Course Structure and Syllabus for
4-Year B.Sc. (Computer Science)
+
1-Year M.Sc. (Programme)

Department of Computer Science

School of Mathematics, Statistics and Computational Sciences

Central University of Rajasthan

NH-8 Jaipur- Ajmer Highway, Bandarsindri
Kishangarh -305817
District-Ajmer, Rajasthan
Website: www.curaj.ac.in

1. The NEP 2020 based Four-Year UG Programme is an 8-semester (4-year) Programme of 160 credits with multiple exit and entry options at the successful completion of courses assigned at theend of each year.

- Students who opt to exit after completion of the first year and have secured 40 credits will be awarded a UG certificate if, in addition, they complete one vocational course of 4 credits during the summer vacation of the first year. These students are allowed to re-enter the degreeprogramme within three years and complete the degree programme within the stipulated maximum period of seven years.
- Students who opt to exit after completion of the second year and have secured 80 credits will be awarded the UG diploma if, in addition, they complete one vocational course of 4 credits during the summer vacation of the second year. These students are allowed to re-enter within a period of three years and complete the degree programme within the maximum period of seven years.
- Students who wish to undergo a 3-year UG programme will be awarded UG degree in the Major discipline after successful completion of three years securing 120 credits.
- Students who wish to undergo a 4-year UG programme will be awarded UG Honours degree in the major discipline after successful completion of four years degree programme with 160 credits.
- Students who meet minimum requirement of CGPA of 7.5, shall be eligible for a UG Honours with Research degree in respective discipline/field of study after successful completion of 160 Credit, including 12/16 Credits from a research project.

Outline of Courses:

1. **Major and Minor Courses:** Major and minor courses are 4 credits courses. An additionalone to two credits may be for tutorials or practices.
2. **Other Courses:** The Other courses are 3 Credits courses which included multi-disciplinary,Ability enhancement (language) and skill enhancement courses.
3. **Value-Added Courses (Common):** Courses under value-added, summer internship/apprenticeship/ community outreach activities of 2 credits.
4. **Final year research project/ dissertation:** Final year research project or dissertation will of 12 credits.

2. Students qualifying and taking one-year Master's degree programme and have secured 200credits will be awarded the Master's degree in Computer Science.

Structure of the 4-Year UG + 1-Year PG Programme

Minimum Credit requirements to Award Certificate/ Diploma/ Degree under each category

S. No.	Broad category of Course	Minimum Credit Requirement					
		UG Certificate (1-year)	**UG Diploma (2-year)**	**UG Degree (3-year)**	**UG Honors Degree (4-year)**	**UG Honors with Research Degree (4-year)**	**PG degree (1-year)**
1	MAJOR	8	30	60	80	80	100
2	MINOR	8	16	24	44	32	48
3	MULTI/INTER DISCIPLINARY COURSES (**IDC**)	6	9	9	9	9	9
4	ABILITY ENHANCEMENT COURSES (**AEC**)	4	8	8	8	8	8
5	SKILL ENHANCEMENT COURSES (**SEC**)	6	9	9	9	9	9
	VALUE-ADDED COURSES (**VAC**)	8	8	8	8	8	8
6	SUMMER INTERNSHIP	4*	4*	2	2	2	2
7	RESEARCH PROJECT/ DISSERTATION	-	-		-	12	16
	Total	**40+4***	**80+4***	**120**	**160**	**160**	**200**

CORE: MAJOR AND MINOR
IDC: INTER DISCIPLINARY COURSES
AEC: ABILITY ENHANCEMENT COURSES (LANGUAGE)
SEC: SKILL ENHANCEMENT
VAC: COMMON VALUE-ADDED COURSES
INT: INTERNSHIP
DISS: DISSERTATION

Course Structure:

Semester-I: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 1: **Foundation of Computer Science**	**Major**	**4**
2	Core 2: From another department	Minor	4
3	IDC 1: From Table of **IDC**	IDC	3
4	SEC 1: From Table of **SEC (**Programming in 'C')	SEC	3
5	AEC 1: (English Language)	AEC	2
6	VAC 1: From Table of VAC	**VAC**	**4**
		Total	**20**

Semester-II: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 3: **Object-Oriented Modelling**	**Major**	**4**
2	Core 4: From another department	Minor	4
3	IDC 2: From Table of IDC (Discrete Structure)	IDC	3
4	SEC 2: From Table of SEC (Programming in C++)	SEC	3
5	AEC 2: (ICT and Tools)	AEC	2
6	VAC 2: From Table of VAC (Computing and Cyber Ethics)	**VAC**	**4**
		Total	**20**

Semester-III: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 5: **Data Structures**	**Major**	**4**
2	Core 6: **Digital Electronics**	**Major**	**4**
3	Core 7: From another department	**Minor**	**4**
4	IDC 3: From Table of IDC (Programming in Python)	IDC	3
5	SEC3: From Table of SEC	SEC	3
6	AEC 3: From Table of AEC (Web Development)	AEC	2
		Total	**20**

Semester-IV: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 8: **Database Management System**	**Major**	**4**
2	Core 9: **Programming in Java**	**Major**	**4**
3	Core 10: **Software Engineering**	**Major**	**4**
4	Core 11: MOOC Course	Major	2

5	Core 12: From another department (Graph Theory)	Minor	4
6	AEC 4: From Table of AEC	AEC	2
		Total	**20**

Semester-V: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 13: **Computer Networks**	**Major**	**4**
2	Core 14: **Operating System**	**Major**	**4**
3	Core 15: **Theory of Computation**	**Major**	**4**
4	Core 16: MOOC Course	**Major**	**2**
5	Core 17: From another department	Minor	4
6	Summer Internship (6 weeks)	**VAC**	**2**
		Total	20

Semester VI: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 18: **Design & Analysis of Algorithms**	**Major**	**4**
2	Core 19: **Computer Organization**	**Major**	**4**
3	Core 20: **Introduction to Artificial Intelligence**	**Major**	**4**
4	Core 21: Project	**Major**	**4**
5	Core 22: From another department	Minor	4
		Total	**20**

Semester-VII: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 23: Machine Learning	**Major**	**4**
2	Core 24: Advanced Algorithms	**Major**	**4**
3	Core 25: Big Data Analytics	**Major**	**4**
4	Core 26: Data Mining and Warehousing	**Major**	4
5	Core 27: (from department/ another department)	Minor/ Elective	4
		Total	**20**

Semester-VIII: UG

S. No.	Title of Courses	Type of Course	Credits
1	Core 28: Natural Language Processing	**Major**	**4**
2	Core 29: Neural Networks & Deep Learning	**Major**	**4**
3	Core 30 (from department/ School/ another department)	Minor/ Elective	4

4	Core 31 (from department/ another department)	Minor/ Elective	4
5	Core 32 (from department/ another department)	Minor/ Elective	4
6	**Project (In place of Core 30, 31, 32)**	SEC	12*
		Total	**20**

Semester-I: PG

S. No.	Title of Courses	Type of Course	Credits
1	**Core 33:** Image Processing & Computer Vision	**Major**	**4**
2	**Core 34: Software Agent and Swarm Intelligence**	**Major**	**4**
3	Core 35: From department	DE	4
4	Core 36: From department	DE	4
5	Core 37: (from department/ another department)	GE	4
		Total	**20**

Semester-II: PG

S. No.	Title of Courses	Type of Course	Credits
1	Core 38: Self Study (MOOC)	**Major**	**4**
2	Core 39: Project Work in Industry or Institution	GE	16
		Total	**20**

* Project of 12 Credits in place of Minor courses for the 4-year UG Honours with Research degree progamme.

List of Major Courses offered by the department

(i) Foundation of Computer Science
(ii) Object-Oriented Modelling
(iii) Data Structure
(iv) Digital Electronics
(v) Database Management System
(vi) Programming in Java
(vii) Software Engineering
(viii) Computer networks
(ix) Operating System
(x) Design and Analysis of Algorithms
(xi) Theory of Computation
(xii) Computer Organization
(xiii) Introduction to Artificial Intelligence
(xiv) Project
(xv) Machine Learning
(xvi) Advanced Algorithms

(xvii) Big Data Analytics
(xviii) Datamining ad Warehousing
(xix) Natural Language Processing
(xx) Neural Network & Deep Learning
(xxi) Image Processing & Computer Vision
(xxii) Software Agent and Swarm Intelligence

List of IDC

(i) Sem II - Discrete Structure
(ii) Sem III - Programming in Python (L+P=2+1)
(iii) Sem IV - Graph Theory

List of AEC

(i) Sem II – ICT and Tools (L+P=1+1)
(ii) Sem III - Web Development (L+P=1+1)

List of SEC offered by the department

(i) Sem I - Programming in 'C' (L+P=2+1)
(ii) Sem II – Programming in C++

List of VAC offered by the department

1. Sem II - Computing and Cyber Ethics

List of Electives:

Fourth Year	Fifth Year
Subject Title	**Subject Title**
Cloud Computing	Data Science Algorithms
Parallel Processing	Dot Net Technologies
Ad-hoc & Wireless Networks	Compiler Design
High Performance Computing	Software Defined Networks
Internet of Things	Mobile Computing
ADBMS	Human Computer Interaction
Software Project Management	Fractal Theory
Distributed Systems	Blockchain & Cyber Security
Computer Graphics	Game Theory
E-Commerce	Open-Source Operating System
Cyber Security	Quantum Computing
Computing and Vedic Mathematics	

Detailed Syllabus

Semester- I

Core 1: Foundation of Computer Science

Course Outline: The course introduces the fundamental principles and concepts of Computer Science.
Course Objectives: On completion of the course the student will be able to:

- Understand the fundamental concepts of computers.
- Learn basic concepts of operating systems including basic commands of DOS, UNIX/LINUX
- Understand office packages including Word, Excel and PowerPoint
- Learn the fundamentals of networking.

UNIT – I: Introduction to Computer, Von Neumann Architecture, Generation of Computer, Storage Device- Primary Memory and Secondary Storage, Random, Direct, Sequential access methods. Concept of High- Level, Assembly and Low Level programming languages, Program Development Steps, Representing Algorithms through flow chart, pseudo code.

UNIT – II: Number systems, Binary number system, Binary to decimal conversion, Decimal to binary conversion ,Binary operations: addition, subtraction, complement of a number - 1's complementary subtraction, 2's complementary subtraction , binary multiplication, binary division, Representation of binary number as electrical signals, octal number system, octal to decimal conversion – decimal to octal conversion, binary to octal conversion, octal to binary conversion, advantages of octal number system, hexadecimal number system, binary to hexadecimal conversion, hexadecimal to binary conversion

UNIT – III: Introduction to social media, Impact of Social Media, Types of Social Media, Social media practices, Social media platforms, Social media monitoring, Blogging, social bookmarking, Building communities-pages & Channel, Hangouts, Hashtag, Viral content, Social media marketing, Social media privacy, Challenges, opportunities and pitfalls in online social network, Security issues related to social media, Flagging and reporting of inappropriate content, Laws regarding posting of inappropriate content, Best practices for the use of Social media, Case studies.

UNIT – IV: Classification of cyber crimes, Common cyber crimes- cyber crime targeting computers and mobiles, cyber crime against women and children, financial frauds, social engineering attacks, malware and ransomware attacks, zero day and zero click attacks, Cybercriminals modus-operandi , Reporting of cyber crimes, Remedial and mitigation measures, Legal perspective of cyber crime, IT Act 2000 and its amendments, Cyber crime and offences, Organisations dealing with Cyber crime and Cyber security in India, Case studies.

Text/Reference Book

1) Sinha, P.K. "Computer Fundamentals". New Delhi: BPB Publications.
2) M. Morris Mano: Digital Logic and Computer Design, Pearson
3) Cyber Crime Impact in the New Millennium, by R. C Mishra , Auther Press. Edition 2010
4) Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Sumit Belapure and Nina Godbole, Wiley India Pvt. Ltd. (First Edition, 2011)
5) Fundamentals of Network Security by E. Maiwald, McGraw Hill.

SEC 1: Programming in 'C'

Course Outline: Basic understanding of computer fundamentals and various types of languages. Various kinds of number representation and flow chart for easy understanding flow of an algorithm. C-language basics, control and looping control structures.

Course Objectives:-

- To learn the basic principles of programming and software development.
- To demonstrate the use of various structured Programming concepts with the help ofprograms.
- To enhance problem-solving and programming skills in C.
- Handling File in "C".

UNIT – I: Structure of C program, A Simple C program, identifiers, basic data types and sizes, Constants, variables, arithmetic, relational and logical operators, increment and decrement operators, conditional operator, bit-wise operators, assignment operators, expressions, type conversions, conditional expressions, precedence and order of evaluation.

UNIT-II : Input-output statements, statements and blocks, if and switch statements, loops- while, do-while and for statements, break, continue, goto and labels, programming examples. Designing structured programs, Functions, basics, parameter passing, storage classes- extern, auto, register, static, scope rules, block structure, user defined functions, standard library functions, recursive functions, header files, example c programs.

UNIT-III: Introduction to Arrays- concepts, declaration, definition, accessing elements, storing elements, arrays and functions, two-dimensional and multi-dimensional arrays, applications of arrays. String and String functions.

UNIT – IV: Derived types- structures & Union- declaration, definition, Pointers- concepts, Character pointers and functions, pointers to pointers, pointers and multidimensional arrays, Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file, Appending to a file.

Text/Reference Books

1) Ritchie & Kernighan, The C Programming language, ANSI C Version 2nd Ed., PHI.
2) Ashok Kamthane, Programming in C, 2nd Ed., Pearson 2011
3) Schildt, C- The Complete Reference, 4th Ed., TMH 2000
4) E. Balaguruswamy, Programming in ANSI C, 6th Ed., TMH 2012
5) Venugopal K. R and Prasad S. R, "Mastering 'C'", Third Edition, 2008, Tata McGraw Hill

Course Outcomes:

- Basic understanding of programming language and storage representation.
- Demonstration of data types, logical operators, arithmetic operators and increment and decrement operators.
- Capability to write C-Program for Simple C-Program

Semester- II

Core 3: Object-Oriented Modelling

Course Outline: An overview - Object basics - Object state and properties, Behavior, Methods, Messages, Object Oriented system development life cycle, Benefits of OO Methodology.

Course objectives: This course will enable students to

- Describe the concepts involved in Object-Oriented modelling and their benefits.
- Demonstrate concept of use-case model, sequence model and state chart model for a given problem.
- Explain the facets of the unified process approach to design and build a Software system.
- Translate the requirements into implementation for Object Oriented design.
- Choose an appropriate design pattern to facilitate development procedure.

UNIT – I: Object orientation introduction: OO development, OO Themes; Evidence for usefulness of OO development; OO modelling history, Modelling as Design technique: Modelling; abstraction; The Three models. Class Modelling: Object and Class Concept; Link and associations concepts; Generalization and Inheritance; A sample class model; Navigation of class models; Advanced Class Modelling, Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived Data; Packages.

UNIT – II: UseCase Modelling and Detailed Requirements: Overview; Detailed object- oriented Requirements definitions; System Processes: A use case/Scenario view; Identifying Input and outputs-The System sequence diagram; Identifying Object Behaviour-The state chart Diagram; Integrated Object-oriented Models.

UNIT – III: Process Overview, System Conception and Domain Analysis: Process Overview: Development stages; Development life Cycle; System Conception: Devising a system concept; elaborating a concept; preparing a problem statement. Domain Analysis: Overview of analysis; Domain Class model: Domain state model; Domain interaction model; Iterating the analysis

UNIT – IV: Use case Realization: Design Discipline within up iterations: Object Oriented Design-The Bridge between Requirements and Implementation; Design Classes and Design within Class Diagrams; Interaction Diagrams-Realizing Use Case and defining methods; Designing with Communication Diagrams; Updating the Design Class Diagram; Package Diagrams-Structuring the Major Components; Implementation Issues for Three-Layer Design. Design Patterns: Introduction, Describing design patterns, the catalogue of design patterns, Organizing the catalogue, solve design problems by design patterns, selecting and using a design pattern, Creational patterns: prototype and singleton (only); structural patterns adaptor and proxy (only).

Text/Reference Books

1) Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education, 2005
2) Satzinger, Jackson and Burd: Object-Oriented Analysis & Design with the Unified Process, Cengage Learning ,2005.
3) Erich Gamma, Richard Helm, Ralph Johnson and john Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software, Pearson Education, 2007.
4) Grady Booch et. al.: Object-Oriented Analysis and Design with Applications,Pearson Education,2007.
5) Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michel Stal: Pattern –Oriented Software Architecture. A system of patterns, John Wiley and Sons.2007.
6) Booch, Jacobson, Rambaugh : Object-Oriented Analysis and Design with Applications,Pearson, 2013

Course outcomes: The students should be able to:

- Describe the concepts of object-oriented and basic class modelling.
- Draw class diagrams, sequence diagrams and interaction diagrams to solve problems.
- Choose and apply a befitting design pattern for the given problem.

IDC 2: Discrete Structures

Course Outline:

Discrete mathematics is the backbone of computer sciences. It provides the base for algorithm development and semantics of programming languages. This course is designed to given an introductory idea of different discrete structures, including graph theory. In particular, this course introduces logic, proofs, sets, relations, functions, counting, and abstract structure, with an emphasis on applications in computer science.

