III Semester

MAS2001: STATISTICAL METHODS AND PROBABILITY THEORY [3 0 0 3]

Course Contents

Probability Theory and Random Variables: Probability (Only One Lecture), Random variables, Cumulative distribution functions, Discrete random variables, Continuous random variables, Independent random variables, Probability mass and density functions, Expectation of random variables, Chebyschev's inequality, Central limit theorem. **Probability distribution:** Binomial, Poisson, Uniform, Normal, Exponential **Theory of Estimation:** Maximum Likelihood and method moment estimation, Sufficient statistics, Bayesian estimation, Confidence intervals for means. **Tests of Statistical Hypothesis:** Introduction, Parameter and Statistic, Standard error, Statistical hypotheses, Critical region, Tests of hypotheses and significance, Type I and Type II errors, level of significance. level of significance, Test about one mean, Test about equality of two means, Test of variances, Chi square test, Analysis of Variance.

References:

1. **Goon, A.M., Gupta, M.K. and Dasgupta, B., An Outline of Statistical Theory, Vol. II, 3rd Edn. World Press, Kolkata, 2005.**
2. **Kendall and Stuart, Advanced Theory of Statistics Vol.-II, Charles Griffin & Co. Ltd, London, 1961.**
3. **Casella, G. and Berger, R.L., Statistical Inference, Second Edn. Thomson Duxbury, 2002.**
4. **Hogg, R.V. and Tanis, E.A., Probability and Statistical Inference, 3rd Edn. Macmillan Publishing Co. Inc., 1988.**
5. **Feller, W., An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition, John Wiley, 1968.**

CSE2101: DATA STRUCTURES AND ALGORITHMS [3 1 0 4]

Course Contents

Introduction: algorithm specification; **Performance analysis:** time and space complexity, asymptotic notation; **C concepts:** pointers, functions, arrays, passing arrays to functions through pointers, dynamic memory allocation, bubble sort , insertion sort , selection sort, structures, arrays of structures, passing structures to functions; **List:** ADT, array and its types, implementation, operations, linked list and its types, implementation and operations; **Stack:** ADT, implementations using array and linked list, operations and its applications; **Queue:** ADT, implementations using array and linked list, operations and its applications; **Tree:** terminologies, different types, representation of binary tree using array and linked structure, binary search tree, different operations (recursive and non-recursive), heap, heap sort, priority queue, AVL trees, B-tree; **Graph:** Introduction, representation, operations and applications; Searching techniques and hashing.

References:

1. **A. S. Tannenbaum, J. Augenstein, Data Structures using C, Pearson India, 2018.**
2. **2. E. Horowitz, S. Sahni, Fundamentals of Data Structures in C, (2e), Universities Press, 2008.**
3. **A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C, (3e), Cengage Learning, 2006.**

CSE2102: Relational Database Management Systems [3 1 0 4]

Course Contents

Introduction: DBMS Concepts, Database System Vs File System, Data Models, Schema & Instance, Schema architecture, Data independence, Data Base Languages and interfaces, Database system applications, Database users, Functions of DBA Data Modeling using the Entity Relationship; **Model:** ER model concepts, Entities, Attributes, Relationship & types, Relationship Constraints, Extended ER-Model Concept - Generalization, Specialization and Aggregation, Transforming ER diagram into the tables; **Relational Data models:** Domains, Tuples, Attributes, Relations, Characteristics of relations, Keys, Key attributes of relation, Relational database, Schemas, Integrity constraints. Referential integrity, Relational Algebra and Relational Calculus; **Relational algebra operators**: Unary, Binary, Set Operations. Tuple oriented and domain oriented relational calculus and its operations; **SQL:** Basic SQL Query, Creating Table and Views, SQL as DML, DDL and DCL, SQL Algebraic Operations, Joins, Sub-Queries, Aggregate Operations, Cursors, Dynamic SQL, Integrity Constraints in SQL, Triggers; **Data Base Design:** Introduction to Normalization, Functional dependency, Normal forms, Decomposition, Armstrong's Axioms, Canonical Cover, Lossless Join & Dependency preservation Problems with null valued and dangling tuples, multivalued dependencies; **Transaction Processing Concepts:** Transaction Properties & States, Schedules, Serial & Concurrent Schedule, Serializability of schedules, conflict & view serializable schedule, Recoverability, Recovery from transaction failures, log-based recovery, checkpoints, Deadlock handling; **Concurrency Control Techniques:** Concurrency control, Concept of Locks, Concurrency Control Protocols - Two Phase Locking Protocols, Time stamping protocols, validation-based protocol, multiple granularities, Multi version schemes, Recovery with concurrent transactions; **File Structures:** File Organization, Indexing, Primary, Clustered, Secondary Indexes, Hashing, Multilevel Indexing with B-Tree, B+ Tree.

