 1990.
5. K. Deb, "Optimization for Engineering Design: Algorithms and Examples", PHI, 1995.

## **CSC304: Programming in Java**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Nill

**Course Outcome:** By the end of the course, students should be able to:

- Understand concept of Java Programming.
- Understand to take input through user in java.
- Understand the interfaces, awt, swing, and beans and will able to create own tool kit using Java.
- Understand the applet and servlet design.

**Introduction to Java:** Java Architecture and Features, Differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords, Data Types, Operators, Expressions, Comments, Using Classes, Controlling Access to Class Members, Basic Program, Decision Making Constructs.

**Java Methods, Arrays, Strings and I/O:** Java, Class Constructors, Method Overloading, final classes, Object class, Garbage Collection, Arrays, Java Strings: The Java String class, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

**Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata:** Single Level and Multilevel Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Packages, Package and Class Visibility, Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

**Exception Handling, Threading, Networking and Database Connectivity:** Exception types, uncaught exceptions, throw, built-in exceptions, creating your own exceptions; Multi-threading: The Thread class and Runnable interface. Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

**Applets and Event Handling:** Java Applets: Introduction to Applets, Writing Java Applets, working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts.

**Advanced Topics:** Java Beans and Web Servers, BDK, Introduction to EJB, Java Beans API, Introduction to Servlets, Lifecycle, JSDK, Servlet API, Servlet Packages: HTTP package, Working with Http request and response, Security Issues. Java Script: Data types, variables, operators, conditional statements, array object, date object, string object, Dynamic Positioning and front end validation. JSP, J2SE.

#### **Textbooks:**

1. Herbert Schildt, "Java, The Complete Reference", TMH, 7<sup>th</sup>ed., 2007.
2. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4<sup>th</sup>ed., 2005.
3. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 1 and 2", Prentice Hall, 9<sup>th</sup>ed., 2012.
4. Bruce Eckel, "Thinking in Java", PHI, 3<sup>rd</sup>ed., 2002.
5. Paul Deitel, Harvey Deitel, "Java: How to Program", Prentice Hall, 10<sup>th</sup>ed., 2011.
6. Bert Bates and Kathy Sierra "Head First Java", Orielly Media Inc., 2<sup>nd</sup>ed., 2005.
7. Elliotte Rusty Harold, "Java Network Programming", O'Reilly publishers, 2000
8. Ed Roman, "Mastering Enterprise Java Beans", John Wiley & Sons Inc., 1999.

## SEMESTER VI

### CSC351: Computer Networks

L | T | P (3 | 0 | 1)

**Pre-requisite:** Basics of Computer Fundamental

#### **Course outcomes:**

1. Understanding Network topologies and network architecture.
2. Demonstrate a clear understanding of the different layers of network architecture.

**Introduction to Computer Networks and Data Communication:** Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite. Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation; multiplexing techniques– FDM, TDM; transmission media.

**Networks Switching Techniques and Access mechanisms:** Circuit switching; packet switching– connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

**Data Link Layer:** Error detection and error correction techniques; data-link control-framing and flow control; Error recovery protocols– stop and wait ARQ, go-back-n ARQ; CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks– repeaters, hubs, switches, bridges, router and gateways.

**Network and Transport Layers:** Routing; routing algorithms; Network layer protocol of Internet– IP protocol, Internet control protocols, Transport services, Transport protocols, Internet Transport Protocols– UDP, TCP.

**Application Layer:** Client–server model, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Telnet, Network Virtual Terminal (NVT), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), User Agent (UA), Mail Transfer Agent (MTA), Multipurpose Internet Mail Extensions (MIME), Post Office Protocol (POP), Simple Network Management Protocol (SNMP), Hypertext Transfer Protocol (HTTP), Uniform Resource Locator (URL), World Wide Web (WWW).

#### **Textbooks:**

1. B.A. Forouzan, “Data Communications and Networking”, THM, 4<sup>th</sup>ed., 2007.
2. Andrew S. Tanenbaum, “Computer Networks”, PHI, 4<sup>th</sup>ed., 2003.

#### **References:**

1. J.F. Kurose, K.W. Ross, “Computer Networking: A Top-Down Approach”, Pearson, 6<sup>th</sup>ed., 2012.

2. Leon Garcia, Widjaja, “Communication Networks: Fundamental Concepts and Key Architectures”, Tata McGraw Hill, 2001.

## **CSC352: Compiler Design**

**L | T | P (3 | 0 | 1)**

**Pre requisite:** Theory of Computation.

### **Course Outcomes:**

1. understand different considerations and phases of compilation
2. the impact of language attributes upon the compilation process
3. the effect of hardware feature on the generated code
4. the practical fundamentals of compiler implementation.

**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 and implementation. Regular grammar & language definition, Transition diagrams, Design of a typical scanner using LEX or Flex.

**Syntax Analysis:** Context free grammars, ambiguity, associability, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing LL(1) grammar, Bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), Design of a typical parser using YACC or Bison.

**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 function and operators, polymorphic function. Run time system: storage organization, activation tree, activation record, parameter passing symbol table, dynamic storage allocation. Intermediate code generation: intermediate representation, translation of declarations, assignments, Intermediate Code generation for 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, specification of machine.

**Code optimization:** Source of optimizations, optimization of basic blocks, loops, global dataflow analysis, solution to iterative dataflow equations, code improving transformations, dealing with aliases, data flow analysis of structured flow graphs.

### **Textbooks:**

1. K. C. Louden, “Compiler Construction, Principle and Practice”, Cengage Publication, 6<sup>th</sup>ed., 2009.
2. Alfred V. Aho, Ravi Sethi, Jeffrey, D. Ullman, “Compilers: Principles, Techniques and Tools”, Pearson, 1998.

3. V. Raghvan, "Principles of Compiler Design", TMH, 2009.
4. Levine, Mason, Brown, "Lex & Yacc", O' Reilly, 1998.
5. S. S. Muchnick Harcourt Asra, "Advanced Compiler Design implementation", Morgan Kaufman, 2006.
6. Allen, "Modern Compiler Implementation in C", Cambridge University Press, 1997.
7. Vinu V. Das, "Compiler Design using FLEX and YACC", PHI, 2005.
8. Cooper, "Engineering a Compiler", Elsevier, 2005.
9. Alan I. Holub, "Compiler Design in C", PHI, 2009.
10. Fisher, "Crafting a Compiler in C", Pearson, 2005.

**CSC354: Project Work/**

**(2 credits)**

This course requires individual/group effort that is overseen by your project supervisor. Weekly meetings will be held to discuss progress and review necessary documents in support of the project. There will be monthly presentations regarding progress of the project. A final presentation followed by viva-voce by external examiner will be held at the end of the semester where the student must submit a project report.



## SEMESTER VII

### CSC401: Computer Graphics

L | T | P (3 | 0 | 1)

**Prerequisites:** Programming, Data Structures

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Graphics.
- Understand the concept of Transformation of Images.
- Understand the concept of rendering related to surface.
- Understand the Graphics Programming

**Introduction:** Computer Graphics– Overview, Basic elements, Animation and Multimedia Applications, Pictures– Representation, Storage and Display, Visualization and Image Processing, RGB color model, Output/Display Devices– Cathode Ray Tube (CRT), Refreshing Display Devices– Raster scan display device– Pixel, Frame Buffer, Color Display, Random scan display device, Plotters, Printers, Digitizers, Tablets, Light Pen, 3D viewing devices, Active and Passive Graphic Devices, Software for Computer Graphics. Lines– Point Plotting Techniques, Points and Lines, Line drawing algorithms– Digital Differential Analyzer (DDA) algorithm, Bresenham’s algorithm, Circle and Ellipse drawing algorithms, Region filling algorithms– Boundary Seed Fill algorithm, Flood Fill algorithm.

**Transformations:** 2D and 3D Transformations– Translation, Rotation, Scaling and other transformations, Matrix Representation of Points, Homogeneous Coordinate System, 2D and 3D Viewing Transformations, Parallel and Perspective Projections, Clipping and Windowing, Line Clipping algorithms– Cohen-Sutherland Line Clipping algorithm, Cyrus-Beck Line Clipping algorithm.

**Curves and Surfaces:** Curve representation, Polygon representation methods, Bezier curves, Bezier surfaces, Spline representations, B-spline methods, Hidden Surface Removal– Z-buffer algorithm, Back face detection, Binary Space Partitioning (BSP) tree method, Scan Line Coherence algorithm, Hidden Line Elimination.

**Surface Rendering:** Illumination/Lighting and Shading Models, Surface Lighting Effects, Basic Lighting Models– Ambient Lighting, Diffuse Lighting, Specular Reflection Lighting Model (Phong Specular Reflection Model), combined effect of Ambient, Diffuse and/or Specular Reflection. Shading– Gouraud Shading, Phong Shading Model, Creating Shadowed Objects, Drawing Shadows, Rendering Texture.

**Graphics Programming:** Graphics Programming using OpenGL, Programming 2D Applications, The OpenGL API, Primitives and Attributes, Color, Viewing, Control Functions, Polygons and Recursion, The Three-Dimensional Gasket, Plotting Implicit Functions, Interaction, Input Devices, Clients and Servers, Display Lists.

**Textbooks:**

1. F. S. Hill, “Computer Graphics Using OpenGL”, Pearson Education, 2<sup>nd</sup>ed., 2007.

2. Donald D. Hearn, M. Pauline Baker, “Computer Graphics with OpenGL”, Pearson Education, 3<sup>rd</sup>ed., 2004.
3. David Rogers, “Procedural Elements of Computer Graphics”, McGraw Hill, 2<sup>nd</sup>ed., 2001.