UNIT I: Set Theory: Introduction, Combination of sets, Multisets, Ordered pairs. Proofs of some general identities on sets. Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Recursive definition of relation, Order of relations. Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions. Growth of Functions. Natural Numbers: Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases. Proof Methods, Proof by counter – example, Proof by contradiction.

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

UNIT III: Lattices: Definition, Properties of lattices – Bounded, Complemented, Modular and Complete lattice. Boolean Algebra: Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra.

UNIT IV: Language of Logic: Proposition, Compound Proposition, Conjunction, Disjunction, Implication, Converse, Inverse & Contrapositive, Bi-conditional Statements, tautology, Contradiction & Contingency, Logical Equivalences, Quantifiers, Arguments Groups, Ring, fields and Lattice

Text/Reference Books

1. C.L Liu and D.P. Mohapatra, Elements of Discrete Mathematics: A Computer Oriented Approach, TMH, 3rd Edition
2. Rosen, Discrete Mathematics and its applications, 6th Edition
3. Schaum's Outlines of Discrete Mathematics, Seymour Lipschutz & Marc Lipson, 2nd Edition

Learning Outcomes

This introductory course will allow students to learn the following:

- Understand the basic principles of sets and operations in sets.
- Fundamental mathematical concepts and terminology underlying a variety of discretestructures.
- Techniques of constructing mathematical proofs and use of propositional and predicate logic.

SEC 2: Programming in C++

COURSE OUTLINE

This course provides in-depth coverage of object-oriented programming principles and techniques using C++. The course begins with a brief review of control structures and data types with emphasis on structured data types and array processing. It then moves on to introduce the object- oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Other topics include overloading, data abstraction, information hiding, encapsulation, inheritance, and polymorphism.

Objectives:

The course is designed to provide complete knowledge of Object Oriented Programming through C++ and to enhance the programming skills of the students by giving practical assignments to be done in labs. The following are the main objectives of this course:

- To learn advanced features of the C++ programming language as a continuation of Cprogramming.
- To learn the basic principles of object-oriented design and software engineering regarding software reuse and managing complexity.
- To demonstrate the use of various OOPs concepts with the help of programs.
- To enhance problem-solving and programming skills in C++.

UNIT I: Introduction to programming paradigms- (Process oriented and Object oriented), Characteristics of OOP- Data hiding, Encapsulation, data security and Polymorphism. Concept of object, class, objects as variables of class data type, difference in structures and class, interms of access to members, private and public members of a class, data & function members. Basics of C++: Structure of C++ programs, introduction to defining member functions within and outside a class, keyword *using*, declaring class, creating objects, simple programs to access & manipulate data members, *cin* and *cout* functions. Decision making constructs (if, switch), Looping (for, while, do…while)

UNIT II: Creating Classes and objects, Modifiers and Access Control, Constructors & destructor functions, Initializing member values with and without use of constructors, constant objects and members function, composition of classes, Static class members.

UNIT III: friend functions and classes, using *this* pointer, creating and destroying objects dynamically using *new* and *delete* operators. Operator overloading: Fundamentals, Restrictions, operator functions as class members v/s as friend functions. Overloading stream function, binary operators and unary operators. Converting between types.

UNIT IV: Inheritance: Base classes and derived classes, protected members, relationship between base class and derived classes, constructors and destructors in derived classes, public, private and protected inheritance, relationship among objects in an inheritance hierarchy, abstract classes, virtual functions and dynamic binding, virtual destructors. Multiple inheritance, virtual base classes, pointers to classes and class members, multiple class members. Templates

Unit V: Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations.

Text/ References Book

1) E. Balagurusamy, Object Oriented Programming with C++, 5th Edition, TMH Education 2011
2) Robert Lafore, Object Oriented Programming, Pearson Publication 2008

3) Rajesh Kumar Shuka, Wiley Publication, 2008
4) Bjarne Stroustrup, The C++ Programming Language, 3rd Edition, Pearson Publication 2002

Learning Outcomes

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

- Understand the features of C++ supporting object-oriented programming.
- Use the characteristics of an object-oriented programming language in a program.
- Use the basic object-oriented design principles in computer problem-solving.
- Use the basic principles of software engineering in managing a complex software project.

AEC 2: ICT and Tools

Objectives:

- Basic understanding of Hardware, Software and Networking fundamentals to under graduate students**.**
- Demonstrate the Detail Hardware configurations of available IT Infrastructure and current status of Hardware configuration includes processor, Memory and Various I/O devices**.**
- Basic understanding of most commonly usage software such as Latex and open source word processor tools.
- Hands on practice of DOS and Linux Operating System Commands.
- Basic understanding and terminology of Computer Networks and usage of Internet.

UNIT-1 Hardware Details:- System Unit, Monitor, Keyboard, Mouse, Flash Memory, Cache Memory and functioning and Internal structure of Hard Disk. Evolution and Types of latest Microprocessor. Printer, Scanner, Functionality and basic differences in printing technology.

UNIT-2 Basic Software Fundamentals:- Basics of software, Types of Software – System software, Application software, Utility Software, Open source software, compiler, assembler, and interpreter. Operating Systems – Functions, Types, Dos, windows, Linux. Basic DOS Commands, Linux Commands.

UNIT-3: Office Processing tools Latex: What is Latex? Latex tools , Installing LaTeX or Web Browsing software. Formatting Words, Lines, and Paragraphs, Designing Pages, Creating Lists, Creating Tables and Inserting Pictures, Create Report, resume and Slides.
Understanding Word Processing: Word Processing Basics, Opening and Closing of documents, Text creation and Manipulation, Formatting of text, Table handling, Spell check, language setting and thesaurus, Printing of word document.
Using Spread Sheet: Basics of Spreadsheet, Manipulation of cells, Formulas and Functions, Editing of Spread Sheet, printing of Spread Sheet. Basics of presentation software, Creating Presentation, Preparation and Presentation of Slides, Slide Show, Taking printouts of presentation/handouts.

UNIT-4 Networking Fundamentals:- What is Computer Network, LAN , WAN and MAN. Networking Devices such as Repeater, Hub, Switch and Router. Introduction about Layers in Computer Networks. Various kinds of addresses in Networking. Demonstration of Wireshark Tools and some of the applications using Wireshark.
Introduction to Internet, WWW and Web Browsers: Concept of Internet, Applications of Internet, connecting to internet, ISP, World Wide Web, Web Browsing software, Search Engines, Understanding URL, Domain name.

Text/Reference Books:

1. P. K. Sinha and Priti Sinha. *Computer fundamentals*. BPB publications.
2. B. Ram, *Computer fundamentals: architecture and organization*. New Age International.
3. D. H. Sanders, *Computers today*, McGraw-Hill.
4. Anita Goel, *Computer fundamentals*. Pearson Education India.
5. Computer Networks, Tanenbaum.
6. Red Hat Linux 9, Bible Chritstopher Negus.
7. LaTeX_Beginners_Guide, Stefan Kottwitz, ISBN 978-1-847199-86-7.

Learning Outcomes:-

- Students should be able to input experimental data into Microsoft Excel Work Sheet in the form of rows and columns.
- Perform calculations in Microsoft Excel using both manually inputting formulas and built-in functions.

- Generate simple and effective tables and graphs to describe experimental data in Microsoft Excel.
- Properly format and organize a formal laboratory report in Microsoft Word.
- Integrate both graphs and tables created in Microsoft Excel into a laboratory report in Microsoft Word and Microsoft Power Point Presentation.
- Students should be able to use equations, sample calculations in Excel using Formulas option.
- Critically evaluate experimental results on a basic level in the form of various graphs to understand easily.

VAC 2: Computing and Cyber Ethics

Course Outline: This course explores technical, and social issues related to cybercrimes, cyber ethics. Cybercrime and laws are broad terms that include offenses where a computer may be the target, crimes where a computer may be a tool used in the commission of an existing offense, and crimes where a computer may play a subsidiary role, such as offering evidence for the commission of an offense. It is also required to know about cyber ethics and its role and significance.

Objectives: The main objectives of this course are:

- To learn the basics of cyber ethics.
- To understand IT Act 2000, its amendments, the offences related to cybercrimes and penalties for these offences.
- To enhance the knowledge related to cyber ethics.
- To understand the rights and responsibilities of IT professionals.

UNIT I: Introduction to Cyber Ethics
Evolution of Computer Technology, Emergence of Cyber Space. Definition of Ethics, Ethics in the Business World, Ethics in Information Technology, Ethical Theories: Virtue Ethics, Duty Ethics, Right Ethics, Utilitarianism, Other Ethical views, Heinz's Dilemma.

UNIT II: Ethics in IT Societies
IT Professionals, IT workers Professional Relationships: Relationships Between IT Workers and Employers, Relationships Between IT Workers and Employers, Relationships Between IT Workers and Employers, Relationships Between IT Workers and Clients, Relationships Between IT Workers and Suppliers, Relationships Between IT Workers & Other Professionals, Professional Codes of Ethics, Certification, Government Licensing, ACM/IEEE Software Engineering Code of Ethics and Professional Practice.

UNIT III: Computer, Internet Crime and Preventive Measures
IT security incidents, Types of Attacks: Viruses, Worms, Trojan Horses, Denial-of-Service (DoS) Attacks; Federal Laws for Prosecuting Computer Attacks, The Information Technology ACT 2000: Offences and Punishment; Implementing Trustworthy Computing: Prevention, Detection, Response, Digital Signatures, Basics of Cryptography, Public Key Cryptography, Shared Key Cryptography.

UNIT IV: Privacy and Intellectual Property
Privacy protection and the law, Information Privacy, Classification of Privacy Laws, International Legal Standards on Privacy, Privacy laws in India, Privacy Issues, Data Encryption, Intellectual Property: Acts Regulating Intellectual Property, Copyrights, Patents, Trade Secrets, Key Intellectual Property Issues.

UNIT V: Ethics of IT Organizations
Key Ethical Issues for Organizations, The Need for Nontraditional Workers, Contingent Workers, H-1B Workers, Outsourcing, Whistle-Blowing, Green Computing.

UNIT VI: Ethical Challenge of Artificial Intelligence & Blockchain
Artificial Intelligence Ethics: Top Nine Ethical Issues in Artificial Intelligence, Five Core Principles to Keep AI Ethical; Blockchain Ethics: Blockchain Definition and Description, Blockchain Anonymity and Privacy, Blockchain for Voting, Blockchain for Transparent Trade Tracing.

Text Books:

1. Stückelberger, C., & Duggal, P. (2018). *Cyber ethics 4.0*. Geneva: Globethics. net.
2. Jonathan, R. (2003). Cyber Law [M].
3. Reynolds, G. (2019). Ethics in Information Technology–6th Edition.
4. Information Technology Act (2000), S. R. Bhansali,, University Book House Pvt. Ltd., Jaipur (2003).
5. Cyber Crimes and Law Enforcement (2003), Vasu Deva, Commonwealth Publishers, New Delhi

References:

1. https://ethics.acm.org/code-of-ethics/software-engineering-code/

Course Outcomes:

After completion of the course students will be able to:

- Understand the basics of cyber ethics.
- Describe Information Technology act and related legislation
- Understand the knowledge related to cyber ethics.

Semester- III

Core 5: Data Structures

COURSE OUTLINE

The course focuses on basic and essential topics in data structures, including array-based lists,linked lists, recursion, stack, queue, heaps, sorting algorithms, and binary tree.

Objectives:

- To impart the basic concepts of data structures and algorithms.
- To introduce various techniques for the representation of the data in the real world.
- To understand basic concepts about the array, stacks, queues, lists, and trees.
- To understand concepts about searching and sorting techniques.
- To understanding writing algorithms and step by step approach in solving problems with thehelp of fundamental data structures.

UNIT I: Arrays: Array as storage element, Row major & column major form of arrays, computation of address of elements of n dimensional array. Arrays as storage elements for representing polynomial of one or more degrees for addition & multiplication, sparse matrices for transposing & multiplication

UNIT II: Stack, queue, dequeue, circular queue for insertion and deletion with condition for over and underflow, transposition of sparse matrices with algorithms of varying complexity

UNIT III: Linear linked lists: singly, doubly and circularly connected linear linked lists insertion, deletion at/ from beginning and any point in ordered or unordered lists. Comparison of arrays and linked lists as data structures.

Linked implementation of stack, queue and dequeue. Algorithms for/of insertion, deletion of stack,queue, dequeue implemented using linked structures. Polynomial representation using linked lists for addition, Concepts of Head Node in linked lists.

Searching: Sequential and binary search.

UNIT IV: Non-Linear Structures: Trees definition, characteristics concept of child, sibling, parent child relationship etc, binary tree: different types of binary trees based on distribution of nodes, binary tree (threaded and unthreaded) as data structure, insertion, deletion and traversal of binary trees, constructing binary tree from traversal results. B-Trees and introduction to B+ Trees. Graph, Traversersing

Text/References:

1. An introduction to data structures with applications By Jean-Paul Tremblay, P. G.Sorenson, TMH
2. A. Drozdek, Data Structures and Algorithms in C++, 3rd Edition, Course Technology
3. Data Structures in C & C++, Tanenbaum, PHI
4. S. Sahni, Data Structure Algorithms and Applications in C++, Wiley 2003.

Learning Outcomes

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

- Use of basic data structures for the storage and retrieval of ordered or unordered data. Datastructures include arrays, linked lists, binary trees, heaps, and hash tables.
- Students develop knowledge of applications of data structures including the ability to implement algorithms for the creation, insertion, deletion, searching, and sorting of each data structure.
- Students know searching and sorting algorithms.
- Students can implement projects using different data structures.

Core 6: Digital Electronics

Unit –I: Codes: Binary, Sum of products and product of sums, BCD, Excess 3, Gray, Alphanumeric codes, Logic Gates: OR, AND, NOT, NAND, NOR, Exclusive – OR, Exclusive – NOR, Mixed logic, Boolean Algebra: Boolean Logic Operations, Basic Law of Boolean Algebra, Demorgan's Theorem, Principle of Duality, Karnaugh map Minimization and Quine-McCluskey method of minimization, Sum of Products, Product of Sums

UNIT II COMBINATIONAL CIRCUIT DESIGN
Design of Half and Full Adders, Half and Full Subtractors, Binary Parallel Adder — Carry look ahead Adder, BCD Adder, Multiplexer, Demultiplexer, Magnitude Comparator, Decoder, Encoder, Priority Encoder.

UNIT III SYNCHRONOUS SEQUENTIAL CIRCUITS
Flip flops — SR, JK, T, D, Master/Slave FF — operation and excitation tables, Triggering of FF, Analysis and design of clocked sequential circuits — Design — Moore/Mealy models, state minimization, state assignment, circuit implementation — Design of Counters- Ripple Counters, Ring Counters, Shift registers, Universal Shift Register.

UNIT IV ASYNCHRONOUS SEQUENTIAL CIRCUITS
Stable and Unstable states, output specifications, cycles and races, state reduction, race free assignments, Hazards, Essential Hazards, Pulse mode sequential circuits, Design of Hazard free circuits.

UNIT V MEMORY DEVICES AND DIGITAL INTEGRATED CIRCUITS
Basic memory structure — ROM -PROM — EPROM — EEPROM –EAPROM, RAM — Static and dynamic RAM — Programmable Logic Devices — Programmable Logic Array (PLA) — Programmable Array Logic (PAL) — Field Programmable Gate Arrays (FPGA) — Implementation of combinational logic circuits using PLA, PAL. Digital integrated circuits: Logic levels, propagation delay, power dissipation, fan-out and fan-in, noise margin, logic families and their characteristics-RTL, TTL, ECL, CMOS

Text / Reference Books

1. A.S.Tannenbaum : Structured Computer Organization, Pearson
2. Thomas C. Bartee : Digital Computer Fundamentals, McGraw-Hill
3. Duglus V Hall : Microprocessors and Interfacing: programming and Hardware, McGraw-Hill, 1986.
4. Introduction to Computer Architecture, Stone S.Galgotia Publicatons 1996.
5. Microprocessor Architecture Programming & Applications, R. Gaonkar, Wiley Eastern-1987.
6. Computer Architecture and Organization by N.P. Carter, 4th Edition, McGraw-Hill, 2014.

Learning Outcomes: At the end of the course, the students will be able:

- to understand theory of Digital Design to provide an insight of how basic computer components are specified.
- to understand the functions of various hardware components and their building blocks
- to understand and appreciate Boolean algebraic expressions to digital design
- in depth understanding of sequential and Combinational circuits
- in depth understanding of realization of different combinational/sequential circuits

IDC 3: Programming in Python

COURSE OUTLINE
Python programming develops the basic skills of programming using python.

Course Objectives

- To introduce the basics of Python programming.
- Laboratory exercises to cover in Lab sessions.

UNIT I: Basics of python programming: python identifiers, indentation, comments in Python, data types, python strings.

UNIT II: Python operators: arithmetic, assignment, relational operators etc. Decision making and loop control structures.

UNIT III: Built-in functions in python, built-in string methods. User-defined functions, keyword arguments. Lambda functions.

UNIT IV: Python lists, tuples, dictionaries. Performing basic operations on lists, tuples and dictionaries.

UNIT V: Python modules, namespace and scoping. File handling, access modes, reading and writing files, renaming and deleting files.

