COEP Technological University

Department of Computer Science and Engineering

Curriculum Structure & Detailed Syllabus

T.Y. B. Tech. Computer Science and Engineering

(Effective from A.Y. 2025-26)

List of Abbreviations

Abbreviation	Title
BSC	Basic Science Course
ESC	Engineering Science Course
PCC	Programme Core Course (PCC)
PEC	Programme Elective Course (PEC)
OE	Open Elective - other than particular program
MDM	Multidisciplinary Minor (MDM)
VSEC	Vocational and Skill Enhancement Course (VSEC)
HSMC	Humanities Social Science and Management
AEC	Ability Enhancement Course
IKS	Indian Knowledge System (IKS)
VEC	Value Education Course (VEC)
RM	Research Methodology (RM)
OJT	Internship
CEA	Community Engagement Activity (CEA)/Field Project
CCA	Co-curricular & Extracurricular Activities (CCA)

CURRICULUM STRUCTURE OF T. Y. B. TECH

Effective A. Y. 2025-26

Semester - V

Sr. No.	Course Type	Course Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									MSE	TA	ESE	CIE	ESE
01	PCC	CT-25001	Software Engineering	3	0	2	1	4	30	20	50	100	--
02	PCC	CT-25002	Database Management Systems	3	0	2	1	4	30	20	50	100	--
03	PCC	CT-25003	Operating Systems	3	0	2	1	4	30	20	50	100	--
04	PEC	As per course	Programme Elective-1	2	0	2	1	3	30	20	50	100	--
05	OE	As per course	Open Elective-3	2	0	0	1	2	30	20	50	--	--
06	OJT	CT-25004	Internship	--	--	2	--	1	--	--	--	100	--
07	VSEC	CT-25005	Project Stage-1	--	--	4	--	2	--	--	--	100	--
08	MDM	As per course	Multidisciplinary Minor-2	3	0	2	1	4	30	20	50	100	--
			Total	**16**	**00**	**16**	**06**	**24**					

Semester – VI

Sr. No.	Course Type	Course Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									MSE	TA	ESE	CIE	ESE
01	PCC	CT-25006	Computer Networks	3	0	2	1	4	30	20	50	100	--
02	PCC	CT-25007	Design and Analysis of Algorithms	3	0	2	1	4	30	20	50	100	--
03	PCC	CT-25008	Artificial Intelligence	3	0	2	1	4	30	20	50	100	--
04	PEC	As per course	Programme Elective-2	3	0	2	1	4	30	20	50	100	--
05	RM	CT-25009	Research Methodology	2	0	0	1	2	30	20	50	--	--
06	VSEC	CT-25010	Project Stage-2	0	0	4	--	2	--	--	--	100	--
07	MDM	As per course	Multidisciplinary Minor-3	3	0	2	1	4	30	20	50	100	--
			Total	**17**	**00**	**14**	**06**	**24**					

**** Exit option to qualify for B.Voc.:** Mini Project / Internship (8 weeks)

DEPARTMENT ELECTIVES

PEC-1	PEC-2
Sem-5	**Sem-6**
Cloud Computing	Advanced System Programming
Parallel Computer Architecture and Programming	Blockchain Technologies
Mobile and Ad-hoc Networks	Geographical Information Systems
Remote Sensing	Machine Learning
Data Visualization	Functional Programming
Advanced Data Structures	Advanced Database Management Systems

MULTIDISCIPLINARY MINORS

Semester	Course Code	Course Title	L	T	P	S	Cr
V		Artificial Intelligence (Machine Learning and Deep Learning Mastery)	3	0	2	1	4
V	CT(MDM)-25002	Introduction to Quantum Computing	3	0	2	1	4
V		Object oriented Programming	3	0	2	1	4
VI		Artificial Intelligence (Natural Language Processing (NLP) and Computer Vision (CV) Mastery, Capstone Project)	3	0	2	1	4
VI	CT(MDM)-25005	Quantum Algorithms and Applications	3	0	2	1	4
VI		Fundamentals of Database Management Systems	3	0	2	1	4

OPEN ELECTIVES LIST

(offered to other departments)

Semester	Course Code	Course Title	L	T	P	S	Cr
V		Fundamentals of Machine Learning	2	0	0	1	2

(PCC) Software Engineering

Teaching Scheme	**Evaluation Scheme**
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	
Self-study: 1 Hr/ Week	Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students will be able to:

1. Describe fundamental concepts of system development lifecycle through SDLC.
2. Design user interface prototypes for real world scenarios using appropriate methods of analysis and design.
3. Devise procedure to assure the quality and maintainability of the product before and after deployment.
4. Develop skill to transfer acquired knowledge across a wide range of industrial and commercial domains and have a basis for further studies in software engineering or in computing related industries.
5. Analyze real world scenario and apply tools and techniques to produce application software solutions from informal and semiformal problem specifications.
6. Develop an ability to work in a team by communicating computing ideas effectively in writing a technical report.

Course content

Software Development Process: **[8 Hrs]**

Software Engineering basics, Software Crisis and Myths, Software Quality Attributes, Software Process and development, Software life cycle and Models, Analysis and comparison of various models, Iterative Enhancement Models, agile process.

Requirement Engineering: **[6 Hrs]**

Requirements Engineering, requirement engineering process, SRS Document, IEEE Standards for SRS, Introduction to Analysis model.

System Architecture and Design Overview: **[8 Hrs]**

Architecture 4+1 view, architecture styles, Design process, Data Flow Diagrams, Entity Relationship Diagrams, Refactoring of designs, quality concepts – modularity, cohesion coupling, Analysis model to Design Model transformation, Standardisation using UML.

Project Management & Software Metrics: **[6 Hrs]**

Project Management Concepts - Planning, Scope, Project Scheduling, Work break-down structure, Milestones, Deliverables, Introduction to Software Metrics-Size-oriented metrics and function point metrics. Effort and cost estimation techniques - LOC-based and Function- point based measures - The COCOMO model.

Testing and Maintenance: **[8 Hrs]**

Validation and Verification activities, Testing Principles and strategies, testing levels- Test Drivers and Test Stubs, Testing types- White Box & Black Box Testing, Software Maintenance, Types of Maintenance, Overview of Software Configuration Management.

Quality Assurance: **[6 Hrs]**

Quality Assurance, Quality Standards Models Overview - ISO, Process improvement Models: CMM & CMMI. Quality Control and Quality Assurance

Self-study: **[12 Hrs]**

Case Study of SRS as per IEEE Standards, Entity Relationship Diagrams, Risk Management, Software Configuration Management Tools, Quality Standards Models - TQM, Six-Sigma.

Textbooks

1. R. S. Pressman, “Software Engineering: A Practitioners Approach”, McGraw Hill, 7th edition, 2010
2. Rajib Mall, “Fundamentals of Software Engineering”, PHI Publication, 3rd edition, 2009
3. “G. Booch, J. Rumbaugh and I. Jacobson. The Unified Modeling Language User Guide, Addison Wesley, 1999.

Reference Books

1. “Shari Pfleeger, “Software Engineering”, 2nd Edition. Pearson’s Education, 2001. · “Ian Sommerville, “Software Engineering”, 6th Edition, Addison-Wesley, 2000
2. “Pankaj Jalote, “An Integrated Approach to Software Engineering”, Narosa publication house
3. “Fred Brooks, “Mythical Manmonths”

Web references

1. www.cs.drexel.edu/~yfcai/CS4517/.

Laboratory assignments

A full-fledged working system in the form of mini project will be implemented following the task list given below. Students in groups of two will be working on mini project. After consultation with the course instructor and finalization of the topic following deliverables are expected under mini project. Task List for the same is as follows:

1. Carry out state of art surveys, selecting appropriate domain, problem identification statement formulation based on research problems or real-world problems, industry-based problems etc.
2. Develop workflow graphs and carry project estimation, calculation of efforts,project planning (schedule) using automated tools.
3. Gather requirements and Write the Software Requirement specification (SRS, IEEE specs) document for the project.
4. Design System architecture using different ERD, DFD, UML diagrams for the proposed system. (Use different open-source tools for design.)
5. Coding/Implementation using suitable Programming Languages/ Technology.
6. Develop Test cases. Propose solution for wrong results in test cases by focusing on regression testing.
7. Write the constraints, advantages and disadvantages of your project over existing system. Write the future scope of your project. Develop help manual for maintenance and usability.

Students will be required to submit a technical report written using LaTEX. The technical report will include description of the project/problem, design of the software, description of problems solved and solution design, result analysis of test cases and conclusions. Students will also be required to demonstrate and present their work in a viva-voce.

(PCC) Database Management Systems

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will be able to:
1. Identify and describe various components of DBMS.
2. Construct Entity-Relationship Model for given applications and demonstrate it as a relational model.
3. Design optimal (SQL) query statement.
4. Apply normalization to database design.
5. Improve efficiency in data retrieval by using indexing and solving analytical problems on serializability, concurrency and recovery.

Course content

Introduction: **[4 Hrs]**

Basic Concepts; Database system application, purpose of database systems, view of data, database languages; Database architecture: components of DBMS and overall structure of DBMS; Various types of databases.

E-R and Relational Model: **[8 Hrs]**

Database design; 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.

Relational Algebra and SQL: **[6 Hrs]**

Relational algebra: 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.

Relational Database Design: **[6 Hrs]**

Basic concept of normalization; Decomposition using functional dependencies (1 to BCNF and 3NF).

Indexing and Hashing: **[6 Hrs]**

Basic concepts, Ordered Indices; Indices: concepts, B+ trees; Static and dynamic hashing. Self-Study: B tree index file, Bit Map indices.

Transactions and Concurrency control: **[10 Hrs]**

Transaction: basic concepts, states, concurrent execution, serializability, recoverability, isolation; Concurrency control: timestamps and locking protocols, validation-based protocols, deadlock handling; Recovery.

Self-Study: **[12 Hrs]**
Network Model, Views, Basics of Multivalued dependencies, Multiple granularity protocols, ARIES algorithm.

Textbooks

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database system concepts”, Fifth Edition, McGraw Hill International Edition, ISBN 978-0073523323.
2. Raghu Ramkrishnan, Johannes Gehrke, “Database Management Systems”, Second Edition, McGraw Hill International Editions, ISBN 978-0072465631.

Reference Books

1. Rob Coronel, “Database systems: Design implementation and management”, Fourth Edition, Thomson Learning Press, ISBN 978-1418835934.
2. Ramez Elmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Third Edition, Pearson Education, 2003, ISBN 978-0321204486.
3. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", Second Edition, Pearson, 2008, ISBN 978-0131873254.

Laboratory assignments

1. Basic SQL Statement: Write the simple SQL Queries on the given schema.
2. Assignment

   a. SQL: Aggregates Statement: Write the SQL queries using aggregates, grouping and ordering statements for given statements on given schema.

   b. Nested Subqueries and SQL Updates Statement: Write the SQL queries forgiven schema using Nested Subqueries and SQL Updates
3. Assignment 3: SQL DDL and updates Statement: Write the DDL and DML statements for the given statements.
4. Assignment 4: Schema creation and constraints Statement: Create the schema and constraints on the given relations using given statements.
5. Assignment 5: Implement a small database application using appropriate front End and Back end. Select any real time problem for database implementation. (Draw the ER diagram for the selected problem in hand. Normalize the database up to the appropriate normal form.) This list is a guideline. The instructor is expected to improve it continuously.

(PCC) Operating Systems

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will be able to:
1. Write programs to deal with processes, files, and hardware resources using appropriate system calls.
2. Illustrate the design issues, solutions and complexity of operating system by compiling, modifying an OS kernel, tracing the sequence of activities on processors, data structures of a file system, race conditions, locking mechanisms and storage techniques.
3. Correlate the computer architecture features with operating system design issues.
4. Make design choices for an operating system with given constraints
5. Discuss suitable changes to existing implementations of operating systems for new features.

Course content

Introduction and Operating Systems structures: **[6 Hrs]**

Operating system components, O.S. Services, System Calls, Virtual Machines, Special purpose operating systems, Open-source operating systems, Boot Procedure. Source code of boot loader.; System Services; Linkers and Loaders; Operating system structures; Calling convention and Executable file formats:

Processes and CPU Scheduling: **[6 Hrs]**

Interrupt and Interrupt handling, Process concept, interleaved I/O and CPU burst; Process states; Co-operating processes, Thread, Thread libraries, Multithreaded programming, Scheduling, Scheduling criterion, Scheduling algorithms, multi-processor scheduling, Real time scheduling, Source code of interrupt handling and scheduler and its dependence on hardware features.