References:

1. **A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, (6e), McGraw Hill, 2013.**
2. **R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, (6e), Addison-Wesley, 2010.**
3. **R. Ramakrishnan, J. Gehrke, Database Management Systems, (3e), McGraw Hill, 2014.**
4. **I. Bayross, SQL, PL/SQL The Programming Language of Oracle, (4e), BPB Publications, 2010.**
5. **J. Date, An Introduction to Database Systems, (8e), Prentice Hall of India, 2006.**

CSE2103: Computer Organization and Architecture [3 1 0 4]

Course Contents

Basic Structure of Computers: Computer types, Functional units, Basic operational concepts, performance; **Machine Instructions and Programs:** Number representation and arithmetic operations, Memory locations and addresses, Instructions and instruction sequencing, Addressing modes, Assembly language programs; **Input-Output Organization:** I/O module schematic, Memory-mapped and I/O mapped device interface, Programmed I/O, Interrupt driven data transfer, Interrupt handling, Direct memory access; **Memory System:** Cache basics, Mapping techniques, Replacement algorithms, Hit rate and miss penalty; **Arithmetic:** Design of fast adders, Multiplication of positive numbers, Signed operand multiplication, Fast multiplication; **Basic Processing Unit and Pipelining:** Single and multiple-bus organization of processor Datapath, Microprogrammed control unit, Hardwired control unit, Basic concepts of pipelining, Structural hazards, Data hazards, Control hazards; **Multi-Core Processors:** Flynn's classification, Superscalar, Vector processor, Multi-core, and GPU.

References:

1. **J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, (6e), Morgan Kaufmann Publishers, 2019.**
2. **M. W. Stallings, Computer Organization and Architecture -Designing for Performance, (9e), Pearson, 2013**
3. **C. Hamacher, Z. Vranesic, S. Zaky, Computer Organization, (5e), Tata McGraw Hill, 2011.**

CSE2121: Object Oriented Programming using Java [3 1 0 4]

Course Contents

Introduction: Objects, Classes, Encapsulation, Polymorphism, Inheritance; **Java Basics:** Compilation and Execution of a Java program, Java Compiler and Interpreter, Data Types in Java, operators and control statements in java; **Class Definition and Object Creation:** Instance-Fields/Attributes, Methods, Access Modifiers, Constructors, Object vs Class Variables, Role of static and final keywords in Java, Type Conversion and Promotion; **Polymorphic Forms:** Method Overloading, Objects as Parameters and return types; **Input-Output :** Reading Input and Output in Java; **Object Class in Java :** String form of an Object via toString() method, Object equality method; **Arrays and Strings in Java:** 1-D Arrays, 2-D and Multi-dimensional arrays, Variable Size array, Dynamic Arrays using Array Lists, Strings in Java via String, StringBuilder and StringTokenizer classes; **Inheritance in Java:** Extending classes, abstract classes, final classes, use of super keyword, Method Overriding, Runtime Polymorphism, Inner Classes – static and non-static nested Classes, Local Classes; **Inheritance via Interfaces:** class vs interface, defining interfaces, implementing multiple inheritance, extend interface; **Exception Handling:** Exceptions, Defining and Creating Exceptions, multiple catch clauses, Use of Exceptions in Real Life Problems; Package in Java: Defining and Creating Packages, importing packages; **Garbage Collector:** Role, definition, explicit call; **Multithreading:** Thread class, Runnable interface, thread life cycle, synchronization, thread priority, creating and running threads.

References:

1. **Herbert Schildt ,Java : The Complete Reference, 12th Edition, McGraw Hill, 2022**
2. **E Balagurusamy, Programming with Java, 6th Edition, McGraw Hill, 2019**
3. **Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11th Edition, Pearson, 2018.**

CSE2122: Object Oriented Programming using Python [3 1 0 4]

Course Contents:

Introduction: Programming a computer, Programming languages; **Python basics:** Getting started with Python, Essentials of a Python program, Integers, Floating-point numbers, Strings; **Variables and scope:** Variables, Modifying values, Type conversion; **Selection control statements:** if statement, Boolean values, operators, and expressions; **Collections:** Lists, Tuples, Sets, Ranges, Dictionaries, Conversion, Sequences; **Loop control statements:** while, for statements, Nested loops, Iterables, iterators and generators, Comprehensions, The break and continue statements; **Functions:** Input parameters, Return values, Default parameters, *args and **kwargs, Decoratorss, Lambdas,Generator functions and yield; **Data Structure in Python:** Array, Linked List, Stack, Queue, Tree, Searching and Sorting; **Object-Oriented programming:** OOP's Concepts, Classes, and Objects: Defining and using a class, Instance attributes, Class attributes, Class decorators, inspecting an object, Constructor, Abstraction, Composition; **Inheritance:** Types of Inheritance. overriding magic methods; **I/O and Errors Handling:** Errors, exceptions, handling exceptions, Debugging programs, Logging, Testing; **Packaging:** Modules, Packages, Documentation; **File Handling:** Introduction, Access Methods, Read and write operation, Working with directories; **Python Libraries:** Pandas, Matplotlib, NUMPY.

