

College of Engineering, Pune

(An Autonomous Institute of Govt. of Maharashtra, Permanently Affiliated to S.P. Pune University)

Department of Computer Engineering and Information Technology

Curriculum Structure & Detailed Syllabus (UG Program)

Final Year B. Tech. Computer Engineering

(Revision: 2015-19, Effective from: A.Y. 2018-19)

INDEX

| Sr. No. | Item | Page No |
|----------------|---|----------------|
| 1 | Program Education Objectives (PEOs) and Program Outcomes (POs) | 2 |
| 2 | Correlation between PEOs and Pos | 3 |
| 3 | List of Abbreviations | 3 |
| 4 | Curriculum Structure & Detailed Syllabi | 4 |

Correlation between the PEOs and the POs

| PO→ PEO↓ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------|---|---|---|---|---|---|---|---|
| I | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| II | | ✓ | | | | ✓ | | ✓ |
| III | | ✓ | ✓ | ✓ | ✓ | | | |

Note: The cells filled in with ✓ indicate the fulfillment/correlation of the concerned PEO with the PO.

List of Abbreviations

| Abbreviation | Title |
|--------------|--|
| S.P. P.U. | Savitribai Phule Pune University |
| A.Y. | Academic Year |
| BSC | Basic Science Course |
| EFC | Engineering Foundation Course |
| MLC | Mandatory Learning Course |
| ILOE | Institute Level Open Elective Course |
| SLC | Self Learning Course |
| HSMC | Humanities/Social Sciences/Management Course |
| LLC | Liberal Learning Course |
| SBC | Skill Based Course |
| PCC | Program Core Course |
| DEC | Department Elective Course |
| LC | Laboratory Course |

**B. Tech. Computer Engineering
Sem VII: Effective AY 2018-19**

| | | | Teaching Scheme | | | |
|---------|-------------|--|-----------------|----------|-----------|-----------|
| Sr. No. | Course Type | Course Name | L | T | P | Credits |
| | ILOE | Institute level Open Elective [Science/Technology/Engg.] [To be offered to other Departments] Introduction to Algorithms | 3 | 0 | 0 | 3 |
| 1 | LLC | Liberal Learning Course | 1 | 0 | 0 | 1 |
| 2 | DEC | Department Elective-II | 3 | 0 | 0 | 3 |
| 3 | DEC | Department Elective-III | 3 | 0 | 0 | 3 |
| 4 | SBC | Project Stage-I | 0 | 0 | 4 | 2 |
| 5 | PCC | Parallel Computer Architecture and Programming | 3 | 0 | 0 | 3 |
| 6 | PCC | Cryptography and Network Security | 2 | 1 | 0 | 3 |
| 7 | LC | Department Elective-II Laboratory | 0 | 0 | 2 | 1 |
| 8 | LC | Department Elective-III Laboratory | 0 | 0 | 2 | 1 |
| 9 | LC | Parallel Computer Architecture and Programming Laboratory | 0 | 0 | 2 | 1 |
| 10 | LC | Cryptography and Network Security Laboratory | 0 | 0 | 2 | 1 |
| | | Total Academic Engagement and Credits | 15 | 1 | 12 | 22 |

**B. Tech. Computer Engineering
Sem VIII: Effective AY 2018-19**

| | | | Teaching Scheme | | | |
|---------|-------------|--|-----------------|----------|-----------|-----------|
| Sr. No. | Course Type | Course Name | L | T | P | Credits |
| | MLC | Intellectual Property Rights | 1 | 0 | 0 | 0 |
| 1 | LLC | Liberal Learning Course | 1 | 0 | 0 | 1 |
| 2 | PCC | Compiler Construction | 3 | 0 | 0 | 3 |
| 3 | DEC | Department Elective-IV | 2 | 1 | 0 | 3 |
| 4 | DEC | Department Elective-V | 3 | 0 | 0 | 3 |
| 5 | LC | Project Stage-II | 0 | 0 | 12 | 6 |
| 6 | LC | Compiler Construction Laboratory | 0 | 0 | 2 | 1 |
| 7 | LC | Department Elective-V Laboratory | 0 | 0 | 2 | 1 |
| | | Total Academic Engagement and Credits | 9 | 2 | 14 | 18 |

Sem VII Minor Course: Database Management Systems

Sem VIII Minor Course: Internet Technologies

Sem VII Honors Course: Advanced Computer Networks

Sem VIII Honors Course: Multicore Technology

List of Department Elective Courses

| | |
|----------|---|
| Sem VII | Department Elective-II CBD: Cloud and Big Data Platforms FP: Functional Programming IR: Information Retrieval SDP: Software Design Patterns IOT: Internet of Things Subjects in Association with Domain Experts |
| Sem VII | Department Elective-III AUP: Advanced UNIX Programming ES: Embedded Systems OOMB: Object Oriented Modeling and Design NLP: Natural Language Processing LP: Language Processors (only for IT) Subjects in Association with Domain Experts |
| Sem VIII | Department Elective-IV SA: System Administration S&V: Storage and Virtualization MAN: Mobile and Ad-hoc Networks CFDR: Computer Forensics and Data Recovery CB: Computational Biology Subjects in Association with Domain Experts |
| Sem VIII | Department Elective-V MT: Multicore Technology ADBMS: Advanced Database Management Systems EC: E-commerce CS: Cyber Security DS: Distributed Systems (only for CE) Subjects in Association with Domain Experts |

Semester-VII

(CT) Parallel Computer Architecture and Programming

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Justify the need of high performance provided by parallel computer architecture.
2. Demonstrate quantitative design principles of parallel computing systems.
3. Measure and analyze performance through different benchmarks.
4. Comprehend and differentiate various computer architectures.
5. Understand the building blocks of cluster setup.

Unit I :Fundamentals of Computer Design: Defining Computer Architecture, Trends in Technology, Trends in Power in Integrated Circuits, Trends in Cost, Dependability, Measuring, Reporting and Summarizing Performance, Quantitative Principles of Computer Design, Classification of Computer Architectures, Recent Computing Trends Top 500 Ratings, Fundamentals of Computer Design, Basic and Intermediate concepts of pipelining , Pipeline Hazards, Pipelining Implementation issues. **[6 Hrs]**

Unit II :Instruction-Level Parallelism and Its Exploitation: Instruction-Level Parallelism: Concepts and Challenges, Basic Compiler Techniques for Exposing ILP, Reducing Branch Costs with Prediction, Scheduling, Overcoming Data Hazards with Dynamic Scheduling, Dynamic Scheduling: Algorithm and Examples, Hardware-Based Speculation, Studies of the Limitations of ILP, Limitations on ILP for Realizable Processors, Hardware versus Software Speculation, ILP Support to Exploit Thread-Level Parallelism. **[7 Hrs]**

Unit III :Data-Level Parallelism in Vector, SIMD, and GPU Architectures: Vector Architecture, SIMD Instruction Set Extensions for Multimedia, Graphics Processing Units, CPU/GPU architecture comparison, Detecting and Enhancing Loop-Level Parallelism, Data parallelism and SPMD programming model, Nvidia GPU Case study and Programming Model, Example of ARM Heterogeneous Architecture. **[7 Hrs]**

Unit IV :Thread-Level Parallelism: Introduction to Shared Memory Architectures, Loosely and Tightly coupled multiprocessors, Centralized Shared-Memory Architectures, Snoopy Bus Cache Coherence, Performance of Shared-Memory Multiprocessors, Distributed Shared Memory and Directory Cache Coherence, Basics of Synchronization, Models of Memory Consistency. **[7 Hrs]**

Unit V :Memory Hierarchy Design: Cache Performance, Basic Cache Optimizations, Virtual Memory, Protection and Examples of Virtual Memory, Advanced Optimizations of Cache

Performance, Memory Technology and Optimizations, Protection: Virtual Memory and Virtual Machines, The Design of Memory Hierarchies, Study of Memory Hierarchies in different Processors. **[6 Hrs]**

Unit VI :Parallel Programming Paradigms for different Architectures: Cluster: Cluster and Network of Workstations (COW and NOW), Different ways of building a cluster, Parallel Programming Models: Message Passing, Data Parallel, MPI/PVM, Shared Memory Programming OpenMP, CUDA Programming, Parallel Algorithm examples: Matrix Multiplication, Sorting, Introduction to Parallel Programming Languages. Warehouse-Scale Computers: Architecture, Programming Model and Workloads **[7 Hrs]**

Text Books:

- John L Hennessy, David A Patterson, “Computer Architecture: A Quantitative Approach”, Sixth Edition, Morgan Kaufmann, 2018, ISBN: 978-0-12-811905-1.
- Kai Hwang, Naresh Jotwani, “Advanced Computer Architecture”, Third Edition, Tata McGraw-Hill Edition, 2016, ISBN: 978-0-07-070210-3.

Reference Books:

- D. E. Culler, J. P. Singh, and A. Gupta, “Parallel Computer Architecture”, Second Edition, Morgan Kaufmann, 2017, ISBN:978-1-4987-7271-6.
- McCool, Michael D., Arch D. Robison and James Reinders, “Structured Parallel Programming: Patterns for Efficient Computation”, Morgan Kaufmann, 2012, ISBN: 978-0-12-415993-8.

(CT-) Parallel Computer Architecture and Programming Laboratory

Teaching Scheme:

Laboratory : 2 Hours/week

Examination Scheme:

Continuous evaluation: 30 Marks

Mini Project/ Assignments: 20 marks

End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Evaluate performance using different benchmarks used in high performance computing systems.
2. Implement shared memory programs using pthreads and OpenMP.
3. Implement message passing programs in distributed environment.
4. Program GPU architecture using CUDA.
5. Demonstrate the different steps involved in building of a simple cluster.

List of Assignments:

1. Study of different benchmarks used to evaluate performance of different systems.
 - a. Study of different performance parameters like execution time, throughput etc.
 - b. Application benchmarks suits like PARSEC, SPLASH, SPEC.
2. Program to understand shared memory paradigm using pthreads and related issues.
 - a. Matrix multiplication using pthreads.
 - b. Synchronization and thread scheduling.
3. Program to execute dense matrix multiplication using pthreads.
 - a. Explicit code writing to handle dense matrix multiplication and execution time.
 - b. Different algorithms for dense matrix multiplication.
4. Program to execute dense matrix multiplication using OpenMP clauses.
 - a. Usage of #pragma directives along with OpenMP clauses.
 - b. Comparison with pthread implementation.
5. Study of Cluster building steps - MPI Cluster setup and overview of different routines.
 - a. Different steps to build a MPI cluster over LAN.
 - b. Master-Slave concept and different MPI routines.
6. Program to implement point to point communication using MPI routines.
 - a. Parallelizing Trapezoidal Rule using MPI_Send and MPI_Receive.
7. Program to implement collective communication using MPI routines.
 - a. Gather, Scatter and Broadcast operations.
 - b. Matrix-Vector multiplication execution.
8. Program to implement Merge sort/ Graph Computation algorithm using MPI routines.
 - a. Steps in parallelizing Merge Sort/ Matrix Partitioning.

- b. Execution time comparison of Merge sort/ Graph Computation with normal implementation.
- 9. Program to execute matrix multiplication using CUDA.
 - a. Basic CUDA host, device and memory constructs.
 - b. Thread- warp, block, grid usage.
- 10. Program to study working of Click parallel programming language.
 - a. Simple program to execute Fibonacci example and performance evaluation.
- 11. Program to implement Map-Reduce parallelism for Warehouse -Scale Computer.
 - a. Parallelism using Map-Reduce programming model.
 - b. Example of word count process with key-value pair.

Reference Books:

- Peter S. Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann, Elsevier Series, 2011, ISBN:978-0-12-374260-5.
- Jason Sanders, Edward Kandrot, "CUDA by Example: An Introduction to General Purpose GPU Programming", 2011, ISBN:978-0-13-138768-3.

(CT)Cryptography and Network Security

Teaching Scheme:

Lectures : 2 Hrs/week

Tutorial : 1 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Explain the concepts related to applied cryptography, including plaintext, ciphertext, symmetric cryptography, asymmetric cryptography, and digital signatures
2. Apply concepts of finite mathematics and number theory.
3. Demonstrate the understanding of common network vulnerabilities and attacks, defence mechanisms against network attacks, and cryptographic protection mechanisms.
4. Detect possible threats to different defence mechanisms and different ways to protect against these threats.

Unit I :Introduction: Cryptography and modern cryptography, Need of security, Security services, Basic network security terminology, Security attacks, Classical cryptosystems and their cryptanalysis, Operational model of network security **[4 Hrs]**

Unit II: Mathematical Foundations: Prime Number, relatively prime numbers, Modular Arithmetic, Fermat's and Euler's Theorem, The Euclidean and Extended Euclidean Algorithms, The Chinese Remainder Theorem, Discrete logarithms **[4 Hrs]**

Unit III :Symmetric Key Ciphers: Symmetric Key Ciphers, Feistel Networks, Modern Block Ciphers, Modes of Operation, Cryptanalysis of Symmetric Key Ciphers: Linear Cryptanalysis, Differential Cryptanalysis **[6 Hrs]**

Unit IV :Asymmetric Cryptography: RSA, Key Distribution and Management, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography, hash functions: The Merkle Damgard Construction, Message Digest algorithms: MD5, Secure Hash algorithm (SHA), Message Authentication Codes **[6 Hrs]**

Unit V: Authentication and Web Security: Digital Signatures, Authentication Protocols, Kerberos, X.509 Digital Certificate Standard, Pretty Good Privacy, Secure Socket Layer, Secure Electronic Transaction. Zero knowledge proof **[4 Hrs]**

Unit VI :Network Security: Intruders, Intrusion Detection, Password Management, Worms, viruses, Trojans, Virus Countermeasures, Vulnerabilities in TCP/IP model, Firewalls, Firewall Design Principles **[4 Hrs]**

Text Books:

- V. K. Pachghare, "Cryptography and Information Security", 2nd edition, PHI Learning, ISBN: 978-81-203-5082-3.
- Charlie Kaufman, Radia Perlman, and Mike Speciner, Network Security: Private Communication in a Public World, Prentice Hall, ISBN 0-13-046019-2.

Reference Books:

- William Stallings, "Cryptography and Network Security, Principles and Practices", Pearson Education, Fifth Edition, ISBN: 0-13-60970-9.
- Christopher M. King, "Security architecture, design deployment and operations", Curtis Patton and RSA Press, ISBN: 0072133856.
- Stephen Northcatt, LenyZeltser, "INSIDE Network Perimeter Security", Pearson Education Asia, Second Edition, ISBN: 978-0735712324.
- Robert Bragge, Mark Rhodes, HeithStraggberg, "Network Security the Complete Reference", Tata McGraw Hill Publication, ISBN: 9780072226973.

(CT) Cryptography and Network Security Lab

Teaching Scheme:

Laboratory: 2 Hrs/week

Examination Scheme:

Continuous evaluation: 50 Marks

Oral Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Analyse the optimal features and time required for an encryption technique.
2. Implement cryptographic algorithms in any programming language.
3. Demonstrate the ability to detect attacks on a system and tackle it.
4. Write a security application to protect a system from some attacks.