UNIT VI: Plotting graphs in python, Introduction to Matplotlib. Developing basic GUI applications using Tkinter.

Text/References Books

1. Introduction to computation and programming using Python, John V. Guttag, MIT Press.
2. Python: The Complete Reference, by Martin C. Brown, McGraw Hill Education
3. Learn Python Programming: An in-depth introduction to the fundamentals of Python, Third Edition, By Fabrizio Romano, Heinrich Kruger

Outcomes:-

At the end of this course, the student will be able to:

- Develop basic programs in Python.
- Plot graphs using Python.

AEC 3: Web Development

COURSE OUTLINE: Web development is building websites and web applications like Facebook, Twitter, or internal web portals within businesses. Web development has two disciplines: front-end and back-end.

Course Objectives: This Subject is useful for Making own Web page and how to host own web site on internet.

UNIT I: Introduction to WWW : Protocols and programs, secure connections, application and development tools, the web browser, What is server, choices, setting up UNIX and Linux web servers, Logging users, dynamic IP Web Design: Web site design principles, planning the site and navigation

UNIT II: Introduction to HTML : The development process, Html tags and simple HTML forms, web site structure Introduction to XHTML : XML, Move to XHTML, Meta tags, Character entities, frames and frame sets, inside browser.

UNIT III: Style sheets : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS, CSS2

UNIT IV: Javascript : Client side scripting, What is Javascript, How to develop Javascript, simple Javascript, variables, functions, conditions, loops and repetition

UNIT V: DHTML : Combining HTML, CSS and Javascript, events and buttons, controlling your browser, Ajax: Introduction, advantages & disadvantages ,Purpose of it ,ajax based web application, alternatives of ajax
UNIT VI: PHP : Starting to script on server side, Arrays, function and forms, advance PHP Databases : Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHP myadmin and database bugs

Text/References Books

- Burdman, “Collaborative Web Development”, Addison Wesley.
- Sharma &Sharma, “Developing E-Commerce Sites”, Addison Wesley
- Ivan Bayross, “Web Technologies Part II”, BPB Publications.
- Steven Holzner,”HTML Black Book”, Dremtech press.
- Web Technologies, Black Book, Dreamtech Press
- Web Applications : Concepts and Real World Design, Knuckles, Wiley-India
- Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel Pearson.

Course Outcomes: After Studying that subject students would have capability to make own web site and host their own web site on internet. Also students would have enough knowledge about what are the technologies used in internet.

Semester- IV

Core 8: Database Management System

Objectives:

The objective of the course is to present an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information from a DBMS. Also provide fundamental knowledge of, and practical experience with, database concepts. Include study of information concepts and the realization of those concepts using the relational data model. Practical experience gained designing and constructing data models and using SQL to interface to user DBMS packages.

UNIT-I: Introduction to database, Overview and History of DBMS, File System vs DBMS, Purpose of Database, Overall System Structure, Entity Relationship Model, Mapping Constraints - Keys - E- R Diagrams.

UNIT-II: Overview of Data Design Entities, Attributes and Entity Sets, Relationship and Relationship Sets, Features of the ER Model-Key Constraints, Participation Constraints, Weak Entities, Class Hierarchies, Aggregation. Relationship Algebra: Selection and Projection, Set Operations, Renaming, Joints, Division, Relation Calculus.

UNIT-III: Relational Database Design: Pitfalls, Normalization Using Functional, Dependencies, First Normal Form, Second Normal Form, Third Normal Form and BCNF.

UNIT-IV: Structured Query Language (SQL), Basic Structure, Set Operations, Aggregate, Functions, Date, Numeric, and Character Functions, Nested Sub queries, Modification of Databases, Joined Relations. Transaction Processing: ACID Properties, Concurrency Control, Recovery.

Text/References:

1. Elmasri R and Navathe SB, Fundamentals of Database Systems, 3rd Edition, AddisonWesley, 2000.
2. Connolly T, Begg C and Strachan A, Database Systems, 2nd Edition, Addison Wesley, 1999
3. Ceri Pelagatti , Distributed Database: Principles and System - (McGraw Hill)
4. Simon AR, Strategic Database Technology: Management for the Year 2000, Morgan Kaufmann,1995
5. A. Silversatz, H. Korth and S. Sudarsan: Database Cocepts 5th edition, Mc-Graw Hills 2005.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

- Differentiate database systems from file systems by enumerating the features provided by database systems and describe each in both function and benefit.
- Define the terminology, features, classifications, and characteristics embodied in databasesystems.
- Student will be able to model an application's data requirements using conceptual modelingtools like ER diagrams and design database schemas based on the conceptual model.
- Student will be able to write SQL commands to create tables and indexes, insert/update/delete data, and query data in a relational DBMS.
- Analyze an information storage problem and derive an information model expressed in the form of an entity relation diagram and other optional analysis forms, such as a data dictionary.
- Demonstrate an understanding of the relational data model.
- Transform an information model into a relational database schema and to use a data definition language and/or utilities to implement the schema using a DBMS.

- Formulate, using SQL, solutions to a broad range of query and data update problems.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
- Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.
- Use a desktop database package to create, populate, maintain, and query a database.
- Demonstrate a rudimentary understanding of programmatic interfaces to a database and beable to use the basic functions of one such interface.

Core 9: Programming in Java

Course Objectives

- To introduce concepts of core java systematically.
- To demonstrate the principles underlying the design of high-level programming languages.

UNIT- 1	Fundamentals of object-oriented programming, Java evolution, Features of Java, Java Development Kit (JDK), Java basics, keywords, constants, variables, and data types.	**(7 Hours)**
UNIT- 2	Operators and expressions, compiling and executing java Programs, operators, control flow Statements, Switch Statement, Looping statements, break and continue.	**(7 Hours)**
UNIT- 3	Classes, objects, and methods, declaring methods in java, constructors, static members, nesting of methods, inheritance, abstract methods and classes, array, String, and vectors.	**(8 Hours)**
UNIT- 4	Interfaces, defining interfaces, extending interfaces, implementing interfaces, Packages, creating packages, managing errors, and exception. Multithreaded programming.	**(7 Hours)**
UNIT- 5	Applets programming, Applet class, Applet and HTML, the Life cycle of an applet.	**(8 Hours)**
UNIT- 6	Graphics programming using AWT, colors, font, Event handling, Components of an event, event classes, event listener.	**(8 Hours)**

Text/Reference Books

1) Schildt Herbert, Java: The Complete Reference, 8th Edition, Tata McGraw-Hill,2011.
2) E. Balagurusamy, Programming with JAVA, a Primer, 4th Edition, 2010, TataMcGraw-Hill Publishing Company Limited, Delhi.
3) Dr.Rao, Nageswara. Core Java: An Integrated Approach, New Edition Kongent SolutionsInc, 2009.

Learning Outcome

- The students will have competence in the use of the Java Programming language.
- The development of small to medium-sized application programs that demonstrate professionally acceptable coding.

Core 10: Software Engineering

Course Objectives:- The Software Engineering course provides students with knowledge and skills that enable them to design, code, test and manage quality-measured software systems. Software Engineering major includes studying and practicing the software development process, in addition to the algorithm and data process needed to develop innovative software that solves a specific problem.

- Knowledge of basic SW engineering methods and practices, and their appropriateapplication.
- A general understanding of software process models such as the waterfall and evolutionarymodels.
- Understanding of software requirements and the SRS documents.
- Understanding of software testing approaches such as unit testing and integration testing.
- Understanding on quality control and how to ensure good quality software.

UNIT I: System Analysis: Characteristics, Problems in system Development, System Level projectPlanning, System Development Life cycle (SDLC), computer system engineering & system analysis, modeling the architecture, system specification. Capability Maturity Model Integration (CMMI)

UNIT II: Requirement Analysis: Requirement analysis tasks, Analysis principles, Softwareprototyping and specification data dictionary finite state machine (FSM) models.
Structured Analysis: Data and control flow diagrams, control and process specification behaviouralmodelling, extension for data intensive applications.

UNIT III: Software Design: Design fundamentals, Effective modular design: Data architectural and procedural design, design documentation, coding – Programming style, Program quality, quantifying program quality, complete programming example

UNIT IV: Testing Strategies and tactics: Testing fundamentals, strategic approach to software testing, Validation testing, system testing, Black-Box testing, white-box testing and their types, basic path testing.

Text/Reference Books

1. R.S. Pressman, Software Engineering: A Practitioner's Approach, Mc Graw Hill
2. P. Jalote, An Integrated Approach to Software Engineering (II Edition)
3. KK Agarwal an Y. Singh, Software Engineering, New Age International Publishers
4. I. Somerville, Software Engineering, Addison Wesle, 2006

Learning Outcomes:-

Upon successful completion of the course, the student will be able to:

- Understand and implement the best practices of software engineering to develop high-performance, maintainable software:
- Capture, analyze and document computing system requirements.
- Translate system requirements into an implementable software design.
- Design test strategies to verify correct and robust software functionality.
- Apply knowledge of mathematics, science, and engineering.
- Use the techniques, skills, and modern engineering tools necessary for engineering practice.
- Appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems.
- Install, configure, troubleshoot, maintain, and upgrade components of computer systems.
- Apply knowledge of security issues to the implementation of information technology solutions.

Core 12: Graph Theory

Course Outline:

Graph theory is an essential course of computer sciences. Different types of graphs, theorems and techniques selected have profound applications in different areas of study in Computer Science and Engineering.

Course Objective: Graph theory has enormous application in modelling problems of Computer Science. Solution to such problems can be easily obtained by appeal to the known results in Graph Theory

Unit I: Introduction to Graph Theory – Definition, Applications, Degree of vertices, Brief History.
Paths and Circuits – Isomorphism, Subgraphs, Disconnected Graphs, Euler Graph, Operation on graphs, Hamiltonian circuit, Travelling Salesman problem.

Unit II: Tree – Properties, Center, Rooted tree, Binary trees, On counting trees, Spanning tree, Fundamental circuits, Shortest spanning tree, Prim's Algorithm, Kruskail Algorithm. Cut-sets – Properties, Finding cut-sets, Connectivity, Separability, Network flows, 1-isomorphism, 2-isomorphism.

Unit III: Planar Graph -Combinatorial Vs Geometrical graphs, Kuratowski's two graphs, Detection of planarity, dual graph, thickness and crossing. Vector Representation of Graph, Circuit and Cut-set vector subspace, Intersection, join.

Unit IV: Matrix Representation – Incidence matrix, Circuit matrix, Cut-set matrix, Path matrix, Adjacency matrix. Coloring – Chromatic number, Chromatic partitioning, chromatic polynomial, Matching, Covering, Four color problem. Five color theorem.

Unit V: Directed Graph – Types, Directed path, Euler diagraph, Tree with directed edges, Matrix representation, Acyclic diagraphs. Graph Enumeration – Types, Polya's counting theorem.

Text/Reference Books

1) Narsingh & Deo, Graph Theory with Applications to Engineering and Computer Science, PHI2004 Publication.
2) V. Balkrishnan, Schaum's Outline of graph Theory, 2004.
3) Jonathan L Gross, Jay Yellen and Mark Anderson, Graph Theory and its Applications, Chapman and Hall, New York.

Learning Outcomes

This introductory course will allow students to learn the following:

- Know some important classes of graph theoretic problems;
- Be able to formulate and prove central theorems about trees, matching, connectivity, colouring
- and planar graphs;
- Be able to describe and apply some basic algorithms for graphs;
- Be able to use graph theory as a modelling tool.

Semester- V

Core 13: Computer Networks

Course outline: The designed course covers the topics of computer networks covers the fundamentals of computer networks , basics of signals , convention of signals from analog to digitaland from digital to analog. Course will give introduction of OSI Model for computer communication system and practical explore of communication protocol model which is TCP/IP layer architecture. Course will cover detail functionalities and basic services provided by each and every layer.

Objectives of the Course: -

- The course demonstrates OSI and TCP/IP Model.
- Clear understanding of Guided Media characteristics and various Network Topology and Hardware building blocks.
- Demonstration of challenges and issues in Data Link Layer functionalities.
- Demonstration and Explanation of routing algorithms in Network Layer.
- Demonstration of Various Application layer concepts.

UNIT I: Introduction to Networks and Layered Architectures (OSI, TCP/IP), Categories of Networks Network performance measures e.g. bandwidth, latency, Delay/bandwidth product. Transmission Media: Guided Media (twisted pair cable, Coaxial Cable, fibre optic cable), Unguided media (radio waves, microwaves, infrared), Topology. Hardware building Blocks of a network e.g. switches, routers, gateways etc.

UNIT II: Data Link Layer: Data Link Layer Design Issues - Error Detection and Correction. Elementary data link protocols - Sliding Window Protocols - Protocols Verification - ChannelAllocation Problem- Multiple Access Protocols

UNIT III: Network Layer: Network Layer Design Issues- Routing Algorithms-Congestion Control Algorithms- Quality of Service -Internetworking
Transport Layer: Transport Services – elements of transport protocols – simple transport protocols.

UNIT IV: Application layers: Domain name system – Electronic mail – The World Wide Web. Introduction to Network security.

Text/References Book:

1. Computer Networks, Andrew S. Tanenbaum , Fourth edition,PHI private Ltd, New Delhi , 2008
2. Computer Networking Top Down approach 3rdeditionBy Jim kurose and keithross

Course Outcomes:

- Students able to understand various layer functionalities.
- Able to understand various addressing schemes.
- Students able to understand various functionalities of each layer.
- Students able to understand difference between TCP and UDP.
- Application layer protocols are going understand properly.

Core 14: Operating System

Course Outline: Operating systems are the heart of the Computer system. They act as an interface between the Hardware and the user. This course is designed to provide in-depth understanding of the operatingsystems.

Course Objectives

- Provide basic understanding of the functions and types of operating systems.
- To introduce the concepts of process management, memory management, file management and deadlocks.
- Do practical exercises on scheduling techniques.
- Laboratory exercises to be covered in Lab sessions.

UNIT-I: Introduction to Operating Systems, Types of operating systems, Multiprogramming, Time-sharing systems, Operating system services, System calls and System programs, Storage structures

UNIT II: Process concepts, process scheduling, operations on process, threads, Inter process communication, precedence graphs, critical section problem, semaphores, classical problems of synchronization, CPU Scheduling,

UNIT-III: Deadlock: Introduction, problem, characterization, prevention, avoidance, detection,recovery from deadlock, Methods for deadlock handling.

UNIT-IV: Memory Management, Single and multiple partitioned allocations, paging segmentation, Virtual Memory Management, Demand paging and Page Replacement Algorithms

Unit –V Device and Storage Management**:** File-System Interface, Mass-Storage Structure, Device Management: Techniques for Device Management, Dedicated Devices, Shared Devices, Buffering, Multiple Paths, Secondary-Storage Structure: Disk Structure, Disk Scheduling, Disk Management.

Unit –VI File-System Implementation: A Simple File System, Logical & Physical File System
File-System Interface: Access Methods, Directory Structure, Protection, Free-Space Management, Directory Implementation

Text/Reference Book:

1) Abraham Silberschatz and P. B. Galvin - Operating system concepts – Addison WesleyPublication
2) A. Tanunbaum, Modern Operating Systems, 3rd Edition, Pearson Publication

Course Outcomes:-

At the end of this course, the student will be able to:

- Develop in-depth understanding of the functions and concepts related to operating systems.
- Demonstrate understanding of CPU and disk scheduling algorithms.
- Understand how different kinds of Operating systems work.

Core 15: Theory of Computation

Course Outline:

The proposed Course theory of computation illustrates various computational models to perform scientific calculations. The proposed model describes how to design mathematical function which admits an algorithm. In the proposed course one of the functions which are membership to find an element belongs to set different classes of set or not. The proposed course describes various kind of automata which are mathematical models to accept or reject sentences belongs to various kinds of formal languages.

Objectives:

- Design various kinds of mathematical models to perform computing strategies.
- Demonstrate the various kinds of automata models to accept or reject strings belong to thevarious kinds of formal languages.
- To identify the limitation of the proposed model and try to find the models overcome thelimitations.
- Understand various design principles of the computing models to estimate functioning ofmodels.
- Decide a function admits an algorithm or not, if it admits a function then develop a step bystep procedure.
- Learn Mathematical models and formal languages to develop compiler various phases such aslexical phase and syntax phase.
- Basic understanding of Finite Automata, Push down Automata and Turing Machine Design.
- Various forms to represent the formal languages and simplification of grammar.

UNIT I: Languages: Alphabets, string, language, basic operations on language, concatenation, Kleene Star

UNIT II: Regular languages model: finite state machine (deterministic and non deterministic), regular grammars, regular expressions, equivalence of deterministic and non deterministic machine and of the three models; Properties: closure, decidability, minimality of automata.

UNIT III: Context Free Grammar, Derivation trees, Simplification of Context Free Grammar, Chomosky Normal Form, Greibach Normal Form, pushdown automata and their equivalence, Properties of Context Free Languages.

UNIT IV: Turing machines, grammars, recursive functions and their equivalence, language acceptability, decidability, halting problem

Text/References:

1. Hofcroft J.E., Ullman J.D., Introduction to Automata Theory, Languages and Computation,Narosa Publishing House.
2. Lewis H. R. and Papadimitriou C. H., Elements of the theory of computation, Pearson EducationAsia
3. Martin J. C., Introduction to Languages and the Theory of Computation, 2e, Tata McGraw-Hill .
4. Daniel I A Cohen, Introduction to computer Theory, Wiley II Edition

Course Outcomes:

- Consider one of the computing paradigms as membership, and then design models for various classes of sets.
- Able to apply the proposed models to find the performance evaluation of scientific computations.
- Understanding the models to design compiler various phases.
- Design various machines/models for simple calculations.