## SEMESTER VIII

### CSC451: Artificial Intelligence

**L | T | P (3|0|1)**

**Prerequisites: NIL**

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Artificial Intelligence.
- Understand to apply the knowledge and reasoning for different components.
- Understand the Expert Systems and their uses.
- Understand the basics of PROLOG.

**Introduction and applications of artificial intelligence, Problem solving:** State space search, Production system, Problem characteristics, Problem system characteristics, Search techniques: Generate and test, Hill climbing, Best first search, A\* algorithm, Problem reduction.

**Knowledge and Reasoning:** Knowledge acquisition, Knowledge engineer, Cognitive behavior, Knowledge representation: Level of representation, Knowledge representation schemes, Formal logic, Inference Engine, Semantic net, Frame, Scripts. Adversarial search, Optimal and imperfect decisions, Alpha, Beta pruning, Logical agents: Propositional logic, First order logic – Syntax and semantics – Inference in first order logic. Uncertain Knowledge and Reasoning: Uncertainty – Acting under uncertainty – Basic probability notation – Axioms of probability – Baye’s rule – Probabilistic reasoning – Making simple decisions.

**Expert systems:** Definition, Role of knowledge in expert system, Architecture, Expert System Development Life Cycle: Problem selection, Prototype construction, Formalization, Implementation, Evaluation.

**Planning and Learning:** Planning: Planning problem – Partial order planning – Planning and acting in non-deterministic domains – Learning: Learning decision trees, Knowledge in learning, Neural networks, Reinforcement learning – Passive and active.

**PROLOG Programming:** Introduction, variables, using rules, Input and Output predicates, Fail and cut predicates, Recursion, Arithmetic operation, Compound object, Dynamic database, Lists, String, File operations.

**Textbooks:**

1. Elaine Rich, Kevin Knight, “Artificial Intelligence”, Tata McGraw Hill.
2. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, Prentice Hall of India.
3. Nils J. Nilsson, “Principles of Artificial Intelligence”, Narosa Publication house.
4. Stuart Russell, Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson Education, 2<sup>nd</sup>ed.
5. Winston, Patrick, Henry, “Artificial Intelligence”, Pearson Education.
6. Gopal Krishna, Janakiraman, “Artificial Intelligence”.

1.

### **SEMESTER IX**

**DSCP-500: PG Pre-research Thesis**

**(14 credits)**

This course requires individual effort that is overseen by your project supervisor. Weekly meetings will be held to discuss progress and review necessary documents in support of the project. There will be monthly presentations regarding progress of the project. A final presentation followed by viva-voce by external examiner will be held at the end of the semester where the student must submit a project report.

### **SEMESTER X**

**CSC552: Master's Thesis**

**(18 Credits)**

The preparation and defense of Master's thesis is the culminating experience in the M.Sc. program. Students are expected to work with a high level of self-motivation. This course requires individual effort that is overseen by your Thesis Advisor. Weekly or bi-weekly meetings will be held to discuss progress and review submitted documents. There will be monthly presentations regarding progress in the thesis work. Once the research and necessary analysis and results compilation are completed, then individual thesis chapters will be written. A final presentation followed by viva-voce by external examiner will be held at the end of the semester where the student is expected to defend his/her thesis. The thesis will be approved upon successful completion of the defense.

## List of Electives:

### CSE101: Big Data Analytics

L | T | P (3 | 0 | 1)

**Prerequisites:** Programming, Data Structures

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Big Data and Analytics.
- Understand how to use Big Data in different application areas.
- Understand how to use hadoop and mapreduce.
- Understand the concepts of different methods and techniques used in Big Data.

**Introduction:** Big Data and its Importance, Four V's of Big Data, Drivers for Big Data, Introduction to Big Data Analytics, Big Data Analytics applications.

**Big Data Technologies:** Hadoop's Parallel World, Data discovery, Open source technology for Big Data Analytics, Cloud and Big Data, Predictive Analytics, Mobile Business Intelligence and Big Data, Crowd Sourcing Analytics, Inter- and Trans-Firewall Analytics, Information Management.

**Processing Big Data:** Integrating Disparate Data Stores, Mapping Data to the Programming Framework, Connecting and Extracting Data from Storage, Transforming Data for Processing, Introduction to MapReduce/Hadoop for analyzing unstructured data, Subdividing Data in Preparation for Hadoop Map Reduce.

**HadoopMapReduce:** Employing Hadoop Map Reduce, Creating the components of HadoopMapReduce jobs, distributing data processing across server farms, Executing HadoopMapReduce jobs, Monitoring the progress of job flows, The Building Blocks of Hadoop Map Reduce, Distinguishing Hadoop Daemons-Investigating the Hadoop Distributed File System, selecting appropriate execution modes: local, pseudo-distributed, fully distributed.

**Advanced Analytics Platform:** Real-Time Architecture– Orchestration and Synthesis Using Analytics Engines, Discovery using Data at Rest, Implementation of Big Data Analytics, Big Data Convergence, Analytics Business Maturity Model. HADOOP ECO-SYSTEM: Pig– Installing and Running, Comparison with Databases – Pig Latin – User-Define Functions – Data Processing Operators – Installing and Running Hive– Hive QL – Tables – Querying Data – User-Defined Functions. Fundamentals of HBase and ZooKeeper-IBMInfoSphereBigInsights and Streams. Visualizations– Visual data analysis techniques, interaction techniques; Systems and applications.

**Textbooks:**

1. Michael Minelli, Michele Chambers, “Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today’s Business”, 1<sup>st</sup>ed., Wiley CIO Series, 2013.
2. ArvindSathi, “Big Data Analytics: Disruptive Technologies for Changing the Game”, 1<sup>st</sup>ed., IBM Corporation, 2012.
3. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics”, 1<sup>st</sup>ed., Wiley and SAS Business Series, 2012.
4. Noreen Burlingame, Little Book of Big Data, 2012.
5. Tom White, “Hadoop: The Definitive Guide”, 3<sup>rd</sup>ed., O’Reilly, 2012.

### **CSE102: Business Itelligence**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Database Basics, Basics of Programming, Data Structures

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Data warehouse.
- Understand the concepts of OLAP / OLTP.
- Understand the concepts of Data Mining.

Business Intelligence: Introduction, Meaning, Purpose and Structure of Business Intelligence Systems. Understanding Multidimensional Analysis Concepts: Attributes, Hierarchies and Dimensions in data Analysis. Understanding Dimensional Data Warehouse: Fact Table, Dimension Tables, Surrogate Keys and alternative Table Structure. What is multidimensional OLAP?

Understanding OLAP: Fast response, Meta-data based queries, Spread sheet formulas. Understanding Analysis Services speed and meta-data. Microsoft’s Business intelligence Platform. Analysis Services Tools. Data Extraction, Transformation and Load. Meaning and Tools for the same.

Creating your First Business Intelligence Project: Creating Data source, Creating Data view. Modifying the Data view. Creating Dimensions, Time, and Modifying dimensions. Parent-Child Dimension.

Creating Cube: Wizard to Create Cube. Preview of Cube. Adding measure and measure groups to a cube. Calculated members. Deploying and Browsing a Cube

Advanced Measures and Calculations: Aggregate Functions. Using MDX to retrieve values from cube. Calculation Scripting. Creation of KPI’s.

Advanced Dimensional Design: Creating reference, fact and many to many dimensions. Using Financial Analysis Cubes. Interacting with a cube. Creating Standard and Drill Down Actions.

Retrieving Data from Analysis Services: Creating Perspectives, MDX Queries, Excel with Analysis Services

Data Mining: Meaning and purpose. Creating data for data mining. Data mining model creation. Selecting data mining algorithm. Understanding data mining tools. Mapping Mining Structure to Source Data columns. Using Cube Sources. Configuring Algorithm parameters.

**Textbooks:**

1. Business Intelligence, Analytics, and Data Science: A Managerial Perspective, Pearson Education

**CSE103: Introduction to IOT**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basics of Networking, Communication, and related protocols

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of IOT Architecture.
- Understand the concepts of different types of platform for service example XaaS, PaaS.
- Understand the concepts of different devices and gateways.

**M2M to IoT:** The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics.

**M2M to IoT – A Market Perspective :** Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

**M2M and IoT Technology Fundamentals :** Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management

**IoT Architecture-State of the Art :** Introduction, State of the art, Architecture Reference Model- Introduction, Reference Model and architecture, IoT reference Model

**IoT Reference Architecture :** Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints-Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control. Industrial Automation- Service-oriented

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

### **Textbooks**

1. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014.

### **References**

1 Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014.

2. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013

## **CSE104: Modeling and Simulation**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Discrete Mathematics

**Course Outcome:** By the end of the course, students should be able to understand the concept of:

- Project Planning.
- System Definition.
- Model Formulation.
- Input Data Collection and Analysis.
- Model Translation.
- Experimentation and Analysis.

**Introduction to System Modeling:** The notion of system, model, simulation, Types of simulations, Illustrative examples, Conceptual and computer models, Verification and validation of models, Simulation experiment, Simulation project life cycle, Description of simulation models, Structure vs. behavior models, Classification of tasks solved within the modeling and simulation process, Detailed example introduction: database server as a typical queuing system. Description of discrete-event systems behavior. Modeling of time. The notion of status, event, activity, process and their interdependencies. Object-oriented model design. Simulation time, control of time advancement, event list. Event driven simulation algorithm. Detailed example: implementation of the database server as a queuing system.



**Random Numbers in Simulation:** Random variables with discrete and continuous probability distribution. Pseudo-random generators. Multiplicative and additive congruential method. Nonuniform random numbers. Testing of pseudo-random generators. Monte Carlo method. Precision. Queueing systems. Entities: queues, service facilities, storages. Properties of input and output stream. Kendall classification of queueing systems. Entity behavior and statistical data sampling during the simulation run.