References:

1. **Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Data Structures and Algorithms in Python, Wiley Publication.**
2. **D. Phillips, Python 3 Object-Oriented Programming Build robust and maintainable software with object-oriented design patterns in Python 3.8, (3e), Packt Publishing, January 2018.**
3. **W. J. Chun, Core Python Applications Programming, (3e), Prentice Hall Publishers, 2012.**
4. **J. Grus, Data Science from Scratch: First Principles with Python, (1e), O'Reilly Media, 2015.**
5. **Alberto Boschetti, Luca Massaron, Python Data Science Essentials: A practitioner's guide covering essential data science principles, tools, and techniques, Third Edition .**

CSE2131: Data Structures and Algorithms Lab [0 0 2 1]

Course Contents:

Implementation of array operations: insertion, deletion, linear search and binary search, matrix operation; **Implementation of singly, doubly, and circular linked lists:** inserting, deleting, and inverting a linked list, Polynomial addition, subtraction and sparse matrix implementation by linked list, Josephus problem; **Stacks and Queues:** adding, deleting elements. **Circular Queue:** Adding & deleting elements, conversion of infix to postfix and Evaluation of postfix expressions using stacks & queues, Implementation of stacks & queues using linked lists; **Recursive and Non-recursive traversal of Trees:** Threaded binary tree traversal, BST and AVL tree implementation; **Implementation of sorting and searching algorithms:** bubble sort, Insertion sort, selection sort, quick sort, heap sort, merge sort, radix sort, Hash table implementation.

References:

1. **A. S. Tannenbaum, J. Augenstein, Data Structures using C, Pearson India, 2018.**
2. **E. Horowitz, S. Sahni, Fundamentals of Data Structures in C, (2e), Universities Press, 2008.**
3. **A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C, (3e), Cengage Learning, 2006.**

CSE2132: Relational Database Management System Lab [0 0 2 1]

Course Contents:

Database Foundations: Introduction, Databases and Data Modelling, Refining the Data Model, Oracle SQL Developer Data Modeler, Introduction to SQL; **Database Design:** Introduction, Entities and Attributes, Super/Sub Types and Business Rules, Relationship Fundamentals, UIDs and Normalization, Arcs, Hierarchies, and Recursive Modelling, Mapping, Creating Database Projects, Presenting Database Projects; **Database Programming with SQL**: Introduction, SELECT, WHERE ORDER BY, and Intro to Functions, Single Row Functions, JOINs, Group Functions, Subqueries, Ensuring Quality Queries Part I, DML, DDL, Constraints, Views, Sequences and Synonyms, Privileges and Regular Expressions, TCL, Final Project, and Exam Review.

References:

1. **A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, (6e), McGraw Hill, 2013.**
2. **I. Bayross, Teach yourself SQL & PL/SQL using Oracle 8i & 9i with SQLJ, BPB Publications, 2010.**
3. **R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, (6e), Addison-Wesley, 2010.**

CSE2201: Design and Analysis of Algorithms [3 1 0 4]

Course Contents:

Introduction: Algorithm Definition and Criteria of Algorithms, Iterative and Recursive algorithms; **Performance Analysis:** Priori and Posteriori Analysis, Asymptotic Notations, Space Complexity, Time Complexity, Performance measurement of iterative and recursive algorithms; **Solving Recurrence Relations:** Substitution Method, Iterative Method, Recursive Tree Method, Master Method; **Divide and Conquer:** Introduction, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Randomized Quick Sort, Integer Multiplication; **Graph Search Algorithm:** Graph representation, Breadth First Search and Depth First Search; **Greedy Strategy:** Introduction, Knapsack Problem, Job Sequencing with Deadlines, Huffman Coding, Union and Find Operation (Set and Disjoint Set), Minimum Cost Spanning Tree Algorithms (Prim's and Kruskal's), Optimal Merge Patterns, Single Source Shortest Path (Dijkstra's Algorithm); **Dynamic Programming:** Introduction, Single Source Shortest Path (Bellman and Ford Algorithm), All Pair Shortest Path (Floyd Wrashal's Algorithm), Optimal Binary Search Trees, 0/1 Knapsack Problem, Travelling Salesperson Problem, Longest Common Subsequence, Matrix Chain Multiplication, Edit distance; **Backtracking:** Introduction, N-Queens Problem, Graph Colouring and Hamiltonian Cycles; **Branch and Bound:** Introduction, FIFO and LC Branch and Bound, 0/1 Knapsack Problem, Travelling Salesman Problem; **String Matching:** Naïve String Matching, Rabin Karp Algorithm, Knuth-Morris-Pratt Algorithm; **Complexity Classes:** NP, NP-Complete and NP-Hard Problems, Cook's Theorem, Polynomial time reductions, Satisfiability, Reduction from Satisfiability to Vertex Cover.