- Understanding the concept of NP-Complete and NP-Hard.

Semester- VI

Core 18: Design & Analysis of Algorithms

Course Outline: The proposed DAA course covers algorithm and its design strategies. The course will illustrate complexity of designed algorithm in space and time of algorithms. Various Asymptotic notations illustrated during the course. Designed strategies have explained by taking well known algorithms, NP-Complete and NP-Hard topic explained.

Objectives:

- Demonstrate various algorithm analysis parameters to explain best, average and worst case.
- Various strategies to explain to design algorithms.
- Demonstration of various Graph Theory algorithms such as Minimal Spanning Tree and Allpair shortest Path.
- Demonstration of P and NP Complete Problems.

UNIT I: Definition & characteristics of algorithms, structures. Difficulties in estimating exact execution time of algorithms. Concept of complexity of program. Asymptotic notations: Big-Oh, theta, Omega-Definitions and examples, Determination of time and space complexity of simple algorithms without recursion. Representing a function in asymptotic notations.

UNIT II: Divide-and-conquer, Dynamic Programming, Greedy methods, Backtracking, Branch- and Bound Technique.

UNIT III: Minimum Spanning Trees, Single-Source Shortest Paths, All-Pairs Shortest Paths,Maximum Flow. String Matching, Computational Geometry.

UNIT IV: P and NP class, NP-completeness and reducibility, NP-complete problems.

Text/References Book:

1) T. Cormen, C. Leiserson, R. Rivest. Introduction to Algorithms, Indian Reprint, PHI
2) V. Aho, J. Hopcraft, J. Ulmann. The Design and analysis of computer Algorithms. AddisonWesley
3) S. Basse, A. V. Gelder, Computer Algorithms: Introduction to design and Analysis, 3rd., PearsonEducation Asia Pvt. Ltd.

Course Outcomes:-

- Students able to design and analyze algorithms before solving problem.
- Able to decide which algorithms are space and time wise efficient algorithms.
- The main outcome of the course is to decide type of the algorithm which can solve theproblem more effectively with less number of computational resources.
- Understand difference between NP-Hard and NP-Complete Problems.

Core 19: Computer Organization

Course Outline:

The outline of the course is basic understand of circuit logic design and storage information invarious formats in the memory, various addressing modes and various registers.

Course objectives:

The course is designed to train the graduates in:

- Architecture of digital computers.
- Architecture of various digital units of a computer.
- Usage of digital computers in industry and research.

UNIT I

Basic architecture of computer, Computer Components and Functions, Interconnection Structures. Bus Interconnections, Von Neumann Concept, Input / Output: I/O Module, Programmed I/O, Interrupt Driven I/O, Direct Memory Access, data transfer schemes for microprocessors

UNIT II

Classification and design parameters, Memory Hierarchy, Internal Memory: RAM, SRAM and DRAM, Interleaved and Associative Memory. Cache Memory: Design Principles, Memory mappings, Replacement Algorithms, Cache performance, Cache Coherence. Virtual Memory, External Memory : Magnetic Discs, Optical Memory, Flash Memories, RAID Levels

UNIT III

Central Progressing Unit (CPU): General Register Organization, Control Word, Example of Micro operation, Stack Organization: Register stack, Memory Stack, Reverse Polish Notation. Instruction Formats, Three, Two, One, Zero Address Instructions, RISC Instructions, Addressing Modes, Addressing Modes Examples with Assembly Language [8085/8086 CPU], CISC Characteristics and RISC Characteristics.

UNIT IV

Parallel Processing, Concept and Block Diagram, Types (SISD, SIMD, Interconnect network, MIMD, MISD), Future Directions for Parallel Processors, Performance of Processors, Pipelining: Data Path, Time Space Diagram, Hazards. Instruction Pipelining, Arithmetic Pipelining

Text/References Book:

1. Computer System Architecture- M. Morris Mano, Pearson Publication 3rd Edition, PHI
2. Computer Organizations and Architecture - William Stallings (Pearson Education Asia),2008
3. Computer Organization and Architecture -John P. Hayes (McGraw -Hill), 1998
4. Computer Organization -V. Carl. Hamacher (McGraw-Hill), 2011
5. Nicolas Carter, Computer Architecture, Schaum's Series, TMH

Course Outcomes:

Graduates after completing the course shall gain:

- Ability to understand architecture of digital computers.
- Ability to apply digital computers in solving complex problems in industry and research.
- Ability to take up advanced course in Computer Architecture.

Core 20: Introduction to Artificial Intelligence

Course Outline:

The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches. Develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning. Students will implement a small AI system in a team environment. The knowledge of artificial intelligence playsa considerable role in some applications students develop for courses in the program.

Objectives :

- To have a basic proficiency in a traditional AI language including an ability to write simple to intermediate programs and an ability to understand code written in that language.
- To have an understanding of the basic issues of knowledge representation and blind andheuristic search, as well as an understanding of other topics such as minimax, resolution, etc. that play an important role in AI programs.
- To have a basic understanding of some of the more advanced topics of AI such as learning, natural language processing, agents and robotics, expert systems, and planning.

UNIT 1: Introduction: Introduction to AI, Historical Development, Turing Test. Problem Solving, Search Algorithms, State-space and Solution Space Search, State space as graph- state v/s node; Evaluating Search Strategies- Time, Space, Completeness, Optimality.
Uninformed search: Breadth First Search, Depth First Search, Iterative Deepening Search, Bi-directional Search, Uniform Cost Search.

UNIT 2: Informed search: Best First Search, Heuristic Search, A* Search, Admissible heuristic, Consistent heuristic, optimality and admissibility, IDA* search, Weighted A* search and inconsistency. Hill Climbing, Local Search, Simulated Annealing, local beam search and Genetic Algorithm.
Adversarial search: Adversarial Search and Game Playing, Min-max Algorithm, Alpha-beta pruning, partially observable games, stochastic games.

UNIT 3: Constraint satisfaction problems: Introduction to CSPs, Constraint Networks, Binary and non-binary constraints, qualitative and quantitative CSPs, Consistencies- Local and global consistencies; Constraint propagation and generalizations − Related Methods: backtracking search; dynamic programming; variable elimination; Handling Spatial and Temporal constraints.
AI planning: Introduction, complexity, PDDL, Domain Independent Planning, Domain Description, PDDL (syntax), forward vs. backward search, planning graph. Graph Plan

UNIT 4: Probabilistic reasoning: Uncertainties in AI; Markov random fields; Markov networks; Baye's Theorem; Bayesian networks – Concepts, Representation and Inference; Hidden Markov Model and Dynamic Bayesian Network. Dempster-Shaffer Framework of Evidential Reasoning.

BOOKS

1. Artificial Intelligence: A Modern Approach (third Edition): S. Russel and P. Norvig.
2. Artificial Intelligence: Foundation of Computational Agents: D Poole and AMckworth.

Course Outcomes:-

Upon successful completion of the course, the student will be able to:

- Design a knowledge based system.
- Ability to apply Artificial Intelligence techniques for problem solving.
- Use classical Artificial Intelligence techniques, such as search algorithms, minimax algorithm, and neural networks.
- Have read and analyzed important historical and current trends addressing artificial intelligence.

Semester- VII

Core 23: Machine Learning

Course Outline: Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self- driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome and a lot of other applications.

Objective of the Machine Learning Course:

- The objective is to familiarize the audience with some basic learning algorithms and techniques and their applications, as well as general questions related to analyzing and handling large data sets.
- Several libraries and data sets are publicly available, that will be used to illustrate the application of machine learning algorithms.
- The emphasis will be on machine learning algorithms and applications, with some broad explanation of the underlying principles.
- To develop the basic skills necessary to pursue research in machine learning.
- To develop the design and programming skills that will help you to build intelligent, adaptive artifacts.

UNIT- 1	Basics: Introduction to Machine Learning - Different Forms of Learning, Basics of Probability Theory, Linear Algebra and Optimization. Regression Analysis: Linear Regression, Ridge Regression, Lasso, Bayesian Regression, Regression with Basis Functions.	**(08 Hours)**
UNIT- 2	Classification Methods: Instance-Based Classification, Linear Discriminant Analysis, Logistic Regression, Large Margin Classification, Kernel Methods, Support Vector Machines, Multi-class Classification, Classification and Regression Trees.	**(08 Hours)**
UNIT- 3	Neural Networks: Non-linear Hypotheses, Neurons and the Brain, Model Representation, Multi-layer Networks, Back-propagation, Multi-class Discrimination, Training Procedures, Localized Network Structure, Deep Learning.	**(08 Hours)**
UNIT- 4	Graphical Models: Hidden Markov Models, Bayesian Networks, Markov Random Fields, Conditional Random Fields. Ensemble Methods:Boosting - Adaboost, Gradient Boosting, Bagging - Simple Methods, Random Forest.	**(08 Hours)**
UNIT- 5	Clustering:Partitional Clustering - K-Means, K-Medoids, Hierarchical Clustering - Agglomerative, Divisive, Distance Measures, Density Based Clustering – DBscan, Spectral Clustering.	**(08 Hours)**
UNIT- 6	Dimensionality Reduction: Principal Component Analysis, Independent Component Analysis, Multidimensional Scaling, and Manifold Learning. Reinforcement Learning: Q-Learning, Temporal Difference Learning	**(08 Hours)**

Text/ Reference Books:

1. Pattern Recognition and Machine Learning. Christopher Bishop.
   Machine Learning. Tom Mitchell.

2. Pattern Classification. R.O. Duda, P.E. Hart and D.G. Stork.
3 Data Mining: Tools and Techniques. Jiawei Han and Michelline Kamber.
4 Elements of Statistical Learning. Hastie, Tibshirani and Friedman.
Springer.

Learning outcomes:

After completing the study of the discipline “Machine Learning ”, the student are expected to:

- Able to understand complexity of Machine Learning algorithms and their limitations;
- Able to understand modern notions in data analysis oriented computing;
- Capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
- Capable of performing experiments in Machine Learning using real-world data.

Core 24: Advanced Algorithms

Course Outline:-

Advanced Algorithm is seen as an empirical thoughts of design and analysis course which is a fundamental and important part of computer science. This course introduces students to advanced techniques for the design and analysis of algorithms and explores a variety of applications that cover several advanced topics like P, NP-Complete problems, Btree, Fibonacci heaps, Disjointsets, hashing, network design, algorithms in machine learning, internet algorithms, and nearest neighbor algorithms. It also boosts various useful ideas, including randomization, probabilistic analysis, amortized analysis, competitive analysis, eigenvalues, high dimensional geometry, random walks etc.

Course objectives:

The course is designed to train the graduates in:

- Advanced topics in algorithm.
- To develop concept, ideas for any problem.
- To be enable to formalize with theoretical computer algorithms.

UNIT- 1	Design Paradigms Overview: Overview of complexity notations, Divide and Conquer method, Greedy and Dynamic Programming	**(08 Hours)**
UNIT- 2	Backtracking, Branch and Bound, Max Flow Problem, String Matching etc.	**(08 Hours)**
UNIT- 3	Brief overview of Notations and Recurrence analysis, Amortized analysis, B- Trees, AVL trees	**(08 Hours)**
UNIT- 4	Dictionaries and tries, Binomial Heaps, Fibonacci Heaps, Disjoint Sets, Union by Rank and Path Compression	**(06 Hours)**
UNIT- 5	Randomized Algorithms and Parallel Algorithms: Randomized Algorithms: Las Vegas and Monte Carlo algorithms, Applications on graph problems, Finger Printing, Pattern Matching, Primality testing algorithm	**(06 Hours)**
UNIT- 6	Introduction, Combinatorial optimization, approximation factor, PTAS, FPTAS, Approximation algorithms for vertex cover, set cover, TSP, subset-sum problem etc., Analysis of the expected time complexity of the algorithms	**(06 Hours)**

Text/References Book:

1) Introduction to Algorithms: T.H. Cormen, C.E.Leiserson and R.L. Rivest
2) Fundamentals of Algorithmics : G.Brassard and P.Bratley
3) Approximation Algorithms: Vijay V.Vazirani
4) Randomized Algorithms: R. Motwani and P.Raghavan
5) Parallel Computing: Theory and Practice: M. J. Quinn
6) Introduction to Parallel Computing: T. G. Lewis and H. El-Rewini

Course Outcomes:

Graduates after completing the course shall gain:

- Ability to understand algorithms.
- Ability to develop concepts, logics towards solving a unknown problem in IT and research.
- Ability to get formalizes theoretical concepts of computer algorithms.

Core 25: Big Data Analytics

Course Objectives:

- To understand the need of Big Data, challenges, and different analytical architectures
- Installation and understanding of Hadoop Architecture and its ecosystems
- Processing of Big Data with Advanced architectures like Spark.

UNIT- 1 Introduction to Big Data, Types of Digital Data, Characteristics of Data, Evolution of Big Data, Data Storage and Analysis, Characteristics of Big Data, Big Data Architecture, Requirement for new analytical architecture, Challenges in Big Data analytics, Need of big data frameworks

UNIT- 2 Requirement of Hadoop Framework, Design principle of Hadoop, Comparison with other system, Hadoop Components, Hadoop versions, HDFS, Map Reduce Programming: I/O formats, Map side join, Reduce Side Join, secondary sorting, Pipelining MapReduce jobs

UNIT- 3 Introduction to Hadoop ecosystem technologies: Serialization: AVRO, Co-ordination: Zookeeper, Databases: HBase, Hive, Scripting language: Pig, Streaming: Flink, Storm

UNIT- 4 MapReduce workflows, unit tests with MRUnit, test data and local tests, anatomy of MapReduce job run, classic Map-reduce, YARN, failures in classic Map-reduce and YARN, job scheduling, shuffle and sort, task execution, MapReduce types, input formats, output formats.

UNIT- 5 Spark Framework, Writing Spark Application, Spark Programming in Scala, Python, R, Java, Application Execution

UNIT- 6 SQL Context, Importing and Saving data, Data frames using SQL, GraphX overview, Creating Graph, Graph Algorithms

Text/References Book:

- Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, John Wiley & Sons, 2013.
- Douglas Eadline,"Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem", Pearson Education, 2016
- Mike Frampton, "Mastering Apache Spark", Packt Publishing, 2015.
- TomWhite,"Hadoop:The Definitive Guide",O'Reilly,4thEdition,2015.
- NickPentreath,MachineLearningwithSpark,PacktPublishing,2015.
- Mohammed Guller, Big Data Analytics with Spark, Apress,2015
- Donald Miner, Adam Shook, "Map Reduce Design Pattern", O'Reilly, 2012

Course Outcomes: The student will be able to

- Discuss the challenges and their solutions in Big Data
- Understand and work on Hadoop Framework and eco systems.
- Explain and Analyse the Big Data using Map-reduce programming in Both Hadoop and Spark framework.
- Demonstrate spark programming with different programming languages.

Core 26: Data Mining and Warehousing

Course Objectives: To develop the basic understanding of Data Mining Algorithms, Applications of Data Mining Algorithms and Information Retrieval

UNIT- 1 Fundamental of Data Mining: Data Mining, History of Data Mining, Data Mining Strategies, Data Mining Techniques, KDD process, Applications of Data Mining, Challenges and Future of data mining. Data Preprocessing and Data Warehousing: Data, information, knowledge, and intelligence, Types of data, Data warehouses, Data cleaning, Data de-normalization, Data transformation, Data quality measures, OLAP technology, OLAP vs OLTP. Data Sampling.

UNIT- 2 The Classification Task: Introduction to classification, Decision trees, Random forests, Naïve Bayes', K-NN, SVM, ANNs, applications of classification. Model evaluation techniques- ROC, Lift Charts, cost and utility, Parsimony, Bagging and Boosting, The model ranking approach.

UNIT- 3 The Clustering Task: Introduction to clustering, Distance measures, types of clustering-hierarchical: agglomerative and divisive, Non-hierarchical: Partition based, Density based, Probability based, K-means clustering, Self-organizing concept, self-organizing maps, SOM algorithm, cluster validation, strength and weaknesses of clustering algorithms, applications of clustering.

UNIT- 4 Association Rule Mining: Concepts of association rules, relevance and functions of association rules, the problem of large data set, Apriori algorithm, scalable association rule mining-FP-Growth algorithm, Applications of ARM, strength and weaknesses of ARM.

UNIT- 5 Information Retrieval: Boolean Retrieval, The term vocabulary and postings lists: Document delineation and character sequence decoding, determining the vocabulary of terms. Dictionaries and tolerant retrieval: Search structures for dictionaries, spelling correction. Scoring, term weighting and vector space model, the vector space model for scoring, variant tf-idf functions.

UNIT- 6 Computing scores in a complete search system: Efficient scoring and ranking, components of an information retrieval system. Evaluation in information retrieval. Relevance feedback and query expansion: Relevance feedback and pseudo relevance feedback, global methods for query reformulation.