Memory management: **[10 Hrs]**

O.S. and hardware interaction, Swapping, Continuous memory management, paging, Segmentation, Virtual Memory Management, Demand Paging, Copyon-write, Page replacement algorithms, Allocation of frames, Thrashing, Kernel memory management, SVR4 architecture, Unified buffer cache. Case study of x86 architecture and implementation of segmentation and paging in an OS kernel, code of fork(), exec(), sbrk(), etc. [10 Hrs]

File Management and Storage Structures: **[6 Hrs]**

File Organization, Concept of files and directories, System calls for file systems, Space allocation issues, Free space management, Data structures like inode and super block, Virtual file system and related object-oriented concepts, Disk layout, Formatting, Recovery, NFS, Efficiency and performance, Distributed file systems, Disk Structure, Disk Scheduling, RAID. Case study of a file system like ext2.

Process Synchronization: **[8 Hrs]**

Critical section problem, Hardware support for mutual exclusion, Semaphores, Deadlock-principle, Deadlock detection, prevention and avoidance, Classical problems in concurrent programming: Producer-consumer, Reader-writer with and without bounded buffer. Design of locking primitives like spinlock, semaphore, read-write locks, recursive locks, etc. Source code of spin locks, semaphores, read-write locks.

Inter process Communication: **[4 Hrs]**

Pipes, Shared memory mechanism, Streams, Asynchronous communication, Signals. A shell program using system calls and pipes.

Self-study: **[12 Hrs]**

Design aspects of Android operating systems like ext3 and ext4; pthread implementation of threads; SVR4 memory management architecture; Network file systems; Synchronization problems in kernels like Solaris; IPC mechanisms in different kernels like Linux, MAC, etc.

Textbooks

1. Abranhan Silberschatz, Peter B Galvin, Greg Gagne; Operating System Concepts, Wiley India Students Edition, 9th Edition, ISBN: 978-81-265-2051-0
2. Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (UniversityofWisconsin-Madison),
3. xv6, a simple, Unix-like teaching operating system, Russ Cox, Frans Kaashoek, Robert Morris; E-book

Reference Books

1. Milan Milenkovic; Operating Systems; Tata McGraw Hill; Second Edition. ISBN: 0- 07-044700-4
2. Andrew S. Tanenbaum; Modern Operating Systems; Prentice Hall of India Publication; 3rd Edition. ISBN: 978-81-203-3904-0
3. Maurice J. Bach; The Design of the Unix Opearating System; Prentice Hall of India; ISBN: 978-81-203-0516-8
4. Uresh Vahalia; Unix Internals, The New Frontiers; Prentice Hall; ISBN: 0-13-101908- 2

Web references

- https://pages.cs.wisc.edu/~remzi/OSTEP/

Laboratory assignments

Suggested List of Assignments:

1. Install two operating systems in dual boot mode using virtual machines. One OS will run a GNU/Linux of your choice on it. The other virtual machine will run any nonLinux operating system.
2. Write a minimal version of a shell. The shell should be able to a) execute a program without the complete path name b) handle pipes c) handle redirection d) handle signals
3. Use debugfs tool to locate a file which was recently deleted on an ext2 file system.
4. Write a program, on the lines of debugfs, to browse an ext2 file system and give the complete name of a file, print it’s inode.
5. Download Linux kernel source code, compile it and reboot your system with the newly compiled kernel. Add a dummy system call to the Linux kernel. Write a conformance test to test your system call.
6. Implement a list type. Write a code using pthreads for concurrent insertions to the list and demonstrate the problem of race. Then rewrite the program to show how race conditions can be solved by using proper synchronization primitives.
7. Write a program which results in a guaranteed deadlock among its threads. Kill the threads using operating system’s commands to solve the deadlock.
8. Write a program using pthreads to demonstrate the producer consumer problem. Implement appropriate synchronization. Show the different results with and without synchronization.
9. Write a program which acts as a chat application between two users on the same computer, using shared memory.
10. Implement the lseek() system call in xv6 kernel.

11. Rewrite the kernel memory manager in xv6 kernel, that is rewrite the kmalloc() and kfree() using another data structure.
12. Suggested list of course projects: Implement mkfs, fsck for ext2 file system; Implement a multithreading library using 1-1, many-many, many-one implementation on Linux; Any of the xv6 based projects: implement demand paging, implement ext2 filesystem, implement signals, Implement shared memory, Implement Kernel threads, Implement mmap(), etc.

Suggested list of demonstrations by instructor.

1. Trace and explain completely the output of trace on running a “Hello World” C program.
2. Write a program to demonstrate the usage of signals – show how processes can wait for each other, kill each other, stop and continue each other.
3. Demonstrate the changing memory map of a process, by using the contents of the /proc file system, and creative use of malloc() function in the code of the process.
4. Demonstrate that not unmounting a file system results in loss of data. Recover possible data using fsck and restore file system consistency. Analyze the recovered data. This list is a guideline. The instructor is expected to improve it continuously.

This list is a guideline. The instructor is expected to improve it continuously.

(OE) Fundamentals of Machine Learning

Teaching Scheme
Lectures: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes
Students will be able to:
1. Understand the basic concepts, tools and techniques of Machine Learning.
2. Apply various data pre-processing techniques to simplify and speed up machine learning algorithms.
3. Analyze different ML models.
4. Implement ML algorithms to solve real life problems.
5. Evaluate the different ML models

Prerequisites: Relevant applied math and statistics: probability theory, probability distribution, Conditional probability, Bayesian probabilities, Data visualization

Course content

Introduction to Machine Learning: [4 Hrs]

Introduction to Machine Learning, Comparison of Machine Learning with traditional programming, ML vs AI vs Data Science. Types of learning: Supervised, Unsupervised, and semi-supervised, reinforcement learning techniques, Important Elements of Machine Learning- Data formats.

Data Preprocessing: Libraries: [5 Hrs]

Pandas, NumPy, Scikit-Learn, Understanding Data: Dealing with missing values, Data formatting, Implementing Data Cleaning: Preprocessing of data: Transformation, Normalization and Scaling, Standardization, Importing and exporting data in Python, Data Splitting: Train-Test Split, Cross-Validation

Supervised Learning: [4 Hrs]

Regression Types of Regression Techniques: Linear regression, Multiple regression, Polynomial regression, Lasso regression, Ridge regression, Evaluation Metrics for Regression Models: MAE, MSE, RMSE, R square

Supervised Learning: [5 Hrs]

Classification: Logistic Regression, K-nearest neighbor, Support vector machine., Random Forest, Binary-vs-Multiclass Classification, Evaluation Metrics and Score: Accuracy, Precision, Recall, F score.

Unsupervised Learning: [4 Hrs]

Introduction to Unsupervised Learning, Definition and Scope of Unsupervised Learning, Types of Unsupervised Learning: Clustering, Dimensionality Reduction, Association Rule Mining, Evaluation metrics and score: elbow method, extrinsic and intrinsic methods

Model Evaluation and Tuning: [5 Hrs]

Bias-Variance Trade off, Overfitting and Underfitting, Cross-Validation Techniques, Hyperparameter Tuning: Grid Search, Random Search, Model Evaluation Metrics, Visualizing ROC, AUC, and Precision-Recall Curves using Matplotlib and Seaborn.

Case Studies and Applications: [4 Hrs]
Evaluation and Tuning of Regression Models (Business Forecasting), Evaluation and Tuning of Classification Models (Healthcare Diagnostics), Comparing Models using Evaluation Metric.

Self-study: Introduction To Neural Networks: **[12 Hrs]**

Artificial Neural Networks: Single Layer Neural Network, Multilayer Perceptron, Back Propagation Learning, Functional Link Artificial Neural Network, and Radial Basis Function Network, Activation functions, Introduction to Recurrent Neural Networks and Convolutional Neural Networks.

Textbooks

3. Bishop, Christopher M., and Nasser M. Nasrabadi, “Pattern recognition and machine learning”, Vol. 4.No. 4. New York: springer, 2006.
4. Ethem Alpaydin, “Introduction to Machine Learning”, PHI 2nd Edition-2013

Reference Books

4. Tom Mitchell, “ Machine learning”, McGraw-Hill series in Computer Science, 1997
5. Shalev-Shwartz, Shai, and Shai Ben-David, “Understanding machine learning: From theory toalgorithms”, Cambridge university press, 2014.
6. Jiawei Han, Micheline Kamber, and Jian Pie, “Data Mining: Concepts and Techniques”, Elsevier Publishers Third Edition, ISBN: 9780123814791, 9780123814807
7. Hastie, Trevor, et al., “The elements of statistical learning: data mining, inference, and prediction”, Vol. 2. New York: springer, 2009.
8. McKinney, “Python for Data Analysis “,O' Reilly media, ISBN : 978-1-449-31979-3

(PEC) DE1- Remote Sensing

Teaching Scheme
Lectures: 2 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

students will be able to:

1. Understand the fundamental principles of remote sensing, including electromagnetic radiation, sensor systems, and data acquisition.
2. Analyze and interpret remote sensing data for various applications such as land cover classification, environmental monitoring, and disaster management.
3. Apply image processing techniques to enhance and classify remote sensing imagery.
4. Evaluate the advantages and limitations of different remote sensing platforms (satellite, aerial, UAV).
5. Demonstrate proficiency in using remote sensing software (e.g., QGIS, ENVI, Google Earth Engine) for data analysis.

Course content

Introduction to Remote Sensing: [4 Hrs]

Definition, history, and significance of remote sensing, Electromagnetic spectrum and its relevance to remote sensing, Interaction of EMR with atmosphere and Earth's surface (absorption, scattering, reflection).

Remote Sensing Platforms and Sensors: [5 Hrs]

Types of platforms: Satellites (Landsat, Sentinel, MODIS), Aerial, UAVs, Sensor characteristics: Spatial, spectral, temporal, and radiometric resolution, Passive vs. active remote sensing (Optical, Thermal, Microwave, LiDAR).

Image Acquisition and Preprocessing: [6 Hrs]

Data acquisition methods and formats (GeoTIFF, HDF, NetCDF), Radiometric and geometric corrections, Image enhancement techniques (histogram equalization, filtering).

Image Classification and Analysis: [6 Hrs]

Supervised vs. unsupervised classification, Feature extraction and object-based image analysis (OBIA), Accuracy assessment (confusion matrix, Kappa coefficient).

Emerging Trends in Remote Sensing: [4 Hrs]

Hyperspectral and microwave remote sensing, Machine learning and AI in remote sensing, Cloud computing for remote sensing (Google Earth Engine).

Applications of Remote Sensing: [5 Hrs]

Land use/land cover mapping, Vegetation monitoring (NDVI, EVI), Disaster management (floods, wildfires, deforestation), Urban planning and climate change studies.

Self-Study: [12 Hrs]

Textbooks

1. "Remote Sensing and Image Interpretation" – Lillesand, Kiefer, Chipman.

2. "Introduction to Remote Sensing" – James B. Campbell.

Reference Books

1. "Remote Sensing of the Environment: An Earth Resource Perspective" – John R. Jensen.
2. "Hyperspectral Remote Sensing: Principles and Applications" – Marcus Borengasser.

Laboratory assignments

Suggested List of Assignments

1. Basics of Remote Sensing Software
   - Install and explore QGIS/ Any other open-source GIS Tool.
   - Load and visualize satellite imagery (Landsat/Sentinel).

2. Image Preprocessing
   - Perform radiometric correction (DN to reflectance).
   - Apply geometric correction (reprojection, resampling).

3. Image Enhancement Techniques
   - Apply histogram equalization and contrast stretching.
   - Perform edge detection and filtering.

4. Spectral Indices Calculation
   - Compute NDVI, NDWI, and other vegetation/water indices.
   - Analyze changes over time using multi-temporal imagery.

5. Image Classification
   - Perform unsupervised classification (K-means, ISODATA).
   - Conduct supervised classification (Maximum Likelihood, SVM).

6. Accuracy Assessment
   - Generate a confusion matrix.
   - Calculate overall accuracy and Kappa coefficient.

7. Mini Project

This is a suggested list. The instructor is expected to continuously update it.

(PEC) DE1- Advanced Data Structures

Teaching Scheme
Lectures: 2 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students will be able to:

1. Illustrate advanced data structures to solve complex problems in various domains.
2. Develop logic for graphical modelling of real time scenarios
3. Select the non-linear data structures for real world scenarios.
4. Implement suitable data structure for any practical problem with given constraints.
5. Analyze the algorithmic solutions for resource requirements and optimization

Course content

Introduction: [4 Hrs]

Abstract data type, Algorithm; Asymptotic notations: time and space complexity, time analysis of recursive algorithms, amortized analysis; Recurrence relation: Master theorem, recursion tree, substitution method. Data Structure: Tree, BST, Graph.

Search Trees: [6 Hrs]

Balanced Binary search trees- (AVL trees, red-black tree, splay trees); Multiway search trees- (B-trees, 2-3 trees); specialized search trees- (Treaps, skip list); Multidimensional search trees- (K-D trees, segment trees).

Heap: [6 Hrs]

Overview, Leftist Heap, skew Heap, Binomial Heap, Fibonacci Heap; Applications Priority Queue, Graph algorithms, Huffman coding

Data Structure for Strings: [6 Hrs]

Introduction to string data structures, Tries, Compressed Tries, Suffix Trees, Suffix Arrays; Applications-Search Engines, Bioinformatics, Pattern Matching: KMP algorithm; Internet Packet Forwarding.