List of Assignments:

1. Study papers on a network security topic and write a study report
 1. Wireless Network Security,
 2. Key Exchange Protocols,
 3. Block chain.
2. Implement any one classical encryption technique in any programming language.
3. Design and implement a symmetric encryption algorithm based on Feistel structure.
4. Demonstrate how Diffie-Hellman key exchange works with Man-In-The-Middle attack.
5. Study different approaches for Anti-virus software and write one document.
 - a) Examine files to look for viruses by means of a virus dictionary
 - b) Identifying the suspicious behavior from any computer program which might indicate infection
6. Study and demonstrate system hacking and write a report.
 - a) How to crack a password?
 - b) How to use Ophcrack / Crowbar / John the Ripper / Aircrack-ng to Crack Passwords
7. Develop a mini project on
 - a) a hack tool to break the security of a system. OR
 - b) a tool to protect the system from the hack tool.

ILOE (CT) Introduction to Algorithms

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Determine different time complexities of a given algorithm
2. Develop algorithms using various design techniques for a given problem.
3. Determine a design technique and design an algorithm using that technique for a given problem.

Unit 1: : Basic Data Structures

Abstract data types, data types; Stack, Queue, Lists, Graphs; various implementations of the data structures **[8 Hrs]**

Unit 2: Asymptotic notation

Objectives of time and space analysis of algorithms; Order notations (O , Ω , θ notations); (Best average and worst case) time complexity of algorithms such as bubble sort, selection sort, insertion sort, heap sort etc.; Time complexity of recursive programs using recurrence relations **[6 hrs]**

Unit 3: Divide and Conquer:

Binary Search, Quicksort, Mergesort, Strassen's matrix multiplication **[6 Hrs]**

Unit 4: Greedy Algorithms:

Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees (Prim's and Kruskal's algorithms), Optimal merge patterns, Single source shortest paths **[8 Hrs]**

Unit 5: Dynamic Programming:

All pairs shortest paths, 0-1 Knapsack, Travelling salesperson problem, Chained matrix multiplication, Longest common subsequence etc **[8 Hrs]**

Unit 6: Backtracking:

8-Queens problem, Sum of subsets, Graph coloring, 0-1 Knapsack problem. **[4 Hrs]**

Text Books

1. Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein, "Introduction to Algorithms", The MIT Press - 3rd edition, ISBN-13: 978-0262033848
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press, second edition (2008), ISBN-13: 978-8173716126

Reference Books

1. Gilles Brassard and Paul Bratley, "Fundamentals of Algorithmics", PHI, ISBN-13: 978-8120311312
2. Jon Kleinberg and Éva Tardos, "Algorithm Design", Pearson Education India, ISBN-13: 978-9332518643

List of Assignment topics:

1. Implementation of stack, queue and list data structures and one application each.
2. Verification of (worst and average) complexity of Quicksort
3. Implementation of Heap sort.
4. Implementation of greedy algorithm for knapsack problem
5. Minimum Spanning tree as an example of Greedy approach [Prim's or Kruskal's algorithm]
6. Implementation of Single Source and All Pairs Shortest algorithms
7. Finding longest common subsequence of given two sequences using dynamic programming
8. Implementation of sum of subsets problems using backtracking

This is an indicative list. The instructor is encouraged to update the list.

Semester-VIII
(CT) Compiler Construction

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of different phases of compilation.
2. Demonstrate the ability to generate and code lexical and syntax analyzer.
3. Analyze and differentiate different parsing techniques and syntax directed translation schemes and choose the optimal parsing technique.
4. Apply different intermediate code generation and code optimization techniques for various statements.

Unit I : Introduction: Translator Issues, Introduction to phases of compilation, Front end and Back end model of compiler, Cross compiler, Incremental compiler, Boot strapping. **[6 Hrs]**

Unit II : Lexical Analysis: Concept of Lexical Analysis, Regular Expressions, Tokens, Lexemes, and Patterns, Block Diagram of Lexical analyser, Hand coding of Lexical analyser, Introduction to LEX Tool and LEX file specification, Error detection and recovery in LEX. **[6 Hrs]**

Unit III : Syntax Analysis: Context Free Grammars(CFG), Concept of parsing, Parsing Techniques, Top-Down Parsers : Introduction, Predictive Parsing - Removal of left recursion, Removal of left factoring, Recursive Descent Parsing, Predictive LL(k) Parsing Using Tables, Bottom Up parsing: Introduction, Shift-Reduce Parsing Using the ACTION/GOTO Tables, Table Construction, SLR(1), LR(1), and LALR(1) Grammars, Introduction to YACC Tool & YACC file specification, Error detection and recovery in YACC. **[8 Hrs]**

Unit IV : Semantic Analysis & Intermediate Representation: Need of semantic analysis, Abstract Parse trees, Syntax directed definitions, Syntax directed translation schemes, Symbol Tables, Symbol Table management, Type Checking, Intermediate code generation: Intermediate languages, Design issues, Intermediate code representations: three address code, abstract & concrete syntax trees,. **[8 Hrs]**

Unit V : Code Optimization: Introduction, Principal sources of optimization, Machine Independent Optimization, Machine dependent Optimization, Various Optimizations: Function preserving transformation, Common Sub-expressions, Copy propagation, Dead-code elimination, Loop Optimizations, Code Motion, Induction variables and strength reduction, Peephole Optimization, Redundant –instruction elimination, Prescient store optimizations; Inter-procedural, whole-program or link-time optimization. **[6 Hrs]**

Unit VI : Run-Time Memory Management & Code generation: Model of a program in execution, Stack and static allocation, Activation records , Issues in the design of code

generation, Target machine description, Basic blocks & flow graphs, Expression Trees, Unified algorithms for instruction selection and code generation. **[6 Hrs]**

Text Books:

- Alfred V. Aho, Monica S. Lam, A. V. R. Sethi and J.D. Ullman, "Compiler Principle, Techniques and Tools" Addison Wesley, Second Edition, ISBN: 978-0321486813.

Reference Books:

- Barrent W. A., J. D. Couch, "Compiler Construction: Theory and Practice", Computer Science series, Asian student edition, ISBN: 9780574217653.
- Dhamdhere D.M., "Compiler Construction Principle and Practice", Macmillan India, New Delhi, 2003, Second Edition, ISBN: 0333904060.
- Ravendra Singh, Vivek Sharma, Manish Varshney, "Design and Implementation of Compiler", New Age Publications, 2008, ISBN: 978-81-224-2398-3.
- Holub, A.J., "Compiler design in C", Prentice Hall, 1994, ISBN: 978-0133049572.
- John Levine, Tony Mason & Doug Brown, "Lex and Yacc", O'Reilly, 1995, Second Edition, ISBN: 978-1565920002.

(CT) Compiler Construction Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

End Sem Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Code lexical analyzers for different languages using hand coding technique and Lex tool
2. Write a parser and semantic analyzer for different Context-Free Grammars using Yacc tool.
3. Implement different representations of Intermediate code
4. Demonstrate ability to optimize intermediate code using different techniques

List of Assignments:

1. Design a lexical analyzer for a subset of C language using Lex tool.
2. Design a hand-coded lexical analyzer for a subset of C language, draw the transition diagrams and then implement the lexical analyzer in C language.
3. Design a scientific calculator using Lex & Yacc or PLY or ANTLR tools.
4. Write a code for finding FIRST & FOLLOW of a grammar.
5. Design a SQL parser / html parser.
6. Implement a SLR parser for a given grammar.
7. Implement a static semantics analyzer.
8. Implement an intermediate code generator in three-address code form represented in quadruples.
9. Implement different optimization techniques on intermediate code.

Minor Course Sem VII: (CT) Database Management Systems

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Construct Entity-Relationship Model for given applications and Relational Model for the same.
2. Design and write SQL queries for given problems
3. Normalization to database design
4. Describe, compare, and solve analytical problems based on storage mechanisms, and transactions

Unit I : Introduction: Basic Concepts: Database System Application, Purpose of Database Systems, View of data, Data base languages, Database Architecture: Components of DBMS and overall structure of DBMS [4 Hrs]

Unit II : E-R and Relational Model: Database Design and the E-R Model: Modeling, entity, attributes, relationships, constraints, Components of E-R Model. Relational Model: Basic concepts. Attributes and domains, concept of integrity and referential constraints, schema diagram.. [6 Hrs]

Unit III : Relational Algebra and SQL: Fundamental relational algebra operations, Additional relational algebra operations, Extended relational algebra operations ,null values, Modification to database, SQL: Basic structure and operations, Aggregate functions ,Nested subqueries , Complex queries, Views. [8 Hrs]

Unit IV : Relational Database Design: Basic concept of normalization, Decomposition using Functional dependencies. [6 Hrs]

Unit V : Indexing and Hashing : Basic of query processing, Basic concepts, Indices, B+ trees and B tree index file, static and Dynamic hashing [6 Hrs]

Unit VI : Transactions and Concurrency control: Transaction: Basic concepts, States, concurrent execution, serializability, Recoverability, Isolation. Concurrency control: timestamps and locking protocols, validation Based protocols, Multiple granularity protocols, Deadlock handling, Recovery [8 Hrs]

Text Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database system concepts", 5th Edition, McGraw Hill International Edition.
- Raghu Ramkrishnan, Johannes Gehrke, "Database Management Systems", Second Edition, McGraw Hill International Editions

Reference Books:

- Rob Coronel, "Database systems : Design implementation and management", 4th Edition, Thomson Learning Press.
- Ramez Elmasri and Shamkant B. Navathe, "Fundamental Database Systems", 3rd Edition, Pearson Education, 2003

Minor Course Sem VIII: (CT) Internet Technologies

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Describe, draw diagrams, solve analytical problems based on, structure of computer networks.
2. Describe, solve analytical problems based on, networking protocols
3. Create WWW pages to serve as front-end for Internet applications
4. Demonstrate the ability to write server side and client side programs.

Unit I: Introduction: Introduction to Internet, Evolution of Internet, Internet applications, Protocols and Standard, Important Internet Services: WWW, File Transfer, Email, DNS, Remote Access: Telnet, SSH, Search Engines, Browser Basics **[6 hrs]**

Unit II: Computer Network – Overview: Introduction to computer network, OSI Model: Layered Architecture, Functions of the layers, Peer-to-Peer Processes, Encapsulation, TCP/IP protocol suite: Addressing, Packet forwarding, Introduction to UDP. **[8 hrs]**

Unit III: Web Essentials: Clients, Servers, Communication, HTTP Request Message, HTTP Response Message, Web Clients, Generations of web applications. **[6 hrs]**

Unit IV : Markup languages : An Introduction to HTML, Fundamental HTML Elements head, body etc. Document publishing, Introduction to XML. Introduction to Cascading Style Sheets, CSS features, CSS syntax, Style properties of text, box, layout, list, table, cursor etc **[6 hrs]**

Unit V : Client-Side Programming : Introduction to JavaScript, Basic Syntax, Variables and Data Types, Statements, Operators, literals, functions. Javascript Objects – properties, references, methods, constructors. Arrays, other built-in objects. Debugging javascript. Browsers. **[6 hrs]**

Unit VI: Server-Side Programming : PHP - Client Request – form data, request headers. Server Response - HTTP Status Codes, HTTP Response Headers. Sessions, Cookies, URL Rewriting. Introduction to Web services. **[8 hrs]**

Text Books:

- Jeffrey C.Jackson, "Web Technologies : A Computer Science Perspective", Pearson Education, 2nd edition, 2007
- B. A. Forouzan and Firouz Mosharraf, Computer Networks, A Top-Down Approach, Tata McGraw-Hill, 2012

Reference Books:

- A S Tanenbaum, "Computer Networks", 4th Edition, Pearson Education, ISBN 9788177581652
- Marty Hall, Larry Brown, "Core Web Programming", Pearson Education, 2nd Edition, 2001.
- Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 4th Edition, 2007.
- H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web: How To Program", Pearson Education, 3rd Edition, 2006.

Honors Course Sem VII: Advanced Computer Networks

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to

6. Understand issues in the design of network processors and design network systems
7. Analyze different possible solutions for communications at each network layer
8. Simulate working of wired and wireless networks to understand networking concepts
9. Develop solutions by applying knowledge of mathematics, probability, and statistics to network design problems
10. Understand and Compare various storage and networking technologies

Unit 1: Internetworking: Routing Algorithms, Congestion Control, Quality of Service, Queue Management, High Speed Networks, Performance Modeling and Estimation

Unit 2: IPv6: IPv4 deficiencies, patching work done with IPv4, IPv6 addressing, multicast, Anycast, ICMPv6, Neighbour discovery, Routing

Unit 3: Software Defined Networking and OpenFlow: Centralized and Distributed Control and Data Planes, SDN Controllers, Data Center Concepts, Network Function Virtualization, Mininet, Programming SDNs, Openflow Switch, Wire Protocol, Openstack Neutron plug-in

Unit 4: Ad Hoc Wireless Networks : MAC Protocols for Ad Hoc Wireless Networks, Routing Protocols for Ad Hoc Wireless Networks, Multicast routing in Ad Hoc Wireless Networks, Transport Layer and Security Protocols for Ad Hoc Wireless Networks, Quality of Service in Ad Hoc Wireless Networks.

Unit 5: Network management Protocols: SNMPv1 Network Management: Organization and Information Models, SNMPv2: major changes, SNMPv3, RMON, Network Management Tools, Systems, and Engineering, Network Management Applications

Unit 6: Storage and Networking: Storage and Networking Concepts, Fiber Channel Internals, Fiber Channel SAN Topologies, Fiber Channel Products, IP SAN Technology, IP SAN Products, Management of SANs, SAN Issues

Text Books:

- Thomas D Nadeau and Ken Grey, Software Defined Networking, O'Reilly, 2013
- Pete Loshin IPv6, Theory, Protocols and Practice, Morgan Kaufmann, 2nd Edition, 2004
- Mani Subramanian, Timothy A. Gonsalves, N. Usha Rani; Network Management: Principles and Practice; Pearson Education India, 2010

References

- William Stallings, High-Speed Networks and Internets, Pearson Education, 2nd Edition, 2002.
- C. Siva Ram Murthy, B.S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall, 2004
- Muthukumaran B, Introduction to High Performance Networks, Tata Mc Graw Hill, 2008
- Tom Clark, Designing Storage Area Networks, A Practical Reference for Implementing Fibre Channel and IP SANs, Addison-Wesley Professional, 2nd Edition, 2003.

Honors Course Sem VIII: (CT) Multicore Technology

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Understand the working principles of multicore architectures.
2. Optimize performance of multicore systems.
3. Specify the necessity of GPU.
4. Comprehend and differentiate between CPU and GPU.
5. Identify and demonstrate the need of domain specific architectures.