Text /Reference Books:

1) Data Mining Concepts & techniques: Jai wei Han and Micheline Kamber, Morgan Kaufman.
2) Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, Introduction to Information Retrieval, Cambridge University Press
3) Data Mining Techniques : Arun K. Pujari, Universities Press, Fourth Edition, ck and ps2016.
4) Mastering Data Mining: M. Berry and G. Linoff, John Wiley & Sons., 2000
5) Data Mining: Methods and Techniques: A B M Shawkat Ali, Saleh A. Wasimi, 2009, Cengage Learning

Course Outcomes: The student will be able to

- Develop the skills to gain a basic understanding of Data Mining Algorithms and their Applications
- Introduce students to Data Mining Algorithms from the engineering perspective.
- To give design methodologies for Data Mining Algorithms
- To provide knowledge for model tuning and over fitting avoidance
- To understand the Data Mining Algorithms and Information Retrieval their implementations
- To demonstrate Data Mining and Information Retrieval Algorithms applications in solving the real-world tasks

Semester- VIII

Core 28: Natural Language Processing

Course Objectives:

- To understand Levels of Language Analysis, Organization of Natural language Systems
- To learn Linguistic Background: An outline of English syntax.
- To learn Grammars and Parsing, Morphological Analysis, Parsing with Features, Various Lexicon Resource & Knowledge Source
- To understand Grammars for Natural Language, Ambiguity Resolution

UNIT I : Introduction to Natural Language Understanding: The study of Language, Evaluating Language Understanding Systems, Different levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems

UNIT II: Linguistic Background: An outline of English syntax.

UNIT IIII: Grammars and Parsing: Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Transition Network Grammars, Top-Down Chart Parsing, Morphological Analysis and the Lexicon.

UNIT IV: Parsing with Features, Augmented Transition Networks, Various Lexicon Resource & Knowledge Source, Study of Word Net and Indo Net

UNIT V: Grammars for Natural Language: Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language, Handling questions in Context-Free Grammars, Hold mechanisms in ATNs. Human preferences in Parsing, Encoding uncertainty, Deterministic Parser, Study of POS Tagger, Stemmer

UNIT VI

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

Books:

1) JAMES ALLEN, Natural Language Understanding, 2/e, Pearson Education, 2003.
2) D. JURAFSKY, J. H. MARTIN, Speech and Language Processing, Pearson Education, 2002.
3) CHRISTOPHER D. MANNING, HINRICH SCHÜTZE, Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, Massachusetts.1999.
4) U. S. TIWARY, TANVEER SIDDIQUI, Natural Language Processing and Information Retrieval, Oxford University Press (2008).
5) AKSHAR BHARATI, VINEET CHAITANYA, RAJEEV SANGAL, Natural Language Processing: A Paninian Perspective

Course Outcome: After completion of this course students will be able to design a model of a prototype language.

Core 29: Neural Networks & Deep Learning

Course Objectives: To develop the basic understanding of ANNs and various ANN Algorithms, Applications of ANN Algorithms in Machine Learning and data analysis.

UNIT- 1 Overview of biological neurons: Structure of biological neurons relevant to ANNs. Fundamental concepts of Artificial Neural Networks: Models of ANNs; Feed-forward & feedback networks; learning rules; Hebbian learning rule, perception learning rule, delta learning rule, Widrow-Hoff learning rule, correction learning rule, Winner –lake all learning rule, etc.

UNIT- 2 Single layer Perception Classifier: Classification model, Features & Decision regions; training & classification using discrete perceptron algorithm, single layer continuous perceptron networks for linearly separable classifications.

UNIT- 3 Multi-layer Feed forward Networks: linearly non-separable pattern classification, Delta learning rule for multi-perceptron layer, generalized delta learning rule, Error back-propagation training, learning factors, Examples. Single layer feedback Networks: Basic Concepts, Hopfield networks, Training & Examples

UNIT- 4 Self-Organizing Networks: Introduction, Self-organizing concept, self-organizing maps, SOM algorithm, adaptive resonance theory (ART), ART algorithm, and variations of ART algorithm. Adaptive pattern classification.

UNIT- 5 Associative memories: Linear Association, Basic Concepts of recurrent. Auto associative memory: retrieval algorithm, storage algorithm; Bi-directional associative memory, Architecture, Association encoding & decoding, and Stability.

UNIT- 6 Introduction to Deep Learning: Deep learning vs. machine learning, significance of deep learning, deep neural networks vs. traditional neural networks. Convolutional neural networks (CNNs): convolution, pooling, padding, and stride. Image classification using CNNs.

Text / Reference Books:

1) Neural networks a comprehensive foundation, Simon Haykin, Pearson Education 2nd Edition 2004.
2) Artificial neural networks - B.Vegnanarayana Prentice Halll of India P Ltd 2005
3) Neural networks in Computer intelligence, Li Min Fu TMH 2003
4) "Neural Networks, Fuzzy Logic and Genetic Algorithms", S. Rajasekaran and G. A. V. Pai, PHI,
5) 2003.
6) Introduction to artificial neural systems", Jacek M. Zurada, 1994, Jaico Publ. House.

Course Outcomes: The student will be able to

- Develop the skills to gain a basic understanding of neural network theory and its applications.
- Introduce students to artificial neural networks an engineering perspective
- To give design methodologies for artificial neural networks
- To provide knowledge for network tuning and over fitting avoidance
- To offer neural network implementations
- To demonstrate neural network applications on real-world tasks

PG Semester- I

Core 33: Image Processing & Computer Vision

Course Objectives:

- know the fundamental techniques for image processing, video processing, and computer vision
- understand the basics of analog and digital video: and apply machine learning in the field of computer vision.

UNIT- 1 Overview of image processing systems, Image formation and perception, Continuous and digital image representation, Image quantization: uniform and non-uniform, visual quantization (dithering)

UNIT- 2 Image contrast enhancement: linear and non-linear stretching, histogram equalization, Continuous and discrete-time Fourier Transforms in 2D; and linear convolution in 2D

UNIT- 3 Image smoothing and image sharpening by spatial domain linear filtering; Edge detection, Discrete Fourier transform in 1D and 2D, and image filtering in the DFT domain

UNIT- 4 Median filtering and Morphological filtering, Color representation and display; true and pseudo color image processing, Image sampling and sampling rate conversion (resize)

UNIT- 5 Image segmentation and Feature Extraction Various methods of image segmentation, edge detection, object proposals, SIFT features. Multi-view Geometry (2 weeks) Shape from stereo and motion, feature matching, surface fitting, Active ranging Object Recognition: Traditional Methods HoG/SIFT features, Bayes classifiers, SVM classifiers

UNIT- 6 Object Recognition: Deep Learning Methods : Image classification, object detection and semantic segmentation, adversarial attacks. Various neural network architectures, visualization techniques. Motion analysis and Activity Recognition: Motion detection and tracking, Inference of human activity from image sequences

Text/ Reference Books:

1) Forsyth and Ponce, “Computer Vision – A Modern Approach”, Second Edition, Prentice Hall, 2011.
2) Emanuele Trucco and Alessandro Verri, “Introductory Techniques for 3-D Computer Vision”, Prentice Hall, 1998.
3) Olivier Faugeras, “Three Dimensional Computer Vision”, MIT Press, 1993.
4) Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 2011
5) Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image Processing, Analysis and Machine Vision”, Third Edition, CL Engineering, 2013.

Course Outcomes: The student will be able to

- know the fundamental techniques for image processing, video processing, and computer vision
- understand the basics of analog and digital video: video representation and transmission
- acquire the basic skill of designing image/video compression
- Familiarize himself/herself with image/video compression standards

Core 34: Software Agent and Swarm Intelligence

Unit I
Introduction to Software Agents: Software agents in business and private area, Economic potential, Definition, Characteristics, Classification.

Unit II
Areas of Influence of software agents
Architecture – BDI architecture, Deliberative agent architecture, Architecture of Reactive agent, A few existing architectures. Mobile architecture – Remote programming, Remote procedure call, Advantages and disadvantages, Technical implementations, Layers of base software, Migration, A few existing architectures – Rao-Georgeff BDI architecture, Brooks Subsumption architecture, Muller Interrap architecture.

Unit III
Communication – Distributed problem solving, Communication methods: Blackboard, Messege passing, KQML. Cooperation Protocols - Contract net systems, Partial Global Planning, Negotiations, Matching and brokering. Learning and Planning, Security.

Unit IV
Honey Bee Colony – Introduction, Decentralized decision making, Where to forage?, Exploration, Exploitation method, Waggle dance, Where to live?, House hunting, Ant Colony Optimization – Introduction, ACO modelling, Solving travelling salesman problem using ACO,

Unit V
Particle Swarm Optimization – PSO modelling, Variants of PSO, Applications, Introduction to Swarm Robotics

Text Books:

1) Intelligent software agents: foundations and applications by Walter Brenner, Rudiger Zarnekow, Hartmut Witting Springer, 1998.
2) Swarm intelligence: introduction and applications By Christian Blum, Daniel Merkle., Springer 2008

PG Semester- II

Core 39: Project Work in Industry or Institution

Objective:-
Before starting internship, the student will gain prerequisite knowledge of working in an industry or academic research. Major projects should encourage students to solve real-life problems.

Outcome:-
Students should be able to discover existing literature and devise new methodologies andmechanisms to solve unsolved problems.

Departmental Elective Course:

CLOUD COMPUTING

Course Objectives:

- Demonstrate the various Distributed technologies to perform the complex task in highly distributed environment.
- Demonstrate the service oriented architecture to provide on-demand services to Internet users.
- Design service level agreements (SLA) to meet the guaranty ¬¬¬services in Cloud Environment.
- Design Energy efficient Scheduling techniques to balance the Workload in a distributed environment.
- Design Energy Efficient model for sustainable cloud platform for next decade various novel service integration paradigm.

UNIT- 1 Introduction Introduction to Cloud Computing, Roots of Cloud Computing: Fundamental concepts of Distributed Systems, Cluster Computing, Grid Computing, and Mobile Computing.

UNIT- 2 Cloud Models Basics of Cloud Computing Concepts, Characteristics of Cloud Computing, Need for Cloud, Cloud Deployment models: private, public, hybrid and community cloud, Cloud Services: Resource-as-a-Service (RaaS), Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS), Examples of each service.

UNIT- 3 Cloud Services RaaS: Usage of Physical resources like servers, networks, data center etc, IaaS: Virtualization,. PaaS: Integrated lifecycle platform: Google App Engine, Microsoft Azure, Anchored life cycle platform: Salesforce platform, SaaS: Characterizing SaaS, Salesforce's software environment.

UNIT- 4 Resource Scheduling for Cloud Computing: - Introduction, Virtual Machine provisioning and Migration Services, Scheduling techniques of Virtual machines for resource reservation, Cloud Service Scheduling hierarchy

UNIT- 5 Economic models for Resource-allocation scheduling , Heuristic Models for task –execution scheduling : Static Strategies , Dynamic Strategies , Heuristic Schedulers.

UNIT- 6 Cloud Applications Cloud Applications, Cloud challenges, Cloud Security and privacy issues, Mobile Cloud Computing, Integration of Cloud with Wireless Sensor Network and its application.

Text/Reference Books:

1. Cloud Computing Bible by Barrie Sosinsky, Wiley Publication, 2011.
2. Cloud Computing: A Practical Approach by Anthony T. Velte Toby J. Velte, Robert Elsenpeter, The McGraw-Hill Publication, 2010.
3. Cloud Computing: Concepts, Technology and Architecture by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, 1st Edition, Prentice Hall.
4. Cloud Computing: Data-Intensive Computing and Scheduling by Frederic Magoules , Jie Pan, and Fei Teng. CRC Press. Taylors & Francis Group.

Course Outcomes:

- Student will able to understand basic concepts required to develop cloud computing applications.
- Student will able to develop applications for cloud computing to provide on-demand services required for users.
- Student will able to understand the service oriented architecture such as IaaS, PaaS and SaaS.
- Student will able to design and implement a novel cloud computing application in simulation environment.
- Student will able to do comparative study and analysis of different economic cloud computing models with existing conventional software developing methodologies.

Departmental Elective Course:

PARALLEL PROCESSING

Objective: After completion of this course students will be able to understand architectural design that provides the parallel computational power to the computer.

Unit-I: Pipeline and Vector Processing: Nonlinear and linear pipelining, Multiprocessor, Multicomputer, Super computer. Array Processors. Scope and Application of Parallel approach.

Unit-II: Paradigms of parallel computing: SIMD, Systolic; Asynchronous - MIMD, reduction paradigm. Hardware taxonomy: Flynn's classifications, Handler's classifications. PRAM model and its variants: EREW, ERCW, CRCW, PRAM algorithms, Sorting network, Interconnection RAMs. Parallelism approaches - data parallelism, control parallelism.

Unit-III: Parallel Processors: Taxonomy and topology - shared memory mutliprocessors, distributed memory networks.

Unit-IV: Processor organization - Static and dynamic interconnections. Embeddings and simulations.

Unit-V: Performance Metrices: Laws governing performance measurements. Metrices - speedup, efficiency, utilization, cost, communication overheads, single/multiple program performances, bench marks.

Unit-VI: Scheduling and Parallelization: Scheduling parallel programs. Loop scheduling. Parallelization of sequential programs. Parallel programming support environments.

BOOKS:

1. M. J. Quinn.Parallel Computing: Theory and Practice, McGraw Hill, New York, 1994.
2. T. G. Lewis and H. El-Rewini. Introduction to Parallel Computing, Prentice Hall, New Jersey, 1992.
3. T. G. Lewis.Parallel Programming: A Machine-Independent Approach, IEEE Computer Society Press, Los Alamitos.
4. Sima and Fountain, Advanced Computer Architectures, Pearson Education.
5. Mehdi R. Zargham, Computer Architectures single and parallel systems, PHI.
6. Ghosh, Moona and Gupta, Foundations of parallel processing, Narosa publishing.
7. Ed. Afonso Ferreira and Jose' D. P. Rolin, Parallel Algorithms for irregular problems - State of the art, Kluwer Academic Publishers.
8. Selim G. Akl, The Design and Analysis of Parallel Algorithms, PH International.

Learning Outcomes:

At the end of this course, the student will be able to:

- Understand uniprocessor computer architecture
- Understand the computer architecture (i.e., pipelining and superscalar processor design and memory hierarchy)
- Understand parallel hardware and parallel software
- Understand shares-memory management
- Understand distributed-memory with MPI
- Understand general-purpose GPU

Departmental Elective Course:

Ad-hoc & Wireless Network

Course Objectives:

- To develop the knowledge of Adhoc and Wireless Sensor Network
- To understand the basics of Ad-hoc & Sensor Networks.
- To learn various fundamental and emerging protocols of all layers.
- To study about the issues pertaining to major obstacles in establishment and efficient management of Ad-hoc and sensor networks.
- To understand various security practices and protocols of Ad-hoc and Sensor Networks.

UNIT- 1 Introduction: What is an Ad Hoc Network?, Types of Ad hoc Mobile Communications , Types of Mobile Host Movements, Challenges Facing Ad hoc Mobile Networks, Ad hoc wireless Internet, Issues in Designing a Routing Protocol for Ad Hoc Wireless Networks, Classifications of Routing Protocols: Table–Driven Routing Protocols, Destination Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP), Source–Initiated On–Demand Approaches, Ad hoc On–Demand Distance Vector Routing (AODV), Dynamic Source Routing (DSR), Temporally Ordered Routing Algorithm (TORA), Location–Aided Routing (LAR), Power–Aware Routing (PAR), Zone Routing Protocol (ZRP).

UNIT- 2 Wireless Sensor Networks: Introduction to Wireless sensor networks, Single-sink single-hop WSN, Single-sink multi-hop WSN, Multi-sink multi-hop WSN, Advantages of ad-hoc/sensor networks, Node and Network Architectures, Wireless Sensor Device Architecture, Network Architectures, Main features of WSANs, Current and future research on WSANs

UNIT- 3 Applications of WSNs: Positioning and animals tracking, Entertainment, Logistics, Transportation, Industrial Control and Monitoring, Home Automation and Consumer Electronics, Security and Military Sensing, Asset Tracking and Supply Chain Management, Intelligent Agriculture and Environmental monitoring, Health Monitoring.

UNIT- 4 Technologies for WSNs: ZigBee technology, Ultrawide bandwidth technology, Bluetooth technology, Comparison among technologies

UNIT- 5 The Physical Layer: Introduction, Wireless Propagation Models: The Free Space Propagation Model, The Two-Ray Ground Model, The Log-Distance Path Model, Energy Dissipation Model, Error Models: The Independent Error Model, The Two-State Markov Error Model, Sensing Models: The Binary Sensing Model, The Probabilistic Sensing Model

UNIT- 6 Communication protocols for WSNs, MAC protocols: Scheduled protocols, LEACH protocol, Guo protocol, TRAMA protocol, Contention-based protocols, Zhong protocol, DMAC protocol, PAMAS protocol, SMAC protocol, Routing protocols: Issues in designing routing protocols, Classification of routing protocols, Flat routing, Flooding and gossiping, SPIN protocol, Directed diffusion protocol, Rumour routing, Gradient-based routing, Hierarchical routing, LEACH protocol,

Text/Reference Books:

1) Roberto Verdone, Davide Dardari, Gianluca Mazzini and Andrea Conti, “Wireless Sensor and Actuator Networks: Technologies, Analysis and Design”, Academic Press, 2008.
2) Miguel A. Labrador and Pedro M. Wightman, “Topology Control in Wireless Sensor Networks-with a companion simulation tool for teaching and research”, Springer Science, 2009.
3) Azzedine Boukerche, “Handbook of Algorithms for Wireless Networking and Mobile Computing”, Chapman & Hall/CRC, 2005.