**Markov Model:** Discrete and continuous Markov model. Birth-Death processes. Steady-state queueing systems of types M/M/1, M/M/?M/M/m, M/Er/1, Er/M/1 and their variants, Models M/G/1, G/M/1, G/M/m, G/G/1, G/D/1, M+D/D/1. Closed systems and queueing networks, Simulation languages for discrete-event systems. Case study and comparison: Simscript, GPSS, SOL.

**Simulation experiments:** Preparation and pre-processing of input data. Statistical data collected during the simulation run. Time dependency of statistics. Histograms. Evaluation and interpretation of results. Model validation and verification. Simulation of digital systems. Abstractions levels of digital system description. Models of signals and functions. Structure vs. behavior. Models of components. Models of delays. Digital systems simulators– methods of implementation. Flow of simulation time. Synchronous and asynchronous algorithm of digital systems simulation. Acceleration of simulation run.

**Textbooks:**

1. A. M. Law, W.D. Kelton, “Simulation, Modeling and Analysis”, McGraw-Hill, 2<sup>nd</sup>ed., 1991.
2. Frank L. Severance, “System Modeling and Simulation: An Introduction”, Wiley India, 2009.

**CSE105: Operation Research**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basics of mathematics.

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Model Building and analysis.
- Understand the basics of Linear Programming.
- Understand the basics of IOT Architecture.
- Understand the basics of Game Theory and Queuing Theory.

**Introduction:** A quantitative approach to decision making, History and definition of Operations Research, Features of Operations Research Approach, Operations Research Approach to solve a problem, Models and Modelling, Advantages of model building, Methodology, Advantages, Opportunities, Shortcomings, Features and Applications of Operations Research.

**Linear programming:** Structure of Linear Programming, Advantages Limitations, Application areas of Linear Programming. Mathematical formulations of LP models for product mix problems, graphical and simplex method of solving LP problems, sensitivity analysis, duality.

**Transportation and Assignment Problem:** Various methods (NWCM, LCM, VAM) of finding initial basic feasible solution and optimal cost. Mathematical model of the Assignment Problem, Hungarian Methods for solving assignment problem, Travelling Salesman Problem.

**Network Analysis:** Network definition and Network diagram, probability in PERT analysis, project time cost trade off, introduction to resource smoothing and allocation.

**Game Theory:** Concept of game, two person zero sum game, pure and mixed strategy games, saddle point, dominance method Solution Methods for Games without Saddle point.

**Inventory Model:** Introduction to inventory control, deterministic inventory model, EOQ model with quantity discount.

**Replacement & Maintenance Models:** Replacement of items, subject to deterioration of items subject to random failure group vs. individual replacement policies.

**Sequencing problem:** Johnsons algorithm for n jobs and 2 machines, n jobs and 3 machines, two jobs and m machine problems.

**Queuing Theory:** Introduction, Concepts relating to queuing systems, basic elements of queuing model, role of Poisson & exponential distribution, concepts of birth and death process. The Structure of a Queuing Systems, Performance Measures of a Queuing System, Probability Distribution in Queuing System, Classifications, Solutions of Single Queuing Model Models(M/M/1).

**Textbooks:**

1. H.A. Taha, "Operations Research: An Introduction", Macmillan, New York, 5<sup>th</sup>ed., 1992.
2. GovindasamiNaadimuthu and Richard E. Johnson, Schaum's Outline of Theory and Problems of Operations Research.
3. Hillier, Frederick S., Gerald J. Lieberman, "Introduction to Operations Research", McGraw Hill Book Company New York, 6<sup>th</sup>ed., 1995.
4. Levin, Richard I., David S. Rubin, Goel P. Stinson and Everett S. Gardner, "Quantitative Approaches to Management", McGraw Hill Book Company, New York, 8<sup>th</sup>ed., 1992.

**Prerequisites:** Basics of Programming, Data Structure

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of Biometrics.
- Understand the tools and techniques used in Biometrics. .

Introduction of Biometric traits and its aim, image processing basics, Geometric Transformations, Linear Interpolation, brightness correction, basic image operations, filtering, enhancement, sharpening, edge detection, smoothening, enhancement, thresholding, localization. Fourier Series, DFT, inverse of DFT.

Biometric system, identification and verification. FAR/FRR, system design issues. Positive/negative identification. Biometric system security, authentication protocols, matching score distribution, ROC curve, DET curve, FAR/FRR curve. Expected overall error, EER, biometric myths and misrepresentations. Selection of suitable biometric. Biometric attributes, Zephyr charts, types of multi biometrics. Verification on Multimodel system, normalization strategy, Fusion methods, Multimodel identification.

Biometric system security and vulnerabilities, circumvention, covert acquisition, quality control, template generation, interoperability, data storage. Recognition systems: Face, Signature, Fingerprint, Ear, Iris etc.

**Textbooks:**

1. Rafael C. Gonzalez, Richard Eugene Woods, *Digital Image Processing using MATLAB*, 2<sup>nd</sup>ed., Tata McGraw-Hill Education, 2010.
2. Ruud M. Bolle, SharathPankanti, Nalini K. Ratha, Andrew W. Senior, Jonathan H. Connell, *Guide to Biometrics*, Springer, 2009.
3. Richard O. Duda, David G. Stork, Peter E. Hart, *Pattern Classification*, Wiley, 2007.

**Prerequisites:** Basics of Linear Algebra, and Mathematics

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of supervised and unsupervised learning
- Understand the concept of classification, Clustering, and related tools and techniques.

Depth estimation and Multi-camera views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration. apparel.

Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT. Image

Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Pattern Analysis: Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Shape from X: Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

**Textbooks:**

1. David A. Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2<sup>nd</sup>ed., Pearson.
2. R.C.Gonzalez, M.G.Thomason, Syntactic Pattern Recognition: An introduction.
3. P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach.
4. R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley.

**CSE108: Digital Image Processing**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basics of Linear Algebra, and Mathematics.

**Course Outcome:** By the end of the course, students should be able to:

- Understand the basics of Image Processing.
- Understand the concept of Image restoration, compression, and Segmentation.

**Introduction:** Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system. Digital Image Fundamentals– A simple image formation model, image sampling and quantization, basic relationships between pixels. Image enhancement in the spatial domain– Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

**Image restoration:** A model of the image degradation/restoration process, noise models, restoration in the presence of noise-only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms, Introduction to the Fourier transform and the frequency domain, estimating the degradation function, Color Image Processing.

**Image Compression:** Fundamentals, image compression models, error-free compression, lossy predictive coding, image compression standards. Morphological Image Processing– Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms.

**Image Segmentation:** Detection of discontinuous, edge linking and boundary detection, thresholding, region-based segmentation. Object Recognition– Patterns and patterns classes, recognition based on decision-theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods– matching shape numbers, string matching.

**Textbooks:**

1. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, “Digital Image Processing using MATLAB”, PHI, 2003.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall, 1989.
3. Digital Image Processing, Rafael C. González, Richard Eugene Woods, Steven L., Pearson, 2010.

**Prerequisites:** CSC 301, CSC 351

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Cloud Computing and compare with traditional Computing
- Remember the key terminologies used in Cloud Computing and understand key concepts
- Describe virtualization architecture and implement the virtualization using open-source tools
- Identify the advantages and disadvantages of various cloud computing platforms and service models.
- Classify security and privacy issues in cloud computing.
- Apply various cloud services to understand elasticity, scalability and availability properties of Cloud services and also their usage towards web service deployments.

**Overview of Computing Paradigm:** History with overview of Computing Paradigm, Cluster Computing, Grid Computing, Distributed Computing, Utility Computing, Cloud Computing versus Traditional Computing

**Introduction to Cloud Computing:** Introduction to Cloud Computing, Different Perspectives on Cloud Computing, Characteristics, Different Stakeholders in Cloud Computing, Cloud NIST Reference Architecture

**Service Level Agreements (SLAs),** Total cost of ownership (TCO), Benefits and limitations of Cloud Computing, Open Challenges

**Virtualization:** Introduction & need of Virtualization, Definition & types of Hypervisors, Characteristics of Virtualized Environments, Virtualization and Cloud Computing, System calls & Ring Privileges, Machine Reference Architecture, Xen Hypervisor Architecture, Pros and Cons of Virtualization

**Cloud Computing Architecture:** Traditional Computing Architecture-Client-Server Architecture, Peer to Peer Architecture, OpenStack-based Cloud Computing Architecture, Cloud Reference Architecture: Service Models Perspective- Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Deployment Models- Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud

**Cloud Security:** Introduction, Cloud Security Issues such as Application-level Security, Network Level Security, Data-level Security, Virtualization Security, Identity Management & Access Control

**Case Studies:** Implementation of Cloud Services: AWS Cloud Services, Google Cloud Services, Apply Cloud Services for Hosting the Website

**Text Books:**

1. Raj Kumar Buyya, Mastering the Cloud Computing, MacGraw Hill Education (India), 2013
2. Tim Mather, SubraKumaraswamy, ShahedLatif: Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance
3. J.R. ("Vic") Winkler: Securing the Cloud
4. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008.

**Reference Books:**

5. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, August 2008.
6. David Chisnall, The Definitive Guide to Xen Hypervisor, Prentice Hall; Reprint edition (9 November 2007)

**Prerequisites:** Programming and Data Structure

**Course Outcomes:** By the end of the course, students will be able to

- Learn basics of speech and text processing.
- Understand sequential modeling and algorithms.
- Understand parsing and ambiguity resolution.
- Understand multilinguality and associated applications.

**Course Outline:**

Biology of Speech Processing; Place and Manner of Articulation; Word Boundary Detection; Argmax based computations; HMM and Speech Recognition.