Unit I :Introduction to Multicore Systems: Fundamentals, The Era of Multicore Machines, Unicores vs Multicores, Understanding Performance, Shared Memory Multicore Systems, Distributed Memory Multicore Systems, Hybrid Systems, Symmetric and Asymmetric Multicore Systems – Overview of Multithreading – Multithreading in different forms – Homogeneous and Heterogeneous Multicore systems – Examples of different Multicore Systems. **[5 Hrs]**

Unit II :Cache Memory: Large Cache Design: Shared vs. Private Caches, Centralized vs. Distributed, Shared Caches – Coherence: Snooping, based cache coherence protocol, directory, based cache coherence protocol, Uniform Cache Access, Non-Uniform Cache Access, S, NUCA, D, NUCA, Inclusion, Exclusion – Examples of different Cache Organization, Consistency Models – Case Study. **[7 Hrs]**

Unit III :Performance and Optimizations for Multicore Systems: Select the right “core”, Improve serial performance, Achieve proper load balancing, Improve data locality, Reduce or eliminate false sharing, Use of affinity scheduling, Lock granularity and frequency, Remove synchronization barriers, Minimize communication latencies, Use of thread pools, Managing thread count, Use of parallel libraries. **[4 Hrs]**

Unit IV :Programming Multicore Systems: Programming models for Multicore Systems – Shared Memory Programming using pthreads, Shared Memory Programming using OpenMP – Use of OpenMP compiler directives – #pragma with different clauses – Understanding parallelized loops – Synchronization Constructs towards dependencies – Function parallel program, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, Performance. **[8 Hrs]**

Unit V :Special Case Graphics Processing Unit: CPU architecture, GPU hardware – CPU and GPU: Design Goals – Compute levels – Case Study: Nvidia GPU – GPGPU, Compute Unified Device Architecture (CUDA) Programming model – Applications of CUDA, Threads, Blocks, Grids – Memory management – Examples – Alternatives to CUDA. **[7 Hrs]**

Unit VI :Domain Specific, Architectures: Guidelines for domain specific architectures – Deep Learning Architecture, Google’s Tensor Processing Unit (TPU) for Deep Neural Networks (DNNs), Pixel Visual Core, a Personal Mobile Device Image Processing Unit. **[5 Hrs]**

List of Assignments:

1. Program for matrix vector multiplication using pthreads.
2. Program to implement data sharing and thread scheduling.
3. Program for matrix vector multiplication using OpenMP clauses.
4. Program to implement sorting using OpenMP clauses.
5. Program to find n^{th} term of Fibonacci sequence to implement function/ task parallel aspect using OpenMP clauses.
6. Setting up CUDA environment on Linux and first CUDA program.
7. Program to implement Matrix-Matrix Multiplication using CUDA.
8. Program to implement sorting using CUDA.
9. Program to Histogram calculation using CUDA.
10. Program to create threads using default stream in CUDA.
11. CUDA for Deep Learning - A Case Study.

Text Books:

- Gerassimos Barlas, “Multicore and GPU Programming: An Integrated Approach”, Morgan Kaufmann, 2015, ISBN: 978-0-12-417137-4.
- Rob Oshana, “Multicore Application Development Techniques: Applications, Tips and Tricks”, Elsevier, 2016, ISBN: 978-0-12-800958-1.
- John L Hennessy, David A Patterson, “Computer Architecture: A Quantitative Approach”, Sixth Edition, Morgan Kaufmann, 2018, ISBN: 978-0-12-811905-1.

Reference Books:

- Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar, “Multi-Core Cache Hierarchies”, Morgan & Claypool Publishers, 2011, ISBN: 9781598297546.
- Daniel J. Sorin, Mark D. Hill, David A. Wood “A Primer on Memory Consistency and Cache Coherence”, Morgan & Claypool Publishers, 2011, ISBN: 9781608455652.
- Shane Cook, “CUDA Programming: A Developer’s Guide to Parallel Computing with GPUs”, Morgan Kaufmann, 2013, ISBN: 978-0-12-415933-4.
- Barbara Chapman, Gabriele Jost, Ruud van der Pas, “Using OpenMP Portable Shared Memory Parallel Programming”, The MIT Press, 2008, ISBN-13: 978-0-262-53302-7.
- David B. Kirk and Wen-mei W. Hwu, “Programming Massively Parallel Processors”, Second Edition, Morgan Kaufmann, 2013, ISBN: 978-0-12-415992-1.

Syllabi
of Elective Subjects
B. Tech.
Computer Engineering
and
B.Tech.
Information Technology
Sem VII and Sem VIII

Various possible tracks of the elective courses
A guideline for the students

| | Department Elective-I | Department Elective-II | Department Elective-III | Department Elective-IV | Department Elective-V |
|--------------------------------------|---|--|---|--|---|
| Tracks | Sem – 6 | Sem – 7 | Sem – 7 | Sem – 8 | Sem – 8 |
| System Software/ Hardware | AM: Advanced Microprocessors | CBD: Cloud and Big Data LP: Language Processors (only for IT) | AUP: Advanced UNIX Programming ES: Embedded Systems | SA: System Administration S&V: Storage and Virtualization | MT: Multicore Technology DS: Distributed Systems (only for CE) |
| Database Technologies | WST: Web Systems and Technologies | CBD: Cloud and Big Data | IR: Information Retrieval | CB: Computational Biology | ADBMS: Advanced Database Management Systems |
| Networking and Security | WST: Web Systems and Technologies | IOT: Internet of Things | AUP: Advanced UNIX Programming | MAN: Mobile and Ad-hoc Networks CFLP: Computer Forensics and Data Recovery SA: System Administration | CS: Cyber Security DS: Distributed Systems (only for CE) |
| Algorithms and Programming | ADS: Advanced Data Structures G&M: Graphics and Multimedia Fundamentals of Digital Signal Processing | SDP: Software Design Patterns FP: Functional Programming | OOMD: Object Oriented Modeling and Design | Advanced Algorithms CB: Computational Biology | ML: Machine Learning |
| Data Science | ADS: Advanced Data Structures | CBD: Cloud and Big Data | IR: Information Retrieval NLP: Natural Language Processing | Advanced Algorithms | ML: Machine Learning |

Department Elective-II(CT) Cloud Computing and Big Data

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Understand various basic concepts related to cloud computing technologies.
2. Identify various cloud-based solutions to meet a set of given requirements.
3. Acquire flavor of micro-services and visualise developing of apps using docker and kubernetes.
4. Understand basic concepts of Hadoop programming models for big data analysis on cloud.

Unit I:Introduction: Cloud computing architecture, cloud delivery models, cloud deployment models, Benefits and challenges, cloud Quality of Service (QoS) and Service Level Agreements (SLAs), overview of virtualization, challenges in X86, virtualization advantages, hypervisor / Virtual Machine Monitor, Hypervisors types, Full virtualization, Para virtualization, Hardware assist virtualization ESXi, Xen, KVM.

[6 Hrs]

Unit II:Cloud Orchestration: Elements of cloud orchestration, backing up VM, restoring VM, clustering and high availability, VM migration, OpenStack: NOVA, Neutron, Keystone Cinder, Swift and Glances, Vmware Suit, Apache CloudStack

[6 Hrs]

Unit III:Platform as a Service: Window Azure, types of app services, Azure compute and storage services, Google CloudPlatform benefits, GCE, AWS, Docker flow, orchestration with Docker, dynamic and legacy linking of containers, architecture of Devops, introduction to configuration and management tools Ansible, Puppet and Chef

[6 Hrs]

Unit IV:Micro services and Kubernetes: Micro services architecture, benefits of micro services, characteristics of micro services, migration to micro services, Kubernetes features, containerization with kubernetes, architecture of Kubernetes cluster, fundamentals of kubernetes

[6 Hrs]

Unit V:Software Defined Networking: Software Defined Networking, history and evolution of SDN, SDN controllers, control and data plane separation OpenFlow, OpenvSwitch, Mininet

[6 Hrs]

Unit VI :Big Data: Big Data for cloud, Characteristics of Big data, Key aspects of a Big Data Platform, Hadoop stack, Spark Ecosystem Walkthrough, HDFS framework, HDFS File Operation Lifecycle, Fundamentals of HBase, HBase Data-Model, Fundamental of Apache Hive, Hive to query Hadoop files, Pig Data Flow Engine, A case study: the GrepTheWeb application

[10Hrs]

Text Books:

- Danielle Ruest and Nelson Ruest, "Virtualization, A beginners Guide", Tata McGraw Hill, ISBN: 978-661-2031-892
- Dinakar Sitaram and Geetha Manjunath, "Moving to the cloud", Elsevier, ISBN: 978-159-7497-251
- Adrian Mouat, "Using Docker: Developing and Deploying Software with Containers", O'reilly Media, ISBN: 978-149-1915-769.
- Tom White, "Hadoop: The Definitive Guide", O'reilly Media, ISBN: 978-144-9311-520.

Reference Books:

- L. Wang, R. Ranjan, J. Chen, and B. Benatallah, "Cloud Computing: Methodology, Systems, and Applications", CRC Press, Boca Raton, FL,USA, ISBN: 978-143-9856-413
- Buyya R., Broberg J., Goscinski A., "Cloud Computing: Principles and Paradigms", John Wiley & Sons Inc., ISBN: 978-0-470-88799-8
- V.K. Pachghare, "Cloud Computing", PHI Publication, ISBN: 978-81-203-5213-1
- James Turnbull, "The Docker Book: Containerization is the new virtualization", ISBN: 978-098-8820-203
- Bernard Marr, "Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results", Wiley, ISBN: 978-111-9231-387
- <http://docs.openstack.org/>
- <http://mininet.org/>
- <https://www.opennetworking.org/>
- <http://pubs.vmware.com/vsphere-50/index.jsp>
- <https://developers.google.com/appengine/>
- <http://www.windowsazure.com/en-us/>
- <http://www.cloudfoundry.com/>
- <http://aws.amazon.com/developers/getting-started/>

Department Elective-II (CT) Cloud Computing and Big Data Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Mini Project: 20 marks

End Semester Exam: 30 Marks

Course Outcomes:

Students will be able to:

1. Practice orchestration tools to aid in both developing and deploying applications.
2. Create application by utilizing cloud platforms such as Google app Engine and Amazon Web Services (AWS).
3. Evaluate techniques and methods for extracting, transformation and loading big data.

List of Assignments:

1. To create and run virtual machines on open source OS [VirtualBox, VMWare Workstation].
 1. To install an operating system in the virtual machine from template
 2. Add storage to create the new virtual disk.
2. To install OpenStack and use it as Infrastructure as a Service.
 1. Create and delete compute resources.
 2. Attach volumes to running instances
 3. Create a network and subnet for the web server nodes.
3. To install hypervisor such as KVM, ESXi.
 1. Deploy VM on hypervisor
 2. Back up or migrate VM.
4. To create AWS EC2 Instances.
 1. Logging into the AWS portal
 2. To attach and detach an EBS volume to an EC2 instance.
 3. To create an S3 Bucket for object storage to EC2 instance.
5. To configure and start VM from azure portal.
 1. Logging into the Microsoft Azure portal
 2. Connect to the Windows virtual machine using Remote Desktop Protocol (RDP) software
6. To install docker on window/linux.
 1. To build docker image from docker hub.
7. Deploy a stateless/stateful application on kubernetes cluster.
8. To install Puppet using the pe.conf file and set up an agent node using the slipstream installation.
 1. Edit /etc/hosts 2. Set SELinux to permissive. 3. Install the Puppet master
 4. Configure your agent node.
9. Perform setting up and Installing Hadoop in its three operating modes:

1. Standalone Mode
 2. Pseudo Distributed Mode
 3. Write Program using Hadoop Spark for Word Count Example, Weather Data Set Example.
 4. Write a Pig Latin script to handle the Weather Data Set problem.
[sort, group, join, project, and filter Weather data]
-
10. To Create Hive tables and Querying Hive tables for Weather Data Set problem.
 1. Create, alter, and drop databases, tables, views, functions, and indexes.
 11. To Create HBase tables that include multiple column families and columns
 1. Alter properties associated with a table
 2. Populate a table with data
 3. Retrieve data from a table
 12. To create and query HBase tables using BigSQL
 1. Populate these tables using Big SQL INSERT and LOAD operations
 2. Query these tables using projections and restrictions.

This is an illustrative list of assignments. The instructor is expected to update the list.

Department Elective-II (CT) Functional Programming

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes -40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Compare various programming paradigms
2. Apply the basics of the lambda calculus and combinators in the implementation of functional languages. Solve a problem using functional programming paradigm
3. Evaluate functions and analyse memory management
4. Apply functional programming in current technology.

Unit I: Introduction: necessity and comparison with other programming paradigms, First class and higher order functions , Combinatory logic, Intuitionistic logic, Lambda Calculus. **[8 hrs]**

Unit II: Typed Systems: Typed languages, statically typed languages, dynamically typed languages, data types and state of data, type casting, type inference. **[6 hrs]**

Unit III: Functions: Inbuilt functions, user defined functions, high order functions on list, recursive function, lazy function evaluation, polymorphism, File I/O, Channels and Concurrency, Pattern matching **[8 hrs]**

Unit IV: Automatic memory management: heaps, limitations of stacks, heap management , garbage collection algorithms. **[6 hrs]**

Unit V: A Little Language of Expressions, Little Quilt, data types, expressions, functions, lexical scope, type checking. **[6 hrs]**

Unit VI: Application of Functional Programming: Hadoop, MapReduce, Google, Facebook , APIs in other languages like PHP, Java, Ruby. **[6 hrs]**

Text Books:

- Ravi Sethi, "Programming Languages: Concepts and Constructs", Pearson Education; 2 edition, ISBN 978-8177584226.
- T W Pratt ,Marvin V. Zelkowitz , Pearson; 4th edition, ISBN 978-0130276780

Reference Books:

- Graham Hutton, “Programming in Haskell”, Cambridge University Press, ISBN, 978-0521692694.
- “Performance and Evaluation of Lisp Systems”, The MIT Press; ISBN 0-262-07093-6.

Department Elective-II (CT) Functional Programming Laboratory

Teaching Scheme:

Practical : 2 Hrs/week

Examination Scheme:

Term Work - 50 marks

End Sem Exam - 50 marks

Course Outcomes:

Students should be able to:

1. Write functional programs for various data structures
2. Write functional programs to use lists and its operations
3. Write functional programs to use concurrency
4. Write functional programs to use file I/O
5. Write type and typeclasses in functional programming languages.

List of Assignments:

1. Implement a program in functional language to demonstrate various data structures
2. Implement a program in functional languages to create, read elements traverse elements and delete elements from a list.
3. Write a function in functional programming languages.
4. Write a program which uses in-built functions in functional programming language.
5. Write a program to read and write file.
6. Write a program to demonstrate concurrency in functional programming.
7. Write a program to demonstrate polymorphism in functional programming.
8. Write a program to demonstrate classes in functional programming.

Some more assignments can be added by the instructor.

Department Elective-II (CT) Information Retrieval

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the basic pre-processing required for Information Retrieval
2. Analyze different approaches for query processing
3. Write a search module for college website and demonstrate effective information retrieval.