Course Outcomes: The student will be able to

- Identify different issues in wireless ad hoc and sensor networks.

- To analyze protocols developed for ad hoc and sensor networks.
- To identify and address the security threats in ad hoc and sensor networks.
- Establish a Sensor network environment for different type of applications.

Departmental Elective Course:

INTERNET OF THINGS

Course Outline:
Internet of Things (IoT) is presently an emerging technology worldwide. Government, academia, and industry are involved in different aspects of research, implementation, and business with IoT. IoT cuts across different application domain verticals ranging from civilian to defense sectors. These domains include agriculture, space, healthcare, manufacturing, construction, water, and mining, which are presently transitioning their legacy infrastructure to support IoT. Today it is possible to envision pervasive connectivity, storage, and computation, which, in turn, gives rise to building different IoT solutions. IoT-based applications such as innovative shopping system, infrastructure management in both urban and rural areas, remote health monitoring and emergency notification systems, and transportation systems, are gradually relying on IoT based systems.

Course Objectives
- To teach state of art of wireless sensor networks
- To discuss importance of communication protocols.
- To teach challenges in routing protocols and overview of protocols across different layers.
- To teach basics of Internet of Things.

Unit I: Introduction: Overview of Wireless Sensor Networks – Characteristics, Applications, Design objectives, challenges. Different types of sensors and applications of wireless sensor networks.

Unit II: Medium Access Control protocols for Wireless sensor networks: Functions of MAC layer, Fundamental MAC protocols, Objectives of MAC protocols, Energy efficiency in MAC design, Fixed assignment protocols, demand assignment protocols.

Unit III: Network and Transport Layer protocols for wireless sensor networks: Fundamentals and Challenges of Routing protocol, routing strategies in wireless sensor networks. Traditional transport protocols, Transport protocols for sensor networks.

Unit IV: Basics on Internet of Things: Introduction, Components of IoT, IoT communication technologies and protocols, developing basic IoT applications.

Unit V: Physical and Data link layer protocols for IoT like Zigbee and Z-Wave. Network layer protocols for IoT like RPL.

Unit VI: Transport layer protocols for IoT. Application layer protocols for IoT like MQTT. Emerging technologies in IoT.

Text/References:
1. Jun Zheng, Abbas, " Wireless sensor networks A networking perspective", WILEY, 2009.
2. Kazem Sohraby, Daniel Minoli, & Taieb Znati, ―Wireless Sensor Networks-Technology, Protocols, And Applications‖, John Wiley, 2007
3. Thomas Haenselmann, ―Wireless Sensor Networks: Design Principles for Scattered Systems‖, Oldenbourg Verlag, 2011
4. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1st 9 Edition, VPT, 2014.
5. E. H. Callaway, Jr. E. H. Callaway, Wireless Sensor Networks Architecture and Protocols: CRC Press

6. F. Zhao and L. Guibas, Wireless Sensor Network: Information Processing Approach, Elsevier.
7. A. Hac, Wireless Sensor Network Designs, John Wiley & Sons 8. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013

Learning Outcomes

- At the end of the course students can be able to:
- Understand technological background of sensor networks.
- Able to design applications using Raspberry Pi.
- Design and apply various existing routing protocols of sensor networks.
- Design the architecture and reference model of IoT.

Departmental Elective Course:

COMPUTING & VEDIC MATHEMATICS

About the subject: The course is mainly based upon the book Lilavati, originally, authored by the mathematician Bhaskaracarya in vedic perid. The Lilavati is a book on arithmetic written in the twelfth century. It has been used as a textbook for 800 years in India.

Objective: The objective of the course is to introduce the methods used for arithmetic in vedic period. After studying this course the students will be able to use vedic methods in arithmetic which are easy to use and/ or whose computer algorithms are of less complexity.

UNIT I
Brief introdction to Bhascarararyacarya and his works
Units of measurement. Indo-Arabic numericals, Place value system, Arithmetic operations of addition, subtarction, multiplication and division, Methods of finding squares, Square root, Methods to find cube, Cube roots.
Fractions - Operations, Addition and subtraction, Multiplication, Division, Squares, Cubes, Square roots, Cube roots.

UNIT II
Rules concerning zero, Reverse process, To find an unknown quantity, Method of transition, square transition. Linear and quadratic equations, The rule of three, Inverse proportion, The rule of five, Rules for Barter, Simple interest, Combinations.

UNIT III
Progression: Arithmetic and geometric progressions, and series, Mensuration

UNIT IV
Volume, Wood Cutting, Volume of a heap of grain, Shadows, Pulverization, Concatenation (Permnutations, Partitions etc.)

UNIT V
Pingal's binary number system, Different types of Meru Prastar (including Pascal triangle).

UNIT VI
Computer programmming based on methods given in Unit I - IV and comparing complexity with the respective modern methods. Square root as numerical approach as prescribed in Sulbasutras.

References

1. Krishnaji Shankara Patwardhan, Somashekhara Amrita Naimpally and Shyam Lal Singh, Lilavati of Bhaskaracarya: A Treatise of Mathematics of Vedic Tradition, Motilal Banarsidaas Pub. Pvt. Ltd., Delhi, 2021, 2001.
2. Kapil Dev Dwivedi, Shyam Lal Singh, The Prosody of Pingala: A Treatise of Vedic and Sanskrit Metrics with applications of Vedic Mathematics (with Hindi and English Translation), Vishwavidyalaya Prakashan, Varanasi, 2008.
3. Bibhutibhusan Datta, The Science of The Sulba, University of Calcutta, 1991.
4. Bhakaracarya, Leelavati, Srivenkateshwar Steem Press, Bombay, 1979.
5. Pt Kedar Nath, छन्द: शास्त्रम, Chaukhambha Publisher, Varanasi, 2002.
6. A B Padmnabha Rao, Bhascaracarya's Leelavati, Chinmay International Foundation Shodha Sansthan Adi Sankara Nilyam, Veliyanad, 2014.
7. K Ram Subramaniam, Ganitanand, Indian Society of History of Mathematics, 2015.
8. Pandit Yudhishtir Mimansa, वैदिक-छन्दोमीमांस, Ramlal Kapoor Trust, Haryana, 2006.
9. N. H. Fadke, लीलावती पुनर्दर्शन, Sarvahak Prakashan, Pune, 1971.
10. Sisheel Trivedi, छंद का आधुनिक रचना विधान, Rashtra Prakashan, Delhi, 1880.
11. Pandit Ganpatideva Shastri, गणितकौमुदि, Chaukhambha SansKrit Series, Varansi, 1969.

The first book covers syllabus from Unit I – III of the course. Second book covers last half of the Unit III. Third book covers some part of Unit IV.

Departmental Elective Course:

SOFTWARE DEFINED NETWORKS

Outline of the Course:-
The proposed course outline is to describe advanced technology in communication based on requirement and need for industry and academia. The designed course covers protocol framework which can support Software oriented networking protocol architecture which supports virtualization. Now-a-days network virtualization play key role in creating virtual local area networks (VLAN) to control the traffic generated by enterprise networks. The proposed course covers to design a state of art technology which can support Software Defined Networking.

Objectives: -

- To design protocol architecture which can meet the challenges of current user demands and needs data transmission.
- To demonstrate the performance of proposed SDN supportive protocols with Open Flow enabled networks.
- To learn simulator basics this can support SDN Functionalities.
- To design and detail study of security attacks which are going to occur in SDN supportive enterprise networks.

UNIT-I: Introduction, Centralized and Distributed Control and Data Planes, Introduction What Do They Do? Distributed Control Planes Centralized Control Planes Conclusions.

UNIT-II: Open Flow: Introduction, Hybrid Approaches Conclusions SDN Controllers Introduction General Concepts Layer 3 Centric Plexxi Cisco One PK Conclusions.

UNIT-III: Network Programmability: Introduction, the Management Interface the Application-Network DivideModern Programmatic Interfaces, I2RSModern Orchestration Data Center Concepts and Constructs.

UNIT-IV: Introduction: The Multitenant Data Centerthe Virtualized Multitenant Data Center SDN Solutions for the Data Center Network VLANsEVPN, VxLan, NVGRE, Conclusions, Network Function Virtualization Introduction Virtualization and Data Plane I/O Services Engineered Path, Service Locations and Chaining, NFV at ETSI, Non-ETSI NFV Work, Conclusions.

UNIT-V: Network Topology and Topological Information Abstraction Introduction, Network Topology, Traditional Methods, LLDP, BGP-TE/LS,ALTO,I2RS Topology Building an SDN Framework, Introduction, Build Code First; Ask Questions Later, The Juniper SDN Framework, IETF SDN Framework(s), Open Daylight Controller/Framework, Policy, Conclusions.

UNIT-VI: Use Cases for Bandwidth Scheduling, Manipulation, and Calendaring, Introduction, Bandwidth Calendaring, Big Data and Application Hyper-Virtualization for Instant CSPF, Expanding Topology, Conclusion, Use Cases for Data Center Overlays, Big Data, and Network Function Virtualization, Introduction, Data Center Orchestration, Puppet (DevOps Solution), Network Function Virtualization (NFV), Optimized Big Data, Conclusions.

Text/Reference Books:

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, Chuck Black
3. Software Defined Networking with Open Flow by SiamakAzodolmolky (Author).

Course Outcomes:

- Understanding between conventional networks and SDN Supportive networks to provide high throughput based on user needs.
- Understanding of Network Virtualization and requirements and changes in hardware design point of view.
- Virtual LAN supportive protocols and its operations to enhance the Quality of Service parameters.
- Understand and identify security vulnerabilities in open flow based networks. Understand prevention mechanism for well-known security attacks in conventional networks.
- Adaptive machine learning techniques to prevent security attacks in SDN.

Departmental Elective Course:

MOBILE COMPUTING

Course Outline: - The proposed course introduces the fundamentals of Wireless Communication, issues challenges in wireless communication. The course detail explanation of various generation of Wireless Networks generation those are 2G, 3G and 4G. The proposed course covers technical details layer wise, which are Physical layer parameters such as modulation, demodulation and multiplexing techniques. MAC Layer issues such as various channel accessing schemes those are pure aloha, slotted aloha and p-persistent. The course covers in detail technical details such as packet formats of IEEE-802.11 standards for Medium accesses control to avoid collisions. Network Layer issues and challenges and details of various routing algorithms such as AODV, DSR and TORA protocols. Various TCP Enhancements for existing TCP Version which are TCP-RENO, Tahoe and SACK protocols for reliable and end-to-end communication for improving the performance.

Objectives:

- The objective is to understand various generations of Mobile Communication such as 2G, 3G and 4G.
- To study various issues and challenges in Physical layer such as analog to digital conversion and various modulation and demodulation techniques.
- Illustration of various physical layer issues like inter symbol interference, ISI Mitigation. Physical layer parameter such as refraction, reflection and signal to noise ratio to improve the quality.
- Demonstrate the Various MAC Layer challenges in Wireless Networks when compared to structured Networks.

- Study of various Routing Layer Protocols suitable for Wireless Ad-Hoc Networks and Protocol operations.
- Study of various TCP Layer issues and challenges for Wireless Networks.

UNIT-I
Introduction, Applications, A short history of wireless Communication, A market for Mobile Communications, Some open research topics, A Simple Reference Model.

UNITT-II Wireless Transmission, Overview, Frequency for radio transmission, Regulations, Signals, Antennas, Signal Propagation, Path Loss of radio Signals, Additional signal Propagation effects, Multi-path Propagation. Multiplexing, Modulation, Spread Spectrum.

UNIT-III
Medium Accesses Control, Motivation for Specialization MAC, Hidden and exposed terminals, near and Far Terminals, SDMA, FDMA, TDMA, CDMA.

UNIT-IV
Wireless LAN, IEEE 802.11: System Architecture, Protocol architecture, Physical Layer, MAC Control Layer, MAC Management, 802.11b, 80.11a,HIPERLAN:

UNIT-V
Bluetooth: User Scenario, Architecture, Radio Layer, Link Manager Protocol, L2CAP, SDP, IEEE 802.15.

UNIT-VI
Mobile Network Layer, Mobile IP, Dynamic Host Configuration Protocol, Mobile Ad-Hoc Networks, Mobile Transport Layer, Classical TCP Improvements.

Text Reference Books:

1) Mobile Communications by JochenH.Schiller.
2) Mobile Computing, Technology Applications and Service Creation by Asoke K Talukder and Roopa R Yavagal.
3) Stojmenovic and Cacute**,** "Handbook of Wireless Networks and Mobile Computing", *Wiley,* 2002, ISBN0471419028.
4) Reza Behravanfar, "Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML", ISBN: 0521817331, Cambridge University Press, October 2004,
5) Adelstein, Frank, Gupta, Sandeep KS, Richard III, Golden, Schwiebert, Loren, "Fundamentals of Mobile and Pervasive Computing", ISBN: 0071412379, McGraw-Hill Professional, 2005.

Departmental Elective Course:

HUMAN COMPUTER INTERACTION

Human Computer Interaction deals with how humans interact with the Computer System. The course will uncover how designs are aesthetically done, details of ergonomics and evaluation techniques

Course Objectives

- Demonstrate how input-output channels work.
- To introduce the details of interaction and design.
- To discuss different evaluation techniques and cognitive methods.
- Laboratory exercises to be covered in Lab sessions.

UNIT-I

The Human: input-output channels, Human memory, thinking, emotions, individual differences, psychology and the design of interactive systems.
The Computer: Text entry devices with focus on the design of key boards, positioning, pointing and drawing, display devices.

UNIT-II

The Interaction: Models of interaction, ergonomics, interaction styles, elements of WIMP interfaces, interactivity, experience, engagement and fun. Paradigms for Interaction.

UNIT-III

Design Process: The process of design, user focus, scenarios, navigation design screen design and layout, iteration & prototyping. Usability Engineering
Design rules: Principles to support usability, standards, guidelines, rules and heuristics, HCI patterns.

UNIT-IV

Evaluation Techniques: Definition and goals of evaluation, evaluation through expert analysis and user participation, choosing an evaluation method.

UNIT-V

Cognitive methods: Goals and task hierarchies, linguistic models, challenges of display based systems, physical and device models, cognitive architectures.

UNIT-VI

Communications and collaborations models: Face to Face communication, conversations, Text based communication, group working.

BOOK:

Human Computer Interaction; Alan Dix et.al, 3rd ed., Pearson.

Outcomes:

At the end of this course, the student will be able to:

- Develop better interfaces that are more usable.
- Demonstrate understanding of design guidelines, principles and standards.

Departmental Elective Course:

FRACTAL THEORY

Course Objectives: Fractals are strange but beautiful objects that appear in nature and arts as results of self-organization and self similarity. In dynamics they are ”responsible” for the presence of highly-irregular, chaotic motions. The course is an introduction to a circle of topics in fractal geometry and chaotic dynamics. After completion of this course students will be able to draw fractals and develop understanding of chaos.

Unit 1: The basic concepts of geometric iteration, principle of feedback processes Fundamentals of Fractals, Types of fractal (mathematical and nature), self-similarity, fractal dimension, multiple reduction copy machines, the chaos game, fractals in nature, and decoding fractals. Chaos wipes out every computer. Chaos in (nature and Math).

Unit 2: Standard mathematical fractals (Seirpinski carpet ,gasket, cantor dust , koch curve etc), limits and self similarity, Fractal dimension, Types of fractal dimension, implementation of standard fractal and calculating their dimensions.

Unit 3: Affine transformation, Transformations, composing simple transformations, classical fractals by IFS, drawing the classical fractals using IFS.

Unit 4: Deterministic Chaos, analysis of chaos, periodic points, sensitivity, fixed points, logistic map, sensitivity dependence of initial condition, implementation and detailed analysis of logistic map (mathematically and in real life).

Unit 5: L-systems, turtle graphics (graphical interpretation of L-Systems), Networked MRCMs, L-Systems tree and bushes, Growing classical fractals with L-Systems and their implementation.

Unit 6: Julia set (Fractal basin boundaries), complex numbers, escape and prisoners set, filled Julia set, Quaternion Julia set, exploring Julia sets by varying complex numbers.
Mandelbrot set, geometric features and properties , study structure of Mandelbrot set. Implementation of Julia set and Mandelbrot set.
Project: Students will complete a final creative project that involves researching an application to fractals and chaos. Students will create something to go along with the project, like artwork, a short story, or a computer generated image.

Text Book:

1) Ya. Pesin and V. Climenhaga, ”Lectures on Fractal Geometry and Dynamical Systems”, Student Mathematical library, 52, AMS, Providence, RI, 2009

Learning Outcomes:

- Iterated Function System
- Escape-time Fractals
- Behavior of Chaotic logistic map
- L-system

Departmental Elective Course:

GAME THEORY

Course Outline:

This course provides an introduction to Game Theory. Game Theory is a mathematical framework that studies strategic interactions amongst self-interested decision makers. It has applications in a wide variety of areas, including statistical decision theory, artificial intelligence (online learning, multi-agent systems), economics and business (auctions, pricing, bargaining), political science (stability of government, military strategy), philosophy (ethics, morality and social norms) and biology (evolution, signaling behavior, fighting behavior).