Words and Word Forms: Morphology fundamentals; Morphological Diversity of Indian Languages; Morphology Paradigms; Finite State Machine Based Morphology; Automatic Morphology Learning; Shallow Parsing; Named Entities; Maximum Entropy Models; Random Fields.

Theories of Parsing, Parsing Algorithms; Constituency Parsing, Dependency Parsing, Robust and Scalable Parsing on Noisy Text as in Web documents; Hybrid of Rule Based and Probabilistic Parsing; Scope Ambiguity and Attachment Ambiguity resolution.

Lexical Knowledge Networks, Wordnet Theory; Indian Language Wordnets and Multilingual Dictionaries; Semantic Roles; Word Sense Disambiguation; WSD and Multilinguality; Metaphors; Coreferences.

Text summarization, Text classification, Sentiment Analysis; Text Entailment; Robust and Scalable Machine Translation; Question Answering in Multilingual Setting; Cross Lingual Information Retrieval (CLIR).

**Textbooks:**

1. James Allen, "Natural Language Understanding", Pearson Education, 2<sup>nd</sup>ed., 2003.
2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
3. C. Manning and S. Heinrich, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
4. Radford, Andrew et. al., Linguistics: An Introduction, Cambridge University Press, 1999.
5. L.M. Ivasca, S.C. Shapiro, "Natural Language Processing and Language Representation".
6. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
7. T. Winograd, "Language as a Cognitive Process", Addison-Wesley.



**Prerequisites:** Basics of Machine Learning, Programming, and Databases

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Data Science.
- Understand the applicability of Data Science in various fields.
- Understand the basics of Data Visualization and recommendation Systems.

Introduction: What is Data Science? Big Data and Data Science, Statistical Inference– Populations and samples, Statistical modeling, probability distributions, fitting a model, Intro to R. Exploratory Data Analysis– Basic tools (plots, graphs and summary statistics) of EDA, Philosophy of EDA, The Data Science Process– Case Study: RealDirect (online real estate firm)

Basic Machine Learning Algorithms– Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes and why it works for Filtering Spam, Data Wrangling: APIs and other tools for scrapping the Web.

Feature Generation and Feature Selection (Extracting Meaning from Data) – Motivating application: user (customer) retention, Feature Generation (brainstorming, role of domain expertise, and place for imagination) - Feature Selection algorithms – Filters; Wrappers; Decision Trees; Random Forests.

Recommendation Systems: Building a User-Facing Data Product, Algorithmic ingredients of a Recommendation Engine, Dimensionality Reduction, Singular Value Decomposition, Principal Component Analysis, Mining Social-Network Graphs, Social networks as graphs, Clustering of graphs, Direct discovery of communities in graphs, Partitioning of graphs, Neighborhood properties in graphs.

Data Visualization: Basic principles, ideas and tools for data visualization. Data Science and Ethical Issues.

**Textbooks:**

1. Cathy O'Neil, Rachel Schutt, Doing Data Science: Straight Talk from the Frontline, O'Reilly, 2014.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2005.
3. David Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.

## **CSE112: Digital Marketing**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basics of Computer Fundamentals, Social Media platform.

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of Digital Marketing.
- Understand the SEO/SMO/ SMM / Back Linking, and related concepts theritically and practically.

An Introduction to Digital Marketing, Digital Marketing basics (Online Payments, Disability Web Access, Surveys & Forms, Affiliate & Voucher Marketing, Crowdsourcing).

Search Engine Optimization: Understanding Search Engine, Search Engine Result Pages, Search Behavior, On-Page Optimization, Off-Page Optimization,

Digital Display Advertising, Online Advertising, Social Media Marketing (Facebook & LinkedIn), Mastering Google (AdWords Advertising, Analytics & Applications).

Micro Blogging, Twitter, Copy Writing for The Web, Social Media & Mobiles, Mobile Marketing, Email Marketing, Video & Audio (Podcasting) Marketing, Strategic & Action Planning.

### **Textbooks:**

1. Ian Dodson, “The Art of Digital Marketing”, Wiley, 2016.

### **References:**

1. A Beginner’s Textbook for Digital Marketing online book.

**Prerequisites:** Discrete Mathematics

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Fuzzy Logic.
- Understand the concept of Fuzzy Arithmetic and Control System.

Introduction to Soft Computing: Evolution of Computing, Soft Computing Constituents– From Conventional AI to Computational Intelligence.

Fuzzy Sets and Uncertainty: Uncertainty and information, fuzzy sets and membership functions, chance verses fuzziness, properties of fuzzy sets, fuzzy set operations.

Fuzzy Relations: Cardinality, operations, properties, fuzzy cartesian product and composition, fuzzy tolerance and equivalence relations, forms of composition operation. Fuzzification and Defuzzification: Various forms of membership functions, fuzzification, defuzzification to crisp sets and scalars.

Fuzzy Logic and Fuzzy Systems: Classic and fuzzy logic, approximate reasoning, Natural language, linguistic hedges, fuzzy rule based systems, graphical technique of inference. Development of membership functions: Membership value assignments: intuition, inference, rank ordering, neural networks, genetic algorithms, inductive reasoning.

Fuzzy Arithmetic and Extension Principle: Functions of fuzzy sets, extension principle, fuzzy mapping, interval analysis, vertex method and DSW algorithm. Fuzzy Optimization: One dimensional fuzzy optimization, fuzzy concept variables and casual relations, fuzzy cognitive maps, agent based models.

Fuzzy Control Systems: Fuzzy control system design problem, fuzzy engineering process control, fuzzy statistical process control, industrial applications.

**Textbooks:**

1. T. J. Ross, “Fuzzy logic with Engineering Applications”, 3<sup>rd</sup> ed. McGraw-Hill, 2011.
2. H. J. Zimmermann, “Fuzzy set theory and its applications”, Springer, 4<sup>th</sup>ed., 2006.
3. George J. Klir and Bo Yuan, “Fuzzy Sets and Fuzzy Logic-Theory and Applications”, Prentice Hall, 1995.
4. Klir, G. and Yuan, B., “Fuzzy Set and Fuzzy Logic: Theory and Applications”, Prentice Hall, 2002.
5. T. Terano, K. Asai, and M. Sugeno, “Fuzzy systems theory and its applications”, 1 ed. San Diego, CA: Academic press, 1992.
6. Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, “Neuro-Fuzzy and Soft Computing”, Prentice-Hall of India, 2003.

## **CSE114: Data Mining and Warehousing**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Basics of Database

**Course Outcome:** By the end of the course, students should be able to:

- Understand the Data Mining basics and knowledge Discovery in Database.
- Understand the pattern identification and knowledge recognition.
- Understand the basics of classification, clustering and their related techniques.

**Evolution of database technology:** Introduction to data warehousing and data mining, difference between operational databases and data warehouses.

**Data warehouse architecture & design:** Data warehousing Components, building a Data warehouse, Mapping the Data warehouse to multiprocessor architecture, DBMS Schema as for Decision Support, Data Extraction, Clean up and Transformation tools, Metadata.

**Data mining:** Data Pre-processing & Data Mining Primitives Data Pre-processing, Data cleaning, Data Integration and Transformation, Data reduction, Discretization and Concept Hierarchy Generation, Data Mining primitives, Languages and System Architectures, Concept Description: characterization and Comparison, Analytical Characterization, Mining Class Comparison.

**Association Rules & Mining Association Rule Mining:** Mining of Single dimensional Boolean association rules, Constraint based association Mining Classification and prediction: Basic issues regarding classification and prediction, Classification by Decision Tree, Bayesian classification, Prediction, Classifier accuracy.

**Cluster Analysis:** Basic issues, clustering using partitioning methods, Hierarchical methods, Density based methods, Grid based methods and model based methods, Algorithms for outlier analysis.

### **Textbooks:**

1. Ralph Kimball, “The Data Warehouse Life Cycle Toolkit”, John Wiley & Sons Inc., 1998.
2. Alex Berson, S.J. Smith, “Data Warehousing, Data Mining & OLAP”, TMH, 1997.
3. W.H. Inmon, “Building the Data Warehouse”, Wiley India, 2011.

**Prerequisites:** Digital Electronics

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Signals and their processing.
- Understand the basics of Discrete-Time Signals, and related techniques.
- Understand the use of Fast Fourier Transformation

**Introduction:** Signals, systems and signal processing, classification of signals, elements of digital signal processing system, concept of frequency in continuous and discrete time signals, Periodic Sampling, Frequency domain representation of sampling, Reconstructions of band limited signals from its samples, general applications of DSP.

**Discrete-Time Signals and Systems:** Discrete-Time Signals, Discrete-Time Systems, LTI Systems, Properties of LTI Systems, linear convolution and its properties, Linear Constant Coefficient Difference equations. Frequency domain representation of Discrete-Time Signals & Systems, Representation of sequences by Discrete Time Fourier Transform (DTFT), Properties of DTFT and correlation of signals, Fourier Transform Theorems.

**The Z-transform and Analysis of Linear Time Invariant System:** Z-Transform, Properties of ROC for Z-transform, Inverse Z-transform methods, Z-transform properties, Analysis of LTI systems in time domain and stability considerations. Frequency response of LTI system, Relationship between magnitude & phase, all pass systems, inverse systems, Minimum/Maximum phase systems, systems with linear phase. Structures for Discrete Time Systems: Block Diagram and signal flow diagram representations of Linear Constant-Coefficient Difference equations, Basic Structures of IIR Systems, Transposed forms, Direct and cascade form Structures for FIR Systems, Effects of Co-efficient quantization.