Unit I : Introduction: Motivation, Data retrieval versus information retrieval, Goals and history of IR, Problems in IR, Modeling- taxonomy of IR models, Retrieval- adhoc and filtering, Basic Models of IR- Boolean, Vector-space and probabilistic retrieval models. **[6 Hrs]**

Unit II : Basic preprocessing and Indexing: Simple tokenizing, Stop-word removal, Stemming, Inverted index and postings, Text-similarity metrics, TF, IDF, TF-IDF (term frequency/inverse document frequency) weighting, k-Shingles, Distance and similarity measures- Minkowski, Manhattan, Euclidean, Jaccard, Pearson, Cosine similarity measure. **[6 Hrs]**

Unit III : Query Operations: Relevance feedback- Global methods and local methods, Global methods- Query expansion/reformulation with a thesaurus or WordNet, Query expansion via automatic thesaurus generation, Techniques like spelling correction, Local methods-Relevance feedback, Pseudo relevance feedback, Indirect relevance feedback, Rocchio algorithm for relevance feedback, Probabilistic relevance feedback, Kullback-Leiber divergence (KL-divergence) retrieval function. **[8 Hrs]**

Unit IV : Indexing and Searching: Introduction, Inverted Files - Construction, Searching, Suffix trees and suffix arrays, Signature files, Boolean queries, Sequential searching- Brute Force, Knuth Morris-Pratt, Shift-or, Suffix Automation, Pattern matching- Regular expression and extended patterns, Pattern matching using indices, Structural queries, Compression- sequential searching, compressed indexing. **[7 Hrs]**

Unit V : Probabilistic Retrieval and Evaluation: Formal model for enhancing retrieval effectiveness by using sample information about the frequency of occurrence and co-occurrence of index terms in the relevant and non-relevant documents, Traditional view of measurement of effectiveness, Performance metrics: recall, precision, and F-measure, Evaluations on benchmark text collections. **[7 Hrs]**

Unit VI : Parallel and distributed IR: Parallel Indexing, Query processing in distributed IR, Spidering, Metacrawlers, Directed spidering, Link analysis (e.g. hubs and authorities, Google PageRank), Shopping agents. **[6 Hrs]**

Text Books:

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, "Introduction to Information Retrieval", Online Edition, Cambridge University Press, ISBN: 978-0521865715.
- Richardo Baeza –Yates, Berthier Ribeiro-Neto "Modern Information Retrieval " Addison – Wesley, Second Edition, ISBN: 978-0321416919.
- C J Van Rijsbergen, "Information Retrieval", An online book by C J Van Rijsbergen, University of Glasgow, ISBN: 978-0408709293.

Reference Books:

- Kowalski, Gerald J, "Information Retrieval Systems Theory and Implementations", Springer, ISBN: 978-0-585-32090-8
- David A. Grossman, Ophir Frieder, "Information Retrieval Algorithms and Heuristics", Springer, ISBN: 978-81-8128-917-9
- Jin Zhang, "Visualization for Information Retrieval", Springer, ISBN-13: 978-3540751472.

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Build inverted index file
2. Implement web based crawler
3. Implement query processing module of a search engine
4. Write a search engine and demonstrate effective information retrieval for a website

List of Assignments:

1. Design and Implement pre-processing module required for building an inverted index file.
2. Building an Inverted Index File for the data on your machine.
3. Installation and study of elastic Search Engine.
4. Installation and study of Lucene and Solr.
5. Design and implement different distance and similarity measures (Euclidean, Jaccard, Manhattan, Cosine) required for finding duplicates or near-duplicates in the datasets.
6. Design and implement a web-based crawler.
7. Design and Implement the query processing module of an search engine.
8. Design Search Engine using Solr and Lucene for your college website.

Department Elective-II (CT) Software Design Patterns

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Identify and implement appropriate solutions to recurring programming problems by consulting technical documentation and specifications, including design pattern catalogues and existing source code.
2. Develop design solutions using creational patterns.
3. Apply structural patterns to solve design problems.
4. Construct design solutions by using behavioural patterns.

Unit I :Basics of design pattern:definition, catalogue of design patterns, organizing the catalogue, solving design problems, selecting a design patterns, and use of design patterns.

[8Hrs]

Unit II :A case Study: Designing a document editor: design problems, document structure, user interface, supporting multiple look-and-feel standards, supporting multiple window system, user operations, spelling checking and hyphenation.

[8Hrs]

Unit III :Creational patterns:object creational, abstract factory, builder, factory method, prototype, singleton.

[6 Hrs]

Unit IV :Structural patterns:object structural, adapter, bridge, composite, decorator, façade, flyweight, proxy.

[6 Hrs]

Unit V :Behavioural patterns:Object Behavioural, chain of responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, and visitor.

[6 Hrs]

Unit VI :Patterns and software architecture:architectural pattern, difference between design patterns and architectural patterns, different architectural patterns , relation between software architecture and patterns.

[6 Hrs]

Text Books:

- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object Oriented Software, ISBN-10: 0-201-63361-2, ISBN-13: 078-5342633610, Pearson 1st edition.
- Allan Shalloway, James Trott, Design Patterns Explained- A New Perspective on Object Oriented Design, Addison Wesley 2nd Edition, ISBN-10: 0321247140, ISBN-13: 978-0321247148

- F. Buschmann et al. Pattern-Oriented Software Architecture, Wiley Volume 1 1996. ISBN-10: 8126516119 , ISBN-13: 978-8126516117

Reference Books:

- Eric Freeman and Elisabeth Freeman, Head First Design Patterns, O'Reilly 1st Edition, ISBN-10: 8173664668, ISBN-13: 978-8173664663
- Stephen Stelting and Olav Maassen , Applied Java Patterns, Prentice Hall 1st Edition, ISBN-10: 0130935387, ISBN-13: 978-0130935380

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Assignments: 50 Marks

End Sem Exam: 50 Marks

Course/Lab Outcomes:

Students will be able to:

1. Apply various patterns to solve real-world problem scenarios and learn sensible trade-offs wherever necessary.
2. Compare the abstraction in patterns with real world system implementations.

List of Assignments:

Students shall submit at-least eight template patterns drawn in (any) modeling language:

1. Model use case scenarios using creational patterns (minimum 2)
2. Draw different types of structural patterns (minimum3)
3. Design pattern templates for behavioral patterns (minimum 2)
4. Create architectural patterns for various applications (minimum 1)
5. Implement abstract factory pattern for at least one case scenario.
6. Draw adapter design pattern for multimedia applications.

The instructor shall update the above practical implementations as per availability and the area of interest.

Department Elective-II(CT) Internet of Things

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the application areas of IOT
2. Analyze data and knowledge management and use of devices in IoT technology
3. Analyze gateways and data management in IoT.
4. Analyze real world IoT design constraints, Industrial automation and commercial building automation in IoT.

Unit I: Introduction and concepts: Origins, Drivers, Applications, Physical and logical design of IoT, IoT enabling technologies: Wireless sensor networks, Cloud computing, Big data analytics, Communication protocols, embedded systems, IoT levels and deployment templates **[8 Hrs]**

Unit II: IoT and M2M: Introduction to M2M, Difference between IoT and M2M, M2M and IoT technology fundamentals: Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a service (XaaS), M2M and IoT analytics, Knowledge management **[7 Hrs]**

Unit III: IoT Architecture-state of the art: Architecture reference model: Introduction, Reference model and architecture, IoT Reference model, M2M to IoT-an architectural overview: Building architecture, Main design principles and needed capabilities, State of the art, Standards considerations **[7 Hrs]**

Unit IV: Data management in IoT: Managing M2M data, Data collection and analysis (DCA), Big data, Semantic sensor networks and semantic annotation of data, Virtual sensors, Complex event processing **[6 Hrs]**

Unit V: Security, Privacy & Trust: IoT security challenge, Spectrum of security considerations, Unique security challenges of IoT devices, Internet of things privacy background, Unique privacy aspects of internet of things, Trust for IoT **[6 Hrs]**

Unit VI: Case studies Illustrating IoT design: Home automation: Smart lighting, Home intrusion detection, Cities: Smart parking, Environment: Whether monitoring system, Weather reporting bot, Air pollution monitoring, Forest fire detection, Agriculture: Smart irrigation, Productivity applications: IoT printer **[6 Hrs]**

Text Books:

- Vijay Madiseti, Arshdeep Bahga, “Internet of Things: A Hands-On Approach”, Universities Press (India) Private Limited, 2016, ISBN: 978 81 7371 954 7

- Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis Karnouskos, Stefan Avesand & David Boyle "From Machine-to-Machine to the Internet of Things", Academic Press, Elsevier, 2014, ISBN: 978-0-12-407684-6

Reference Books:

- Karen Rose, Scott Eldridge, Lyman Chapin, "The Internet of Things: An Overview", Internet Society, 2015
- Adrian McEwen, Hakim Cassimally, "Designing the Internet of Things", Wiley, 2014, ISBN 978-1-118-43062-0
- Daniel Kellmereit, "The Silent Intelligence: The Internet of Things", 2013, ISBN 0989973700

Department Elective-II (CT) Internet of Things Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Demonstrate to program on embedded systems
2. Demonstrate how to communicate with other mobile devices using various communication platforms
3. Analyze estimation of the performance of the developed scalable cloud storage and execution platform for IoT on the real sensors data

IOT Kit contains following items:

1. Arduino Uno Rev 3 (Made in Italy) with USB Cable
2. ESP8266 (ESP01) Serial WiFi Module
3. 1-Channel Relay Module
4. Breadboard (Regular)
5. Assorted Jumper Wires (20)
6. LM35 Temperature Sensor
7. Assorted LEDs (10)

List of Assignments:

1. Embedded Programming

- 1.1. Toggling LEDs
- 1.2. Transmitting a string through UART
- 1.3. Controlling LEDs blinking pattern through UART
- 1.4. Echo each character typed on serial terminal.
- 1.5. Digital IO configuration.
- 1.6. Timer based LED Toggle.
- 1.7. On-chip Temperature measurement through ADC.

2. RF experiments

- 2.1. Point-to-Point communication of two Motes over the radio frequency.
- 2.2. Multi-point to single point communication of Motes over the radio frequency.

3. Experiments on interfacing with UbiSense

- 3.1. I2C protocol study
- 3.2. Reading Temperature and Relative Humidity value from the sensor.
- 3.3. Reading Light intensity value from light sensor.
- 3.4. Reading of atmospheric pressure value from pressure sensor.
- 3.5. Proximity detection with IR LED.
- 3.6. Generation of alarm through Buzzer.

3.7. Transmitting the measured physical value from the UbiSense over the Air.

4. Experiments on interfacing with Ubi-DAQ

- 4.1. Timestamp with RTC
- 4.2. IO Expander
- 4.3. Relay control.
- 4.4. I2C based 12-channel ADC
- 4.5. EEPROM read and write

5. WSN Applications

- 5.1. Demonstration of a Peer-to-Peer network topology using Coordinator and end device network device types.
- 5.2. Demonstration of Peer-to-Peer communication between Coordinator and end device through Router.
- 5.3. Establishing Many-to-One Communication (Star Network Topology)
- 5.4. Establishing Tree Network Topology
- 5.5. Establishing Cluster Tree Network

6. IOT applications

- 6.1. Porting 6LoWPAN stack on Ubimote for enabling it with IPv6
- 6.2. 6LoWPAN network formation with motes and PC
- 6.3. IP based lighting control through Data Acquisition Card
- 6.4. IP based sensor monitoring through Ubi-Sense

Each section any two assignments should be implemented by using IoT research lab kit as per Lab instructor.

Department Elective-III (CT) Advanced Unix Programming

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Correlate, describe and critically appreciate design of Unix commands, system calls and Operating System internal data structures
2. Write code snippets using system calls to emulate Unix commands, short system utilities and applications
3. Predict the output for a given code snippet consisting of various system calls

Unit I:Introduction: Role of OS, CPU Protection Rings, POSIX APIs, System calls, System Call Handler, Parameter Passing, Kernel Wrapper Routines, System architecture, Files and directories, User identification, Programs and processes, Signals, POSIX, SVR4, ISOC limits, POSIX limits; sysconf, pathconf, fpathconf. **[6 Hrs]**

Unit II:File Management:File data structures, File Control Operations, File Status Flags, Duplicating File Descriptors, Nonblocking I/O, File Types, set-user-id, set-group-id, Sticky bit, File attributes, File Ownerships, Access Permissions,Process File Mode Creation Mask, File Size, File truncation, System data files: passwd, shadow, etc. File Systems, hard/Symbolic Links, Directory Files, File Times, Special Device Files, FIFO files, **System calls:** open, creat, close, lseek, read, sync, dup, fcntl, ioctl, stat, access, umask, chown, link, symlink, utime, getcwd.**[8 Hrs]**

Unit III:Process Control: Context of a process, process states and Transitions, Environment of a UNIX Process, Process Termination, Command-Line Arguments, Environment List, Memory Layout, Performing a Nonlocal Goto, Processes Identifiers, Process Creation, Termination, Monitoring Child Processes, File Sharing between Parent and Child, Zombie process, Race Conditions, Process Execution, Process Credentials, Set-User-ID and Set-Group-ID Programs, Saved Set-User-ID and Saved Set-Group-ID, Retrieving and Modifying Process Credentials, Process Time, Interpreter Files, Executing a Shell Command, system and security, Processes Accounting. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, Controlling Process,Foreground and Background Process Groups, Job Control, Shell Execution of Programs, Orphaned Process Groups, System calls: setjmp, longjmp, getrlimit, setrlimit,fork, vfork, exit, wait, exec, system, tcgetpgrp. **[12 Hrs]**

Unit IV: Daemon Processes: Introduction, Daemon Characteristics, Coding Rules.**Threads:** Concepts, Identification, Creation, Termination, Synchronization, Limits, Reentrancy, Thread specific data, Thread I/O. **[4 Hrs]**

Unit V: Signals: Signal Types and Default Actions, Defining Signal Dispositions, Sending Signal, SIGCLD Semantics, Re-entrant functions, Displaying Signal Descriptions, alarm, Waiting for a Signal, Signal Set, Signal Mask, Pending Signals, Waiting for a Signal Using a Mask, Handling Signal with unconditional Jump, Suspending Process Execution for a Fixed Interval, Implementation of abort, system function handling signals, System calls and functions: signal, kill, raise, alarm, sigprocmask, sigpending, sigaction, sigsetjp, abort, sigsuspend, SIGCLD, sleep; **Interprocess Communication Techniques:** *Pipe, popen, pclose*, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores, Shared Memory, POSIX IPCs, Sockets. **[10 Hrs]**

Unit VI: Terminal I/O, Psuedo Terminals: Special input characters, Terminal attributes, Baud rate, Line control, Canonical mode, Window Size, Pseudo terminal devices, streamsbased pseudo terminals, `pty_fork`, `pty`, `stty`, `termcap`, `terminfo`, `curses`. **[4 Hrs]**

Text Books:

- E W. Richard Stevens, Stephen A Rago, Advanced Programming in the UNIX Environment, Addison-Wesley, 3rd Edition, 2013, ISBN-13: 978-0-321-63773-4.
- Michael Kerrisk, The Linux Programming interface, A Linux and UNIX® System Programming Handbook, 2010, ISBN-10: 1-59327-220-0.

Reference Books:

- Maurice J. Bach, The Design of Unix Operating System, PHI, 2009, ISBN-13: 978-0132017992 .
- Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd Edition, O'Reilly, 2006, ISBN-10: 8184040830 Terrence Chan, UNIX System Programming Using C++, Prentice Hall India, 1999, ISBN-10: 0133315622.
- Kay A Robbins and Steve Robbins, Unix Systems Programming, Pearson Education, 2004, ISBN-10: 0130424110.
- Marc J. Rochkind, Advanced UNIX Programming, 2nd Edition, Pearson Education, 2005, ISBN-10: 0131411543.