Advanced Graph: [6 Hrs]

Graph implementation, Disjoint set union problem - Array implementation, linked list implementation, rooted tree implementation, path compression, Hypergraphs

Self-Study: [12 hrs]

Maximal flow problem: Ford Fulkerson Algorithm, Hamiltonian Path and circuit problem, A* for AI path finding, Computational Geometry: Geometric Data Structure, Plane Sweep Paradigm.

Textbooks

1. Peter Brass, "Advanced Data Structures", Cambridge University Press, ISBN- 978-0- 521-88037-4
2. Handbook of Data Structures and Applications; by Dinesh P. Mehta (Editor), Sartaj Sahni (Editor); Chapman and Hall/CRC; ISBN-10: 1584884355; ISBN-13: 978- 1584884354

Reference Books

1. E. Horowitz, S. Sahni, S. Anderson-freed, “Fundamentals of Data Structures in C”, Second Edition, University Press, ISBN 978-81-7371-605-8
2. T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, 2nd edition, Prentice-Hall India, 2001.
3. S. Sahni, Data Structures, Algorithms and Applications in C++,2nd edition, Universities Press,2005.
4. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd edition, Addison Wesley Educational Pub
5. J. P. Tremblay and P. G. Sorenson, “An Introduction to Data Structures with applications”, Second Edition, Tata McGraw Hill, 1981

Laboratory assignments

Suggested List of Assignments

1. Write the following functions for a binary search tree implementation: Searches the maximum value in the tree, preorder traversal without using recursion, Search the string in the tree and returns a pointer to the node, print the binary tree so that it looks like a tree.
2. Implement red-black tree for banking system-Efficient transaction processing. Perform the operations such as insert, search, sort, update based on timestamp
3. Implement search tree which is suitable for E-commerce platform - a product search system in an e-commerce database
4. Start with an empty AVL tree and perform series of insertions like: December, January, April, March, July, August, October, February, November, May, June. Display the tree.
5. Develop C functions to insert and delete into/from a max heap under the assumption that a dynamically allocated array is used, the initial capacity of this array is 1, and array doubling is done whenever we are to insert into a max heap that is full.
6. Use Treap to implement a priority-based task scheduler where each task has a unique ID (key) and a priority. Higher-priority tasks should be completed first while maintaining the properties of a Binary Search Tree (BST).
7. Implement Hospital Emergency Queue to cater services to the patients such as Insert, delete and find highest priority
8. Create Binomial Heap and Insert keys in Binomial Heap, Find Minimum key of Binomial Heap, unite two Binomial Heap: Take Binomial Heaps as parameter and unite it. Extract Minimum Key, Decrease Key, Delete Key
9. A large logistics company faces the following challenges: 1. Efficiently scheduled delivery trucks based on priority and delivery deadlines. 2. Handle real-time changes and increased delivery volumes. 3. Ensure efficient load balancing across multiple distribution hubs. 4. Allow dynamic merging of delivery queues when combining or splitting routes. Implement a suitable data structure.
10. Create a dictionary of English words. Insert all words from the dataset into the trie. Perform Search and auto-suggest operation.

Suggested list of course projects: E-commerce Platform, Memory Management, Dynamic Visualizations of various data structures, Shortest Pathfinding Using A* for Optimal Routing, etc.

Suggested list of demonstrations by instructor.

1. Write a program to demonstrate the balanced Binary search tree and various rotations after insertion or deletion of an element. Analyze the time and space complexity.
2. Write a program to demonstrate the usage of multi-way search trees – Implement insertion, deletion, searching operations.
3. Demonstrate the variants of heap and its operations.
4. Implement the tries data structure with insertion and deletion operation. Analyze the time and space complexity. This list is a guideline.

The instructor is expected to improve it continuously.

(PEC1) Cloud Computing

TeachingScheme
Lectures:3Hrs/Week
Laboratory: 2 Hrs/ Week

EvaluationScheme
Theory:MSE: 30Marks,TA: 20Marks,ESE: 50 Marks
Laboratory:CIE: 50Marks, ESE: 50 Marks

Course outcomes

1. Understand key concepts and definitions in cloud computing.
2. Interpret core principles and illustrate them with real-world examples.
3. Demonstrate learned methods, tools, or frameworks to solve practical problems
4. Evaluate and analyze case studies to design alternative solutions, using innovative approaches

Course content

Introduction: [6Hrs]

History of Centralized and Distributed Computing - Overview of Distributed Computing, Cluster computing, Grid computing Distributed Computing and Enabling Technologies, Cloud Fundamentals: Cloud Definition, Evolution, Architecture, Applications, deployment models, and service models

Virtualization: **[6Hrs]**

Virtualization basics, virtualization technologies and architectures, types of virtualization, para-virtualization, full-virtualization, Issues with virtualization and Multi-tenancy, Introduction to Micro-services, Containers and Kubernetes

Migration and Interoperability: **[6Hrs]**

Broad Aspects of Migration into Cloud, Migration of virtual Machines and techniques, Fault Tolerance Mechanisms, Interoperability approaches, Service Monitoring, Issues with interoperability, Vendor lock-in, SLA Management, Metering Issues

Resource Management and Load Balancing: **[6Hrs]**

Server consolidation, Dynamic provisioning and resource management, Resource Optimization, Resource dynamic reconfiguration, Scheduling Techniques, Capacity Management to meet SLA Requirements, and Load Balancing, load balancing techniques.

Cloud Computing Security: [6Hrs]

Infrastructure Security- Network level security, Host level security, Application level security, Data security, Storage-Data privacy and security Issues in cloud computing

Cloud Simulators- CloudSim and GreenCloud **[6Hrs]**

Introduction to Simulator, understanding CloudSim simulator, CloudSim Architecture -User code, CloudSim, GridSim, SimJava, Understanding Working platform for CloudSim, Introduction to GreenCloud

Self-study **[12Hrs]**

Case study of Service model using Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, Eucalyptus

Textbooks

1. Virtualization, A beginners Guide, Danielle Ruest and Nelson Ruest, Tata McGraw Hill, ISBN: 978-661-2031-892

2. Moving to the cloud, Dinakar Sitaram and Geetha Manjunath, Elsevier, ISBN: 978-159- 7497-251
3. Using Docker: Developing and Deploying Software with Containers,Adrian Mouat, O'reilly Media, ISBN: 978-149-1915-769

Reference Books

1. Cloud Computing: Methodology, Systems, and Applications, L. Wang, R. Ranjan, J. Chen, and B. Benatallah, CRC Press, Boca Raton, FL,USA, ISBN: 978-143-9856-413
2. Cloud Computing: Principles and Paradigms, Buyya R., Broberg J., Goscinski A., John Wiley & Sons Inc., ISBN: 978-0-470-88799-8
3. The Docker Book: Containerization is the new virtualization, James Turnbull, ISBN: 978- 098-8820-203
4. Cloud computing for dummies- Judith Hurwitz , Robin Bloor , Marcia Kaufman ,Fern Halper, Wiley Publishing, Inc, 2010
5. Cloud Computing (Principles and Paradigms), Edited by Rajkumar Buyya, James Broberg, Andrzej Goscinski, John Wiley & Sons, Inc. 2011

Web references

1. Bernard Golden, Amazon Web Service for Dummies, John Wiley & Sons, 2013.
2. Raoul Alongi, AWS: The Most Complete Guide to Amazon Web Service from Beginner to Advanced Level, Amazon Asia- Pacific Holdings Private Limited, 2019.
3. Sriram Krishnan, Programming: Windows Azure, O'Reilly,2010.
4. Rajkumar Buyya, Christian Vacchiola, S.Thamarai Selvi, Mastering Cloud Computing , MCGraw Hill Education (India) Pvt. Ltd., 2013.
5. Danielle Ruest, Nelson Ruest, ―Virtualization: A Beginner‟s Guide‖, McGraw-Hill Osborne Media, 2009.
6. Jim Smith, Ravi Nair , "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.
7. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010
8. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", McGraw-Hill Osborne Media, 2009.
9. Tom White, "Hadoop: The Definitive Guide", Yahoo Press, 201Bernard Golden, Amazon Web Service for Dummies, John Wiley & Sons, 2013.
10. Raoul Alongi, AWS: The Most Complete Guide to Amazon Web Service from Beginner to Advanced Level, Amazon Asia- Pacific Holdings Private Limited, 2019.
11. Sriram Krishnan, Programming: Windows Azure, O'Reilly,2010.
12. Rajkumar Buyya, Christian Vacchiola, S.Thamarai Selvi, Mastering Cloud Computing , MCGraw Hill Education (India) Pvt. Ltd., 2013.
13. Danielle Ruest, Nelson Ruest, ―Virtualization: A Beginner‟s Guide‖, McGraw-Hill Osborne Media, 2009.
14. Jim Smith, Ravi Nair , "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.
15. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation,Management, and Security", CRC Press, 2010
16. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", McGraw-Hill Osborne Media, 2009.
17. Tom White, "Hadoop: The Definitive Guide", Yahoo Press, 201
18. https://www.ibm.com/think/topics/cloud-computing
19. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-292.pdf
20. https://cloud.google.com/

Laboratory assignments

1. Set up a simple client-server model to demonstrate centralized vs. distributed computing. Compare performance and fault tolerance between the two setups.
2. Deploy a sample application using IaaS, PaaS, and SaaS (e.g., VM on AWS, App Engine on GCP, Office 365). Identify differences in control, scalability, and cost.
3. Install and configure VirtualBox/VMware to run multiple OS instances.Demonstrate para-virtualization vs. full virtualization.
4. Containerize a simple web app using Docker and deploy it on Kubernetes. Show scaling and self-healing features of Kubernetes.
5. Migrate a local VM to a cloud platform (e.g., AWS EC2 or Azure VM). Document migration steps and challenges (network, compatibility).
6. Implement a load-balanced web service with failover using cloud tools. Test service continuity during simulated failures.
7. Configure auto-scaling groups in AWS or GCP. Observe dynamic provisioning under varying workloads.
8. Apply security measures (firewall rules, IAM roles, encryption) to a cloud-hosted app. Demonstrate protection at network, host, and application levels.
9. Run experiments on **CloudSim** to simulate resource allocation and scheduling. Compare different scheduling techniques (e.g., time-shared vs. space-shared).
10. Analyze and present a case study of GCP, AWS, or Azure service models. Highlight deployment models, pricing, and real-world applications

(PEC) DE1- Parallel Computer Architecture and Programming

Teaching Scheme
Lectures: 2 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students will be able:

1. To understand different parallel architectures and models of computation.
2. To analyze the various classes of algorithms and gain knowledge about various
3. parallel programming paradigms
4. To program shared memory multicore computer system using OpenMP.
5. To develop practical programming skills using MPI.
6. To apply knowledge of parallel programming to solve different problems.

Course content

Introduction to Parallel Computers: **[5 Hrs]**

Need for parallel computing - SM-SIMD algorithms - Shared memory SIMD - Tree and mesh interconnection computers - Classifying MIMD Algorithms - parallel computational models such as PRAM - LMCC - Hypercube - Cube Connected Cycle - Butterfly - Perfect Shuffle Computers - Tree model - Pyramid model - Fully Connected model - PRAM - CREW - EREW models - simulation of one model from another one.

Introduction to Parallel Programming: **[6 Hrs]**

Strategies - Mechanism - Performance theory - Parallel Programming Patterns: Nesting pattern - Parallel Control Pattern - Parallel Data Management - Map: Scaled Vector - Mandelbrot - Collative: Reduce - Fusing Map and Reduce - Scan - Fusing Map and Scan - Data Recognition: Gather - Scatter - Pack - Stencil and Recurrence - Fork-Join – Pipeline.

Threading Fundamental concepts: **[5 Hrs]**

Parallel programming models - Designing for threads - Scheduling - Threading and parallel programming constructs – Synchronization Threading APIs.

OpenMP Programming: **[6 Hrs]**

OpenMP - Threading a loop - Thread overheads - Performance issues - Library functions - Solutions to parallel programming problems - Data races - deadlocks and live locks - non-blocking algorithms - Memory and cache related issues.

MPI Programming: **[6 Hrs]**

Distributed memory – Cluster -MPI Model - collective communication - data decomposition – communicators and topologies - point-to-point communication - MPI Library.

Self-study: **[12 Hrs]**

Self-Study: C-DAC’s PARAM Utkarsh Supercomputer case study and programming models. Intel’s TBB or any other point in context of recent findings.

Textbooks

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Second Edition, 2017, Pearson.
2. Michael J. Quinn, Parallel Computing: Theory and Practice, Second Edition, 2017, McGraw Hill

Education.

Reference Books

1. Herlihy, M., Shavit, N., Luchangco, V., & Spear, M., The art of multiprocessor programming, Second Edition, 2021, Morgan Kaufmann.
2. Michael McCool, James Reinders, Arch Robison, Structured Parallel Programming: Patterns for Efficient Computation, 2012, Morgan Kaufmann.