**Filter Design Techniques:** Design of Discrete-Time IIR filters from Continuous-Time filters- Approximation by derivatives, Impulse invariance and Bilinear Transformation methods; Design of FIR filters by windowing techniques, Illustrative design examples of IIR and filters. Discrete-Fourier Transform: Representation of Periodic sequences: The discrete Fourier series and its Properties Fourier Transform of Periodic Signals, Sampling the Fourier Transform, The Discrete-Fourier Transform, Properties of DFT, Linear Convolution using DFT.

**Fast Fourier Transform:** FFT-Efficient Computation of DFT, Goertzel Algorithm, radix2 and radix 4 Decimation-in-Time and Decimation-in-Frequency FFT Algorithms. Architecture of DSP Processors— Harvard architecture, pipelining, Multiplier-accumulator (MAC) hardware, architectures of fixed and floating point (TMSC6000) DSP processors.

**Textbooks:**

1. Alan V. Oppenheim, Ronald W. Schaffer, “Digital Signal Processing”, Pearson, 1<sup>st</sup>ed., 2015.
2. Sanjit K. Mitra, “Digital Signal Processing: A Computer-based Approach”, McGraw-Hill, 4<sup>th</sup>ed., 2013.
3. Andreas Antoniou, “Digital Filters: Analysis, Design, and Applications”, TMH, 2<sup>nd</sup>ed., 2001.

## **CSE116: Probability and Statistical Inference**

**L | T | P (3 | 1 | 0)**

**Prerequisites:** Mathematics

**Course Outcomes:** By the end of the course, students will be able to

- Understand the concept of random variables and various discrete and continuous distributions.
- Apply probability distributions and central limit theorem to solve problems.
- Understand data properties using various statistical measures.
- Estimate population parameter (point and confidence interval) using statistical techniques.

### **Course Outline:**

Probability and Randomness, Random Experiment, Sample Space, Random Events, Probability Definition– Axiomatic definition, Frequency Definition. Conditional probability, Independent events, Bayes’ theorem and related problems. Expectation, Standard deviation, Variance, Co-variance, Pearson’s coefficient, Chebyshev’s inequality

Discrete random variables, Probability mass function, Cumulative distribution function and distributions – Bernoulli, Binomial, Geometric, Negative Binomial, Poisson. Poisson approximation of Binomial distribution.

Continuous random variables, Probability density function and distributions – Uniform, Exponential, Gamma, Normal. Normal approximation of Binomial distribution.

Central Limit Theorem, Law of large numbers (Weak law); Random variables simulation, Monte-Carlo methods.

Population and Sample, simple statistics – mean, median, percentiles, quartiles, mode, standard deviation, variance; Statistical Inference: parameter estimation – maximum likelihood, estimation of standard errors. Confidence Intervals estimation.

Hypothesis Testing, Type I and Type II errors, Z-tests, T-tests, P-value, F-distribution and F-tests.

### **Textbooks:**

1. Sheldon Ross, “A First Course in Probability”, 9<sup>th</sup>ed., Pearson Education, 2013.
2. Vijay K. Rohatgi, A. K. Md. EhsanesSaleh, “An Introduction to Probability and Statistics”, 2<sup>nd</sup>ed., Wiley, 2008.
3. Michael Baron, “Probability and Statistics for Computer Scientists”, 2<sup>nd</sup> ed, CRC Press.

### **References:**

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, Springer, 2<sup>nd</sup>ed., 2017.

2. Vijay K. Rohatgi, “Statistical Inference”, 2003.
3. Bradley Efron, Trevor Hastie, “Computer Age Statistical Inference: Algorithms, Evidence and data Science”, Cambridge University Press, 2016.

## **CSE117: Cryptography and Network Security**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** basics of mathematics such as probability, number theory, and linear algebra

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Cryptography.
- Understand the different type of Authentication and messages.
- Understand the basics of web and system security.

**Introduction to cryptography:** Private key cryptography, Conventional Encryption models, Classical encryption techniques, Substitution cipher, Transposition cipher, Cryptanalysis, Stereography, Stream and block ciphers, Modern block cipher: principles, Shannon’s theory of confusion and diffusion, Fiestal structure, DES, Strength of DES, Triple DES, AES, IDEA, Key distribution, Diffie-Hellman algorithm, Public key cryptography, RSA algorithm, Elliptic curve cryptography, Elgamal cryptosystem.

**Message Authentication and Hashing:** Authentication requirements, Message Digest Algorithms-MD4, MD5, Hash functions, Security of hash functions, Message Authentication Codes (MAC), Secure hash algorithm (SHA). Digital Signatures: Digital Signatures, Authentication protocols, Digital signature standards.

**Authentication Applications:** Kerberos and X.509, Public Key Infrastructure (PKI), Concept of Digital Certificate, Types of PKI, Electronic mail security-pretty good privacy (PGP), S/MIME. IP Security: Architecture, Authentication header, encapsulating security payloads, combining security associations, key management.

**Web and System Security:** Secure socket layer (SSL), Transport layer security, Secure electronic transaction (SET). System Security: Intruders, Intrusion Detection System, Password Management, Viruses and related threats, Distributed Denial of Service Attacks, Firewalls, Firewall design principles, Trusted systems.

### **Textbooks:**

1. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education, 6<sup>th</sup>ed., 2013.
2. B. Forouzan, Cryptography and Network Security, TMH, 2<sup>nd</sup>ed., 2010.
3. AtulKahate, Cryptography and Network Security, TMH, 7<sup>th</sup>ed., 2013.
4. Johannes A. Buchmann, Introduction to Cryptography, Springer, 2<sup>nd</sup>ed., 2009.
5. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1996.

**Prerequisites:** Data structures and algorithms.

**Course Outcomes:** By the end of the course, students will be able to

- Understand the necessary mathematical abstraction to solve problems.
- Come up with analysis of efficiency and proofs of correctness.
- Comprehend and select algorithm design approaches in a problem specific manner.

**Course Outline:**

Review of Analysis Techniques, Asymptotic notations; Standard notations and common functions; Recurrences and Solution of Recurrence equations, Master method; Amortized Analysis: Aggregate, Accounting and Potential Methods.

Advanced Data Structures: Red Black Trees, B-Tree, Augmenting Data Structure, Priority Queues, Binomial Heap, Fibonacci Heap, Mergeable Heaps, Data Structure for Disjoint Sets and Union-Find Algorithm.

String Matching Algorithms: Naïve String Matching, Rabin-Karp, String matching with finite automata, Knuth-Morris-Pratt (KMP) Algorithm, Boyer-Moore algorithm.

Number Theoretic Algorithms: Factorization, GCD, Modular Arithmetic, Solving modular linear equations; The Chinese remainder theorem; Powers of an element; RSA cryptosystem; Primality testing; Integer factorization.

Graph Algorithms: Bellman-Ford Algorithm; Single source shortest paths in a DAG; Johnson's Algorithm for sparse graphs; Flow networks and Ford-Fulkerson method; Maximum bipartite matching.

Probabilistic algorithms; Randomizing deterministic algorithms, Randomized Quicksort, Algorithms for Computational Geometry problems, Convex Hull. Approximation Algorithms, Polynomial Time Approximation Schemes.

**Textbooks:**

1. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3<sup>rd</sup>ed., Prentice-Hall of India, 2010.
2. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning, 2002.
3. Ellis Horowitz, Sartaj Sahni, S.Rajasekharan: Fundamentals of Computer Algorithms, 2<sup>nd</sup>ed., Universities press, 2007.



**Prerequisites:** Basics of Fourier Transformation, calculus

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Information Theory.
- Understand the concepts of different type of code such as Linear Block code, Cyclic Codes, Convolutional Codes.

**Information Theory:** Uncertainty, Information, Entropy: characterization and related properties, Huffman codes, Robustness of coding techniques, Discrete Memoryless Channel, Mutual Information, Channel Capacity, Shannon's Theorems, Fundamental theorem of information theory, Gaussian Channel, Limits to Communication.

**Linear Block Codes:** Groups, Fields and Vector Spaces, Construction of Galois Fields of Prime Order, Syndrome Error Detection, Standard Array and Syndrome Decoding, Hamming Codes.

**Cyclic Codes:** Polynomial Representation of Codewords, Generator Polynomial, Systematic Codes, Generator Matrix, Syndrome Calculation and Error Detection, Decoding of Cyclic Codes.

**Structure and Properties of Convolutional Codes:** Convolutional Encoder Representation, Tree, Trellis, and State Diagrams, Distance Properties of Convolutional Codes, Punctured Convolutional Codes and Rate Compatible Schemes.

**Decoding of Convolutional Codes:** Maximum Likelihood Detection, The Viterbi Algorithm, Automatic Repeat Request Strategies: Basic Techniques, Hybrid ARQ.

**Textbooks:**

1. J. A. Thomas and T. M. Cover: Elements of information theory, Wiley, 2006.
2. J. H. van Lint: Introduction to Coding Theory, 3<sup>rd</sup> ed., Springer, 1998.
3. F. J. MacWilliams and N.J. Sloane: Theory of Error Correcting Codes, Parts I and II, 1977.
4. D. Stinson: Combinatorial Designs: Constructions and Analysis, Springer, 2003
5. P. J. Cameron and J. H. van Lint: Designs, Graphs, Codes and their Links, Cambridge University Press, 2010.
6. C. Fragouli and E. Soljanin: Network Coding Fundamentals, Now Publisher, 2007.

**Prerequisites:** Basics of Python, Linear Algebra

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Machine Learning
- Understand the different features such as Generative Learning, Support Vector Machines , Discriminative Learning.
- Understand the concepts of Deep Learning.

**Introduction:** Overview of machine learning, related areas, applications, software tools, Parametric regression: linear regression, polynomial regression, locally weighted regression, numerical optimization, gradient descent, kernel methods.