References:

1. **T. H. Cormen, C. E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, (3e), MIT press, 2009**
2. **E. Horowitz, S. Sahni, S. Rajasekaran, Fundamental of Computer Algorithms, (2e), Universities Press, 2007.**

CSE2202: Operating Systems [3 1 0 4]

Course Contents:

Introduction: Definition of operating systems, Single and multi-processor systems, Operating system services, System commands and system calls, Interrupt, System boot, Operating system structure, Types of OS, Multi-user, Multitasking, Embedded, Real-time, Network, Distributed; **Process and Thread**: Process concept, Operations on processes, Inter-process communication, UNIX pipes, Multithreading, Multithreaded models, Programs using PThread; **Process Scheduling**: Basic concepts, Scheduling criteria, Scheduling algorithms; **Synchronization**: Critical section problem, Dekker's algorithm, Peterson solution, Synchronization hardware, Semaphores, Classical problems of synchronization, Deadlock, Methods for handling deadlock- prevention, avoidance, detection, and recovery; **Memory Management**: Address binding, Logical vs Physical address space, Swapping, Contiguous memory allocation, Paging, Structure of Page Table, Segmentation, Demand Paging, Page Replacement Policies, Allocation of Frames, Thrashing; **File System Interface and Implementation**: File Concept, Access Methods, Directory and Disk Structure, File System Mounting, File System Structure, File System Implementation, Allocation Methods, Free Space Management; **Disk Management**: Disk Scheduling Algorithms, Disk Management, Swap Space Management; **Case Studies**: Linux, Windows, iOS, Android.

References:

1. **A. Silberschatz, P. B. Galvin, G. Gagne, *Operating System Concepts*, (9e), Wiley, 2014.**
2. **A.S. Tanenbaum, H. Bos, *Modern Operating Systems*, (4e), Pearson, 2015.**
3. **W. Stallings, *Operating Systems: Internals and Design Principles*, (9e), Pearson, 2018.**

CSE2221: Cryptography [3 1 0 4]

Course Contents:

Introduction: Basic objectives of cryptography, Secret-key and public-key cryptography, One-way trapdoor one-way functions, Cryptanalysis, Attack models, Classical cryptography; **Block ciphers:** Modes of operation, Public-key parameters: Modular arithmetic, GCD, primality testing, Chinese remainder theorem, modular square roots, finite fields; DES and its variants, AES, Linear and differential cryptanalysis; **Message digest:** Properties of hash functions, MD2, MD5 and SHA-1, Keyed hash functions, Attacks on hash functions. Pseudorandom Number Generation; **Intractable problems:** Integer factorization problem, RSA problem, Modular square root problem, Discrete logarithm problem, Diffie-Hellman problem, known algorithms for solving the intractable problems; **Public-key encryption:** RSA, ElGamal scheme, Elliptic and hyperelliptic, curve cryptography, Side channel attacks, Diffie-Hellman and MQV key exchange; **Digital signatures:** RSA, DSA and NR signature schemes, blind and undeniable signatures; **Entity authentication:** Passwords, Challenge-response algorithms, Zero-knowledge protocols; **Transport-Level Security:** Web Security Issues, Secure Sockets Layer (SSL), Transport Layer Security (TLS), Electronic Mail Security, Pretty Good Privacy (PGP); **IP Security:** IP Security Overview, IP Security Policy, Encapsulating Security, Payload, Combining Security Associations, Internet Key Exchange.

References:

1. **B. A. Forouzan, D. Mukhopadhyay, *Cryptography and Network Security*, McGraw Hill, 3rd Edition, 2008.**
2. **W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall, 5th edition, 2010.**

CSE2222: High Performance Computing [3 1 0 4]

Course Contents:

Introduction to Parallel Computing: Definition and applications of parallel computing, Scope of Parallel Computing, Parallel Programming Platforms: Implicit Parallelism, Trends in Microprocessor and Architectures, Limitations of Memory, System Performance; **Types of HPC systems:** clusters, grids, clouds, HPC Applications, HPC Scalable and parallel architectures – Shared memory and distributed memory; **HPC ecosystem:** hardware, software, networks, and Interconnect technologies, InfiniBand, Ethernet, etc. **case study: HPC facility in INDIA. Parallel programming**: Principles of Parallel Algorithm Design, Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions; **Parallel Programming Models:** Message Passing Interface (MPI), Introduction to OpenMP, OpenMP – Threads vs Process, fork-join model, OpenMP – Race Conditions, False Sharing and Critical Section, OpenMP Data Environment; **Case Study:** Pi calculation and OpenMP; **CUDA programming**: Introduction to CUDA programming, CPU vs GPU, The Age of Parallel Processing, the Rise of GPU Computing, A Brief History of GPUs, Early GPU, GPU Architecture, CUDA Architecture, Using the CUDA Architecture, Applications of CUDA, Introduction to CUDA C-Write CUDA programming execution model, CUDA programming in Python – global memory coalescing and shared memory utilization, Multi-GPU CUDA Programming; **Analytical Models of Parallel Programs:** Analytical Models, Sources of overhead in Parallel Programs; **Performance Metrics for Parallel Systems:** speedup, efficiency, scalability, Profiling, and performance analysis tools, Minimum execution time and minimum cost, optimal execution time; **Dense Matrix Algorithms:** Matrix- Vector Multiplication, Matrix-Matrix Multiplication; **Distributed Memory Parallelization**: Introduction to Open MPI, the message passing programming model – Send (), Receive (), MPI applications, clusters, and benchmarks, Pi Calculation using MPI, MPI blocking/non-blocking communication; **Hands-on Projects:** Implementation of parallel algorithms using MPI, OpenMP.