Laboratory assignments

Suggested List of Assignments:

1. Study of different benchmarks used to evaluate performance of different systems.
2. Performance statistics observation using perf utility.
3. Program to execute matrix multiplication using pthreads.
4. Program to execute matrix multiplication using OpenMP and comparison with pthread program.
5. Program to execute Pi computation and prefix sum using OpenMP.
6. Program to execute section, task and synchronization constructs of OpenMP.
7. Case Study of Cluster building steps - MPI Cluster setup and overview of different routines.
8. Program to implement point to point communication using MPI routines.
9. Program to implement collective communication using MPI routines.

(CT-DE1) Mobile and Ad-Hoc Networks

Teaching Scheme
Lectures: 2 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will be able to:

1. Describe primitive Ad-Hoc network, its specific characteristics and features exhibited by
1. the networks.
2. Identify the different types of mobile Ad-Hoc networks and their challenges.
3. Differentiate the Media Access Control protocols for Ad-Hoc networks.
4. Identify the different transport layer functionalities and protocol and related security.
5. Trace the issues & challenges in providing QoS in wireless network and their classification.

Course content

Introduction: **[5 Hrs]**

Fundamentals of wireless communication technology, Characteristics of the wireless channel, multiple access techniques, IEEE 802 networking standard, IEEE 802.11 standard, IEEE 802.16 Standard

Cellular Wireless Networks and Wireless Internet: **[5 Hrs]**

The cellular concept, Cellular architecture, Generations of cellular systems, Wireless in local loop, Wireless ATM, Wireless internet, Mobile IP, TCP in wireless domain, WAP, Optimizing web over wireless.

Introduction To Ad-Hoc Networks: **[4 Hrs]**

Characteristics, applications, Issues in Ad Hoc wireless networks medium access protocols: design issues, Goals and classification, Contention based protocols- with reservation, Scheduling algorithms and Protocols using directional antennas.

Routing Protocols: **[5 Hrs]**

Design issues, Goals and classification. Proactive Vs reactive routing, Unicast routing algorithms, Multicast routing algorithms, hybrid routing algorithms, Energy aware routing algorithm, Hierarchical routing, QoS aware routing.

Other Features: **[5 Hrs]**

Issues in designing transport layer Ad-Hoc networks, TCP over wireless Ad-Hoc networks, Security issues in Ad-hoc networks, Secure routing protocols, Energy management in AdHoc networks, Battery management, Transmission power management and system power management.

Wireless Sensor Networks: **[4 Hrs]**

Architecture, Data dissemination, Data gathering, MAC protocols for sensor networks, Location discovery, Quality of a sensor network.

Self-study: **[12 Hrs]**

Quality of service in Ad Hoc wireless networks, Energy management in Ad Hoc wireless networks, Recent advances in wireless networks.

Textbooks

1. C. Siva Ram Murthy and B. S. Manoj, Ad hoc Wireless Networks Architectures and protocols, 2nd edition, Pearson Education. 2007
2. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan sSojmenovic, Mobile Ad-hoc Networking, Wiley-IEEE Press, 2004.

Reference Books

1. F.Zhao, L. Guibas, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, 2004
2. Mohammad Ilyas, The Handbook of Ad-hoc Wireless Networks, CRC press, 2002.
3. Research, "Wireless Commun, and Mobile Comp. Special Issue on Mobile Ad-hoc Networking Research, Trends and Applications, Vol. 2, no. 5, 2002, pp. 483 – 502. 5.
4. A survey of integrating IP mobility protocols and Mobile Ad-hoc networks, Fekri M. bduljalil and Shrikant K. Bodhe, IEEE communication Survey and tutorials, no: 12007.

Laboratory assignments

List of Assignments. Using any simulation tool demonstrates the following:

1. Create a sample wireless topology.
2. Create a mobile Ad-hoc network.
3. Implement the Ad-hoc On-demand Distance Vector protocol.
4. Implement the Transmission Control Protocol.
5. Implement the User Datagram Protocol.
6. Implement the Low Energy Adaptive Hierarchy protocol.
7. Implement the Power Efficient Gathering in Sensor Information System.
8. Implement the Sensor Protocol for Information via Negotiation (SPIN).

ADDITIONAL EXPERIMENTS:

1. Implement a Power Efficient and Delay Aware MAC
2. Implement a Predictive Wake-up MAC protocol.
3. Implement a Proactive and Reactive based MAC protocol.
4. Implement a Scheduling based protocol for WSNs.

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

(CT-DE1) System Administration

Teaching Scheme	**Evaluation Scheme**
Lectures: 2 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	
Self-study: 1 Hr/ Week	Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students will be able to:

1. Install various Linux/Unix distribution servers and desktop systems on commodities hardware, and carry out basic administration tasks of the 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.
4. Learn how to manage the Linux file system and the storage of data.
5. Provide security administration for Linux.

Course content

Basic System Administration: **[5 Hrs]**

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.

Network Administration: **[5 Hrs]**

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 Send Mail, Issues with mail services, POP and IMAP Server.

File system Administration: **[5 Hrs]**

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.

Security Administration: **[5 Hrs]**

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.

Server Administration: **[3 Hrs]**

Apache webserver configuration, Database servers: MySQL and PostgreSQL

Devices Administration: **[3 Hrs]**

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.

Self-study: **[4 Hrs]**

LDAP, Proxy servers

Textbooks

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

Laboratory assignments

1. 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.
2. 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.
3. Demonstrate use of three-tier architecture using LAMP and WAMP suite using same code base.
4. Setup a disk management system using LVM such that all options of LVM are demonstrated.
5. Write a program to browse an ext4 file system and locate data of a deleted file.
6. Setup a proxy server and firewall with set of policies to block video content in the network.
7. 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.
8. Setup an email server and demonstrate use of at least 3 email clients to send email using it.
9. Setup SE Linux.

This list is a guideline. The instructor is expected to improve it continuously

(CT-DE1) Data Visualization

Teaching Scheme	**Evaluation Scheme**
Lectures: 2 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	
Self-study: 1 Hr/ Week	Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students will be able to:

1. Understand the fundamentals of data visualization and its significance.
2. Apply data preprocessing techniques and aesthetics principles to prepare data for visualization.
3. Design appropriate visualizations to represent distributions, amounts, and proportions.
4. Analyse complex datasets using advanced visualization techniques.
5. Create and interpret dashboards using Power BI for business intelligence.

Course content

Introduction to Data Visualization: **[6 Hrs]**

Importance of Data Visualization in Data Science & Business; Data for Graphics; Design Principles; Value for Visualization; Categorical, Time Series, and Statistical Data Graphics; Introduction to Visualization Tools.

Data Pre-processing and Aesthetics: **[6 Hrs]**

Data Cleaning and Aggregation; Advanced Visuals: Pairplots, Violin Plots, Heatmaps; Interactive Visualizations; Graphical Perception Theory; Grammar of Graphics (Layering, Aesthetics, Geometries); Aesthetics and types of data; Scales map data values onto aesthetics;

Visualization Design: **[6 Hrs]**

Visualizing amounts: Bar plots; Grouped and stacked bars; Dot plots and heat maps; Visualizing distributions: Histograms and density plots, visualizing a single distribution, Visualizing multiple distributions at the same time, Empirical cumulative distribution functions and q-q plots, Empirical cumulative distribution functions, Highly skewed distributions, Quantile–quantile plots.

Advanced Visualization Design: **[6 Hrs]**

Grouped and Visualizing many distributions at once: Visualizing distributions along the vertical axis, Visualizing distributions along the horizontal axis; Visualizing proportions: A case for pie charts, side-by-side bars, stacked bars and stacked densities, Visualizing proportions separately as parts of the total; Visualizing nested proportions; Visualizing associations among two or more quantitative variables: Scatter plots, Correlograms, Dimension reduction, Paired data.

Visualizing time series, trends and geospatial data: **[6 Hrs]**

Time series: Individual time series, Multiple time series and dose–response curves, Time series of two or more response variables; Trends: Smoothing, Showing trends with a defined functional form, Detrending and time-series decomposition; Geospatial data: Projections, Layers, Choropleth mapping, Cartograms.

Power BI for Business Intelligence & Dashboarding: **[6 Hrs]**

Power BI Components: Desktop, Service, Gateway; Data Loading from Excel, SQL, Web APIs. Power Query for Data Shaping; Visuals: Cards, KPIs, Gauges, Slicers, Drill-down; DAX for Custom Measures and Calculations; Power BI Embedded and Row-Level Security.

Self-study: **[12 Hrs]**

Case Studies Projects: Supply Chain & Logistics Dashboard. Dataset: Shipment and delivery logs; Tools: Power BI or Dash; Focus: Route mapping, delivery status KPIs, filtering by region

Textbooks

1. Claus O. Wilke - Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures 1st Edition, ISBN-10- 1492031089
2. Kieran Healy-"Data Visualization: A Practical Introduction", Princeton University Press, ISBN 978-0-691-18161-5, ISBN (pbk.) 978-0-691-18162-2, 2018

Reference Books

1. Edward R. Tufte – The Visual Display of Quantitative Information
2. Alberto Cairo – The Functional Art: An Introduction to Information Graphics and Visualization

Laboratory assignments

Lab sessions will consist of practically implementing concepts studied in theory lectures.

1. Load a dataset (e.g., Iris, Titanic, Sales). Analyse various features present in dataset and create various charts like Bar Chart (categorical data), Line Chart (time series) and Box plot analysis for the given dataset.
2. Take any sample dataset clean and aggregate it, create various multivariate analysis plots to gain insight in the dataset. Compare different chart types for the same data.
3. Demonstrate complex visual designs using various plots to present proportions and relationships among various features in dataset.
4. Take any time series dataset and display data points over time allowing for the visualization of trends, cycles, variations and prepare a report for the same.
5. Mini Project for getting Business Insights Dashboard Using Visualization Techniques can be implemented using various tools like Power BI.

Above list serves as a guideline and is intended to be continuously refined by the instructor.

(MDM-02) Artificial Intelligence

(Machine Learning and Deep Learning Mastery)

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 100 Marks

Course outcomes
Students will be able to:

1. Understand Supervised Learning techniques such as Regression, Classification (Logistic Regression, SVM).
2. Apply Unsupervised Learning methods like Clustering (K-Means, DBSCAN).
3. Utilize Decision Trees and Random Forests for problem-solving.
4. Build neural networks from scratch, understanding the forward and backward passes.
5. Handle large datasets through preprocessing, data wrangling, and feature selection.
6. Execute hands-on projects using real-life datasets for ML/DL applications.
7. Convert machine learning and deep learning projects into research papers, focusing on writing impactful papers.

Course content

Machine Learning Techniques: [12 Hrs]

Supervised Learning: Regression, Classification (Logistic Regression, SVM), Unsupervised Learning: Clustering (K-Means, DBSCAN), Decision Trees and Random Forests

Deep Learning Techniques: [14 Hrs]

Build a neural network from scratch: Forward pass, Build a neural network from scratch: Backward pass, Assemble and train the entire neural network, Testing the entire neural network

Big Data Techniques: [6 Hrs]

Handling Large Datasets: Data Preprocessing, Data Wrangling, Feature Selection

Machine Learning - Deep Learning Lab: [5 Hrs]

Hands-on Projects: Real-life datasets for ML/DL applications

Projects to Research Papers: [5 Hrs]

Converting Projects into Research Papers: Writing impactful papers.

Self-Study: [12 Hrs]

Reinforcement Learning basics (MDP, Q-Learning), Semi-Supervised and Self-Supervised Learning, Transfer Learning concepts, Explainable AI (XAI) and model interpretability techniques (SHAP, LIME). Introduction to CNN, RNN, LSTM, GRU, Attention Mechanism and Transformers, and Generative Models (Autoencoders, GANs).

Textbooks

1. “Introduction to NLP and CV” by Dr. Raj Dandekar, Dr. Rajat Dandekar, Dr. Sreedath Panat. VizuaraAILabsorg

Reference Books

1. “Natural Language Processing with Python” by StevenBird. O’Reilly
2. JayAlammarblogs, JayAlammar. Github, Youtube

Laboratory assignments
Suggested List of Assignments:

1. Perform data preprocessing and feature selection on a real-world large dataset and document each step clearly.
2. Implement Linear Regression from scratch and evaluate its performance using MSE and R^2 score.
3. Develop a Logistic Regression model for binary classification and analyze results using confusion matrix and F1-score.
4. Train and evaluate a Support Vector Machine model using linear and RBF kernels.
5. Implement K-Means clustering from scratch and visualize the cluster formation.
6. Apply DBSCAN clustering and compare its performance with K-Means using cluster evaluation metrics.
7. Build a Decision Tree classifier and study overfitting and feature importance.
8. Implement a Random Forest model and compare its accuracy with Decision Tree.
9. Design a feedforward neural network and perform forward propagation using NumPy.
10. Implement backpropagation and train the neural network using gradient descent.
11. Develop an end-to-end Machine Learning or Deep Learning mini project using a real-life dataset.