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Mini Project: 20 marks

End Semester Exam: 30 Marks

Course Outcomes:

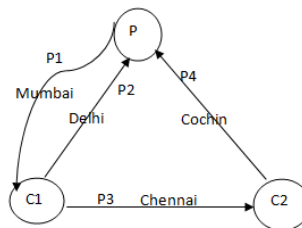
Students will be able to:

1. Develop utilities using system calls, execute and test in POSIX environments

List of Assignments:

1. Using the system calls for regular files
 - a. Emulate the command: **cat >>file1**. Test the program by executing three times by giving different inputs to be stored in the file.
 - b. Write a program to create a file with a hole: write any 10 bytes at an offset of 10 and another 10 bytes at an offset of 30. Using "system" function, invoke "od" command and view the contents. Later copy the contents of the file to another file without writing the bytes of 0. Once again verify the contents by invoking "system" with "od".
2. Using system calls on file attributes, symbolic links
 - a. Create a file with permissions R, W, X for User, R, W for Group and R for Others. Display the permissions as a string. Later Remove the W from group and R from Others and set the Setuid bit on. Login as a different user and check the success of Read of access and open function.
 - b. Create a list of symbolic links like $e \rightarrow f$, $d \rightarrow e$, $c \rightarrow d$, $b \rightarrow c$, $a \rightarrow b$. Input a symbolic link file name (say a). Display all the links until it is not possible to follow.
3. Using system calls on directory and file properties
 - a. Does calling stat function change any of the time values? Verify with a program.
 - b. Input a directory name as command line argument. Check its validity and then display all files in that directory as well as the files in all subsequent subdirectories.
4. Exercises on using system calls on exec and session
 - a. This program works like a shell. Display a prompt and the user enters a command at the prompt. The command should be read from the standard input and will be executed. Do not use system API. The program will be quit when logout is entered at the prompt.
 - b. Call fork. Let the child create a new session. Verify that the child becomes the process group leader and it does not have a controlling terminal

- c. Write a program to verify that when a process in an orphaned process group attempts to read() from the controlling terminal, the read() fails.
5. Exercises on using system calls on signal
 - a. Catch the SIGTERM signal, ignore SIGINT and accept the default action for SIGSEGV. Later let the program be suspended until it is interrupted by a signal.
 - b. Create a child process. Let the parent sleeps of 5 seconds and exits. Can the child send SIGINT to its parent if exists and kill it? Verify with a sample program.
 - c. Child inherit parent's signal mask when it is created, but pending signals for the parent process re not passed on". Write appropriate program and test with suitable inputs to verify this.
 6. Exercises on using system calls on IPC – pipe
 - a. A pipe setup is given below that involves three processes. *P* is the parent process, and C1 and C2 are child processes, spawned from *P*. The pipes are named *p1*, *p2*, *p3*, and *p4*. Write a program that establishes the necessary pipe connections, setups, and carries out the reading/writing of the text in the indicated directions.



7. Terminal Handling
 - a. Write a program that calls `tty_raw` and terminates (without resetting the terminal mode). Then use the `reset(1)` command to restore the terminal mode.
 - b. Write a program that calls `pty_fork` and have the child exec another program that you must write. The new program that the child execs must catch SIGTERM and SIGWINCH. When it catches a signal, the program should print that it did; for the latter, it should also print the terminal's window size. (This task can be extended further to have parent send signals to PTY slave).

This is an illustrative list of assignments. The instructor is expected to update the list.

Department Elective-III (CT) Embedded Systems

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Understand functioning of general purpose computing system and embedded system while amplifying its architecture and features.
2. Evaluate ARM processor and system architecture along with its features and peripherals based on requirements of embedded system architecture.
3. Analyze the need for designing embedded operating system and process of embedded system development.

Unit I:Embedded Systems Overview: Introduction, Definition, Characteristics & Salient Features, Classification, Overview of Embedded System Architecture & Recent Trends, Application Areas
[6 Hrs]

Unit II:Embedded System Architecture: Hardware Architecture: Basic building blocks, Devices and Communication Interfaces, Software Architecture: Embedded Operating System, Applications Software & Communication Software (Firmware)
[8 Hrs]

Unit III:ARMCortex-M3/M4 processor: Introduction ARM architecture and Cortex : M series, Introduction to the Tiva family viz. TM4C123x and its targeted applications, Tiva block diagram, address space, on-chip peripherals, Simplified view: block diagram, programming model: Registers, Operation modes, Exceptions and Interrupts, Reset Sequence
[8 Hrs]

Unit IV:ARM Cortex M4 Peripherals: I/O pin multiplexing, GPIO control, Watchdog Timer, System Clocks and control, Hibernation Module on TM4C, Introduction to Interrupts, Interrupt vector table, interrupt programming, Timers, ADCs, UART and other serial interfaces: USB, PWM, RTC, DMA, QEI on TM4C microcontroller
[7 Hrs]

Unit V:Embedded Operating System: General Purpose Operating Systems vs Embedded Operating Systems, Overview of Embedded Operating Systems, detailed study of Real Time Operating System(RTOS) such as TI RTOS, Task Scheduling, Task Synchronization, Inter- task communication concepts on Embedded TI RTOS platform
[7 Hrs]

Unit VI:Embedded System Development: Development Process, Requirement Engineering, Design, State Transition Diagram,Implementation, Integration and Testing, Cross Platform Development, Process of generating Single Executable Image of Firmware, Boot Sequence, Hardware and Software Development Tools. Case Studies of Typical Embedded Systems : Digital Camera and Washing Machine
[8 Hrs]

Text Books:

- K.V.K. Prasad, Embedded / Real Time Systems: Concepts, Design and Programming Black Book, Dreamtech Press, 2005 publications ISBN 10: 8177224611 / 13: 9788177224610
- Joseph Yiu's The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors – Third Edition, published by Elsevier ISBN: 9789351071754, 9351071758
- Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers, 2014, Jonathan W Valvano Createspace publications ISBN: 978-1463590154.
- Embedded Systems: Introduction to ARM Cortex - M Microcontrollers, 5th edition Jonathan W Valvano, Createspace publications ISBN-13: 978-1477508992
- Vahid F. and Givargies T., Embedded Systems Design, John Wiley X. Sons, 2002, Third edition, ISBN:978-81-265-0837-2

Reference Books:

- John B Peatman, Design with PIC Microcontrollers, Pearson Education, 1998, ISBN:978-81-7758-551-3
- Liu, Real-Time Systems, Pearson Education, 2000. 5. Technical Manuals of ARM Processor Family available at ARM Website on Net
- Doug Abbott, Linux for Embedded and Real-time Applications, Newnes, Fourth Edition, ISBN: 978-0128112779
- Raj Kamal: Embedded Systems – Architecture: Programming and Design: McGraw Hill Education, Second Edition, ISBN:978-0070667648

Teaching Scheme:

Laboratory : 2 hrs per week

Examination Scheme:

Continuous evaluation: 50 Marks

End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Write and demonstrate a working solution for a given problem
2. Think of all possible inputs to an application and handle all possible errors properly
3. Develop a system using the concepts of RISC architecture and ARM processors
4. Write software programs for ARM processor to develop a suitable application

List of Assignments:

List of Experiments based on Cortex M4 Tiva Microcontroller:

- A. Basic Interfacing experiments using Tiva
 1. Blink an LED.
 2. Blink an LED using delay generated by Timer.
 3. Generating PWM using the in-built PWM module.
 4. Configuring PWM and ADC to control speed of DC motor
 5. Echo back the data using the UART module
- B. Experiments using Embedded TI RTOS:
 1. Blink an LED using Swi
 2. Task Scheduling using RTOS on a TM4C microcontroller
 3. Task Switching using Embedded RTOS
 4. Inter-Task Communication using mailboxes

This is an illustrative list of assignments. The instructor is expected to update the list.

Department Elective-III(CT) Object Oriented Modeling and Design

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Specify, analyze and design the use case driven requirements for a particular system.
2. Identify, formulate and solve software development problems: software requirements, specification (problem space), software design, and implementation (solution space).
3. Model the event driven state of object and transform them into implementation specific layouts.
4. Analyze and explore different analysis and design models, such as OO Models, Structured Analysis and Design Models, etc.
5. Apply knowledge software engineering methods, such as object-oriented analysis and design methods with a clear emphasis on UML.

Unit I : Introduction to object diagrams, Class diagrams, Classes and relationships, Interfaces and ports, Templates, Active objects, Advanced relationships generalization, Inheritance, Association, aggregation, and Dependencies. **[10 Hrs]**

Unit II : Introduction to UML, UML history, UML new features, Rational unified process emphasizing inception, Elaboration, Construction, Transition phases, Architectural approaches, Use case centric, Architecture driven, Iterative approach, UML meta model, Extensibility mechanisms like stereotypes, Tagged values, Constraints and Profiles, OCL, Overview of all diagrams in UML. **[10 Hrs]**

Unit III : Composite structure diagrams including composite structures, Collaborations, Interaction diagrams, Interaction overview diagrams including interactions, Signals, Exceptions, Regions, partitions, Sequence diagrams, Communication diagrams. **[8 Hrs]**

Unit IV : State machine diagrams, States, Encapsulation of states, Transitions, Submachine, State generalization, Timing diagrams, Activity diagrams, Activities, Sub activities, Signals, Exceptions, Partitions, and Regions. **[10 Hrs]**

Unit V : Support for modelling, Architecture in UML, Package diagrams, Component diagrams, Deployment diagrams, Applications of UML in embedded systems, Web applications, Commercial applications, UML 2.0 for each diagram the Need, Purpose, Concepts, Notation, Forward Engineering, Reverse Engineering and Application. **[8 Hrs]**

Unit VI : Concepts of distributed operating system, COM and CORBA, Introduction to Object Oriented Database. **[4 Hrs]**

Text Books:

- Grady Booch, James Rumbaugh, Ivar Jacobson “Unified Modeling Language User Guide”, The (2nd Edition) (Addison-Wesley Object Technology Series) (Hardcover), 2013.

- Michael Bilaha, James R Rambaugh: Object Oriented Modelling and Design, PHI, 2007, 2edition, ISBN-10: 8131711064

Reference Books:

- Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third Edition (Paperback) , Addison Wesley, 2011
- Dan Pilone, Neil Pitman "UML 2.0 in a Nutshell", (In a Nutshell (O'Reilly)) Paperback
- Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado "UML 2 Toolkit (Paperback) "
- Jim Arlow, Ila Neustadt "UML 2 and the Unified Process : Practical Object-Oriented Analysis and Design" (2nd Edition) (Addison-Wesley Object Technology Series) (Paperback)

Department Elective-III (CT) Object Oriented Modeling and Design Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Assignments: 50 Mark

End Term Exam: 50 Marks

Course/Lab Outcomes:

Students will be able to:

1. Create the OO design of a system from the requirements model in terms of a high-level architecture description, and low-level models of structural organization and dynamic behavior using UML class, object, and sequence diagrams.
2. Transform Use Cases into Object Oriented software Realizations through OO Analysis and OO Design.
3. Apply techniques of state machines and design patterns to your designs.

List of Assignments:

1. Elicit requirements towards the identified project statement.
2. Capture Functional Requirements with Use Cases for the project / problem statement
3. Design Use Case Scenarios and use case diagrams and documentation
4. Perform Static Analysis Modeling and Dynamic Analysis Modeling
5. Perform Static Design Modeling and Dynamic Design Modeling
6. Apply different Design Patterns, Object Constraint Language, Persistence for your problem statement.

Students shall submit minimum five UML based implementations on: three models to describe a system from different viewpoints.

The instructor shall update the above practical implementations as per availability and the area of interest.

Department Elective-III(CT) Natural Language Processing

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Continuous evaluation – 100 marks

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of basic text processing techniques in NLP.
2. Analyze morphological analyzers and stemmers.
3. Build language models and demonstrate Word Sense Disambiguation using wordnet.
4. Design, implement and evaluate part-of-speech taggers and parsers.

Unit I : Introduction: Introduction, Motivation, Word tokenization, Word normalization, Word level morphology- **morphological analysis and synthesis**, Stemming - Porters algorithm, Levenshtein distance measure. **[6 Hrs]**

Unit II: POS Tagging: Sequence labeling tasks of NLP, POS tagging, POS tag sets, Hidden Markov Model, Viterbi algorithm, Baum Welch Algorithm. **[6 Hrs]**

Unit III: Language Modeling: Introduction to N-gram, Probability estimation for n-gram, Evaluation and perplexity, Smoothing techniques, Named-Entity recognition. **[6 Hrs]**

Unit IV: Parsers: Constituency and dependency parsers, Constituency parser -Syntactic structure, Parsing methodology, Different parsing algorithms, Parsing in case of ambiguity, Probabilistic parsing, CKY algorithm, Issues in parsing, Dependency parsing- Syntactic structure, Parsing methodology, Transition-Based Dependency Parsing, Graph-Based dependency parsing, Evaluation, Co-reference resolution, Named-entity recognition. **[10 Hrs]**

Unit V: WordNet: Word Senses, Word relations, Word similarity and thesaurus methods, Word sense disambiguation, Knowledge base and supervised WSD, WordNet , Unsupervised based WSD. **[6 Hrs]**

Unit VI: Applications of NLP: Question/Answering system, Text summarization, Sentiment Analysis, Information extraction **[6 Hrs]**

Text Books:

- Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Second Edition, Prentice Hall, 2008, ISBN: 978-0131873216.
- Allen James, "Natural Language Understanding", Second Edition, Benjamin/Cumming, 1994, ISBN: 978-0805303346.
- Chris Manning and Hinrich Schuetze, "Foundations of Statistical Natural Language Processing", MIT Press, ISBN: 978-0262133609.

Reference Books:

- Journals: Computational Linguistics, Natural Language Engineering, Machine Learning, Machine Translation, Artificial Intelligence.
- Conferences: Annual Meeting of the Association of Computational Linguistics (ACL), Computational Linguistics (COLING), European ACL (EACL), Empirical Methods in NLP (EMNLP), Annual Meeting of the Special Interest Group in Information Retrieval (SIGIR), Human Language Technology (HLT).

Teaching Scheme:

Laboratory :2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of basic text processing techniques in NLP.
2. Analyse morphological analyzers and stemmers.
3. Build language models and demonstrate Word Sense Disambiguation using wordnet.
4. Design, implement and evaluate part-of-speech taggers and parsers.

List of Assignments:

1. Study, analyze and demonstrate the relative performance of different state of art POS taggers using graphs.
2. Design n-gram language model for English language.
3. Implement Viterbi Algorithm for Part of Speech tagging for English Language.
4. Study, analyze and demonstrate the relative performance of state of art parsers using graphs of precision recall F-measure.
5. Perform the Name-entity tagging on English language text for person name, organization name, Location name, Date, Time, email id etc.
6. Design and implement a co-reference resolution model.
7. Design and implement a dependency parser.
8. Study of knowledge base WordNet and apply it for word sense disambiguation.

The instructor shall update the above practical implementations as per availability and the area of interest.