Reference:

1. **Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, "Introduction to Parallel Computing", 2nd edition, Addison-Wesley,2003, ISBN: 0-201-64865-2.**
2. **Jason sanders, Edward Kandrot, "CUDA by Example", Addison-Wesley, ISBN-13: 978-0-13-138768.**
3. **Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 2013 ISBN: 9780124159884.**
4. **David Culler Jaswinder Pal Singh, Parallel Computer Architecture: A Hardware/Software Approach, Morgan Kaufmann,1999, ISBN 978-1-55860-343-1**

CSE2240: Automata and Compiler Design [PE1] [3 0 0 3]

Course Contents:

Introduction: Three basic concepts – language, grammar and automata; Chomsky Hierarchy; **Finite Automata:** Deterministic Finite Automata (DFA) and Non-Deterministic Finite Automata (NFA), Mealy and Moore machines construction and equivalence; **Regular Sets and Regular Grammars:** Regular Expressions, Equivalence of regular expressions and regular languages, Regular Grammar and FA, Closure properties of Regular Languages, Pumping Lemma for Regular Languages; **Context Free Languages (CFL) and Grammars (CFG):** Ambiguity, Methods for Transforming Grammars, Chomsky Normal Form (CNF) and Greibach Normal Form (GNF); **Push Down Automata:** Nondeterministic Pushdown Automata (NPDA), Design of NPDA, Equivalence of PDA and CFLs, Closure properties and decision problems of CFLs; **Introduction to Turing machine:** Definition, Turing Machining as Language acceptors, Types of Turing machine, Recursively Enumerable and Recursive Languages and their closure properties, Concept of insolvability & reducibility, Halting Problem, Post correspondence Problem, Rice theorem, P and NP, Polynomial-Time reductions and NP-Completeness; **Introduction to Compiler Design:** Structure of a Compiler, Lexical Analysis, Recognition of Tokens.

References:

1. **P. Linz, *Introduction to Formal Languages and Automata*, Johnes and Bartlett , 6th Edition, Indian Student Edition, 2016.**
2. **A. Aho, J. Ullman, M. S. Lam, R. Sethi, *Compilers : Principles, Techniques and Tools,* Pearson Education, 2nd Edition, 2007.**
3. **J. Martin, *Introduction to Languages and the Theory of Computation,* Tata McGraw Hill, 4th Edition, 2010.**
4. **M. Sipser, *Introduction to the Theory of Computation*, Cengage Learning, 3rd Edition, 2014.**

CSE2241: Data Visualization Techniques [PE1] [3 0 0 3]

Course Contents

Introduction: Introduction of visual perception, visual representation of data, Gestalt principles, information overloads; **Visual representations**: Creating visual representations, visualization reference model, visual mapping, visual analytics; **Design of visualization applications**: Classification of visualization systems, Interaction and visualization techniques misleading, Visualization of one, two and multi-dimensional data, text and text documents; **Visualization:** Visualization of groups, trees, graphs, clusters, networks, software; **Metaphorical visualization:** Visualization of volumetric data, vector fields, processes and simulations; **Visualization of maps**: geographic information, GIS systems, collaborative visualizations, evaluating visualizations.

References:

1. ***Ward, Grinstein Keim, Interactive Data Visualization: Foundations, Techniques, and Applications. Natick: A K Peters, Ltd, 1st Edition, 2010***
2. ***Kieran Healy, Data Visualization: A Practical Introduction, 1st Edition, 2018***

CSE2231: Design and Analysis of Algorithms Lab [0 0 2 1]

Course Contents:

Sorting & Searching Algorithm: insertion sort, selection sort, binary search; **Basic data structures:** stacks and queues, graphs and trees, binary trees; **Algorithmic paradigms:** Recursion, divide-and-conquer, Merge sort, Quick sort; **Greedy:** Knapsack, Huffman encoding, dynamic programming, lower bounds and optimal algorithms; **Heaps:** Heaps, priority queues, min-max heaps, heap sort; **Dynamic search structures:** Binary search trees, height balancing, B-trees; **Algorithms on arrays:** Linear-time median finding, sorting in linear time (counting sort, radix sort, bucket sort), String matching (Rabin-Karp and Knuth-Morris-Pratt algorithms); **Graph algorithms Traversal:** (BFS, DFS, topological sort), Minimum spanning trees (Prim and Kruskal algorithms), shortest paths (Dijkstra's and Floyd-Warshal algorithms); **Mini-Projects & Case Studies**.