(MDM-02) – Quantum Computing

Teaching Scheme	**Evaluation Scheme**
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	
Self-study: 1 Hr/ Week	Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students should be able to,

11. Analyze simple states of superposition and the effect of doing the measurement in different basis states.
12. Design and analyze quantum circuits with single and two-qubit gates.
13. Equip students with the linear algebra background required for this course.
14. Analyze quantum circuits with superposition and entanglement.
15. Analyze quantum algorithms and complexity.
16. Understand QAOA and error correction scheme.

Course content

Course Introduction: **[2 Hrs]**

Comparison about classical physics and Quantum Physics, Quantum Mechanics and computations.

Superposition and Single Qubit: **[7 Hrs]**

Superposition, Polarization of light, Single qubit notation, Measurement of Qubit, BB84 Quantum Key Distribution, Bloch Sphere Notations, Q-sphere, Difference between Bloch and Q-sphere.

Quantum Gates and Circuits: **[7 Hrs]**

Model of computation (movement on Bloch Sphere) X, Y, Z, H gates CNOT, Toffoli, Fredkin SWAP gate Simple circuits Quantum Adder Reversible circuits, Basics of quantum circuit design.

Basics of Linear Algebra: **[7 Hrs]**

Dirac Notation Vectors Complex Conjugate & Norm Analysing Pauli gates Analysing Cascade of gates Analysing Two-qubit gates Tensor Product (example), Introduction of Three qubit gates.

Superposition and Entanglement: **[5 Hrs]**

Introduction of quantum mechanics, Quantum state formation, Entangled States, Testing for Entangled States, Bell Pair and Bell States EPR Paradox & Bell Theorem Conditional Instructions Quantum Teleportation Superdense Coding.

Quantum Algorithms: **[5 Hrs]**

Deutsch Deutsch-Jozsa Bernstein, Vazirani Grover, Simon’s Algorithm, Period Finding, Shor’s Algorithm, QFT (Basics).

QAOA and Error Correction **[6 Hrs]**

Introduction of Quantum Approximate Optimization Algorithm (QAOA), Maxcut problem, Overview of QAOA Optimizations for QAOA, what is Quantum errors, analyse the simple error correction scheme, Unique challenges in QEC Shor’s bit-flip code Shor’s phase-flip code Shor 9-qubit code Steane code Concatenation code.

Self-Study: **[12 Hrs]**

Explore advanced quantum algorithms like Grover's, Shor's, Deutsch-Jozsa, and Simon's, and practice analyzing quantum circuits with entanglement and superposition. Study Quantum Approximate Optimization Algorithm (QAOA), quantum error correction codes (Shor, Steane), and experiment with single- and multi-qubit gate operations.

Textbooks

1. P. Wittek, Quantum Computing. Education, 2007
2. Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University press, 2007.
3. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020

Reference Books

1. "Quantum Computing: A Gentle Introduction" by Eleanor Rieffel and Wolfgang Polak, MIT press (2014)
2. Quantum Computation and Quantum Information, M. A. Nielsen &I. Chuang, Cambridge University Press (2013)

Laboratory assignments

1. Practical 1: Introduction of Available Quantum Platforms and Programming languages Contents— IBM Quantum Platform, Quantum Cloud Connects, Local machine connections, AWS quantum, Google quantum, Quantum Technologies. Case study on various quantum technologies
2. Practical 2: Installation of Qiskit Platform Contents Qiskit installation via local classical computer, Installation of Miniconda/Anaconda v.0, Install IBM quantum platform through qiskit. Learn the steps of qiskit installation and IBM quantum platform to access quantum hardware through local computer desktop.
3. Practical 3: Qubit Visualization/Bloch Sphere Single Qubit State Contents—Basic Rotations on One Qubit and Measurements on the Bloch Sphere Design following single qubit states.
   1. Bit Flip
   2. $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
   3. $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
4. Practical 4: Quantum Circuits Using Multi Contents— Design and Model of computation (movement on Bloch Sphere) X, Y, Z, H gates CNOT, Toffoli, Fredkin SWAP gate Simple circuits Quantum Adder Reversible circuits, Basics of quantum circuit design. Construct the Bell state:

$$|\Phi^{+}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

5. Practical 5: Write a function that builds a quantum circuit on 3 qubits and creates the GHZlike state:

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle):$$

6. Practical 6: Execution of task on Real Quantum hardware.

   Contents — Map the problem to a quantum- using a quantum primitive function, Analyze the results. Python code for connecting various quantum hardware, Check availability of hardware, Search for less busy hardware.
7. Practical 7: Execute simple “hello world” Circuit on /quantum computer.
8. Practical 8: Implementation of Quantum Algorithms Contents— Deutsch Deutsch-Jozsa Bernstein, Vazirani Grover, Simon’s Algorithm, Period Finding, Shor’s Algorithm, QFT (Basics)

   Design and implement Oracles and the Deutsch -Jozsa algorithm
9. Practical 9: Design quantum teleportation.
10. Practical 10: Design and analyse circuit for Grover search.

(MDM-02) – Object Oriented Programming

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Students should be able to,

1. Design a class hierarchy using object oriented thinking for a given problem.
2. Create object oriented application code for a given problem.
3. Write small pieces of code demonstrating various object oriented programming concepts.
4. Compare, annotate, and comment on various object oriented programming concepts.
5. Demonstrate ability to write concurrent programs for given problems.
6. Apply different UML diagrams for given problems.

Course content

Introduction: **[8 Hrs]**

Various programming paradigms: Procedural, object-oriented, logic and functional, concurrent programming. Classes, Objects, Methods; Data types provided by OO languages; Abstract Data Types

Abstraction Mechanisms: **[8 Hrs]**

Encapsulation; Constructors, Destructors; Polymorphism; Access specification: Private, public, protected members.

Inheritance: **[6 Hrs]**

Types of inheritance: single, multilevel, multiple, hierarchical and hybrid; Class hierarchies; Virtual Functions; Virtual Base class.

Standard Libraries: **[6 Hrs]**

Templates; Generic Programming using generic function and 6class; Packages; Interfaces. Iterators; Containers

Exception Handling & Multi-Threaded Programming: **[6 Hrs]**

Exception handling, Exception types, Concurrent Programming, Basic Concepts of Concurrent Programming, Threads.

Introduction to OOAD: **[5 Hrs]**

Unified Process – UML diagram, use case, class diagrams, interaction diagrams, state diagrams – activity diagrams – package, component and deployment diagrams.
Design: Unified Modelling language; use case diagrams; Class Diagrams

Self-Study: **[12 Hrs]**

Introduction to Design Patterns, Types of Design Patterns, Application on various design patterns

Textbooks

1. Cay S Horstmann and Gary Cornell, Core Java Vol-1 and Vol-2, 9th Edition, Pearson Education India, ISBN-10: 9332518904 and 9332518890
2. Kenneth Barclay and John Savage, Object-Oriented Design with UML and Java, Elsevier Science. ISBN:

9780080497556

3. Ronald Leach, Object-Oriented Design and Programming with C++: Your Hands-On Guide to C++, Academic Press, ISBN: 9781483214122
4. M.Ben Ari, Principles of Concurrent Programming, Prentice-Hall International, ISBN:0137010788

Reference Books

1. Herbert Schilt, “JAVA Complete Reference”, 7th Edition, Tata McGraw Hill, ISBN: 9780070636774
2. Scott W. Ambler, The Elements of UML (TM) 2.0 Style, ISBN- 978-0521616782
3. Sharon Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser, Mark Hoeber, “The Java Tutorial” Addison Wesley Professional, 2006, Print ISBN-10: 0-321-33420-5
4. Eckel B., "Thinking in Java", 3rd Edition, Pearson Education, 2012
5. E. Balagurusamy, Object Oriented Programming with C++, 6th Edition, McGraw Hill, ISBN-10: 125902993X

Laboratory assignments

Suggested List of Assignments:

1. Define a class to represent a bank account. Include the following details like name of

   the depositor, account number, type of account, balance amount in the account. Write methods to assigr initial values, to deposit an amount, withdraw an amount after checking the balance, to display name account number, account type and balance.
2. Write a program to implement following. Create a base class called person consisting

   of name and code. Create 2 child classes a. Account with member pay and b. Admin with experience anc inherit the base class. Create a class Employee with name, code, experience and pay by inheriting the above class.
   a. Write Python script to display
   b. Current date and time,
   c. Current year,
   d. Month of year,
   e. Week number of the year,
   f. Weekend of the week,
   g. Day of year,
   h. Day of the month and Day of week.
3. Write a program that has an abstract class Polygon. Derive two classes Rectangle and Triangle fron Polygon and write methods to get details of their dimensions hence calculate the area.
4. Write a menu driven to read, add, subtract, multiply, divide and transpose two matrices.
5. Write a program that has a class TIME. Enter the time when a user started an online test and completec the test. Subtract the two-time values and display the duration in which the test was completed. Throv exceptions whenever need arises (like invalid data, or if start time is greater than completion time).

6. Construct a class diagram for a geometrical document. Add at least 10 relationships (associations anc generalizations). Use associations and association end names whereever required. Also use qualifiec associations and show multiplicity. You do not need to show attributes or operations. As you prepare the diagrams, you may add classes. Be sure to explain your diagrams.
7. Scenario: An extension ladder has a rope, pulley, and latch for raising, lowering, and locking the ex-tension. When the latch is locked, the extension is mechanically supported and you may safely climk the ladder. To release the latch, you raise the extension slightly with the rope. You may then freely raise o lower the extension. The latch produces a clacking sound as it passes over rungs of the ladder. The latch may be reengaged while raising the extension by reversing direction just as the latch is passing a rung Construct a state diagram of an extension ladder.
8. Draw a Scenario: A hockey league is made up of at least four hockey teams. Each hockey team is composed of six to twelve players, and one player captains the team. A team has a name and a record Players have a number and a position. Hockey teams play games against each other. Each game has a score and a location. Teams are sometimes lead by a coach. A coach has a level of accreditation and a number of years of experience, and can coach multiple teams. Coaches and players are people, anc people have names and addresses. Construct a UML Class Diagram representing the above problem domain for a hockey league.

This is only a suggestive list. Instructor is encouraged to update the list of assignments and projects. The purpose of the assignments should be to lead to a meaningful project. The students should be encouraged to build different projects.

(PCC) Computer Networks

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will able to:

1. Examine the challenges associated with each layer of the TCP/IP stack and critically assess various application layer protocols utilized on the Internet.
2. Gain a comprehensive understanding of the transport layer's roles, focusing on TCP and UDP protocols, mechanisms for reliable data transfer, and strategies for congestion and flow control.
3. Develop network designs incorporating subnetting as per given specifications, and understand the key network layer protocols employed in the Internet.
4. Understand link layer protocols and core concepts of wireless networking and Ethernet switching and trace the complete end-to-end journey of packets across the Internet.
5. Evaluate the challenges specific to wireless networks and the protocols designed to address them.

Course content

Introduction to Computer Networks and Internet: [4 Hrs]

Physical Mediums, Concept of Network Protocols, Layered Architecture, network core – Packet Switching vs Circuit Switching, Various Losses in the Internet, ISPs, IXPs, Routing, Heterogeneous Networks, Brief History of Computer Networks.

Application Layer: [7 Hrs]

Basics of Socket Programming, Transport Layer Programming Interface (TCP, UDP), Protocols: HTTP (Overview, Persistent and Non-Persistent, Message Format, Cookies, Cachess), SMTP (Overview, Message Formats), IMAP, POP, DNS; FTP; Telnet, SSH; Peer-to-Peer Applications.

Transport Layer: [4 Hrs]

Relationship Between Transport and Network Layer, TCP and UDP; Multiplexing and Demultiplexing; Principles of Reliable Data Transfer; Go-Back-N and Selective Repeat; TCP: Segment Structure, Round Trip Time Estimation, Reliable Data Transfer, State Transitions, Flow Control, Congestion Control, Fairness; UDP: Segment Structure

Network Layer, Subnets: [4 Hrs]

Concept of IP Address, Netmask, Subnet; CIDR; Design of a LAN and WAN; Subnetting Problems.

Network Layer, Routers, IPv4, IPv6: [8 Hrs]

Functions of a Router: Forwarding and Routing; Inside of a Router: Port Processing, Switching; IPV4: Datagram Format, Fragmentation; Network Address Translation; IPv6 Introduction; Multicasting. Network Layer, Routing algorithms: Link State, Distance Vector Routing; OSPF, BGP, RIP; Routing Policies.

Link Layer: **[6 Hrs]**

Review of fundamentals of link layer protocols; Ethernet Switches, LANs, Link Layer Switches, Complete tracking of traversal of a packet over internet between two applications.

Wireless Networks: **[6 Hrs]**

Wireless Links and Network Characteristics, Bit Error Rate, SNR; Problems of Wireless Links (Interference, Signal Strength fading, Multipath Propagation, Hidden Terminal problem), 802.11 (Architecture, MAC protocol, Frame Structure).