Course Overview:
The novel concepts of game theory and how to find different equilibrium solutions to different types of games will be extensively covered in this course. These will be explained and elucidated with relevant examples.
This course provides a rigorous treatment of solution concepts for games with perfect and imperfect information including rationalxizability, Nash and subgame perfect Nash equilibria. It covers topics such as auction, VNM utility function, bargaining game etc. It also discusses cooperative game solution concepts-core, Shapley value and bayesian game with Cournot's duopoly.

UNIT 1- Games with Perfect Information-Strategic Games; Nash Equilibrium and Existence Properties; Some Games in Normal Form, Nash Equilibria in Zero-Sum Games, Bräss' Paradox, and more on Mixed Strategies, Games in Extensive Form, Market Equilibrium and Pricing.

UNIT 2- Electoral Competition: Median Voter Theorem; Auctions: Definitions and The role of Knowledge; Decision Making and Utility Theory; Mixed Strategy Equilibrium;

UNIT 3-The Paretian System Equilibrium, and Walrasian General Equilibrium Theory, Von Neumann and Morgenstern Utility Function, Theory of Risk Aversion, Equilibrium Theory.

UNIT 4- Sealed Bid Auctions, VCG Procedures, Generalized Vickrey Auctions, VCG Procedures, Cournot Competition and Stackelberg Equilibrium; Arrow's Impossibility Theorem, Gibbard-Satterthwaite Theorem, Bargaining Game with Alternating Offers; Bargaining Game with Alternating Offers (General Utilities); Nash Bargaining Solution; Stable Marriages; Multi-Item Auctions;

UNIT 5-Cooperative Game Theory: Cores; Stable Sets and Shapley Value.

UNIT 6- Strategic Games with Imperfect Information-Bayesian Games; Cournot's Duopoly with Imperfect Information; Radio Spectrum, With Arbitrary Distribution of Valuations

TEXT/REFERENCE BOOKS:

1) "Fun and Games: A Text on Game Theory", Ken Binmore, A.I.T.B.S Publishers.
2) "A Course in Game Theory", Martin J. Osborne and Ariel Rubinstein, MIT Press.
3) Prajit Dutta, Strategies and Games, MIT Press

Learning Outcomes:

On successful completion of this course, students will be able to model competitive real world phenomena using concepts from game theory and identify optimal strategy and equilibrium solution for such models. They will be ready to explain the potential or proven relevance of game theory and its impact in various fields of human interaction which involve conflict of interest between two or more participants.

Departmental Elective Course:

Blockchain & Cyber Security

Course Objectives:

To develop the Blockchain & Cyber Security

UNIT- 1 Introduction to Blockchain: Digital Trust, Asset, Transactions, Distributed Ledger Technology, Types of networks, Components of blockchain (cryptography, ledgers, consensus, smart contracts) Introduction to security, attacks, computer criminals, security services.

UNIT- 2 Cryptography: Substitution ciphers, transposition cipher, confusion, diffusion, symmetric and asymmetric encryption. DES, odes of DES. Hash function, key exchange, digital signatures and certificates.

UNIT- 3 PKI and Cryptography: Private keys, Public keys, Hashing, Digital Signature, Principles of Public Key Cryptosystems, Factorization, RSA Algorithm, security analysis of RSA, Exponentiation in Modular Arithmetic. Key Management in Public Key Cryptosystems: Distribution of Public Keys, Distribution of Secret keys using Public Key Cryptosystems. Discrete Logarithms, Diffie-Hellman Key Exchange.

UNIT- 4 Cryptocurrency: Bitcoin creation and economy, Limited Supply and Deflation, Hacks, Ethereum concept and Ethereum classic, Hacks Why it is so revolutionary.

UNIT- 5 Blockchain Applications: Building on the Blockchain, Ethereum Interaction - Smart Contract and Token (Fungible, non-fungible), Languages, Blockchain-as-a-service.

UNIT- 6 Blockchain Use Cases: Finance, Industry and Blockchain in Government Security and Research Aspects: Blockchain Security (DDos), Research Aspects in Blockchain, AI, Blockchain and Big Data.

Text/Reference Books:

1) Bahga, A., & Madisetti, V. (2017). Blockchain Applications: A Hands-On Approach. VPT.
2) Stalling Williams: Cryptography and Network Security: Principles and Practices, 4th Edition, Pearson Education, 2006.
3) Kaufman Charlie et.al; Network Security: Private Communication in a Public World, 2nd Ed., PHI/Pearson.

Course Outcomes: The student will be able to

- Understand what and why of Blockchain
- Explore the major components of Blockchain and Identify a use case for a Blockchain application
- Create your own Blockchain network application

Departmental Elective Course:

COMPILER DESIGN

Course Objectives:

- To understand various models to produce tokens which are inputs syntax phase.
- Try to understand various parsing techniques such as top-down and bottom-up parsing techniques.
- Symbol Table generation and mechanisms to store information while scanning source code from various phases of the compiler.
- Semantic analysis to check the meaning of the sentences in a particular sentence.

UNIT- 1 Overview of Compilation: Phases of Compilation – Lexical Analysis, Regular Grammar and regular expression for common programming language features, pass and Phases of translation, interpretation, bootstrapping, data structures in compilation – LEX lexical analyzer generator.

UNIT- 2 Top down Parsing: Context-free grammars, Top down parsing – Backtracking, LL (1), recursive descent parsing, Predictive parsing, Preprocessing steps required for predictive parsing.

UNIT- 3 Bottom up parsing: Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing, handling ambiguous grammar, YACC – automatic parser generator.

UNIT- 4 Semantic analysis: Intermediate forms of source Programs – abstract syntax tree, polish notation and three address codes. Attributed grammars, Syntax directed translation, Conversion of popular Programming languages language Constructs into Intermediate code forms, Type checker.

UNIT- 5 Symbol Tables: Symbol table format, organization for block structures languages, hashing, and tree structures representation of scope information. Block structures and non-block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records.

UNIT- 6 Code optimization: Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding, DAG representation.
Data flow analysis: Flow graph, data flow equation, global optimization, redundant sub expression elimination, Induction variable elements, Live variable analysis, Copy propagation. Object code generation: Object code forms, machine dependent code optimization, register allocation and assignment generic code generation algorithms, DAG for register allocation.

Text/ Reference Books:

1) Principles of compiler design -A.V. Aho . J.D.Ullman; Pearson Education.
2) Modern Compiler Implementation in C- Andrew N. Appel, Cambridge University Press
3) Modern Compiler Design- Dick Grune, Henry E. BAL, Cariel T. H. Jacobs, Wiley dreamtech.
4) Engineering a Compiler-Cooper & Linda, Elsevier.
5) Compiler Construction, Louden, Thomson.

Course Outcomes: The student will be able to

- Students are able to understand the functionality of compiler design various phases.
- Able to learn functionalities of various phases.
- Able to design phases of compiler as a programming exercise.
- Able Design various parsing techniques such as SLR, LALR and CLR.

Departmental Elective Course:

Dot Net Technologies

Course Objectives:

- Identify the Basics of .Net Framework, Architecture, and programming
- Build GUI applications using .NET Framework and Visual Basics/C#.

UNIT- 1 Introduction to .Net, .Net Framework Features & Architecture, CLR, Common Type System, MSIL, Types of Assemblies, Class Libraries. Event Drive Programming, Methods and Events, Related with Mouse and Keyboard.

UNIT- 2 Programming into Visual Studio, Toolbox, Properties Window, Form Designer, Form Layout, Immediate Window.

UNIT- 3 VB.Net language, Variables, Data Types, Scope & Lifetime of a Variable, Arrays, Types of Array, Control Array, Subroutine, Functions, Passing Argument to Functions, Optional Argument, Returning Value from Function.

UNIT- 4 Conditional and Loop Statement. Loading, Showing and Hiding Forms, Working with Multiple Forms, Controlling one Form within Another, Overview of C#, Structure of C# Program, C# in .Net.

UNIT- 5 GUI Programming, Windows Form and Properties, Methods, Events, Text Box Control, Label Control, Button Control, List box, Combo Box, Checked Box, Picture Box, Radio Button, Scroll Bar, Timer Control, Common Dialog Control, Designing Menus, MDI Forms.

UNIT- 6 ADO .Net Architecture, Create Connection, Accessing Data Using Data Adapters and Datasets, Using Command & Data Reader, Data Bind Controls, Displaying Data in Data Grid. Data Form Wizard, SQL queries, Database Using Ado.Net Object Model, Connection Object, Command Object, Add, Delete, Move & Update Records to Dataset, Executing Queries.

Text/ Reference Books:

1) Steven Holzner, Visual Basic .NET Programming Black Book, Dreamtech Publications
2) E. Balagurusamy, Programming in C# A primer, Tata McGraw-Hill Publishing Company Limited, Delhi.
3) Jeffrey R. Shapiro, Visual Basic.NET: The Complete Reference, McGraw Hill Education

Course Outcomes: Students will be able to

- Design and Develop GUI based Applications using Vb.Net and C#.
- Integrate different components of the .net framework, including the database.

Departmental Elective Course:

Data Science Algorithms

Course Objectives:

- To develop the basic understanding of various Data Science Algorithms, Applications of Data Science Algorithms.

UNIT- 1 Introduction Data Science: The Art of Data Science, Volume, Velocity, Variety, Machine Learning, Supervised and Unsupervised Learning, Predictions and Forecasts, Innovation and Experimentation , The Dark Side-Big Errors, Privacy. Theories, Models, Intuition, Causality, Prediction, Correlation

UNIT- 2 Classification: K-nearest neighbors (K-NNs), Text classification using K-nearest neighbors, Naïve Bayes Theorem and Extended Naïve Bayes Theorem for Classification.

UNIT- 3 Decision Trees (DTs): Information theory, information entropy and information gain. Decision tree construction using ID3 algorithm and classification using DTs. Random Forest-construction and classification.
UNIT- 4 Clustering: Types of clustering, clustering using AGNES and DIANA, clustering using k-means, k-means vs. k-mediods. Density based clustering using DBSCAN. Clustering documents using clustering.

UNIT- 5 Regression Analysis: Linear regression, gradient descent algorithm for regression model. Non-linear regression model.

UNIT- 6 Temporal Data Analysis: Temporal data, Temporal Sequence analysis, Time Series Analysis, Analyzing data trends using regression, creating a time-dependent model.

Text/ Reference Books:

1) Data Science Algorithms in a Week, David Natingga, Packt Publishing Ltd., Birminggham
2) Data Science: Theories, Models, Algorithms, and Analytics, S.R. Das
3) Algorithms for Data Science Book by B. C. H. Steele, John Chandler, and Swarna Reddy
4) Data Science For Dummies Book by Lillian Pierson, A Willey Brand

Course Outcomes: The student will be able to

- Enrich their knowledge in the area of Data Science and its allied areas.
- Understand the various Data Science Algorithms
- Understand the applications of various Data Science Algorithms
- Case study based on any one of the Data Science Algorithm

Departmental Elective Course:

Software Project Management

Course Objectives:

- To develop the knowledge of Distributed Systems and how they are utilized in contemporary computing

UNIT- 1 Introduction Project Management (PM) Fundamentals, People, Process, and Product, Technology Classic mistakes, PMI Processes, Software project phases, Organizational structures, Project charter Statement of Work (SOW)

UNIT- 2 Planning Phase Development lifecycle models, Matching lifecycles to projects, Project plans Work Breakdown Structures (WBS)

UNIT- 3 Estimation and Budgeting Estimation, Budgeting, Project selection, NPV, ROI, Payback models, Scheduling: Project network diagram fundamentals, PERT techniques, Gantt charts, Critical chain scheduling

UNIT- 4 Risk and Change Management Risk management, Change control, More MS-Project

UNIT- 5 Development Management Team models, Requirements process, Configuration management, Software metrics, Programming languages & tools, managing conflict and motivating, MS-Project: Assigning Resources

UNIT- 6 System Test Process Test specifications, Black box and white box testing, Test scripts, Unit and integration testing, Acceptance test specifications, Test tools, Final Phases & Other Issues: Project Recovery, Documentation, Cutover/Migration, Post Project Reviews, Closing

Text/ Reference Books:

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

Course Outcomes: The student will be able to

- Learn Conventional Software Management and Evolution of Software Economics
- Understand Project Organizations and Responsibilities
- Understand the evolution and applications of operations in various fields, mathematically formulate linear programming problems and solve them using different techniques
- Construct a project network and apply program evaluation review technique and critical path method to find date of completion of project and other project related metrics

Departmental Elective Course:

ADBMS

Course Objectives:

- To develop the understanding of various Advanced Database Systems and how they are utilized in present era of computing.

UNIT- 1 Introduction: Distributed Data processing, Distributed Database Systems (DDBMSs), Promises of DDBMSs, Complicating factors and Problem areas in DDBMSs. Overview of relational database system. Distributed DBMS Architecture: DBMS Standardization, Architectural models for Distributed DBMS, Distributed DBMS Architecture.

UNIT- 2 Distributed Database Design: Alternative design Strategies, Distribution design issues, Fragmentation, Allocation. Semantic Data Control: View Management, Data security, Semantic Integrity Control.

UNIT- 3 Overview of Query Processing: Query processing problem, Objectives of Query Processing, Complexity of Relational Algebra operations, characterization of Query processors, Layers of Query Processing. Introduction to Distributed Transaction Management: Definition of Transaction, Properties of transaction, types of transaction.

UNIT- 4 Distributed Concurrency Control: Serializability theory, Taxonomy of concurrency control mechanisms, locking based concurrency control algorithms.

UNIT- 5 Parallel Database Systems: Database servers, Parallel architecture, Parallel DBMS techniques, Parallel execution problems, Parallel execution for hierarchical architecture. Database Interoperability: Database Integration, Query processing

UNIT- 6 Distributed Object Database Management systems: Fundamental Object concepts and Object models, Object distribution design. Architectural issues, Object management, Distributed object storage, Object query processing. Transaction management. Introduction to other Advanced Database Systems:
Multimedia Databases, Spatial Databases, Deductive Databases, etc.

Text/ Reference Books:

1) Principles of Distributed Database Systems, M.TamerOzsu, Patrick Valduriez, 2nd Edition, 1999.
2) Distributed Databases principles and systems, Stefano Ceri, Giuseppe Pelagatti, TMH, 2008.
3) Database System Concepts, 7th Edition, Abraham Silberschatz, Henry F. Korth, S. Sudarshan
4) Fundamentals of Database Systems, Pearson, Elmasri Ramez, Navathe Shamkant

Course Outcomes: The student will be able to

- Enrich their the knowledge in the area of various Advanced Database Systems
- Understand the Distributed query processing.
- Understand Distributed Transaction management and Distributed concurrency control.
- Get the idea of Parallel Database Systems and Distributed Object Database Management systems and their architecture
- Understand the concept of Concurrency Control , Recovery, and security mechanism in Distributed environment
- Undergo the case study of Distributed Database System

Departmental Elective Course:

High Performance Computing

Course Objectives:

- Demonstrate the scientific application execution methodology in Highly Distributed Computing Environment.
- To study Processor Architecture and Memory Hierarchies which support for HPC.
- To learn Programming strategies for parallel computing to solve highly complex scientific problems.
- To understand the parallel computer concepts different types of parallel architecture, hard ware design and compilers principles.
- Illustration of well-known mathematical examples to understand the basic concepts of parallel computation which are highly required to solve scientific applications.
- Detail study of various kinds of mathematical examples where parallel computations are involved, for example in linear algebra solving system of equations and matrix decomposition, Fourier transforms.

UNIT- 1 Single-processor Computing, The Von Neumann architecture, Modern processors, Memory Hierarchies, Multi core architectures, Locality and data reuse, Programming strategies for high performance, Power consumption, Review questions.

UNIT- 2 Parallel Computing, Introduction, Quantifying parallelism, Parallel Computers Architectures, Different types of memory access, Granularity of parallelism, Parallel programming, Topologies, Multi-threaded architectures ,Co-processors, Remaining topics, Computer Arithmetic, Integers, Real numbers, Round-off error analysis, Compilers and round-off, More about floating point arithmetic, Conclusions.

UNIT- 3 Numerical treatment of differential equations, Initial value problems, Boundary value problems, Initial boundary value problem, Numerical linear algebra, Elimination of unknowns, Linear algebra in computer arithmetic, LU factorization, Sparse matrices, Iterative methods, Further Reading.

UNIT- 4 High performance linear algebra, Collective operations, Parallel dense matrix-vector product, LU factorization in parallel, Matrix-matrix product, Sparse matrix-vector product, Parallelism in solving linear systems from Partial Differential Equations (PDEs), Computational aspects of iterative methods , Parallel preconditions ,Ordering strategies and parallelism, Operator splitting, Parallelism and implicit operations ,Grid updates ,Block algorithms on multi core architectures.