**Generative learning:** Gaussian parameter estimation, maximum likelihood estimation, MAP estimation, Bayesian estimation, bias and variance of estimators, missing and noisy features, nonparametric density estimation, Gaussian discriminant analysis, naive Bayes.

**Discriminative learning:** linear discrimination, logistic regression, logit and logistic functions, generalized linear models, softmax regression, Regularization and its utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression

**Support vector machines:** functional and geometric margins, optimum margin classifier, constrained optimization, Lagrange multipliers, primal/dual problems, KKT conditions, dual of the optimum margin classifier, soft margins, kernels, quadratic programming, SMO algorithm.

**Graphical and sequential models:** Bayesian networks, conditional independence, Markov random fields, inference in graphical models, belief propagation, Markov models, hidden Markov models, decoding states from observations, learning HMM parameters, Factor analysis, independent component analysis, multidimensional scaling, manifold learning.

**Deep Learning:** Overview, Convolutional Networks, Recurrent Nets, Autoencoders, Recursive Neural Tensor Nets, Platforms for Deep Learning, Deep Reinforcement Learning.

**Textbooks:**

1. Alex Smola, S.V.N. Vishwanathan, “Introduction to Machine Learning”, Cambridge University Press, 2008.
2. Christopher Bishop, “Pattern Recognition and Machine Learning”, Springer Verlag, 2006.
3. T. Hastie, R. Tibshirani, J. Friedman, “Elements of Statistical Learning”, Springer, 2001.
4. K. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
5. Ethem Alpaydin, "Introduction to Machine Learning", The MIT Press, 2<sup>nd</sup>ed., 2009.
6. Tom M. Mitchell, "Machine Learning", Tata McGraw-Hill Education, 2013.
7. Francois Chollet, “Deep Learning with Python”, Manning Publications Company, 2017.

**Prerequisites:** basic understanding of statistics, mathematics, and machine learning concepts.

**Course Outcome:** By the end of the course, students should be able to:

- Understand the evolution of Neural Networks.
- Understand the basics of ANN.
- Understand the concepts of backpropagation networks, Competitive Learning, and Neuro-Fuzzy Learning.

**Introduction to Neural Networks:** Humans and Computers, Organization of the Brain, Biological Neuron, Features, structure and working of Biological Neural Network, Biological and Artificial Neuron Models, Characteristics of ANN, Comparison of BNN and ANN, Models of neuron, Hodgkin-Huxley Neuron Model, Integrate-and-Fire Neuron Model, Spiking Neuron Model, McCulloch-Pitts Model, Perceptron, Adaline model, Basic learning laws, Topology of neural network architecture, Historical Developments, Potential Applications of ANN.

**Basic functional units of ANN for pattern recognition tasks:** Basic feedforward, Basic feedback and basic competitive learning neural network. Pattern association, pattern classification and pattern mapping tasks. Linear responsibility X-OR problem and solution. Analysis of pattern mapping networks summary of basic gradient search methods. Pattern storage networks, stochastic networks and simulated annealing, Boltzmann machine and Boltzmann learning.

**Backpropagation Networks:** Architecture of feed forward network, single layer ANN, multilayer perceptron, back propagation learning, input, hidden and output layer computation, backpropagation algorithm, applications, selection of tuning parameters in BPN, Numbers of hidden nodes, learning. Activation & Synaptic Dynamics– Introduction, Activation Dynamics models, Synaptic Dynamics models, Stability and Convergence, Recall in Neural Networks.

**Competitive Learning Neural Networks:** Components of CL network, pattern clustering and feature mapping network, ART networks, Features of ART models, character recognition using ART network.

Applications of ANN– Pattern classification, Recognition of Olympic games symbols, Recognition of printed Characters, Neocognitron– Recognition of handwritten characters, NETTALK.

**Neuro-Fuzzy Modeling:** Adaptive Neuro-Fuzzy Inference Systems, Coactive Neuro-Fuzzy Modeling– Classification and Regression Trees, Data Clustering Algorithms, Rulebase Structure Identification, Neuro-Fuzzy Control, Case studies.

**Textbooks:**

1. L. Fausett, “Fundamentals of Neural Networks: Architectures, Algorithms & Applications”, Prentice-Hall,1994.
2. James A. Freeman and David M. Skapura, “Neural Networks Algorithms, Applications, and Programming Techniques”, Pearson, 2003.
3. B. Yegnanarayana, “Artificial Neural Networks”, PHI, 2006.

4. Rajasekaran, Pai “Neural networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications”, PHI, 2011.
  5. Stephen I. Gallant, “Neural Network Learning & Expert Systems”, MIT Press, 1995.
  6. John Hertz, Anders Krogh, Richard G. Palmer, “Introduction to the theory of Neural Computation”, Addison-Wesley, 1991.
  7. J.-S.R. Jang, C.-T. Sun, E. Mizutani, “Neuro-Fuzzy and Soft Computing”, Pearson, 1996.
- Haykin, S., Neural Networks - A Comprehensive Foundation, 2<sup>nd</sup>ed., Macmillan, 1999.

### **CSE122: Mobile Ad-hoc Networks**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** basic understanding of wireless, network devices, and networking protocols.

**Course Outcome:** By the end of the course, students should be able to:

- Understand the MAC addresses and related protocols .
- Understand the basics of Transport Layer.
- Understand the concepts of Routing.

**Introduction:** Ad-hoc Networks– Definition, Characteristics, Features, Application, Characteristics of Wireless Channel, Ad-hoc Mobility Models– Indoor and Outdoor Models. Medium Access Protocols, MAC Protocols, Design Issues, Goals and Classification, Contention Based Protocols– With Reservation, Scheduling Algorithms, Protocols Using Direction Antennas. IEEE Standards– 802.11a, 802.11b, 802.11g, 802.15.HIPERLAN.

**Network Protocols:** Routing Protocols– Design Issues, Goals and Classification, Proactive vs. Reactive Routing, Uncast Routing Algorithms, Multicast Routing Hierarchical Routing, Quos Aware Routing.

**End-End Delivery and Security:** Transport Layer– Design Issues, Classification, Ad-hoc Transport Protocols. Security Issues in Ad-hoc Network– Issues and Challenges, Network Security Attacks, Secure Routing Protocols.

**Cross Layer Design and Integration of Ad-hoc for 4G:** Cross Layer Design– Need for Cross Layer Design, Cross Layer Optimization, Parameter Optimization, Techniques, Cross Layer Cautionary Perspective. Integration of Ad-hoc with Mobile IP Networks. Mesh Networks, Vehicular Area Networks. Ad Hoc On-Demand Distance-Vector Protocol– Properties, Unicast Route Establishment, Multicast Route Establishment, Broadcast, Optimizations and Enhancements.

**Link Reversal Routing:** Gafni-Bertsekas Algorithm, Lightweight Mobile Routing Algorithm, Temporally Ordered Routing Algorithm, Preserving Battery Life of Mobile Nodes, Associativity Based Routing, Effects of Beaconing on Battery Life, Research Paper on Recent Trends in MANET.

#### **Textbooks:**

1. C-K Toh, “Ad Hoc Mobile Wireless Networks: Protocols and Systems”, Pearson, 1<sup>st</sup>ed., 2007.
2. C. Siva Ram Murthy, B. S. Manoj, “Ad Hoc Wireless Networks: Architectures and Protocols”, Prentice Hall, 2004.

3. Stefano Basagni, Marco Conti, Silvia Giordano, Ivan Stojmenovic, "Mobile Ad Hoc Networking", Wiley, 2010.
  4. AzzedineBoukerche, "Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks", Wiley-Blackwell, 2008.
  5. Yi Pan, Yang Xiao, "Ad Hoc and Sensor Networks", Nova Science Publishers, 2005.
- Carlos de MoraisCordeiro, Dharma PrakashAgrawal, "Ad Hoc and Sensor Networks: Theory and Applications", World Scientific, 2<sup>nd</sup>ed., 2013.

### **CSE123: Cloud Architecture**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** CSE109

**Course Outcome:** By the end of the course, students should be able to:

- Understand the terminologies in distributed computing enabling technologies
- Differentiate SMP and MPP architectures
- Describe the concepts of parallel computing
- Analyze various distributed computing models and message based communications
- Implement virtualization concepts
- Differentiate between Openstack, AWS and GAE cloud architectures.

**Distributed System Models and Enabling Technologies:** Scalable Computing Service over the Internet. Technologies for Network-based Computing. System Models for Distributed and Cloud Computing. Software Environments for Distributed Systems and Clouds. Performance, Security, and Energy-Efficiency.

**Computer Clusters for Scalable Computing:** Clustering for Massive Parallelism. Computer Clusters and MPP Architectures. Design Principles of Computer Clusters. Cluster Job and Resource Management. Case Studies of Supercomputers and MPP Systems.

**Principles of Parallel Computing:** Eras of Computing, Parallel vs. Distributed Computing, Elements of Parallel Computing, What is Parallel Processing?, Hardware Architectures for Parallel Processing, Approaches to Parallel Programming, Levels of Parallelism, Laws of Caution,

**Distributed Computing :** Elements of Distributed Computing, General Concepts and Definitions, Components of a Distributed System, Architectural Styles for Distributed Computing, Models for Inter-Process Communication, Technologies for Distributed Computing, Remote Procedure Call, Distributed Object Frameworks, Service Oriented Computing

**Virtual Machines & Clusters and Data Centre Virtualization:** Implementation Levels of Virtualization. Virtualization Structures/Tools and Mechanisms. Virtualization of CPU,

Memory and I/O Devices. Virtual Clusters and Resource Management. Virtualization for Data Center Automation.