Department Elective-III(CT)Language Processors

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Continuous Evaluation – 100 marks

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of program execution process.
2. Apply understanding of working and error handling of different phases of compilation process to design compiler for any language.
3. Implement language translators in high-level programming languages like c, c++, java
4. Implement different phases of compiler using tools like Lex, Yacc and hand-coding in any programming language.

Unit I : Language Processors: Introduction, Language Processing activities, Fundamentals of Language Processing, Fundamentals of Language specification, Language Processing development tools Data Structures for Language Processing. **[6 Hrs]**

Unit II : Macros, Linkers & Loaders:Macros: Macro Processor Algorithm and Data Structures, General Macro processing concepts; Loaders: Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features - Relocation, Program Linking Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage. **[7 Hrs]**

Unit III : Compiler - Lexical Analysis: Translator process and issues, Compilation process, Front-end and back-end model, Concept of Lexical Analysis, Regular Expressions, Tokens, Lexemes, and Patterns, Block Diagram of Lexical analyzer, Hand coding and automatic generation of lexical analyser. **[6 Hrs]**

Unit IV : Syntax Analysis: Role of parsers and issues of separating lexical and syntax Analysis, Context Free Grammars(CFG), Parsing techniques, Automatic generation of parsers, Error detection and recovery. **[7 Hrs]**

Unit V : Static Semantic Analysis and Intermediate Code Generation: Need of semantic analysis, Syntax directed translation schemes for declaration processing, type analysis, scope analysis and intermediate code generation. Intermediate code generation for declaration, assignment, iterative statements, case statements, arrays, structures, conditional statements, Boolean expressions, procedure/function definition and call. **[7 Hrs]**

Unit VI : Code Optimization and Code Generation: Principal sources of optimization, Machine Independent optimization, Machine dependent optimization, Various optimizations techniques, Model of a program in execution, Stack and static allocation, Activation records, Issues in the design of code generation, Target machine description, Basic blocks & flow graphs, Expression trees. **[7 Hrs]**

Text Books:

- D. M. Dhamdhere: "Systems programming and operating system", Tata McGraw Hill, 2001, Second Edition, ISBN: 9780074635797.
- Alfred V. Aho, Monica S. Lam, A. V. R. Sethi and J.D. Ullman, "Compiler Principle, Techniques and Tools" Addison Wesley, Second Edition, ISBN: 978-0321486813.

Reference Books:

- Andrew W. Appel "Modern Compiler Implementation in C", Cambridge University Press 1998, ISBN: 978-0521607650.
- John R Levine, "Linkers and Loaders", Morgan Kaufmann, Revised Edition, ISBN: 978-1558604964
- John Levine, Tony Mason & Doug Brown, "Lex and Yacc", O'Reilly, 1995, Second Edition, ISBN: 978-1565920002.

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

End Semester Exam: 30 Marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of program execution process.
2. Apply understanding of working and error handling of different phases of compilation process to design compiler for any language.
3. Implement language translators in high-level programming languages like c, c++, java
4. Implement different phases of compiler using tools like Lex, Yacc and hand-coding in any programming language.

List of Assignments:

1. Design simple macro processor. The processor should handle macros with and without parameters
2. Design a nested macro processor [Call to a macro processor inside macro definition].
3. Study different tools viz. LEX and YACC or PLY or ANTLR and design a calculator (text or graphics) or a document editor any of the tools.
4. Assignment on compiler
 - a) Design a Lexical and Syntax Analyzer for a subset of C/C++ Language using LEX and Yacc.
 - b) Implement an intermediate code generator (Triple, Quad) for a subset of C/C++
 - c) Implement any two optimization techniques on Intermediate Code Generated:
 - I. Constant expression evaluation.
 - II. Local copy propagation.
 - III. Common sub expression elimination.
 - IV. Loop invariant code movement.

The instructor shall update the above practical implementations as per availability and the area of interest.

Department Elective-IV(CT) System Administration

Teaching Scheme:

Lectures : 2 Hrs/week

Tutorial: 1 Hr/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

Carry out following tasks, with special emphasis on GNU/Linux based systems:

1. Install various Linux distributions and Windows based servers and desktop systems on commodity hardware, and carry out basic administration tasks of user, network, process, storage management.
2. Setup a LAN based laboratory using DHCP.
3. Install and configure a web server, database server, DNS, NFS, NIS, LDAP based system, secure a Desktop and a server system completely using existing tools.

Unit I:Basic System Administration: Partitioning, Installation of multiple operating systems on desktops, Various operating system services: Cron, CPU utilization, User management, Backup, Log management, Boot loader, Process management, File system namespace, Kernel upgrades,Shell scripting. **[6Hrs]**

Unit II:Network Administration:Configuration of network hardware, Linux router,Managing routes, DHCP - Dynamic Host Configuration Protocol, DNS-Domain Name Server, NFS - Network File System, NIS - Network Information Service, Email Setup-SendMail,Issues with mail services, POP and IMAP Server. **[6Hrs]**

Unit III:Filesystem Administration:Formatting, Partitioning, Managing file systems, Defragmentation, Quotas, Journaling file system, Logical volume management, Disk layouts, File system check,Archiving and compressing files,SAN, NAS, Case Studies: ext2, ext4,ntfs, samba, cifs, lvm, fat32.**[6Hrs]**

Unit IV: Security Administration: Methods of attack, Network security tools- Nmap, Snort, Nessus, Wireshark/tcpdump, Firewall, IPtables, NAT, IP filtering, Setting up linux firewall, Mandatory access control (MAC), SELinux, Cryptographic security tools Authentication mechanisms, LDAP, Proxy servers. **[5Hrs]**

Unit V:Server and Devices Administration:Apache web server configuration, Database servers : MySQL and PostgreSQL, The sysfs file system, Device resources,udev: Device files, Installing and configuring printers, Managing printers via the web interface, Scanners, PCI devices, LAN cards, Device troubleshooting, Plug and Play devices. **[5Hrs]**

List of Assignments:

- Set a desktop with following software configuration options: triple boot with Windows, Ubuntu and Fedora operating systems; Grub timeout set to 5 seconds with default Fedora Linux; Each step of boot process secured with passwords; Following software installed in each OS: office, internet browser, c compiler, terminal, ssh server, telnet server, web server, database server.
- Set up a LAN based computer laboratory with 5 computers - using DHCP first and then using static IPs; one of the machines should work as DHCP server; set up LDAP+NFS based authentication for managing single identity on all computers; setup quotas on NFS systems.
- Demonstrate use of three-tier architecture using LAMP and WAMP suite using same code base.
- Setup a disk management system using LVM such that all options of LVM are demonstrated.
- Write a program to browse an ext4 file system and locate data of a deleted file.
- Setup a proxy server and firewall with set of policies to block video content in the network.
- Setup apache web server to serve 3 websites, with 3 domain names at a time from a single machine; demonstrate the use of at least 5 configuration options of apache.
- Setup an email server and demonstrate use of at least 3 email clients to send email using it.
- Setup SE Linux.

The instructor is expected to update the list.

Text Books:

- Evi Nemeth, Garth Snyder, Ben Whaley, Trent R. Hein, "UNIX and Linux System Administration Handbook", Pearson Education, Fourth edition, ISBN-13: 978-8131761779
- Richard Petersen, "Linux: The Complete Reference, Sixth Edition", Mcgraw-Hill Education Sixth edition, ISBN-13: 978-0070222946
- Arnold Robbins, Nelson H. F. Beebe, "Classic Shell Scripting", Shroff/O'Reilly First edition, ISBN-13: 978-8173668463
- Wale Soyinka, "Linux Administration: A Beginner's Guide", McGraw-Hill Osborne Media Publication Sixth Edition, ISBN-13: 978-0071767583
- Olaf Kirch & Terry Dawson, "Linux Network Administrator's Guide", O' Reilly 2nd Edition June 2000, ISBN-10: 1565924002, ISBN-13: 978-1565924000
- W.Preston, "Using SANs and NAS", O' Reilly; First Edition, February 2002, ISBN-10: 0596001533, ISBN-13: 978-0596001537

Reference Books:

- Richard Blum, Christine Bresnahan, "Linux Command Line and Shell Scripting Bible", Wiley India Pvt. Ltd., Second edition, ISBN-13: 978-8126533831;
- Jean-Paul Tremblay, Paul. G. Soresan, "An introduction to data structures with Applications", Tata Mc-Graw Hill International Editions, 2nd edition 1984, ISBN-0-07-462471-7

Department Elective-IV (CT) Storage & Virtualisation

Teaching Scheme:

Lectures : 2 Hrs/week

Tutorial: 1 Hr/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Apply, compare and contrast basic concepts of hardware storage, NAS-SAN, virtualization concepts.
2. Analyze cloud computing setup with its vulnerabilities and applications using different architectures.
3. Assess cloud Storage systems and cloud security, the risks involved, its impact and develop suitable cloud application.
4. Design and implement Virtualization environment and apply cloud resource management.

Unit I : Basics of Systems and Storage Storage Challenges, Issues- Data sources, challenges of data growth and data availability, Performance and manageability requirements, Data virtualization, OS and device drivers - Kernel, firmware, RDMA, boot sector, device partitioning, UNIX file system, virtual memory, namespace, metadata, buffer cache, defragmentation. **[4 Hrs]**

Unit II : Storage Hardware and SCSI Protocol Storage hardware and SCSI Protocol, SAN, NAS, Fibre Channel and iSCSI, Storage Hardware Building Blocks - Device types, HBA, switches, hubs, routers, GBIC, Introduction to various Storage protocols, Overview of IDE, SAS, SATA, SCSI, FC, FCoE, iSCSI, Infiniband, FCP, FC-IP, iFCP, Fibre Channel Protocol Stack & Concepts- FC ,Mapping Protocols - iSCSI, FCP- SCSI mapping to underlying transport, Connection Management, PDU, TOE. **[4 Hrs]**

Unit III :SAN & NAS: SAN concepts, DAS, SAN architecture, concepts of zoning, name server, SCN, WWN, routing, FC-SAN, IP-SAN and Applications, NAS concepts, NAS architecture, Protocols (CIFS, NFS), Performance, Scalability and Usability, Appliances. **[4 Hrs]**

Unit IV- Storage Virtualization Concepts: Data Centre end-to-end view, Overview of stack, clustering, virtual Machines, cloud storage and virtualization, RAID levels, I/O stack, OS abstraction, Storage Pooling, Storage Provisioning, Online Grow/Shrink Storage Virtualization, Metadata management, Transaction consistency, I/O maps, I/O path considerations, Data consistency, Crash recovery, Application interfaces, Linux container. **[5 Hrs]**

Unit V- Applications and Use Cases for Storage Virtualization: Storage Virtualization use cases, applications, data replication, RPO/RTO, Snapshots, Data protection , Restore, Archival, Compliance considerations. Capacity Management , Storage provisioning, De-duplication, Thin provisioning, Storage Tier, ILM, Data classification, Storage grid. **[5 Hrs]**

Unit VI- Cloud Computing: Virtualized environments features, taxonomy of virtualization techniques, virtualization, Cloud Computing basics, Cloud architectures, Services and applications, Cloud Computing mechanisms, MapReduce programming model, Benefits and challenges, Cloud resource management and scheduling, Cloud migration, Service management and SLAs, Security, and monitoring, Cloud Platforms in industry, Case study of AWS, Google Cloud, Microsoft Azure and IBM Softlayer. **[6 Hrs]**

List of Assignments:

1. Create synchronous remote replication functionality from host1 to host2. Implement the functionalities :
 - a. Data written on local disk should be synchronously copied to remote disk
 - b. A write from an application should be completed only when data has been written to both source and replicated node.
2. Writing a simple File System which provide functionality of "ls, mkdir, pwd" posix commands. Apart from that it should support print operation . Write an application programme to test these functionality.
3. Using the hard disk on one host create a snapshot for the same and store in a flat file on other m/c. Provide a mechanism to restore the data from flat file.
4. Write a "Examination data server" using MySQL open source database for storing students examination report, which is a snapshot of all present student's report of college.
The report consist of six field :Name, Id ,Branch, Grade{ in each semester} ,Final Grade, Backlog Courses . Write a client program to query on various parameters. Server should generate a report for same {e.g. The no of students having CGPA(final grade) 7.0 and above in all branches/particular branches., The No. of students not cleared in Mathematics yet , Name of topper in every branch etc..}. Use pthreads to support multiple client queries. Also implement write functionality {update in case of old students / addition (new students) / deletion(final semester student who has passed out } and use reader/writer lock.

This is an illustrative list of assignments. The instructor is expected to update the list.

Text Books:

- Storage Networks: The complete Reference. Robert Spalding TMH.
- Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs, Second Edition Publisher: Addison-Wesley Author: Tom Clark.

Reference Books:

- Cloud Computing: Concepts, Technology & Architecture, Thomas Erl, Prentice Hall, 2013.
- Cloud Computing: Principles and Paradigms, Wiley Series on Parallel and Distributed Computing, Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, John Wiley and Sons, 2011, ISBN 978-0-470-88799-8.
- Cloud Computing: Theory and Practice, Dan C Marinescu, Elsevier, 2013.
- Cloud Computing Bible, Barrie Sosinsky, Wiley Publishing, 2011.
- Virtualization Essentials , Matthew Portnoy, John Wiley and Sons, 2012.
- Computer Systems - A Programmer's Perspective, Randall Bryant and David O'Hallaron Pearson Education, 2003.
- The Design and Implementation of the 4.4 BSD Operating System, McKusick, Bostic, Karels, Quateman, 1996.

Department Elective-IV(CT) Mobile and Ad-Hoc Networks

Teaching Scheme:

Lectures : 2 Hrs/week

Tutorial: 1 Hr/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the current topics in MANETs and WSNs, both from an industry and research point of views.
2. Demonstrate the principles of mobile adhoc networks (MANETs) and what distinguishes them from infrastructure-based networks.
3. Analyze how proactive routing protocols function and their implications on data transmission delay and bandwidth consumption.

Unit I: Introduction:Fundamentals of wireless communication technology, Characteristics of the wireless channel, multiple access techniques, IEEE 802 networking standard, Wireless LANs and PANs, IEEE 802.11 standard, IEEE 802.16 Standard, Bluetooth, HomeRF **[6 Hrs]**

Unit II:Cellular Wireless Networks and Wireless Internet:Cellular concept, Cellular architecture, Generations of cellular systems, Wireless in local loop, Wireless ATM, Wireless internet, What is wireless internet, Mobile IP, TCP in wireless domain, WAP, Optimizing web over wireless. **[6 Hrs]**

Unit III :Introduction To Ad-Hoc Networks:Characteristics, Applications, Issues in Ad Hoc wireless networks medium access protocols: design issues, Goals and classification, Contention based protocols- with reservation, Scheduling algorithms, Protocols using directional antennas, HIPERLAN **[5Hrs]**

Unit IV:Routing Protocols: Design issues, Goals and classification. Proactive Vs reactive routing, Unicast routing algorithms, Multicast routing algorithms, hybrid routing algorithm, Energy aware routing algorithm, Hierarchical routing, QoS aware routing **[6 Hrs]**

Unit V:Wireless Sensor Networks:Architecture, Data dissemination, Data gathering, MAC protocols for sensor networks, Location discovery, Quality of a sensor network, Physical design of IoT, Logical design of IoT, IoT enabling technology**[5 Hrs]**

Text Books:

- C.Siva Ram Murthy and B.S.Manoj, Ad hoc Wireless Networks Architectures and protocols, 2nd edition, Pearson Education. 2007

Reference Books

- F.Zhao, L. Guibas, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, 2004

- Stefano Basagni, Marco Conti, Silvia Giordano and Ivan sSojmenovic, Mobile Ad-hoc Networking, Wiley-IEEE Press, 2004.
- Mohammad Ilyas, The Handbook of Ad-hoc Wireless Networks, CRC press, 2002.