Self-study: **[12 Hrs]**

BitTorrent Protocol; Content Distribution Networks, Queueing: Causes, Delays; Transport layer implementation Overview in OS Kernels; VLANs; Routing Protocols for Ad Hoc Wireless Networks.

Textbooks

1. J.F. Kurose and K. W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson, ISBN-13: 9780201976991
2. Behrouz A. Forouzan, Firouz Mosharraf, Computer Networks: A TopDown Approach, Tata McGraw-Hill Education Pvt. Ltd, ISBN 10: 1259001563 / ISBN 13: 9781259001567

Reference Books

1. Larry Peterson Bruce Davie, Computer Networks A Systems Approach, Elsevier, ISBN: 9780123850591
2. Behrouz A. Forouzan, Firouz Mosharraf, Computer Networks: A TopDown Approach, Tata McGraw-Hill Education Pvt. Ltd, ISBN 10: 1259001563 / ISBN 13: 9781259001567
3. Kevin R. Fall, W. Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols, Pearson, ISBN-13: 978-0321336316/ISBN-10: 0321336313
4. Behrouz Forouzan, Data Communications and Networking, Tata McGraw-Hill, ISBN 13: 978-0073250328/ISBN-10: 0073250325
5. William Stallings, "Data and computer Communication", Pearson Education, ISBN-81 297-0206-1
6. Alberto Leon Garcia and Indra Widjaja, "Communication Networks, Fundamental Concepts, and Key Architectures", Tata McGraw-Hill, ISBN-10: 007246352X
7. Peter Loshin, IPv6 Theory, Protocol, and Practice, Elsevier, ISBN: 9781558608108

Laboratory assignments

1. Implement a TCP-IP client-server application. The server accepts a string as a request and sends the capitalized version back.
2. Configure a network on your computer using DHCP and static IP. Learn the following networking commands and configuration files: ifconfig, ping, host, traceroute, Telnet, netstat, nslookup, ssh, scp, wget, /etc/hosts, /etc/network/interfaces.
3. Install and demonstrate the following networking applications: web server (apache2 or nginx), ftp server,

ssh server, email client.

4. Access the website http://go.com (or any website on HTTP, not HTTPS). Use Wireshark to inspect the flow of packets for a GET request using a browser. Identify all packets for this request. Capture the data from all packets and construct the HTML page. Trace the TCP sequence numbers and verify that they match the expected behaviour.
5. Send an email using SMTP over the server at new.toad.com (an open SMTP server).
6. Implement a packet sniffer to capture packets of a specified link layer, network layer, transport layer, or application layer protocol.
7. Write a program to simulate either Go-Back-N or Selective Repeat protocols.
8. Implement a given subnet using the ns3 simulator and demonstrate the flow of packets.
9. Critically analyse The COEP network design.
10. Course Mini project: Implement any one of these: a web server (HTTP Protocol), email client (IMAP and POP protocols), a BitTorrent client, DNS server and client (like nslookup), etc, simulations of transport layer/network layer protocols; Adding features/bug fixes in apache2/nginx like projects.

This list is a guideline. The instructor is expected to improve it continuously.

(PCC) Design and Analysis of Algorithms

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will able to:

1. Understand the fundamental principles of algorithm design and analysis, including time and space complexity.
2. Apply asymptotic notation to evaluate and compare the performance of algorithms
3. Explain standard algorithm design techniques and analyze their performance.
4. Analyze the correctness and efficiency of algorithms, including worst-case, average-case, and best-case analysis.
5. Formalize computational problems, apply reductions, and evaluate problem complexity within established classes.

Course content

Introduction: **[6 Hrs]**

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, Amortized analysis.

Design Techniques-I: **[7 Hrs]**

Divide and Conquer: Quicksort, Mergesort, Strassen's matrix multiplication; Greedy Algorithms: Knapsack problem, Job sequencing with deadlines, Optimal merge patterns, Single source shortest paths.

Design Techniques-II: **[8 Hrs]**

Dynamic Programming: All pairs shortest paths, 0-1 Knapsack, Traveling salesperson problem, Chained matrix multiplication, Longest common subsequence, Bellman-Ford algorithm, Subset sum.

Design Techniques-III: **[7 Hrs]**

Backtracking: N-queens problem, Hamiltonian cycles, Graph Coloring; Branch-and-Bound: 0/1 Knapsack problem, Traveling salesperson problem.

Selected Algorithms from various areas: **[6 Hrs]**

String Matching: The naïve string-matching algorithm, The Robin-Karp algorithm, The Knuth- Morris-Pratt algorithm; Network Flow Algorithms: Ford-Fulkerson algorithm, Push-relabel algorithm

Complexity Theory: **[6 Hrs]**

Polynomial time, NP-hard and NP-complete problems, proving NP completeness using reduction technique (e.g. SAT, Independent Set, 3VC, etc.)

Self-study: **[12 Hrs]**

Sorting in linear time, Elementary graph algorithms, Minimum spanning tree, Number -Theoretic algorithms: GCD algorithm, Chinese remainder theorem, Primality testing.

Textbooks

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press, 2nd edition (2008), ISBN-13: 978- 8173716126
2. Thomas Cormen, Charles Leiserson, Ronald Rivest and Cliford Stein, “Introduction to Algorithms”, PHI, 3rd edition, ISBN-13: 978- 8120340077

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, ISBN13: 978-9332518643

Laboratory assignments

Lab sessions will consist of solving numerical problems based on the algorithms discussed in theory lectures, discussions around finding complexities of algorithms similar to ones studied in the theory lectures.

1. Apply reduction techniques to demonstrate that specific problems are NP-Complete.
2. Study and solve recurrence relations using methods such as substitution, recursion tree, and the Master Theorem.
3. Trace and analyze the execution of sorting algorithms: Heap Sort, Quick Sort, and Merge Sort.
4. Implement greedy algorithms: Fractional Knapsack, Job Sequencing with Deadlines, Huffman Coding, and Optimal Merge Pattern.
5. Develop programs for graph algorithms: Single-source shortest paths using Dijkstra’s algorithm, Minimum Spanning Tree using Prim’s and Kruskal’s algorithms.
6. Implement dynamic programming solutions for: Matrix Chain Multiplication, Longest Common Subsequence (LCS), 0/1 Knapsack, All-Pairs Shortest Paths (Floyd-Warshall) and Bellman-Ford algorithm.
7. Implement string matching using Naive and Knuth-Morris-Pratt (KMP) algorithms.
8. Apply the branch-and-bound technique to solve problems such as 0/1 Knapsack and the Travelling Salesperson Problem (TSP).
9. Use backtracking techniques to solve: N-Queens Problem, Graph Coloring, Hamiltonian Path, Travelling Salesperson Problem.
10. Implement and analyze network flow algorithms such as Ford-Fulkerson and Push-Relabel.
11. Evaluate the amortized complexity of given data structure operations or algorithms using amortized analysis.

This list serves as a guideline and is intended to be continuously refined by the instructor.

(PCC) Artificial Intelligence

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will be able to:
1. Compare AI with human intelligence and traditional information processing and discuss its strength and limitations.
2. Apply the basic principles, models and algorithms of AI to recognize, model and solve problems
3. Demonstrate knowledge of basics of the theory and practice of Artificial Intelligence.
4. Design and carry out an empirical evaluation of different algorithms on problem formalization and state the conclusions that the evaluation supports.
5. Apply AI techniques and problem-solving strategies to common AI applications.

Course content

Introduction: [6 Hrs]

What is AI, History, AI problems, Production Systems, Problem characteristics, Intelligent Agents, Agent Architecture, AI Application (E-Commerce, & Medicine), AI Representation, Properties of internal representation, Future scope of AI, Issues in design of search algorithms.

Heuristic search techniques: [6 Hrs]

Heuristic search, Hill Climbing, best first search, mean and end analysis, Constraint Satisfaction, A* and AO* Algorithm, Knowledge Representation: Basic concepts, Knowledge representation Paradigms, Structured representation of knowledge, ISA hierarchy, Frame notation, Resolution

Learning & Planning: [8 Hrs]

What is Learning, Types of Learning (Rote, Direct instruction Analogy, Induction, Deduction), Planning: Block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, Least commitment strategy.

Advanced AI Topics: [8 Hrs]

Game playing: Min-max search procedure, Alpha beta cutoffs, waiting for Quiescence, Secondary search, Natural Language Processing (NLP): Introduction, Steps in NLP, Syntactic Processing, Semantic analysis, Discourse & Pragmatic Processing. Perception and Action: Perception, Action, Robot Architecture, Machine Learning: Definition of Learning Systems. Goals and applications of machine learning. Aspects of developing a learning system: training data, concept representation, function approximation.

Neural Networks and Deep Learning: [8 Hrs]

Neurons and biological motivation. Linear threshold units. Perceptrons: representational limitation and gradient descent training. Multilayer networks and backpropagation, Hidden layers and constructing intermediate, distributed representations, Overfitting, learning network structure, Convolutional Neural Networks (CNNs) for image processing, Recurrent Neural Networks (RNNs) for sequential data

AI Ethics, Future Trends, Applications and Case Studies: [6 Hrs]

Bias and fairness in AI, Ethical considerations in AI deployment, Future trends: Explainable AI, AI in healthcare, autonomous systems, AI in robotics, computer vision, and speech recognition, Case studies: Self-driving cars, recommendation systems, fraud detection.

Self-Study: **[12 Hrs]**

Textbooks

1. Elaine Rich and Kerin Knight, Artificial Intelligence, 3rd Edition, McGraw Hill. ISBN13: 9780070087705
2. Eugene, Charniak, Drew Mcdermott, Introduction to artificial intelligence, AddisonWesley. ISBN 0-07-052263-4.

Reference Books

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 3rd Edition. ISBN 0-13-103805-2.
2. Tom Mitchell, Machine Learning, McGraw Hill. ISBN-10: 1259096955
3. Herbert A. Simon, The Sciences of the Artificial, MIT Press, 3rd Edition, 1998. ISBN: 9780262190510.
4. George F Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Edu., 4th Edition. ISBN-13: 978-0-321-54589-3.
5. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016. ISBN: 0262035618.
6. Reinforcement Learning: An Introduction – Richard Sutton, Andrew Barto

Laboratory assignments

Suggested List of Assignments:

1. Implement A* algorithm.
2. Implement AO* algorithm.
3. Implementation of other Searching algorithms.
4. Implementation of Min/MAX search procedure for game Playing.
5. Implementation of variants of Min/ Max search procedure.
6. Implementation of mini-Project using the concepts studied in the AI course.

This list is a guideline. The instructor is free to assign new assignments. This list is a guideline. The instructor is expected to improve it continuously.

(RM) Research Methodology

Teaching Scheme
Lectures: 2 Hrs/ Week
Self-study: 1 Hrs/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Explain the basic concepts of research, research process, and research problems.
2. Identify and differentiate types of research and research designs.
3. Apply sampling methods and data collection techniques for research.
4. Prepare research reports and follow ethical practices in research.

Course content

Introduction to Research **[6 Hrs]**

The concept of research, characteristics of good research, Application of Research, Meaning and sources of Research problem, characteristics of good Research problem, Research process, outcomes, application of Research, Meaning and types of Research hypothesis, Importance of Review of Literature, Organizing the Review of Literature.

Types of Research and Research Design **[6 Hrs]**

Types of research and applied; qualitative and quantitative. Meaning and need of research design; types of research design. Components and features of a good research design; experimental, survey, and case study research designs.

Sampling, Data Collection and analysis **[6 Hrs]**

Types and sources of data, Primary and Secondary data. Methods of data collection. Concept of sampling, sampling frame, sample, characteristics of a good sample, and sampling. Applications of statistics in research, measures of central tendency and measures of dispersion.

Research Report **[6 Hrs]**

Research report and its structure, journal articles, Components of journal article. Explanation of various components. Structure of an abstract and keywords. Thesis and dissertations. components of thesis and dissertations. Referencing styles and bibliography

Self-study: Ethics in Research **[12 Hrs]**

Plagiarism - Definition, different forms, consequences, unintentional plagiarism, copyright infringement, collaborative work. Qualities of good Researcher.

Textbooks

1. Kothari, C. R. (2004). Research Methodology: Methods and Techniques (2nd ed.). New Age International Publishers, New Delhi.

Reference Books

1. Montgomery, D. C. (2017). Design and Analysis of Experiments (9th ed.). Wiley, New York.
2. Creswell, J. W., & Creswell, J. D. (2018). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (5th ed.). SAGE Publications.
3. Walliman, N. (2017). Research Methods: The Basics (2nd ed.). Routledge, London.

4. Leedy, P. D., & Ormrod, J. E. (2015). Practical Research: Planning and Design (11th ed.). Pearson Education.
5. Kerlinger, F. N., & Lee, H. B. (2000). Foundations of Behavioral Research (4th ed.). Harcourt College Publishers.
6. Goddard, W., & Melville, S. (2004). Research Methodology: An Introduction. Juta & Company Ltd.