**Design of Cloud Computing Platforms:** Cloud Computing and Service Models. Data Centre Design and Interconnection Networks. Architecture, Design Cloud Platforms: OpenStack, GAE, AWS. Cloud Security and Trust Management.

**Textbooks:**

2. Kai Hwang, Jack Dongarra, and Geoffrey C. Fox 2011, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things: Clusters, Grids, Clouds, and the Future Internet, Morgan Kaufmann.
3. Rajkumar Buyya 2013, Mastering Cloud Computing: Foundations and Applications Programming, First Ed., Morgan Kaufmann Waltham, USA.
4. Dan C. Marinescu., Cloud computing, Elsevier/Morgan Kaufmann Boston.
5. San Murugesan (Editor), Irena Bojanova (Editor) 2015, Encyclopedia on Cloud Computing, First Ed., Wiley-Blackwell
6. NIST 2013, Cloud Computing Synopsis and Recommendations, CreateSpace Independent Publishing Platform.

**Prerequisites:** Operating System, Computer Hardware and Networking

**Course Outcome:** By the end of the course, students should be able to:

- Understand the terminologies in distributed computing
- Understand the different type of Processors, single as well multi-core processors
- Understand the concepts of computational Grids.

**Introduction:** Parallel Computing Architectures, Paradigms, Issues, & Technologies, architectures, topologies, organizations, Parallel Programming Using Shared Memory, memory coherence, race conditions and deadlock detection, synchronization, multithreaded programming, Parallel Programming using Message Passing, synchronous/asynchronous messaging, partitioning and load-balancing.

**Advanced Processors and Interconnects:** Multicore Processors and High-bandwidth Networks, Parallel and distributed architectures, Distributed and parallel algorithms, Fundamental problems in parallel and distributed computing, fundamental concepts and reasoning principles for parallel and distributed computer systems.

**Distributed Programming Algorithms:** Fundamental issues and concepts, synchronization, mutual exclusion, termination detection, clocks, event ordering, locking, CORBA, JavaRMI, Web Services, shared spaces.

**Clusters of Computers:** Server Clusters, High Availability, and Disaster Recovery, synchronization, fault tolerance, coordination and consensus, Virtual Machines and Virtualized Datacenters.

**Peer-to-Peer Computing:** P2P systems, Familiarity with concurrent programming primitives (semaphores, locks, monitors), Overlay networks, and Content Distribution.

**Computational Grids and Applications:** National or global computing Grids and Applications.

**Textbooks:**

1. M J Quinn, Parallel Programming in C with MPI and OpenMP.
2. AnanthGrama, George Karypis, Vipin Kumar, and Anshul Gupta, Introduction to Parallel Computing, 2<sup>nd</sup>ed., 2003.
3. David Kirk, Wen-Mei W. Hwu, Wen-meiHwu, Programming Massively Parallel Processors: a hands-on approach, Morgan Kaufmann, 2010.
4. William Gropp, Ewing Lusk, and Anthony Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, 2<sup>nd</sup>ed., 1999.
5. Norm Matloff, Programming on Parallel Machines: GPU, Multicore, Clusters and More.
6. K. Hwang and Z. Xu, Scalable Parallel Computing, McGraw-Hill, 1998.
7. G. Coulouris, J. Dollimore, Distributed Systems Concepts and Design, Addison Wesley.
8. Ian Taylor: From P2P to Web Services and Grids, Springer-Verlag, 2005.
9. F. Berman, G. Fox, and T. Hey (Editors), Grid Computing, Wiley, 2003.

**CSE125: Advanced Graph Theory**

**L | T | P (3 | 1 | 0)**

**Prerequisites:** Discrete Mathematics.

**Course Outcomes:** By the end of the course, students will be able to

- Know important classes of problems in graph theory.
- Formulate and prove fundamental theorems on graphs.
- Acquire in-depth knowledge of covering, connectivity and coloring problems in graphs.

**Course Outline:**

Graphs, Types, Isomorphism, Paths, Cycles, Vertex degrees and Counting, Euler and Hamilton graphs, minors, topological minors.

Matching and covering: vertex cover, edge cover, path cover, Gallai-Millgram and Dillworth's theorem, dominating sets, independent sets, matching, maximum matching, Hall's theorem, Tutte's theorem, k-factor graphs.

Connectivity: vertex connectivity, edge connectivity, blocks, 2-connected, 3-connected graphs and their structures, Menger's theorem.

Coloring and Planarity: vertex coloring, Brook's theorem, line graphs and edge coloring, Vizing's theorem, upper bounds on chromatic numbers, structure of k-chromatic graphs, color-critical graphs, planar graphs.

Special classes of graphs, perfect graphs, chordal graphs, interval graphs; basics of spectral graph theory, eigenvalues and graph parameters, strongly regular graphs.

Spectral Graph theory: Basic properties of graph spectrum; Cheeger's inequality and approximation of graph expansion; Expander graphs and applications to super concentrators and pseudo randomness; Error correcting codes and expander codes; Small set expansion, Unique Games Conjecture and Hardness of approximation.

**Textbooks:**

1. Statys Jukna, Extremal Combinatorics: With Applications in Computer Science, Springer, 2<sup>nd</sup>ed., 2013.
2. R.P. Grimaldi, B.V. Ramana, Discrete and Combinatorial mathematics – An applied introduction, Pearson Education (2007).
3. Richard A Brnaldi, Introductory Combinatorics, Pearson Education, Inc. (2004).
4. Miklos Bona, Introduction to Enumerative Combinatorics, McGraw Hill (2007).



5. A walk through Combinatorics – An introduction to enumeration and graph theory, World Scientific Publishing Co. Pvt. Ltd. (2006).
6. J.H. Vanlint, R.M. Wilson, A course in Combinatorics, Cambridge University Press.
7. R. Diestel, "Graph Theory", Springer, 2<sup>nd</sup> ed., 2000.
8. N. Alon and J. Spenser, "Probabilistic Methods", John Wiley and Sons, 2<sup>nd</sup> ed., 2000.

## **CSE126: Cyber Forensics**

**L | T | P (3 | 1 | 0)**

**Prerequisites:** Cyber Security, and Cryptography

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of Cyber Forensics.
- Understand the concepts of Data Forensics, E-Mail Forensics and Steganography.

**History of Forensics:** Computer Forensic Flaws and Risks, Rules of Computer Forensics, Legal issues, Digital Forensic Principles, Digital Environments, Digital Forensic Methodologies.

**Computer Crime:** 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: Processing Crime and Incident Scenes, Working with Windows and DOS Systems, Computer Forensics Tools: Software/ Hardware Tools.

**Data Forensics:** Recovering deleted files and deleted partitions, deleted file recovery tools, deleted partitioned recovery tools, data acquisition and duplication, data acquisition tools, hardware tools, backing up and duplicating data.

**E-Mail Forensics and Steganography:** Forensics Acquisition, Processing Local mail archives, Processing server level archives, classification of steganography, categories of steganography in Forensics, Types of password cracking.

**Validating Forensics Data:** Data Hiding Techniques, Performing Remote Acquisition, Network Forensics, Email Investigations, Cell Phone and Mobile Devices Forensics

### **Textbooks:**

1. Kevin Mandia, Chris Proise, "Incident Response and Computer Forensics", Tata McGraw Hill, 2006.
2. Peter Stephenson, "Investigating Computer Crime: A Handbook for Corporate Investigations", Sept 1999.

3. Anthony Reyes, Jack Wiles, “Cybercrime and Digital Forensics”, Syngress Publishers, Elsevier 2007.
4. John Sammons, “The Basics of Digital Forensics”, Elsevier 2012
5. Linda Volonins, ReynaldsAnzaldua, “Computer Forensics for dummies”, Wiley Publishing 2008.
6. Nelson, Phillips, Enfinger, Steuart, “Computer Forensics and Investigations”, Cengage Learning, 2008.
7. R.Vacca, “Computer Forensics”, Firewall Media, 2005.
8. Richard E. Smith, “Internet Cryptography”, Pearson Education, 3<sup>rd</sup>ed., 2008.
9. Marjie T. Britz, “Computer Forensics and Cyber Crime: An Introduction”, Pearson Education, 1<sup>st</sup>ed., 2012.

## **CSE127: Software Defined Networks**

**L | T | P (3 | 1 | 0)**

**Prerequisites:** Basics of Operating System

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of Network Visualization.
- Understand the concepts of Control Plane, Data Plane, and Data Center Network.

**Introduction:** History and Evolution of Software Defined Networking (SDN), Separation of Control Plane and Data Plane, IETF Forces, ActiveNetworking. Control and Data Plane Separation: Concepts, Advantages and Disadvantages, the Open Flow protocol.

**Network Virtualization:** Concepts, Applications, Existing Network Virtualization Framework (VMWare and others), Mininet based examples.

**Control Plane:** Overview, Existing SDN Controllers including Floodlight and Open Daylight projects. Customization of Control Plane, Switching and Firewall Implementation using SDN Concepts.

**Data Plane:** Software-based and Hardware-based; Programmable Network Hardware. Programming SDNs: Northbound Application Programming Interface, Current Languages and Tools, Composition of SDNs. Network Functions Virtualization (NFV) and Software Defined Networks: Concepts, Implementation and Applications.

**Data Center Networks:** Packet, Optical and Wireless Architectures, Network Topologies. Use Cases of SDNs: Data Centers, Internet Exchange Points, Backbone Networks, Home Networks, Traffic Engineering. Use Cases and Looking Forward.

**Textbooks:**

1. SDN: Software Defined Networks, an Authoritative Review of Network Programmability Technologies, By Thomas D. Nadeau, Ken Gray Publisher: O'Reilly Media, 2013.