References:

1. **T. H. Cormen, C. E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, (3e), MIT press, 2009**
2. **E. Horowitz, S. Sahni, S. Rajasekaran, Fundamental of Computer Algorithms, (2e), Universities Press, 2007.**

CSE2232: Operating Systems Lab [0 0 2 1]

Course Contents:

Red Hat: Install Red Hat Enterprise Linux using scalable methods, Access security files, file systems, and networks; **Shell Programming:** execute shell scripting and automation techniques; **System Calls:** File and process, I/O Redirection, IPC using Pipe and Signals; **PThread API:** Multithreaded programs; **Synchronization**: Programs using PThreads and Semaphores; **CPU Scheduling**: FCFS, SJF, Priority, RR; **Deadlock:** Bankers Algorithm; **Memory Management**: LRU, FIFO, OPT; **Creating a Virtual Machine:** Virtual Machine Files and Snapshots, Virtual Machine Cloning and Exporting.

References:

1. **W. R. Stevens, S. A. Rago, Advanced Programming in the UNIX Environment, (3e), Addison Wesley, 2013.**
2. **S. Das, Unix Concepts and Applications, (4e), McGraw Hill, 2006.**
3. **K. A. Robbins, S. Robbins, Unix Systems Programming: Communication, Concurrency, and Threads, (2e), Prentice Hall, 2004**
4. **https://www.redhat.com/en/services/training/rh124-red-hat-system-administration-i**

V Semester

CSE3101: Computer Networks [3 1 0 4]

Course Contents

Introduction, History and Development of Computer Networks, Concept of Layered Architecture (e.g. OSI Model and TCP/IP Model). **Application Layer:** Basic Communication Applications and Protocols. **Transport Layer:** Services and Protocols, Reliable Protocol Design Concepts. **Network Layer:** Services, Routing Algorithms and Protocols, Inter domain and Intra domain Routing, Multicasting, IP Addressing, Concept of Sub Networks. **Link Layer:** Services, Channel Access Protocols, Link layer Addressing, Interconnection devices (e.g. Hub, Bridge, Switch, Routers), ATM and MPLS networks, Concept of LAN, LAN Implementations. **Physical Layer:** Physical Media, Data Communication Basics, Line Encoding Techniques. **Wireless LAN:** Introduction to WLAN technologies. **IEEE 802.11:** System Architecture, protocol Architecture, 802.11a,802.11b. 3G&4G Mobile Network. **Wireless Sensor Network:** Architecture, Applications.

Reference Books :

1. **James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach, (6e), 2013, Pearson.**
2. **Andrew S. Tanenbaum, Computer Networks, Fourth Edition, Pearson Education, 2018.**
3. **B A Forouzan, and F Mosharraf, Computer Networks: A Top-Down Approach, TMH, 2014.**

CSE3102 : Software Engineering [3 1 0 4]

Course Contents

Introduction: Software processes and its models (waterfall, incremental development, spiral model, re-use-oriented model, prototype), Process activities, Process improvement (CMM Levels). Agile Development models. **Requirement Engineering:** Functional and non-functional requirements, requirement elicitation, use case development, requirement analysis, and validation, requirement review, or requirement change, SRS document. **Size Estimation:** Software Size, LOC and function point, cost and effort estimation, COCOMO. **System modelling:** Interaction models, Structural models, Behavioural models. **Software Design:** Overview, Modularity (cohesion and coupling), information hiding, functional independence, SOA, SAAS. **Software Coding:** Function-oriented and object-oriented coding. Software Quality: McCall’s Quality Factors, Quality Control, Quality Assurance, Software Reliability. **Software Evolution:** Evolution process, legacy system. **Software Testing:** verification, validation, Development testing (unit testing, component testing, system testing, Test Driven Development (TDD), Release Testing (Requirement testing, scenario testing, performance testing), User testing (alpha, beta, and acceptance testing), Regression Testing, Stress Testing. **Software Maintenance**: Maintenance prediction, Reengineering, Refactoring.

References:

1. **Roger S. Pressman, Bruce R. Maxim, Software Engineering: A Practitioner's Approach, (9e), 2023**
2. **Sommerville, Software Engineering (10e), Pearson, 2017.**

CSE3121: Artificial Intelligence & Soft Computing [3 1 0 4]

Course Contents

Introduction: Fundamental Concepts: Agents, environments, general model, Problem solving techniques, Introduction to intelligent agents. **Automated Reasoning**: Foundations of knowledge representation and reasoning, objects, relations, events, actions, time, and space; predicate logic. **Uncertain Knowledge**: Bayesian Networks, Basics of decision theory, sequential decision problems, elementary game theory. **Problem Solving through Search:** Forward and backward, state-space, blind, heuristic, problem-reduction, A, A*, AO*, minimax, constraint propagation. **Neural networks**: Perceptron, Back Propagation. **Fuzzy set theory**: Fuzzy sets, set-theoretic, fuzzy rules, reasoning, and interference. **Evolutionary Techniques**: Genetic algorithms, Swarm Algorithm, Ant Colony Optimization. **Emerging Trends in AI:** Explainable AI and interpretable models, Generative adversarial networks (GANs), Transfer learning and meta-learning, AI for sustainable development and social good. **Ethics and Responsible AI**: Ethical considerations, bias in AI.