Web references

1. https://shodhganga.inflibnet.ac.in
2. https://iksindia.org/
3. https://ieeexplore.ieee.org/
4. https://shodhgangotri.inflibnet.ac.in/

(MDM) – Quantum Computing

(Quantum Algorithms and Applications)

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: CIE: 25 Marks, ESE: 25 Marks

Pre-requisite: Quantum Computing, Machine learning, Cryptography, Material Science
Course Objectives: Graduate will be able to understand the core quantum algorithms and their application across disciplines.

Course outcomes
Students will able to:
1. Demonstrate the need for core quantum algorithms.
2. Apply quantum algorithms to analyse various Machine Learning subroutines, algorithms.
3. Understand the pros and cons of Quantum Cryptography
4. Integrate quantum machine learning to investigate new approach in material science discipline.
5. Compare the classical complexities in various disciplines such as ML, Cryptography and Material Science.

Course content

Course Introduction: **[2 Hrs]**

This course covers quantum machine learning, Cryptography, Material Science, which accelerates and improves machine learning performed on the "classical" computers we use on a daily basis by utilizing the capabilities of quantum computing and quantum physics.

Introduction of Quantum Algorithms: **[5 Hrs]**

Quantum Computing basics, Algorithm Basics, Quantum Algorithms: Deutsch-Jozsa, Simon, Grover, Shor, and their cryptanalytic implications. Implication of Grover’s and Simon’s algorithms towards classical symmetric key cryptosystems, Implication of Shor’s algorithm towards factorization and Discrete Logarithm based classical public key cryptosystems.

Classical ML vs. Quantum Machine Learning: **[8 Hrs]**

Introductory ideas about Turing machine, computational complexity. Basics of classical machine learning, Machine Learning terminologies, Introduction to different strands of quantum machine learning. Introduces feature vector, data encoding, some general property of Hadamard operation, measuring output.

Quantum Encoding, Classification and Clustering: **[5 Hrs]**

Algorithm for amplitude encoding of data in superposition, quantum random access memory (QRAM), Quantum variational circuits, quantum parameterized circuits with various gates, ansatz, Feature maps, Quantum encoding techniques, Quantum Kernels. Building Quantum Classifiers, uploading data in quantum states, quantum feature map design, quantum classification, quantum clustering, preprocessing models, training classical model, training quantum model, handling sampling noise, selection of appropriate feature map, QSVM, QK-mean

Variational Classifiers to Linear Classifiers in QML: **[4 Hrs]**

Introduces the classical concepts of linear classifier, the two formulations of its optimisation function, along with the widely used Kernel trick. Extends the idea of the classical kernel trick to a quantum version, first explaining the concept of quantum feature maps and applying them in introducing a model called Variational Quantum Classifier

(VQC). VQC is indeed a linear classifier, kernel trick use in a VQC; quantum kernel estimator (QKE).

Introduction to Quantum Cryptography: **[7 Hrs]**

QKD (Quantum Key Distribution), BB84, Ekert, Semi-Quantum QKD protocols and their variations, Issues of Device Independence, Other cryptologic issues, such as Quantum secret sharing and multiparty computation, Introductory topics in Post-Quantum Cryptography.

Quantum Material Science: **[8 Hrs]**

Discovery of new materials, Chemical compounds with specific properties, such as high thermal conductivity or superconductivity, QML for Chemical Compounds analysis, Case study on Computational complexities in prediction with classical computing methods in material science discipline.

Self-Study: QC Quantum Algorithms and Applications: **[12 Hrs]**

Explore why oracle separations matter, compare classical Monte Carlo vs quantum speedup, Implement advanced algorithms using Qiskit/Cirq, Analyze data loading (QRAM) challenges, Study quantum speedups in kernel methods, Identify open problems in quantum algorithm design

Lab- Quantum Algorithm using Simulation

Unitary matrices of quantum gates, Bloch sphere visualization, Tensor product structure, Product vs entangled states, Schmidt decomposition (conceptual), Projective measurements, Density matrices, Kraus operators, T1, T2 times and physical decoherence, Phase kickback, Limitations of HHL on NISQ devices, Parameterized quantum circuits, Cost landscapes & barren plateaus, Reproducibility in quantum experiments, Benchmarking NISQ algorithms

Textbooks

1. Quantum Supervised Machine Learning (Quantum Science and Technology), Schuld, Maria, Petruccione Francesco
2. Book on Quantum Cryptography by Vidick and Wehner
3. Quantum Mechanics for Material Science, by Gianluca Stefanucci, Springer Link: Quantum Mechanics for Material Science: An Introduction | SpringerLink

Reference Books

1. P. Wittek, Quantum Machine Learning. Education, 2007
2. Quantum Computation and Quantum Information, Michael Nielsen and Isaac Chuang, (Cambridge Univ Press, 2010)
3. Machine Learning with Quantum Computers, Maria Schuld, Francesco Petruccione, (Springer,2021)

Laboratory assignments

Pre-requisite: Quantum Algorithms and QML, Database management

Course Objectives: Learners will be able to design and implement practical solution for various real-world complex problem using quantum mechanism.

1. Course Introduction: This course helps learners to find classically challenging problems to find quantum solution for it using simulators.
2. Introduction to Variational Algorithms and Circuits: Variational quantum algorithms, in particular Variational Quantum Eigen solvers, how to create and work with parameterized circuits and quadratic programs in Qiskit, how to solve optimization problems using QAOA
   - Practical 1: Construct parameterized quantum circuits and measure performance.
   - Practical 2: Construct parameterized quantum circuits and assign the values and perform Pauli rotations on parameters.
3. Training a Classical Model on a Real Dataset: Select dataset, Upload dataset in qiskit platform, Exploratory data analysis as preprocessing, Training a Classical Machine Learning Model,

- Practical 3: Download dataset of your choice and perform exploratory data analysis on it. (Example datasets: Iris, Brest cancer, Digits.

4. Data Encoding and Quantum Feature map: Quantum parameterized circuits with various gates, ansatz, Feature maps, Quantum encoding techniques such as zfeaturemap, zzfeaturemap, paulifeatureMap, Understand the quantum advantages on data using featuremaps with various conditions.
   - Practical 4: Design zfeaturemap on given dataset features and display quantum circuit for that.
   - Practical 5: Data Encoding Encode the data point x= (−0.1,0.2) using the `ZZFeatureMap` with 4 repetitions and default data mapping function.
   - Practical 6: Obtain quantum circuits using zzfeaturemap on given dataset for linear, circular and block entanglement.
5. Training a Quantum Models, Classification and Clustering: Quantum variational circuits, Building Quantum Classifiers, use objective functions against iteration, Quantum kernels and estimation, QSVM, Pegasos QSVM, QK-mean.
   - Practical 7: Set Up the Quantum Kernel and Quantum Kernel Trainer, Train the Quantum Kernel, Fit and Test the Model, Visualize the Kernel Training Process. Obtain QSVM results.
6. Case study on Quantum Cryptography: Introduction of QKD (Quantum Key Distribution), BB84.
   - Practical 8: Case study on quantum key distribution. QKD and bb84 protocol simulation using qiskit
7. Case study on challenges in material science: Introduction for various classically challenging tasks in material science discipline.
   - Practical 9: Simulate given chemical component using qiskit.
   - Practical 10: Develop a problem statement based on the difficulties in material science engineering after consulting with subject-matter experts.

(MDM-03) -Artificial Intelligence

(Natural Language Processing (NLP) and Computer Vision (CV) Mastery, Capstone Project)

Teaching Scheme	**Evaluation Scheme**
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	
Self-study: 1 Hr/ Week	Laboratory: CIE: 100 Marks

Course outcomes

Students will able to:

1. Understand basic Natural Language Processing (NLP) techniques, including text preprocessing, tokenization, and parsing.
2. Apply various NLP techniques like N-grams, TF-IDF, and Word Embeddings (Word2Vec, GloVe).
3. Explore advanced language models such as RNNs, LSTMs, and Transformers for NLP applications.
4. Understand fundamental Computer Vision (CV) techniques, including image preprocessing and feature extraction.
5. Apply advanced CV techniques like Convolutional Neural Networks (CNN), Transfer Learning, and Object Detection.
6. Explore cutting-edge topics in CV like Generative Adversarial Networks (GANs) and Vision Transformers.
7. Implement NLP and CV models using TensorFlow and Keras in hands-on lab projects.
8. Work on an industrial capstone project involving real-world NLP or CV applications.

Course content

Natural Language Processing (NLP) Techniques: [18 Hrs]

Introduction to NLP: Text Preprocessing, Tokenization, and Parsing, NLP Techniques: N-grams, TF-IDF, Word Embeddings (Word2Vec, GloVe), Language Models: RNN, LSTM, Transformer.

Computer Vision (CV) Techniques: [18 Hrs]

Introduction to Computer Vision: Image Preprocessing, Feature Extraction, CV Techniques: CNN, Transfer Learning, Object Detection, Advanced Topics: GANs for Image Generation, Vision, Transformers

NLP & CV Lab: [6 Hrs]

Hands-on Lab: Implementing NLP and CV Models using TensorFlow, Keras.

Capstone Project: [6 Hrs]

Industrial Capstone Project: Implementing a Real-world NLP or CV Project

Self-Study: [12 Hrs]

Explore advanced NLP techniques like BERT, GPT, Word Embeddings, and Transformers, and practice text analysis, sentiment analysis, and language modeling. Study advanced Computer Vision topics such as CNN architectures, Transfer Learning, Object Detection, GANs, and Vision Transformers with practical experiments.

Textbooks

1. “Introduction to NLP and CV” by Dr. Raj Dandekar, Dr. Rajat Dandekar, Dr. Sreedath Panat. VizuaraAILabsorg

Reference Books

1. "Natural Language Processing with Python" by StevenBird, O'Reilly
2. JayAlammarblogs, JayAlammar, Github,Youtube

Laboratory assignments

Suggested List of Assignments:

1. Clean a text dataset by removing punctuation, numbers, and stopwords.
2. Break text into words using tokenization and show the sentence structure.
3. Create N-grams from text and count their frequency.
4. Calculate TF-IDF scores for words in a text dataset.
5. Generate Word2Vec or GloVe embeddings and find similar words.
6. Build a simple RNN model to predict the next word in a sentence.
7. Use LSTM to classify text into categories like positive or negative.
8. Apply a pretrained Transformer model for text classification.
9. Load images and apply preprocessing like resizing and normalization.
10. Build a basic CNN to classify images into different categories.
11. Use Transfer Learning with a pretrained model (VGG16/ResNet) on your images.
12. Do a small project combining NLP or CV and show results with graphs or predictions.

(MDM-03) – Fundamentals of Database Management Systems

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
Students will be able to:
1. Identify and describe various components of DBMS.
2. Construct Entity-Relationship Model for given applications and demonstrate it as a relational model.
3. Design optimal (SQL) query statement.
4. Apply normalization to database design.
5. Improve efficiency in data retrieval using indexing and solve analytical problems on serializability, concurrency and recovery.

Course content

Introduction: **[7 Hrs]**

Basic Concepts; Database system application, purpose of database systems, view of data, database languages; Database architecture: components of DBMS and overall structure of DBMS; Various types of databases.

Relational Algebra And SQL: **[7 Hrs]**

Relational algebra: fundamental relational algebra operations; SQL: basic structure and operations, aggregate functions, nested subqueries, complex queries.

E-R and Relational Model: **[8 Hrs]**

Database design; E-R model: modelling, entity, attributes, relationships, constraints, components of E-R model; Relational model: basic concepts, attributes and domains, concept of integrity and referential constraints, schema diagram.

Relational Database Design: **[6 Hrs]**

Theory of normalization (1NF to 3NF).

Indexing and Hashing: **[6 Hrs]**

Basic concepts, Ordered Indices; Indices: concepts, B+ trees.

Transactions and Concurrency control: **[7 Hrs]**

Transaction: basic concepts, states, concurrent execution, serializability, recoverability; Concurrency control: Basics, Recovery: Lock based Recovery.

Self-Study: **[12 Hrs]**
Views, Network Model, Decomposition using functional dependencies, Static and dynamic hashing, deadlock handling.

Textbooks

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database system concepts”, Fifth Edition, McGraw Hill International Edition, ISBN 978-0073523323.

2. Raghu Ramkrishnan, Johannes Gehrke, “Database Management Systems”, Second Edition, McGraw Hill International Editions, ISBN 978-0072465631

Reference Books

1. Rob Coronel, “Database systems: Design implementation and management”, Forth Edition, Thomson Learning Press, ISBN 978-1418835934.
2. Ramez Elmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Third Edition, Pearson Education, 2003, ISBN 978-0321204486.
3. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", Second Edition, Pearson, 2008, ISBN 978-0131873254.