2. Software Defined Networks: A Comprehensive Approach, by Paul Goransson and Chuck Black, Morgan Kaufmann, 2014, eBook.
  3. Paul Göransson, Chuck Black, Software Defined Networks: A Comprehensive Approach, Elsevier, 2014.
  4. Thomas D. Nadeau, SDN: Software Defined Networks, 1<sup>st</sup>ed., O’reilly.
  5. SiamakAzodolmolky, “Software Defined Networking with Open Flow”, Packt Publishing, 2013.
- Fei Hu, “Network Innovation through Open Flow and SDN: Principles and Design”, CRC Press, 2014.

## **CSE128: Evolutionary Algorithms**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** Programming

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of Genetic Algorithm.
- Understand the concept of Swarm Optimization.
- Understand the concepts of Differential Evolution, and Genetic Programming.

Genetic Algorithms: Historical development, GA concepts – encoding, fitness function, population size, selection, crossover and mutation operators, along with the methodologies of applying these operators. Binary GA and their operators, Real Coded GA and their operators

Particle Swarm Optimization: PSO Model, global best, Local best, velocity update equations, position update equations, velocity clamping, inertia weight, constriction coefficients, synchronous and asynchronous updates, Binary PSO.

Differential Evolution: DE as modified GA, generation of population, operators and their implementation.

Genetic programming (GP): Steps in GP, individual representation, initial population, tree creation methods, fitness assessment, individual section methods, GP operators, GP parameters

Introduction to parallel genetic programming, distributed genetic programming, parallel distributed GP.

### **Textbooks:**

1. Gen, M. and Cheng, R. “Genetic Algorithms and Engineering Design”, Wiley, New York.
2. David E. Goldberg, “Genetic Algorithm in Search, Optimization and Machine Learning”.
3. Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, Frank D. Francone, “Genetic programming: An introduction– On the Automatic Evolution of Computer Programs and its Applications”, Morgan Kauffman.

**Prerequisites:** Cryptography

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of Bock Chain.
- Understand the concept of Decentralization.
- Understand the concept of Bitcoin, Ethereum, and Hyperledge.

**Introduction to Blockchain:** Definitions of blockchains, The history of blockchain, Generic elements of a blockchain, Features of a blockchain, Applications of blockchain technology, Types of blockchain, Benefits and limitations of blockchain.

**Decentralization:** Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization, Smart contract, Decentralized organizations, Platforms for decentralization.

**Cryptography and Technical Foundations:** Cryptographic primitives, Asymmetric cryptography, Public and private keys, Hash functions, Secure Hash Algorithms (SHAs), Merkle trees, Patricia trees ,Distributed hash tables (DHTs) ,Digital signatures.

**Bitcoin:** Bitcoin definition, Bitcoin Transactions, Bitcoin Blockchain, Bitcoin payments, Bitcoin limitations, Other crypto currency: Namecoin, Litecoin, Zcash.

**Ethereum:** Ethereum clients and releases,The Ethereum stack, Ethereum blockchain ,Currency (ETH and ETC) ,Forks, Gas ,The consensus mechanism, Elements of the Ethereum blockchain, Precompiled contracts, Mining , Applications developed on Ethereum.

**Hyperledger:** Hyperledger as a protocol, Hyperledger Fabric, Sawtooth lake, Corda Architecture ,State objects 376 Transactions ,Consensus , Flows , Components.

### **Textbooks**

1. Mastering Blockchain – Imran Bashir, Packt Publishing.

### **References**

- 1 Building Blockchain Projects-Narayan Prusty, Packt Publishing.

## **CSE130: Quantum Computing**

**L | T | P (3 | 1 | 0)**

**Prerequisites:** NIL

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concepts of Quantum computing.
- Understand the concept of Quantum physics, Circuits, and Algorithms.
- Understand the concepts of Noise detection and correction.

**Introduction to Quantum Computation:** Quantum bits, Bloch sphere representation of a qubit, multiple qubits. 10 5 Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation.

**Background Mathematics and Physics:** Hilber space, Probabilities and measurements, entanglement, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis.

**Quantum Circuits:** single qubit gates, multiple qubit gates, design of quantum circuits.

**Quantum Information and Cryptography:** Comparison between classical and quantum information theory. Bell states. Quantum teleportation. Quantum Cryptography, no cloning theorem.

**Quantum Algorithms:** Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor factorization, Grover search.

**Noise and error correction:** Graph states and codes, Quantum error correction, fault-tolerant computation.

### **Textbooks**

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press. 2002
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific. 2004
3. Pittenger A. O., An Introduction to Quantum Computing Algorithms

## **CSE131: Research Methodology**

**L | T | P (3 | 0 | 1)**

**Prerequisites:** NIL

**Course Outcome:** By the end of the course, students should be able to:

- Understand the concept of research.
- Understand the concept of data collection and selection for research.
- Understand the applicability of research for public at large.

Introduction to Research Methods in science – Philosophy of Science, Research methods and Creative Thinking, Evolutionary Epistemology, Scientific Methods, Hypotheses Generation and Evaluation, Code of Research Ethics, Definition and Objectives of Research, Various Steps in Scientific Research, Research presentations Types of Research – Research Purposes – Research Design – Survey Research – formulation of scientific problems and hypotheses – selection of methods for solving a scientific problem Case Study Research.

How to perform a literature review – Sampling Methods – Data Processing and Analysis strategies - Data Analysis with Statistical Packages – Statistical Analysis – Hypothesis-testing – Generalization and Interpretation.

Research Reports - Structure and Components of Research Report, Types of Report, Layout of Research Report, Mechanism of writing a research report – Requirements of a good dissertation.

**Textbooks:**

1. Oates, B.J., (2005). Researching Information Systems and Computing. Sage Publications, UK.
2. Zobel, J. (2004). Writing for Computer Science - The art of effective communication. 2<sup>nd</sup> ed., Springer, UK.
3. Garg, B.L., Karadia, R., Agarwal, F. and Agarwal, U.K., 2002. An introduction to Research Methodology, RBSA Publishers.
4. Kothari, C.R., 1990. Research Methodology: Methods and Techniques. New Age International.
5. Sinha, S.C. and Dhiman, A.K., 2002. Research Methodology.

**CSE132 Cloud Security**

**L | T | P (3 | 1 | 0)**

**Prerequisites: CSE123**

**Course Outcomes:** By the end of the course, students will be able to:

- Understand the characteristics in terms of the systems, protocols and mechanisms in Cloud
- Comprehend the security & privacy issues with reference to Cloud Computing
- Identify the vulnerabilities, threats and attacks in Cloud Environment and the defense mechanisms
- Examine intrusion detection systems and approaches in Cloud Computing
- Implement open-source attacking and security tools

**COURSE OUTLINES:**

*Introduction to Cloud Computing:* History and Underlying Technologies, Definitions & Characteristics, Cloud Deployment Models, Cloud Service Platforms, Challenges Ahead

*Introduction to Cloud Security:* Definition, vulnerabilities and need of Cloud Security, Cloud Security Concepts: Multi-tenancy, Virtualization, Data Outsourcing, Trust Management, Metadata Security, Cloud Security Standards, CSA Cloud Reference Model, NIST Cloud Reference Model

*Cloud Security & Privacy Issues:* Introduction, Cloud Security goals: Confidentiality, Integrity, Availability, Authentication, Authorization, Auditing, Access Control, Cloud Security Issues: Application Level, Application Level, Virtualization Level, Data security, Identity management and access control, Improper cryptographic keys management, Service level agreement (SLA), Regular audit and compliances, Cloud and CSP Migration, SLA, Hardware issues, Security Requirements for Privacy, Privacy issues in Cloud

*Threat Model, Attacks, Defense Systems and Security Technique:* Threat Model and Taxonomy of Cloud Attacks, Virtual Machines-level Attacks, Virtual Machine Monitor-Level Attacks, Peripheral - Level Attacks, Virtual Storage-Level Attacks, Tenant Network-Level Attacks, Case studies on attack dataset

*Classification of Intrusion Detection Systems in Cloud:* Evolution of Cloud-Intrusion Detection System (IDS), TVM-based IDS, VMM-based IDS, Network-based IDS, Distributed IDS, Research Challenges

*Intrusion Detection Techniques in Cloud:* Taxonomy of Intrusion Detection Techniques in Cloud, Misuse Detection, Anomaly Detection, Virtual Introspection-based, Hypervisor Introspection-based, Hybrid Techniques

*Tools & Case Studies:* Overview of Tools (Attack/Security) in Cloud, Network-Level Attack Tools, VM-Level Attack Tools, VMM Attack Tools, Network Security Tools, VM Security Tool, VMM Security Tools, Case Study of LibVMI : A Virtualization-Specific Tool

*Virtual Machine Introspection and Hypervisor Introspection:* Virtual Machine Introspection (VMI): VM Hook based, VM-State Information based, Hypercall verification based, Guest OS kernel debugging based, VM interrupt analysis based, Hypervisor Introspection (HVI): Nested Virtualization, Code Integrity Checking using hardware-support, Memory Integrity Checking using Hardware/Software Support, Revisiting the VMM Design, VM-Assisted Hypervisor Introspection

#### **BOOKS:**

- 1. Cloud Security: Attacks, Techniques, Tools and Challenges, Published by Preeti Mishra, Emmanuel S. Pilli, R. C. Joshi by Taylor and Francis 2022**
- 2. Cloud Security and Privacy by [Tim Mather](#), [Subra](#), [Shahed Latif](#) (Publ. Orielly Media), 2009**

#### **Reference Books:**

- 1. Mastering Cloud Computing by Raj Kumar Buyya, Vecchiola & Selvi (Published by Mc Graw Hill Education Pvt. Ltd) – 2013**
- 2. Securing the Cloud By Vic (J.R.) Winkler 1st edition , 2011**