List of Assignments for tutorials:

1. Introduction to network simulators used for wireless Ad-Hoc and sensor networks
2. Study various trace file formats of network simulators
3. Implement and compare various MAC layer protocols
4. Implement and compare AODV and DSR routing algorithms in MANET
5. Implement DSDV routing algorithms in MANET
6. Implement signal strength based link management routing protocols
7. Calculate and compare average throughput for various TCP variants
8. Implement and compare various routing protocols for wireless sensor networks
9. Study Ethereal / Wireshark software and analyze dump files

Department Elective-IV (CT) Computer Forensics and Data Recovery

Teaching Scheme:

Lectures: 2 Hrs/week

Tutorial: 1 Hr/Week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Explain various computer forensic techniques/phases, Intellectual Property Rights
2. Analyse different file systems used in different operating systems
3. Demonstrate the knowledge of forensic examination related to Microsoft Windows and Unix systems
4. Demonstrate the knowledge of forensic examination of social media, browsers, emails
5. Apply various tools during real world forensic investigation

Unit I:Introduction: Introduction to computer forensics, Evidence: assessment, acquisition, examination, Handling real world investigations: email accounts, social networking profiles, credit card fraud, source code and confidential information theft, piracy, pornography, virus attacks, etc., Evidence collection, analysis tools and case studies. **[4 Hrs]**

Unit II:Search, seizure and examination of digital evidence:Procedure for search & seizure of computers and digital devices, Detailed procedures for obtaining a forensic image of storage devices, Standards for preservation of digital data, Forensic examination of computer-based electronic and digital evidence: evidence extraction and analysis, Chain of custody. **[4 Hrs]**

Unit III:Data encryption and recovery: Encryption and Decryption methods for data capturing, Recovering deleted evidences, Cyber crime investigation involving the computer, internet and networks: E-mail, websites, chatrooms, etc., Password breaking, Keyloggers, IP tracing, etc. **[4 Hrs]**

Unit IV:Network and web forensics: Introduction, Investigating logs, Network traffic monitoring and web attacks, Mobile device forensics, Email investigation, Browser forensics, Social media forensics (eg. facebook, linkedIn), Case studies and tools. **[8 Hrs]**

Unit V:File system forensic analysis: Basic of file system, Forensic analysis of FAT, NTFS, Ext, UFS file systems, Steganography and image file forensics, Vehicular tracking (GPS) devices forensic examination practices. **[6 Hrs]**

Unit VI:Windows and Unix forensic, IT Act, IPR: Fundamentals of forensics for Microsoft Windows, UNIX derivatives: Event log, Registry, etc., IT Act.2000, 2008, Issues in cyberspace: IPR, Property, Trademarks and domain names, Domain name dispute resolution policy, Role of ICANN, Case studies. **[4 Hrs]**

Text Books:

- “Guide to Computer Forensics and Investigations”, Bill Nelson Amelia Phillips Christopher Steuart, 4th Edition, Course Technology, Cengage Learning, ISBN-13: 978-1-435-49883.
- “Computer Forensics: Computer Crime Scene Investigation”, John R. Vacca, 2nd Edition, Charles River Media Publication, ISBN-13: 978-1-58450-389-7.
- Brian Carrier, “File System Forensic Analysis”, Pearson education, 1st Edition, ISBN-13: 978-0321268174
- E. Casey, Handbook of Digital Forensics and Investigation, Academic Press, 1st Edition, 2010, ISBN-13: 978-0123742674

Reference Books:

- “Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet”, 3rd Edition, Eoghan Casey, ISBN: 978-0-120374268-1
- E. P. Dorothy, “Real Digital Forensics for Handheld Devices”, Auerback Publications, 1st Edition, 2013, ISBN-13: 978-1439898772.
- “Digital Forensics for Legal Professionals: Understanding Digital Evidence From The Warrant To The Courtroom”, Larry Daniel, Lars Daniel, Syngress, 1st Edition, ISBN: 978-1-59749-643-8

List of Assignments:

1. Perform case study of open source computer forensics tools.
2. Exercise on e-mail forensic analysis:
 - a. header analysis: identify values of different header fields like SPF, DKIM, DMARC and check whether an email is spam or not.
 - b. tracking: track different hops through which mail passed to receiver, time and location when receiver opened the mail and information about receiver’s device and browser.
 - c. recovery- deletes an e-mail from trash/bin of your email client and recovers it.
3. Exercise on browser forensic analysis:

Use SQLite DB Console client to execute queries on browser database files and identify following-

 - a. Top 10 websites accessed by user.
 - b. Last 10 websites visited.
 - c. Most commonly used search term/keyword.
 - d. List down the cookies stored by different websites.
 - e. List down the bookmark links saved by user.
 - f. List of saved passwords/ auto form fill details.

Use incognito/private mode of browsing and find the same information again.
4. Implement network forensics: capture live network traffic using any open source tool like Wireshark, examine packets of different protocols like TCP,UDP, HTTP,HTTPS etc. and document your findings.

5. Case Study on forensic analysis of Windows system: Registry, Event log.
6. Perform an experiment to acquire a forensically sound "image" of the USB device, check its validation/ integrity (by calculating MD5/SHA/CRC 32 etc.) and analyze the forensic image acquired. Before acquiring the image, write protect your device.
The USB drive should contain different types of files like text file, pdf file, audio file, image file, word document, excel document, powerpoint document, a small video etc. Once the image is obtained, analyze it using some free (preferably open source) forensic analysis tool and answer the following: how to access contents of files, what metadata information about files was available, can you see the deleted contents of the device as well, how to recover deleted files, was recovery partial or full, perform some keyword searching, how did you make the device write-protected.
Submit a report in .pdf format (include the screen shots also).
7. Exercise on forensic analysis of any one file system(FAT, NTFS, Ext, UFS etc.):
 - a. Take a hex dump of a partition/volume formatted via particular file system.
 - b. Identify the values of data structures of file system like partition table, allocate entries etc.
 - c. Examine unallocated space and find files stored there.

This is an illustrative list of assignments. The instructor is expected to update the list.

Department Elective-IV (CT) Computational Biology

Teaching Scheme:

Lectures : 2 Hrs/week

Tutorial : 1 Hrs/ week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the concept of information generation from protein sequences, DNA sequences, whole genome.
2. Extract information from large databases and to use this information in computer modeling.
3. Evaluate various elements of computational biology such as genomic networks, algorithms, and models
4. Design and develop current applications of computational biology.

Unit I:Introduction: MolecularBiology Introduction, Cell, Nucleus, Genes, DNA, RNA, Proteins, And Chemical structure of DNA, RNA, Transcription and Translation Process. Protein Structure and Functions, Nature of Chemical Bonds Molecular Biology tools, Polymerase chain reaction
[6 Hrs]

Unit II:Sequence Alignment: Simple alignments, Gaps, Scoring Matrices, Global and Local Alignments, Smith-Waterman Algorithm, Multiple sequence Alignments, Gene Prediction, Statistical Approaches to Gene Prediction
[4 Hrs]

Unit III:Genome Algorithms:Genome Rearrangements, Sorting by Reversals, Block Alignment and the Four-Russians Speedup, Constructing Alignments in Sub-quadratic Time, Protein Sequencing and Identification, the Peptide Sequencing Problem
[6 Hrs]

Unit IV:Microarray Data Analysis: Microarray technology for genome expression study, Image analysis for data extraction, Data analysis for pattern discovery, gene regulatory network analysis
[4 Hrs]

Unit V:Phylogenetic: Neighbor's relation method, Neighbor-joining method, Maximum likelihood Approaches, Multiple Sequence Methods Structural Biology, Sequence, organisms, 3D structures, complexes, Assemblies, Case Studies, examples
[6 Hrs]

Unit VI:Drug Discovery &Next Gen Sequencing:Similarities/differences between drugs and receptors, protein-ligand docking, Massively Parallel Signature Sequencing (MPSS), SOLiD sequencing, Single molecule real time (SMRT) sequencing
[4 Hrs]

Text Books:

- Dan E. Krane, Michael L. Raymer, "Fundamental Concepts of Bioinformatics," Pearson Education, Inc. Fourth Edition, 9780805346336.
- Harshawardhan P. Bal, "Bioinformatics Principles and Applications", Tata McGraw-Hill, seventh reprint, 9780195692303.

Reference Books:

- Teresa Attwood, David Parry-Smith, "Introduction to Bioinformatics", Pearson Education Series, 9788180301971
- R. Durbin, S. Eddy, A. Krogh, G. Mitchison., "Biological Sequence Analysis: Probabilistic Models of proteins and nucleic acids", Cambridge University Press, 9780521629713.

List of Assignments to be conducted during tutorials:

1. Study any biological database freely available on Internet; prepare a detailed report of it.
2. Implement string matching algorithm from available ones. Find time & space complexity.
3. Design and implement Smith-Waterman pairwise sequence alignment algorithm using any language (C, C++, Java, python, etc.) and test it for some arbitrary sequences of Protein.
4. Design and implement Needleman-Wunsch pairwise sequence alignment algorithm using any language (C, C++, Java, python, etc.) and test it for some arbitrary sequences of Protein.
5. Prepare detail report on Bioinformatics tools other than BLAST and FASTA Assignment
6. Solve computational problems for reconstructing the phylogeny: Character based and distance based.
7. Write Program to find the parsimony length of a given tree topology.
8. Find multiple sequence alignment using ClustalW2.
9. Perform data mining using Clementine on any arbitrary sequences.
10. Prepare detail report on Case study: A model consists of a rooted tree which models the evolution relationship (Cavender-Felsenstein model and Jukes-Cantor model).

This is an illustrative list of assignments. The instructor is expected to update the list.

Department Elective-V(CT) Multicore Technology

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Continuous Assignment/Quizes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Understand the working principles of multicore architectures.
2. Optimize performance of multicore systems.
3. Specify the necessity of GPU.
4. Comprehend and differentiate between CPU and GPU.
5. Identify and demonstrate the need of domain specific architectures.

Unit I :Introduction to Multicore Systems: Fundamentals, The Era of Multicore Machines, Unicores vs Multicores, Understanding Performance, Shared Memory Multicore Systems, Distributed Memory Multicore Systems, Hybrid Systems, Symmetric and Asymmetric Multicore Systems, Overview of Multithreading, Multithreading in different forms, Homogeneous and Heterogeneous Multicore systems, Examples of different Multicore Systems. **[5 Hrs]**

Unit II :Cache Memory: Large Cache Design: Shared vs. Private Caches, Centralized vs. Distributed, Shared Caches, Coherence: Snoopy cache coherence protocol, Directory cache coherence protocol, Uniform Cache Access, Non-Uniform Cache Access, S-NUCA, D-NUCA, Inclusion, Exclusion, Examples of different Cache Organization, Consistency Models, Case Study. **[7 Hrs]**

Unit III :Performance and Optimizations for Multicore Systems: Select the right core, Improve serial performance, Achieve proper load balancing, Improve data locality, Reduce or eliminate false sharing, Use of affinity scheduling, Lock granularity and frequency, Remove synchronization barriers, Minimize communication latencies, Use of thread pools, Managing thread count, Use of parallel libraries. **[5 Hrs]**

Unit IV :Programming Multicore Systems: Programming models for Multicore Systems, Shared Memory Programming using pthreads, Shared Memory Programming using OpenMP, Use of OpenMP compiler directives, #pragma with different clauses, Understanding parallelized loops, Synchronization Constructs towards dependencies, Function parallel program, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, Performance. **[9 Hrs]**

Unit V :Special Case – Graphics Processing Unit: CPU architecture, GPU hardware, Design Goals of CPU and GPU, Compute levels, Case Study of Nvidia GPU, GPGPU, Compute Unified Device Architecture (CUDA) Programming model, Applications of CUDA, Threads, Blocks, Grids, Memory management, Examples, Alternatives to CUDA. **[9 Hrs]**

Unit VI :Domain Specific-Architectures: Guidelines for domain specific architectures, Example Domain: Deep Neural Networks, Deep Learning Architecture, Google's Tensor Processing Unit (TPU) for Deep Neural Networks (DNNs), Pixel Visual Core, a Personal Mobile Device Image Processing Unit. **[5 Hrs]**

Text Books:

- Gerassimos Barlas, "Multicore and GPU Programming: An Integrated Approach", Morgan Kaufmann, 2015, ISBN: 978-0-12-417137-4.
- Rob Oshana, "Multicore Application Development Techniques: Applications, Tips and Tricks", Elsevier, 2016, ISBN: 978-0-12-800958-1.
- John L Hennessy, David A Patterson, "Computer Architecture: A Quantitative Approach", Sixth Edition, Morgan Kaufmann, 2018, ISBN: 978-0-12-811905-1.

Reference Books:

- Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar, "Multi-Core Cache Hierarchies", Morgan & Claypool Publishers, 2011, ISBN: 9781598297546.
- Daniel J. Sorin, Mark D. Hill, David A. Wood "A Primer on Memory Consistency and Cache Coherence", Morgan & Claypool Publishers, 2011, ISBN: 9781608455652.
- Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufmann, 2013, ISBN: 978-0-12-415933-4.

Department Elective-V Laboratory (CT) Multicore Technology Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Assignments/Mini Project: 20 marks

End Semester Exam: 30 Marks

List of Assignments:

1. Program to implement matrix vector multiplication using pthreads.
 - a. Fork-Join mechanism.
 - b. Need to explicitly write code for synchronizing threads and barrier in matrix vector multiplication.
2. Program to implement data sharing and thread scheduling.
 - a. Thread scheduling and process/ thread affinity and its impact on performance.
3. Program to execute matrix vector multiplication using OpenMP clauses.
 - a. Structure of OpenMP #pragma style clauses.
 - b. Different basic clauses to parallelize loop based matrix vector multiplication program.
4. Program to implement sorting using OpenMP clauses.
 - a. Sorting algorithm implementation using shared and private clauses.
 - b. Execution time comparison with normal sorting implementation.
5. Program to find n^{th} term of Fibonacci sequence.
 - a. Function/ task parallel aspect using OpenMP clauses.
6. Setting up CUDA environment and first CUDA program.
 - a. Different steps to set CUDA environment on Windows using Microsoft Visual Studio.
 - b. Different steps to set CUDA environment on Linux with gcc.
7. Program to implement Matrix-Matrix Multiplication using CUDA.
 - a. Basic CUDA host, device and memory constructs understanding.
 - b. Thread- warp, block, grid usage for Matrix-Matrix Multiplication.
8. Program to implement sorting using CUDA.
 - a. Sorting algorithm implementation with different device/ thread size and its impact on execution time.
 - b. Comparison with OpenMP and normal implementation.
9. Program to implement Histogram calculation using CUDA.
 - a. Letter frequency histogram from string programming using CUDA.
10. Program to create threads using default stream in CUDA.
 - a. Streams to accelerate application.