References:

1. **S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, (4e) PHI, 2020.**
2. **E. Rich, K. Knight, S. B. Nair, Artificial Intelligence, (3e), Tata McGraw Hill, 2017.**
3. **G. F. Luger, Artificial Intelligence-Structures and Strategies for Complex Problem Solving, (6e), Addison-Wesley Pearson Education, 2012.**

CSE3122: Visual Computing [3 1 0 4]

Course Contents

Introduction to Computer Graphics: Computer Graphics Systems, Graphics Primitives: Line, Circle and Ellipse. **2D and 3D Geometric Transformation**: Homogeneous 2D and 3D coordinates, Homogeneous 2D and 3D transformations, pinhole imaging & perspective projection, Lighting and Shading. **Image Enhancement and Restoration:** Introduction to Image Enhancement , grayscale transformations, brightness interpolation, Image histogram, Spatial domain enhancement, point operations, Log transformation, Power-law transformation, Fourier transform, 2D-DFT. DCT, wavelet transform, PCA. Restoration Noise models, Restoration using Inverse filtering and Wiener filtering. **Filters:** Basic filters, Linear shift-invariant filters, convolution, Gaussian filters, derivative filters, sharpening filters, Bilateral filter, Discrete image formation, image interpolation, canny edge detection, detection of corners. **Segmentation and Representation:** Introduction to Segmentation, edges and lines, similarity, correlation, thresholding, template matching, edge-based segmentation, region-based segmentation, evaluation issues, mean shift segmentation, graph cut segmentation. Overview of representation scheme, boundary descriptors, regional descriptors, Image morphology. **Applications** such as Biometric Authentication, Face Recognition.

References:

1. **D. Hearn, M. P. Baker, Computer Graphics with OpenGL, (4e), Pearson Education, 2013.**
2. **R. C. Gonzalez, R. E. Woods, Digital Image Processing, (4e), Pearson Education, 2018.**
3. **A. K. Jain, Fundamentals of Digital Image Processing, Pearson Education, 2015.**
4. **S. Jayaraman, S. Esakkirajan, T Veerakumar, Digital Image Processing, Tata McGraw Hill Education, 2020.**

CSE3131: Computer Networks Lab [0 0 2 1]

Course Contents

Introduction to NetSim. Networking Devices: Networking Device Components, Switch and Router. **Routing:** Static Routing, RIP, OSPF. **NAT:** Configuring NAT and Troubleshooting**. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), Network Utilities.**

References:

1. **Behrouz A. Forouzan, Data Communication and Networking Protocol, TCP/IP Protocol Suite, 4th Ed., TMH, 2010**
2. **W. Stallings, Data & Computer Communications (9e), Pearson Education Inc., Noida, 2017.**

CSE3132: Software Engineering Lab [0 0 2 1]

Course Contents

Introduction to software engineering principles, Waterfall, Iterative, and Agile, techniques for gathering requirements. explore requirement engineering and analysis using IBM Engineering Requirements Management DOORS to capture requirements through UML diagrams and validate them with stakeholders. **The software design and architecture:** architectural styles, Layered, Client-Server, and Microservices, focusing on creating and evaluating design artifacts. D**esign patterns** Observer-Observable, Immediate, and Factory patterns. **Agile development** Scrum and Kanban, sprint planning and backlog management. **Software development and testing** project implementation, version control with Git/GitHub, and automated testing frameworks. **Deployment and maintenance** CI/CD pipelines, documentation strategies, and handling software changes. Case studies and live projects, where students will work with real-time customer requirements, present their projects, and discuss challenges and lessons learned.

References:

1. **Lan Sommerville, Software Engineering, (10e) Pearson Education, 2015.**
2. **Roger S. Pressman and Bruce R. Maxim, Software Engineering: A Practitioner's Approach, (10e) , McGraw Hill Education, 2019.**

CSE3140: Cloud Infrastructures & Virtualization [PE2] [3 0 0 3]

Course Contents

Introduction to Cloud Computing: Cloud Computing Overview Definition, History, Evolution, Benefits and Limitations of Cloud Computing, Key Characteristics of Cloud Computing, Cloud Service Models, IaaS, PaaS, SaaS: Concepts, Examples, and Comparison Cloud Deployment Models Public, Private, Hybrid, and Community Cloud Major Cloud Providers**. Virtualization and Cloud Fundamentals:** Virtualization Basics Definition, Types (Server, Storage, Network, and Desktop Virtualization) Virtual Machines (VMs) and Hypervisors (Type 1 vs. Type 2) Storage Virtualization Concepts and Techniques, Network Virtualization Concepts and Techniques, Benefits of Virtualization in Cloud Environments.