Laboratory assignments

List of Assignments:

1. Assignment 1: Basic SQL
   Write the simple SQL Queries on the given schema.
2. Assignment 2
   a: SQL: Aggregates
   Write the SQL queries using aggregates, grouping and ordering statements for given statements on given schema.
   b: Nested Subqueries
   Write the SQL queries for given schema using Nested Subqueries
3. Assignment 3: SQL DDL and updates
   Write the DDL and DML statements for the given statements.
4. Assignment 4: Schema creation and constraints
   Create the schema and constraints on the given relations using given statements.
5. Assignment 5: Submit the synopsis for small database application.
6. Assignment 6: Implement a small database application
   Implement a small database application using appropriate front End and Backend. Select any real time problem for database implementation. (Draw the ER diagram for the selected problem in hand. Normalise the database upto appropriate normal).

(PEC-02) - Data Science

Teaching Scheme
Lectures: 3 Hrs./ Week
Laboratory: 2 Hrs./ Week
Self-study: 1 Hr/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes
1. Understand the fundamentals of Data Science.
2. Analyze and preprocess data using appropriate statistical measures and data preparation techniques.
3. Perform exploratory data analysis on the given data.
4. Design and analyse basic data models using regression techniques and optimization methods.
5. Apply similarity measures and frequent itemset mining techniques to discover patterns in data.
6. Understand and apply data stream processing techniques to manage high-velocity data.

Course content

Introduction: [6 Hrs]

Introduction to Data Science, Examples, Data Sources, Challenges, Applications, Technologies, Introduction to Data Modeling, Statistical Data Modeling, Computational Data Modeling, Statistical limits on data- Bonferroni's principle.

Data gathering, knowing and preprocessing: [8 Hrs]

Data gathering: structured and unstructured data, data objects and attribute types, statistical description of data: mean, median, mode, range, quartiles, variance, standard deviation and inter quartile range, data similarity and dissimilarity, data preprocessing: structured and unstructured data, types, attributes, data cleaning, data integration, data reduction, transformation, discretization.

Exploratory Data Analysis: [8 Hrs]

Descriptive and inferential statistics, Chart types- Single var: Dot plot, Jitter plot, Error bar plot , Box-and-whisker plot, Histogram, Kernel density estimate, Cumulative distribution function, Two variable: Bar chart, Scatter plot, Line plot, Log-log plot, More than two variables: Stacked plots, Parallel coordinate plot, Inferential statistical analysis: Analysis of Variance (ANOVA), Interpretation of ANOVA results (F-statistic, p-value), Chi-Square Test for independence.

Data Modeling: [8 Hrs]

What is data model? Function approximation, hypothesis representation, objective / loss function, linear regression, logistic regression, gradient descent.

Similarity Measures, Distance Measures and Frequent Itemsets: [8 Hrs]

Feature extraction - TF, IDF, TF-IDF, Hash functions, Similarity measuring techniques- Shingling, Min-hashing, Locality Sensitive hashing, Distance measures- Triangle Inequality, Euclidean Distance, Cosine Distance, Jaccard Distance, Edit Distance measures, Frequent Itemsets, the Market-Basket Model, Association Rules, A-Priori Algorithm.

Data Streams: [8 Hrs]

Stream data model, stream sources, stream queries, issues in stream processing, sampling data in a stream, stream filtering: bloom filter.

Self-study: [12 Hrs]

Ethical Issues and Bias in Data Science, Over-fitting, under-fitting and model evaluation metrics: Bias-variance tradeoff, MSE, RMSE, Regularization Techniques (L1 & L2): Ridge and Lasso regression (conceptual understanding, FP Growth Algorithm, Sliding window model in stream processing.

Textbooks

1. "Mining of Massive Datasets", Jure Leskovec, Anand Rajaraman, and Jeffery David Ullman, Cambridge University Press, 2 edition (13 November 2014), ISBN-10: 1107077230, ISBN-13: 978-1107077232
2. "Data Mining: Concepts and Techniques", Jiawei Han, Micheline Kamber, 3rd Edition, Morgan Kaufmann, ISBN-13: 978-9380931913

Reference Books

1. "Foundations of Data Science", Avrim Blum, John Hopcroft, and Ravindran Kannan, Hindustan Book Agency, (online free version) January 2020, ISBN-10: 9386279800
2. "Introduction to Machine Learning", Ethem Alpaydin, PHI Learning Pvt Ltd, Third edition, 2016, ISBN-978-81-203-5078-6

Web references

5. https://docs.databricks.com/aws/en/notebooks/eda-tutorial
6. https://www.kaggle.com/code/imoore/intro-to-exploratory-data-analysis-eda-in-python
7. https://deepnote.com/app/code-along-tutorials/A-Beginners-Guide-to-Exploratory-Data-Analysis-with-Python-f536530d-7195-4f68-ab5b-5dca4a4c3579?utm_content=f536530d-7195-4f68-ab5b-5dca4a4c3579

Laboratory assignments

1. Choose a data set and perform following steps for the selected problem statement. (Mini project)
   a) Define problem statement
   b) Data gathering
   c) Data preprocessing
   d) Data exploration and analysis
   e) Data modeling
   f) Model evaluation

   Some samples are listed as follows:
   i. Exacerbations in Chronic Obstructive Pulmonary Disease: Identification and Prediction Using a Digital Health System.
   ii. Ad2Vec: Similar Listings Recommender for Marketplace.
   iii. Clothes Classification with the DeepFashion Dataset and Fastai.
   iv. Recommendation lab applications for some select commercial applications.
2. Select existing achieved tasks in open online platforms or social competitions and demonstrate the competition result comparisons. (minimum 3 tasks).

This list is a guideline. The instructor is expected to improve it continuously.

(PEC-2) – Blockchain Technologies

Teaching Scheme	**Evaluation Scheme**
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	
Self-study: 1 Hr/ Week	Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

Course Outcomes

Students will be able to:

1. Discuss the basics of Blockchain and cryptography.
2. Discuss the internal working of various cryptocurrencies.
3. Demonstrate smart contract development in solidity using Ethereum Platform
4. Introduce Hyperledger Frameworks and Fabric Concepts
5. Analyze the importance of blockchain in finding the solution to real-world problems
6. Summarize the different technologies and latest trends in Blockchain.

Course content

Introduction to Blockchain: **[8 Hrs]**

History and evolution of Blockchain, need for Blockchain, Blockchain basics, changing landscape of digitization, cryptographic concepts, hashing, public and private blockchains, smart contracts, and consensus mechanisms.

Introduction to Cryptocurrencies: **[8 Hrs]**

Internal working of cryptocurrencies: Bitcoin, Ethereum, Litecoin, Dogecoin, Altcoins; transaction lifecycle, wallets, mining, gas concept, security and limitations of cryptocurrencies.

Blockchain Platforms Ethereum: **[8 Hrs]**

Blockchain Platform introduction, what is Ethereum, Ethereum feature, Components of Ethereum Ecosystem, Ethereum Programming Languages, Runtime Byte Code, Blocks and Blockchain, How Smart Contracts Work. Introduction to Tokenization: What is token, technology behind tokenization, how blockchain tokenization can help in enterprise systems, Tokenizing Shares and Fund Raising, challenges to tokenization

Blockchain Platforms Hyperledger: **[8 Hrs]**

What is Hyper ledger, features of a Hyper ledger blockchain, How Does Hyper Ledger Fabric Work, The Architecture of Hyper ledger Fabric System, Benefits of Hyper ledger Fabric, Differences Between Ethereum and Hyper ledger

Blockchain Use Cases: **[8 Hrs]**

Blockchain applications in Finance, Banking, Retail, Land Records, Loyalty Programs, Agriculture, NFT and Supply Chain, implementation of any of the use case on the chosen Blockchain platform.

Emerging Trends in Blockchain: **[8 Hrs]**

DeFi, utility tokens, integration of Blockchain with other technologies like IoT, AI, ML, Data Science, etc., Blockchain as a service, privacy and security issues in Blockchain.

Self-Study: **[12 Hrs]**

Blockchain Development Tools: Ethereum Development Tools, Solidity Tools, Hyperledger Fabric Tools, Web3.js and Ethers.js, Corda Development Tools, NEO Development Tools, Binance Smart Chain (BSC) Tools, Solano.

Textbooks

1. “Blockchain for Dummies”, Tiana Laurence, Second Edition, 2017, Wiley Publication, ISBN 978-1119555018
2. “Blockchain Quick Reference: A guide to exploring decentralized blockchain application development”, Brenn Hill, Second Edition, 2015, Pearson Education India; ISBN 81-203-0596-5

Reference Books

1. “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Arvind Narayanan, 2016, Princeton University Press
2. “Mastering Blockchain”, Imran Bashir, 2018, Packt Publication, ISBN 978-1788839044

Laboratory assignments

1. Create a simple Blockchain node setup in any cloud environment (AWS, Azure, GCP, etc.)
2. Implement a Blockchain network on the Multichain platform
3. Use Geth to Implement Private Ethereum Blockchain.
4. Write a Blockchain application to transfer the ether cryptocurrency from one user to another.
5. Build Hyperledger Fabric client application for a supply chain application.
6. Build Hyperledger Fabric application and write a smart contract to transfer ownership of assets in a supply chain.
7. Create a case study of Blockchain being used in illegal activities in real world.
8. Build an application on Corda platform for the student loan disbursal.
9. Implement simple Smart Contracts in Remix IDE.

(CTS (DE)-) ADVANCED SYSTEM PROGRAMMING

Teaching Scheme
Lectures: 3 Hrs/ Week
Laboratory: 2 Hrs/ Week

Evaluation Scheme
Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: CIE: 50 Marks, ESE: 50 Marks

Course outcomes

1. Explain system software and different system level programming concept
2. Simulate Two-pass and One-pass of Assemblers
3. Illustrate expansion of Macro Processors in program.
4. Demonstrate different schemes of loaders
5. Analyse the importance of different development tools.
6. Differentiate between different translators.

Course content

Introduction to System software: [6 Hrs]

Introduction to System software, Software Hierarchy System concepts, system components & architecture, Systems programming structures, Evolution of system Software, Introduction to Assembler, Linker and Loader, operating system Batch, timesharing, multiprogramming, multi-tasking distributed and real time. Recent Trends in Software

Assembler: [8 Hrs]

System Software and Assemblers Definition, Language Translators. Assemblers Structure of a single Pass assembler Design of two pass assembler, Table of incomplete instruction, back patching. Data structures used for design of One and Two pass assembler, Design and Implementation of two pass assembler, Error handling and Symbol Table management in assembler, Handling constants, literals, labels and Procedures, One pass assembler design and comparison with two pass assembler design, Cross assembler

Macro Processor: [6 Hrs]

Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Macro Parameters – Positional, Keyword, Actual, Design and implementation of simple macro processor, Nested Macro processor – Macro call within macro definition and macro definition within macro definition, Design and implementation of nested macro processor. General Macro processing concepts - Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Macro Processor Design Options - Recursive Macro Expansion, implementation Examples - MASM Macro Processor

Linkers and Loaders: [6 Hrs]

Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine- Dependent Loader Features - Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader, Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage, Bootstrap Loaders, Implementation Examples - MS-DOS Linker

Scanning and Parsing: [6 Hrs]

Programming Language Grammars, Classification of Grammar, Ambiguity in Grammatic Specification, Scanning, Parsing, Top Down Parsing, Bottom up Parsing, Language Processor Development Tools, LEX, YACC

Interpreters & Debuggers: [6 Hrs]

Benefits of Interpretation, Overview of Interpretation, The Java Language Environment, Java Virtual Machine, Types of Errors, Debugging Procedures, Classification of Debuggers

Self-study [12 Hrs]

Virtual Machines vs Containers, Dynamic Linking and Shared Libraries, Static vs Dynamic Linking, Shared Object Files, Runtime Loading, Modern Debugging and Profiling Tools, GDB Advanced Features, Valgrind, Performance Profiling using gprof, Binary Analysis and Reverse Engineering, ELF File Format, obj dump, nm utilities, Introduction to Reverse Engineering Concepts

Textbooks

1. D.M Dhamdhere: Systems Programming and Operating Systems (Second Edition), Tata McGraw Hill Pub.Co.Ltd.2000
2. Donovan: Systems Programming McGraw Hill Education; (1 July 2017)

Reference Books

1. BECK (Author) System Software: An Introduction to Systems Programming, 3e Pearson India; 3rd edition (1 January 2002)

Web references

1. https://www.gnu.org/software/Oracle Java Virtual Machine Specification

Laboratory assignments

1. Implementation of single pass assembler on a limited set of instructions for a given assembly language program
2. Designing a two-pass assembler for X86 based machine and ELF. Students should handle data segment, code segment, symbol table management as well as error handling.
3. Expand the above assignment to cover procedure.
4. Design a simple (without nesting) macro processor and nested macro processor [Call to a macro processor inside macro definition].
5. Design a bootstrap loader
6. Assignment on LEX and YACC
7. Use the tool to analyse ELF files, to print them, to instrument (change) them to have features like profiling.