- b. The feature of device overlap.
- 11. A Case Study on CUDA for Deep Learning .
 - a. CUDA for Machine Learning with a typical problem.
 - b. Deep Learning in Domain Specific Architectures.

Reference Books:

- Barbara Chapman, Gabriele Jost, Ruud van der Pas, “Using OpenMP Portable Shared Memory Parallel Programming”, The MIT Press, 2008, ISBN-13: 978-0-262-53302-7.
- David B. Kirk and Wen-mei W. Hwu, “Programming Massively Parallel Processors”, Second Edition, Morgan Kaufmann, 2013, ISBN: 978-0-12-415992-1.
- Jason Sanders, Edward Kandrot, “CUDA by Example”, Addition Wesley, 2011, ISBN 978-0-13-138768-3.

Department Elective-V (CT) Advanced Database Management System

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Examine different parallelism techniques for parallel databases.
2. Analyze Distributed Transaction and Query Processing.
3. Write OLAP queries.

Unit I :Introduction: Database System Architectures: Centralized and Client-Server Architectures, Server System Architecture, Parallel System, Distributed System, Network Types.

[6 Hrs]

Unit II: Parallel Databases: I/O Parallelism, Inter-query and Intra-query Parallelism, Inter-operational and Intra-operational Parallelism, Design of Parallel systems.

[6 Hrs]

Unit III: Distributed Databases: Homogeneous and Heterogeneous databases, Distributed DBMS architectures, Storing data in distributed DBMS, Distributed catalog management.

[6 Hrs]

Unit IV: Distributed Transactions: Distributed Transactions, Distributed Query processing: Non-join queries and joins, Updating distributed data: Synchronous and asynchronous replication, Distributed concurrency and recovery.

[8 Hrs]

Unit V: Data Warehouse and OLAP: Introduction to Decision support, Data Warehousing, Creating and maintaining a warehouse, OLAP: Multidimensional data model, OLAP Queries, Database design for OLAP, Implementation techniques for OLAP: Bitmap Indexes, Join Indexes, Views and decision support, Top N queries, Online aggregation.

[8 Hrs]

Unit VI: NoSQL Databases: Features of NoSQL databases, Types of NoSQL databases: Key-Value databases, Columnar Databases, Document Databases, Graph Databases.

[6 Hrs]

Text Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database system concepts", 5th Edition, McGraw Hill International Edition, ISBN-10: 9332901384, ISBN-13: 978-9332901384
- Raghu Ramkrishnan, Johannes Gehrke, "Database Management Systems", Second Edition, McGraw Hill International Editions, ISBN-10: 0072465638, ISBN-13: 978-0072465631

Reference Books:

- Rob Coronel, Database systems: “Design implementation and management”, 4th Edition, Thomson Learning Press.
- J. D. Ullman, “Principles of Database Systems”, 2nd Edition, Galgotia Publication, ISBN-10: 0716780690, ISBN-13: 978-0716780694
- R. Elmasri and S. Navathe, “Fundamentals of Database System”, Pearson, 7th Edition, ISBN-10: 933258270X, ISBN-13: 978-9332582705

Department Elective-V Lab (CT) Advanced Database Management Systems Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Assignments/Mini Project: 20 marks

End Semester Exam: 30 Marks

List of Assignments:

1. Write a program to demonstrate fragmentation and replication of databases.
2. Demonstrate relational operation sort-merge join algorithm.
3. Assignment to demonstrate OLAP queries.
4. Design and implement an application to demonstrate distributed databases.
5. Design and Implement a small application for Android Tablet.
6. Hadoop installation and programming using MapReduce.

Department Elective-V (CT) E- Commerce

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate an understanding of the foundations and importance of E-commerce
2. Understand the impact of Information and Communication technologies, especially of the Internet in business operations
3. Comprehend risk, legal issues and privacy in E-Commerce and Assess electronic payment systems
4. Analyze the critical building blocks of E-Commerce and different types of prevailing business models employed by leading industrial leaders.
5. Evaluate the opportunities and potential to apply and synthesize a variety of E-Commerce concepts and solutions to create business value for organizations, customers, and business partners.

Unit I: Introduction: E-Commerce: meaning advantages & disadvantages, incentives for engaging in electronic commerce, impact of e-commerce on business and e business, electronic commerce framework, types of e-commerce, web background **[6 Hrs]**

Unit II: Risk and Legal Issues in E-Commerce: risks and barriers in the adoption of e-business environment, the impact of ICT in contemporary business operations, entrepreneurial development in e commerce, cloud computing and e Commerce, e-commerce in India – laws for e-commerce in India, cryptocurrency and e commerce **[6 Hrs]**

Unit III: Ethical and Social and Political issues related to Electronic Commerce: Protecting privacy, protecting Intellectual property, copyright, trademarks and patents, taxation and encryption policies **[6 Hrs]**

Unit IV: E-Commerce Business Models: key element of a business model, major B2C business models, major B2B business models, business models in emerging e-commerce areas

E-Government: issues in e-governance applications, benefits and reasons for the introduction of e- governance, e-governance models **[8 Hrs]**

Unit V: Electronic Payment Systems: types, digital wallet, digital cash, payment through card system, e-cheque, e-cash, e-payment threats & protections, smart cards electronic payment systems, risk in electronic payment Systems, secure electronic transaction protocol, e-marketing : home –shopping, e-marketing, tele-marketing, digital marketing. **[8 Hrs]**

Unit VI: Electronic Data Interchange (EDI): meaning, benefits, concepts, application, EDI model, protocols, e-procurement, future of e commerce and latest technology: Storefront digitizing, chatbot, beacon technology, direct social media purchase, voice assistants, drone and droid delivery, augmented reality device etc. **[6 Hrs]**

Text Books:

- Kenneth C. Laudon, Carol G. Traver, "E-Commerce", Pearson Education, ISBN 97881317812.
- Doing Business on the Internet E-COMMERCE (Electronic Commerce for Business): S. Jaiswal, Galgotia Publications, ISBN 9788175153059.
- E-Business, Bookseller Code (AG) OXFORD, 1st edition Parag Kulkarni, Sunita Jahirabadkar & Pradip Chande, ISBN 9780198069843.

Reference Books:

- P. T. Josef, "Electronic Commerce- A managerial perspective" Prentice-Hall International, ISBN 8120320891.
- Kamlesh K. Bajaj, Debjani Nag, "Electronic Commerce: The cutting edge of business", Tata McGraw-Hill Publishing Co. Ltd, 2000, ISBN 9780070585560.
- Jeffrey F. Rayport, Bernard J. Jaworski, "e-Commerce", Tata McGraw Hill, 2002, ISBN 9780072510249.
- Pete Loshin, Paul A. Murphy, "Electronic Commerce", Jaico Publishing House, 2000, ISBN 9788172246662.
- Ravi Kalakota, Andrew B. Whinston, "Frontiers of Electronic Commerce", Addison Wesley, 2002, ISBN 0201845202.

Department Elective-V Laboratory (CT) E-Commerce Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Assignments/Mini Project: 20 marks

End Semester Exam: 30 Marks

List of Assignments:

1. Study of different e Commerce Platform e.g. Prestashop, WooCommerce, Kickstart, OpenCart or any other. Installation of any one EC platform (Any one you want)
2. Adding categories, product and product details, attributes, tags to the e commerce development tool you have chosen in activity 1.
3. Adding header, footer, slider and any other design as per your project to the e commerce development tool you have chosen in activity 1
4. Adding theme and various module management to the e commerce development tool you have chosen from activity 1
5. Restoring DB, payment taxes promotion discount to the e commerce development tool you have chosen from activity 1
6. Adding order management to the e commerce development tool you have chosen from activity 1
7. Integrating all above module and adding some other module as per requirement of product to present as mini project

Department Elective-V (CT) Cyber Security

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Define the need of Cyber Security.
2. Explain the IT act, Application Security vulnerabilities and its mitigation techniques.
3. Demonstrate the knowledge of penetration testing, and social networking security.
4. Analyse the malwares, social networking websites and impact of cyber-crime on e-commerce.

Unit I:Introduction: Nature and scope of computer crime, Understanding how cyber criminals and hackers work, Different types of cyber-crimes, Introduction to digital signatures, Cryptography, Digital certificate and public key infrastructure, IT Act., Impact of cyber-crime on e-governance and e-commerce. **[6 Hrs]**

Unit II: Malware reverse engineering: Overview of malware reverse engineering, Types of malware, Malicious code families, Latest trends in malware analysis, Basic static and dynamic analysis, Malware analysis techniques, Case study. **[6 Hrs]**

Unit III:Web application security: Introduction to web application security: Attacks, vulnerabilities and mitigation, Client-side security, Server-side security, Application security: HTTPS, HSTS etc., Security engineering: Passwords and their limitations, Attacks on passwords: CAPTCHA, OTP. **[8 Hrs]**

Unit IV:Advanced security topics:Secure email systems: PGP, SMIME, DKIM, DMARC, DNSSec, SMTP STS etc., Privacy and security for online social networks, Database security, Browser security, Mobile device security. **[8 Hrs]**

Unit V: Ethical hacking and penetration testing: Security Technologies: IDS, IPS, Ethical hacking, Penetration testing fundamentals: Reconnaissance, scanning, gaining access, maintaining access, Covering tracks. **[6 Hrs]**

Unit VI:Case studies: Cloud security, Operating system security, Security of social networking websites, IoT devices security, E-commerce websites security. **[6 Hrs]**

Text Books:

- Hossein, “Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management”, Wiley, Volume 3 edition, ISBN-13: 978-0470323069.

- Georgia Weidman, “Penetration testing: A Hands-On Introduction to Hacking”, No Starch Press, 2014, ISBN-13: 978-1593275648.
- Michael Sikorski and Andrew Honig, “ Practical Malware Analysis”, No Starch Press, 1st Edition, 2012, ISBN-13: 978-1593272906

Reference Books:

- “Practical Internet of Things Security” by Brian Russell, Drew Van Duren, Packt publishing, 2016, ISBN: 9781785889639
- T. Mather, S. Kumaraswamy, S. Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance”, O'Reilly Series, 2009, ISBN-13: 978-0596802769.
- “Cyberlaw: the Indian perspective”; Pavan Duggal; Saakshar Law Publications, 1st edition, 2002, ISBN: 8189121022, 9788189121020.

Department Elective-V Laboratory (CT) Cyber Security Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Assignments/Mini Project: 20 marks

End Semester Exam: 30 Marks

List of Assignments:

1. Perform literature survey of recent research papers on cyber security.
2. Perform case study of any two cyber-crime cases in India and write a report illustrating Indian cyber laws (IT Act, IT Act 2008. IPC) relevant to these cases.
3. Perform malware reverse engineering in an isolated environment using sandbox and document the findings.
 - a. Use any sample malware file.
 - b. Study malware behaviour: its working, how it spreads. Find its features/characteristics.
 - c. Identify the changes made by malware on the system. (for e.g., changes in event log, registry etc.). Difference between infected and normal system.
 - d. Describe detection method or alert system for malware
 - e. Write down steps to remove the malware and make system safe again.
4. Perform penetration testing using Kali Linux on virtual machine and write a report. Follow each stage of penetration testing: Planning and reconnaissance, scanning, gaining access, maintaining access and analysis. Once the Kali is installed, use following tools. Submit a report answering following questions:
 - a. Maltego: How did you perform the following things?
 - Associate an e-mail address to a person
 - Associate websites to a person
 - Verify an e-mail address
 - Gather details from Twitter, including geolocation of pictures
 - b. Vega: Provide a target website and scan that website for vulnerabilities.
 - c. NMAP: Scan your local network and provide screenshot of the report
 - d. Tamper Data plugin in FireFox: Gather information of GET and POST request for gmail.com
 - e. Metasploit framework: List out all the exploits provided by metasploit framework. Explain preconditions and expected end results for each one. Provide screenshots also. Do not use any exploit without proper permission.
5. Implement any IDS/IPS system.

This is an illustrative list of assignments. The instructor is expected to update the list.

Department Elective-V(CT) Distributed Systems

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Assignment/Quizzes – 40 marks

End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Identify challenges faced while designing a distributed system,
2. Apply appropriate algorithms in distributed systems in various scenarios,
3. Analyse the trends in distributed systems
4. Trace communication protocols in distributed systems.
5. Design distributed algorithms.
6. Develop an application using a technology from distributed system.

Unit 1 : Introduction: motivation, goals, advantages , disadvantages, hardware concepts, software concepts, design issues, middleware, overview of distributed systems . **[6 hrs]**

Unit 2 : Communication: client server model, middleware and client server model, relation of Network models with distributed system (TCP/IP, OSI, ATM etc.) , Remote Procedure Call, group communication and its protocol (IS-IS) **[6 hrs]**

Unit 3: Synchronization: clock synchronization, logical clocks, Lamport's algorithm, global state, vector algorithm, election algorithms, mutual exclusion algorithms, dead locks in distributed systems, deadlock avoidance, prevention and detection. **[8 hrs]**

Unit 4: Threads, system models, processor allocation, workstation model, Processor Pool Model, Hybrid Model, real time distributed systems, time triggered systems, event driven systems, distributed shared memory, consistency models, page based distributed shared memory. **[8 hrs]**

Unit 5 : Distributed File System , design, implementation, Trends. **[6 hrs]**

Unit 6: Applications of distributed systems : distributed object based systems, distributed co-ordination based systems, distributed document based systems. **[6 hrs]**

Text Books:

- Andrew S. Tanenbaum & Maarten van Steen," Distributed Systems –Principles and Paradigms" , PHI, ISBN-978-81-203-2215
- Pradeep K. Sinha, "Distributed Operating Systems Concepts and Design", PHI Publication

Reference Books:

- Andrew S. Tanenbaum. "*Distributed Operating Systems*", Pearson Education
- George Coulouris, Jean Dollimore & Tim Kindberg, "Distributed Systems, Concepts and Design"
- Galli D.L., "Distributed Operating Systems: Concepts and Practice", Prentice-Hall.
- Mullender S., "Distributed Systems", Addison Wesley.

Department Elective-V Laboratory (CT) Distributed Systems Laboratory

Teaching Scheme:

Laboratory : 2 hours per week

Examination Scheme:

Continuous evaluation: 50 Marks

Assignments/Mini Project: 20 marks

End Semester Exam: 30 Marks

List of Assignments:

1. Implement a client server model to read/write a file present at single server from multiple clients. Trace packets using tool like tcpdump and show the data which is passed on and from server from and to client.
2. Implement a RPC program to read/write a file present at single server from multiple clients. Trace packets using tool like tcpdump and show the data which is passed on and from server from and to client. Also show the protocol used.
3. Configure of NFS client and server in order to read and write a file present on NFS server. Trace packets using tool like tcpdump and show the data which is passed on and from server from and to client. Also show the protocol used.
4. Implement an application using distributed system techniques like DCOM/CORBA/TIB-RENDVOUS etc. [mini-project]

This is an illustrative list of assignments. The instructor is expected to update the list.