UNIT- 5 Applications, Molecular dynamics, Force Computation, Parallel Decompositions, Parallel Fast Fourier Transform, Integration for Molecular Dynamics, Sorting, Brief introduction to sorting Odd-even transposition sort, Quicksort, Bitonic sort, Graph analytics, Traditional graph algorithms, Real world' graphs, Hypertext algorithms, Large-scale computational graph theory

UNIT- 6 N-body problems, The Barnes-Hut algorithm, The Fast Multipole Method, Full computation, Implementation Monte Carlo Methods, Parallel Random Number Generation, Examples, Computational biology Dynamic programming approaches, Suffix tree.

Text /Reference Books:

1) Introduction to High Performance Scientific Computing Evolving Copy - open for comments Victor Eijkhout, Edmond Chow, Robert van de Geijn.
2) High Performance Computing (RISC Architectures, Optimization & Benchmarks), Charles Severance, Kevin Dowd, Oreilly.
3) High Performance Computing (RISC Architectures, Optimization & Benchmarks), Georg Hager, Gerhard Wellein, CRC Press.
4) Introduction to High-Performance Scientific Computing (Scientific and Engineering

Computation), Lloyd D. Fosdick, Elizabeth R. Jessup

Course Outcomes: The student will be able to

- Able to understand the difference between sequential architecture and parallel architecture to execute the scientific applications.
- Understand the way to develop parallel algorithm and way of execution on parallel computing environment.
- Analysis of time and space complexity for a particular mathematical problem in sequential as well as parallel.
- Writing programs for to solve Partial differential equations (PDE) and Matrix decomposition.
- Solve some Computational biology applications using Dynamic programming approaches.

Departmental Elective Course:

Cyber Security

Course Outline: - The designed course covers basic security aspects to protect network from various kind of attacks. Various kinds of encryption and decryption techniques to protect data frommalicious users. Course covers DES and RSA logarithms which are basic symmetric and asymmetric algorithms. Course also covers various authentication mechanisms.

Objectives:

- To understand security fundamentals.
- To learn Cryptography Principles such as key exchange, digital signature and certificates.
- Demonstration of various encryption decryption schemes by taking plain text.
- Generation Message Authentication codes for different kinds of messages.

UNIT I: Introduction to security, attacks, computer criminals, security services.
Cryptography: Substitution ciphers, transposition cipher, confusion, diffusion, symmetric and asymmetric encryption. DES, odes of DES. Hash function, key exchange, digital signatures and certificates.

UNIT II: Public Key Cryptosystems: Principles of Public Key Cryptosystems, Factorization, RSA Algorithm, security analysis of RSA, Exponentiation in Modular Arithmetic. Key Management in Public Key Cryptosystems: Distribution of Public Keys, Distribution of Secret keys using Public Key Cryptosystems. Discrete Logarithms, Diffie-Hellman Key Exchange.

UNIT III: Message Authentication & Hashing: Birthday Paradox and General case of Duplications, Basic functions of Message Authentication and Hashing, Introduction to Hash & MAC algorithms.
Digital Signatures: RSA Based, El Gamal Signatures, Undeniable Signatures.
Authentication: Model of Authentication Systems, Impersonation, Substitution and spoofing games, Authentication schemes for mutual authentication based on shared secret, two-way public key, one-way public key, Mediated Authentication, One way Authentication

UNIT IV: Security in networks: Threats in networks, network security controls, firewalls, intrusion detection system Administrating security: Security planning, risk analysis, physical security, Ethical issues in security

UNIT V: Cyber security increasing threat landscape, Cyber security terminologies- Cyberspace on-state attack vector, attack surface, threat, risk, vulnerability, exploit, exploitation, hacker., non-state actors, Cyber terrorism, Protection of end user machine, Critical IT and National Critical Infrastructure, Cyberwarfare, Case Studies

Text/Reference Books

1. Stalling Williams: Cryptography and Network Security: Principles and Practices, 4th Edition, Pearson Education, 2006.
2. Kaufman Charlie et.al; Network Security: Private Communication in a Public World, 2nd Ed., PHI/Pearson.
3. Pieprzyk Josef and et.al; Fundamentals of Computer Security, Springer-Verlag, 2008.
4. Trappe & Washington, Introduction to Cryptography, 2nd Ed. Pearson.

Course Outcomes:-

- Students will able to understand the security features to improve the reliability incommunication system.
- Able to find the encrypt message by using various encryption mechanisms.
- Various message authentication codes and message digest algorithms to improve the security aspects in data transmission.

Departmental Elective Course:

Ecommerce

Course Objective: This course focuses on Presents concepts and skills for the strategic use of e-commerce and related information technology from three perspectives: business to consumers, business-to-business, and intra-organizational. Examination of e-commerce in altering the structure of entire industries, and how it affects business processes including electronic transactions, supply chains, decision making and organizational performance.
In addition, some of the major issues associated with e-commerce—security, privacy, intellectual property rights, authentication, encryption, acceptable use policies, and legal liabilities. Students will build their own web presence and market it using an online platform.

UNIT-I: **Introduction to E-commerce**
Meaning and Concept; Objectives; Advantages and Disadvantages; Technical components andfunctions of e-commerce; E-Commerce and E-Business; Traditional Commerce vs. E-Commerce; Forces Driving E-Commerce; Growth of E-Commerce; E-Commerce Opportunities for Industries; Future of E-Commerce.

UNIT-II: **E-Commerce Model and Websites**
Forms of E-Commerce- Business to Consumer; Business to Business; Business to Government; Other Models – Brokerage Model, Aggregator Model, Community Model and Value Chain Model; Transaction Process: Basic concepts of EDI; Applications of EDI.

UNIT-III : **Electronic Payment System and E-Security & Privacy**
Special Features of Electronic Payment System; Types of E-Payment Systems-E-Cash, E-Cheque, Credit Card, Smart Card and Electronic Purses, Risk and E-Payment Systems; Secure Electronic Transaction (SET)
Security Risk of E-commerce; Types of Intruders; Types of Threats; Security Tools-Cryptography; Software packages for privacy; Security algorithms; Digital Signature and Firewalls.

Unit-IV Web Designing

Preparing Web pages and Website, Use of HTML and DHTML and scripting languages; Websites Generation- Concept and Meaning; Objectives and Advantages; Types of Websites; Website Designing Principles; Methods of Promoting Website; Searching the Website; Factors for Growth of Websites

Text/References Books:

1. William Stallings, Cryptography and Network security Principles and practice.
2. Kalakota, Ravi and Whinston, Andrew B., Electronic Commerce – A Manager's Guide, Pearson Education, Inc
3. Diwan, Prag and Sunil Sharma: Electronic Commerce – A manager's Guide to E-Business, Vanity Books International, Delhi.
4. Janal, D.S.: On-line Marketing Hand Book, Van Nostrand Reinhold, New York.
5. Kosivr, David: Understanding Electronic Commerce, Microsoft Press, Washington.
6. Robert W. Sebesta, Programming the World Wide Web, Sixth Edition, Published byAddison-Wesley. Copyright © 2011 by Pearson Education, Inc.
7. Mlnoli and Minol: Web Commerce Technology Handbook, Tata McGraw Hill, New Delhi.
8. Schneider Gary P: Electronic Commerce, Course Technology, Delhi.

Learning Outcome: At the end of this course, students should be able to

1. Identify and apply relevant problem solving methodologies.
2. Identify advantages and disadvantages of technology choices such as merchant serversoftware and electronic payment options
3. Discuss electronic commerce and the stakeholders and their capabilities and limitations inthe strategic convergence of technology and business.
4. Analyse features of existing e-commerce businesses, and propose future directions orinnovations for specific businesses.

Departmental Elective Course:

COMPUTER GRAPHICS

Objective:

After completion of this course students will be able to draw 2 dimensional graphical objects using geometrical algorithms and performs operations on them.

S. No.	Content	Hours
1	**Basic of Computer Graphics:** Basic of Computer Graphics, Applications of computer graphics, Display devices, Random and Raster scan systems, Graphics input devices, Graphics software and standards	6
2	**Graphics Primitives:** Points, lines, circles and ellipses as primitives, scan conversion algorithms for primitives, Fill area primitives including scanline polygon filling, inside-outside test, boundary and flood-fill, character generation, line attributes, area-fill attributes, character attributers. Aliasing, and introduction to Anti Aliasing (No anti aliasing algorithm).	7
3	**Two Dimensional Graphics:** Transformations (translation, rotation, scaling), matrix representation, homogeneous coordinates, composite transformations, reflection and shearing, viewing pipeline and	8

	coordinates system, window-to-viewport transformation, clipping including point clipping, line clipping (Cohen-Sutherland, liang- bersky, Mid-point), polygon clipping	
4	**Three Dimensional Graphics:** 3D display methods, polygon surfaces, tables, equations, meshes, curved lies and surfaces, quadric surfaces, spline representation, cubic spline interpolation methods, Bezier curves and surfaces, B-spline curves and surfaces.3D scaling, rotation and translation, composite transformation, viewing pipeline and coordinates, parallel and perspective transformation, view volume and general (parallel and perspective) projection transformations.	8
5	**Illumination and Colour Models:** Light sources – basic illumination models – halftone patterns and dithering techniques; Properties of light – Standard primaries and chromaticity diagram; Intuitive colour concepts – RGB colour model – YIQ colour model – CMY colour model – HSV colour model – HLS colour model; Colour selection.	6
6	**Animations & Realism:** Design of Animation sequences – animation function – raster animation – key frame systems – motion specification – morphing – tweening. **Fractal Graphics:** Tiling the plane, Recursively defined Curves, Koch curves, C curves, Dragons, space filling curves, turtle graphics, ray tracing.	6
	Total	41

Text Book/References:

1. Computer graphics, Donald Hearn and Paulin Baker.
2. Computer graphics, Schaum's Series.
3. Computer graphics: Principles and Practice - James D Foley.
4. Principles of Interactive Computer Graphics – William M Newman
5. Mathematical Elements for Computer Graphics, David Rogers, J Alan Adams
3. Geometric Tools for Computer Graphics - Philip J Schneider, David H Eberly.

Learning Outcomes:

1. To understand basic display Devices, Input Devices.
2. To learn Line Drawing Algorithms, Circle and Ellipse Drawing Algorithms, 2D Transformations, Line and Polygon clipping, Color Fill Methods, 2D Projections.
3. To learn introduction of Fractal Graphics.

Departmental Elective Course:

OPEN SOURCE OPERATING SYSTEM

Course Outline: - The course is designed to develop an understanding of Free and Open Source Software. It will discuss how shell variables and shell scripting can be used.

Course Objectives

- To introduce the concept of Free and Open Source Software.
- To understand the role of Kernel and shell.
- To demonstrate different commands in Linux.

UNIT- 1	Introduction: open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean any cost. History : BSD, The Free Software Foundation and the GNU Project.	**(7 Hours)**

UNIT- 2	Open Source Development Model Licences and Patents: What Is A License, Important FOSS Licenses (Apache, BSD, GPL, LGPL).	**(7 Hours)**
UNIT- 3	Open source operating system: Overview of Open Source operating systems, Introduction to Linux and Unix, Flavours of *NIX Operating systems.	**(7 Hours)**
UNIT- 4	Introduction to Linux: File system, Shell and Kernel,vi editor, shell variables, command line programming.	**(8 Hours)**
UNIT- 5	Filters and commands: Pr, head, tail, cut, paste, sort, uniq, tr, join, etc, grep, egrep, fgrepetc., sed,awk etc.,granting and revoking rights.Any other relevant topic.	**(8 Hours)**
UNIT- 6	Shell Programming: Types of Shells, Shell Meta Characters $#, $*, $?, Shell Variables, Shell Scripts. Debugging scripts, echo, read, operators, keywords, Integer Arithmetic and String Manipulation, Decision Making: if-else-elif-fi, case-esac. Loop Control; while, for, until, break & continue Functions, I/O Redirection and Piping, Exception Handling. Creating shell programs for automating tasks, file handling,trapping signals etc	**(8 Hours)**

BOOKS:

1. Brian W. Kernighan, The Practice of Programming, Pearson Education.
2. Bach Maurice J, Design of the Unix Operating system, PHI.
3. Daniel P. Bovet, Understanding the Linux Kernel, Oreilly.

Outcomes:-

At the end of this course, the student will be able to

- Demonstrate the understanding of different *NIX operating systems.
- Work on Linux using different commands and develop shell scripts.

Departmental Elective Course:

QUANTUM COMPUTING

Course Objectives

CO#1. To introduce the fundamentals of quantum computing
CO#2. The problem-solving approach using finite dimensional mathematics.

Course Outcome

CO#1. Basics of complex vector spaces
CO#2. Quantum mechanics as applied in Quantum computing.
CO#3. Architecture and algorithms
CO#4. Fundamentals of Quantum computations

Syllabus

Unit I. Complex numbers and its geometrical representations, Complex vector spaces, inner products and Hilbert spaces, Hermitian and unitary matrices, Tensor products of vector spaces Deterministic Systems

Unit II. Dirac formalism, superposition of states, entanglement Bits and Qubits. Qubit operations,

Hadamard Gate, CNOT Gate, Phase Gate, Z-Y decomposition, Quantum Circuit Composition, Basic Quantum circuits.

Unit III. Quantum Algorithm - I: Quantum parallelism, Quantum Evolution, Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Simon's periodicity algorithm.
Unit IV. Quantum Algorithm - II: Grover's search algorithm, Shor's Factoring algorithm. Application of entanglement, teleportation, superdence coding.

Unit V. Quantum programming languages, Probabilistic and Quantum computations, introduction to quantum cryptography and quantum information theory.

Unit VI. Quantum gates and algorithms: Universal set of gates, quantum circuits, Solovay-Kitaev theorem, Deutsch-Jozsa algorithm, factoring , Programming a quantum computer: The IBMQ, coding a quantum computer using a simulator to carry out basic quantum measurement and state analysis.

Text Books
1. Quantum computing explained, David McMahon, Wiley-interscience, John Wiley & Sons, 2008
2. Quantum computing for computer scientists, Noson S. Yanofsky, Mirco A. Mannucci, Cambridge University Press 2008.
3. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press,Cambridge, 2020
Reference Books
1. Quantum computation and quantum information, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press 2010
2. Introduction to Quantum Mechanics, 2nd Edition, David J. Griffiths, Prentice Hall New Jersey 1995

Departmental Elective Course:

Distributed Systems		
Course Pre-requisites: The students should have knowledge of Operating System and Computer Networking		
1	Basic concepts of Distributed System	
Course Objectives:		
	To develop the knowledge of Distributed Systems and how they are utilized in contemporary computing	
Course Outcomes: The student will be able to		
1	Elaborate the types of Distributed System and their components	
2	Understand the underlying architecture of Distributed System	
3	Distinguish between different distributed file system	
4	Understand and apply the Distributed clock synchronization using various algorithms	
5	Understand the concept of Concurrency Control , Recovery, and security mechanism in Distributed environment	
6	Study the case study of Distributed object based systems	
Course Content:		
UNIT- 1	**Introduction to Distributed System:** Concepts, Architectures, transparency Self-management in Distributed system ,Thread, Virtualization, Client, server, code migration Semantics, Remote Procedure Calls, Communication, Naming,	**(10 Hours)**

UNIT- 2	Monolithic kernel, layered systems, virtual machines. Process based models and client server model. The micro-kernel based client-server approach. Inter-process communication and Remote Procedure Call. Tasks and Threads. Examples from LINUX, Solaris 2 and Windows NT.	**(10 Hours)**
UNIT- 3	**File System:** Flat naming, Structure naming and Attribute based naming, Security, Concurrency control and recovery, local area network, distributed languages and communication primitives, case studies of distributed systems.	**(10 Hours)**
UNIT- 4	**Distributed Synchronization:** Clocks and Election algorithm, Consistency Model, Consistency Protocol, Resilience, Reliable communication, Distributed Commit, recovery in Distributed systems. Security issues in distributed systems. Deadlock in distributed systems.	**(10 Hours)**
UNIT- 5	Distributed Operating Systems, Distributed File System, Sun NFS, and the Coda files system. NTFS, UNIX ext2 and ext3.	**(10 Hours)**
UNIT- 6	**Case studies** of Distributed object based systems (CORBA), Distributed web based Systems.	**(10 Hours)**
Internal Assessment:		
Part- A	UNIT TEST- 1 :- UNIT- 1, 2,3	
	UNIT TEST- 2 :- UNIT- 4, 5,6	
PART- B	Assignments: Students should perform theoretical / experimental assignment/s from the list below	
	1. Types of Distributed System and their components	
	2. Understanding the underlying architecture of Distributed System	
	3. To undergo the case studies of distributed systems	
	4. Distributed Transaction Management and Distributed Deadlock Managing	
	5. Case study of Disturbed File System	

Text Books:

Tanenbaum, A. S. and Van Steen, M. “Distributed Systems Principles and” (ISBN 0-13-088893-1), Prentice Hall 2002.

Reference Books:

P. K. Sinha, “Distributed Operating Systems,” PHI. Paradigms.

Bacon, J., “Concurrent Systems”, 2nd Edition, (ISBN 0-201-177-676), Addison Wesley 1998.

Silberschatz, A., Galvin, P. and Gagne, G., “Applied Operating Systems Concepts”, 1st Edition,” (ISBN 0-471-36508-4), Wiley 2000.

Coulouris, G. et al, “Distributed Systems: Concepts and Design, 3rd Edition,” (ISBN 0-201-61918-0), Addison Wesley 2001