Cloud Compute Service: Introduction to Cloud Compute Services, Types of compute services, Amazon EC2 (Elastic Compute Cloud), Azure Virtual Machines, Oracle Compute Instances (VMs), Google Compute Engine. **Cloud Storage Service:** Introduction to Cloud Storage Services, Types of Storage services, Amazon S3, Azure Blob Storage, OCI Object Storage, Google Cloud Storage.

References :

1. **Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture (1e), PHI, 2013.**
2. **Karen Tovmasyan, Mastering AWS CloudFormation, Packt Publishing, 2020.**
3. **Michael J. Kavis, Architecting the Cloud (1e), Wiley, 2014.**

CSE3141: Predictive Analytics [PE2] [3 0 0 3]

Course Contents

Introduction: Definition, importance, and scope of predictive analytics, Predictive vs. descriptive vs. prescriptive analytics, **Role of predictive analytics in industries:** Finance, healthcare, marketing, and IT, Ethical considerations in predictive modeling. **Data Preparation and Exploration:** Data collection methods and challenges in real-world datasets, Data preprocessing: Cleaning, handling missing values, and outliers, Exploratory Data Analysis (EDA): Visualization, Feature engineering and selection for improved model performance. **Logistic Regression and Inference:** Logistic regression for binary class classification problems, Estimation of regression coefficient of logistic regression, Concept of discriminant analysis, LDA, and QDA, Multinomial Logit Regression, Generalized Linear Regression, Poisson Regression, Negative Binomial Regression. **Time Series Analysis:** Basics of time series data: Trends, seasonality, and noise, ARIMA, SARIMA, and exponential smoothing techniques, Forecasting applications: Demand prediction and stock price analysis. **Model Validation and Optimization:** Cross-validation and train-test splits, Performance metrics: RMSE, MAE, F1-score, AUC-ROC, Hyperparameter tuning: Grid search and random search, Techniques to handle overfitting and underfitting.

References:

1. **Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, Introduction to Statistical Learning with Applications in R, (1e), Springer, 2017.**
2. **Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, (3e), Shroff/O'Reilly publisher, 2022.**
3. **Foster Provost and Tom Fawcett, Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking, Shroff/O'Reilly publisher, 2013**

CSC3148: User Experience Design [PE2] [3 0 0 3]

Course Contents

Introduction to UX Design: Definition and importance of UX, Difference between UX and UI, Historical evolution of UX, The role of UX in software development; **Human-Centered Design & Design Thinking:** Human-centered design process, The Design Thinking model (Empathize, Define, Ideate, Prototype, Test), Personas and user scenarios, Use cases and task flows; **Information Architecture:** Organizing content and navigation, Site maps, card sorting, and user flows, Wireframes vs mockups vs prototypes; **Interaction Design Principles:** Affordances, signifiers, and feedback, Mental models and cognitive load, Consistency and usability heuristics (Jakob Nielsen's heuristics); **Usability Evaluation:** Types of evaluation: formative and summative, Usability testing methods, Heuristic evaluation, Surveys, interviews, and A/B testing, **Accessibility and Inclusive Design:** Principles of accessible design (WCAG overview), Designing for diverse users (age, disability, tech literacy), Assistive technologies (screen readers, keyboard navigation); **UX in the Real World:** UX in web vs mobile application, UX for enterprise systems, Agile UX and integration with development processes, Ethics in UX design (dark patterns, privacy).

References:

1. **D. A. Norman, The Design of Everyday Things, Revised and Expanded ed., Cambridge, MA: MIT Press, 2013.**
2. **J. J. Garrett, The Elements of User Experience: User-Centered Design for the Web and Beyond, (2e), Berkeley, CA: New Riders, 2010.**
3. **Cooper, R. Reimann, D. Cronin, and C. Noessel, About Face: The Essentials of Interaction Design, (4e), Indianapolis, IN: Wiley, 2014.**
4. **S. Krug, Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, (3e), Berkeley, CA: New Riders, 2013.**

CSC3147: Essentials of Cyber Security [PE2] [3 0 0 3]

Course Contents

Introduction to Cybersecurity: importance and challenges in cyber security, cyberspace, cyber threats, cyber warfare, CIA triad, cyber terrorism, cyber security of critical infrastructure, cyber security organizational implications; **Security Policies and Procedures:** developing security policies, standards, guidelines, security awareness training, employee education programs, incident response and security incident handling procedures; **Network and System Security:** overview of network security concepts and goals, evolution of network security threats and trends, network security fundamentals, wireless network security, securing network devices; **Tools and Methods used in Cybercrime :** proxy servers and anonymizers, phishing, password cracking, key loggers, spy wares, virus, worms, trojan horse, backdoors, steganography, SQL injection, buffer overflow attacks on wireless network; **Cyber Ethics and it's Various Laws:** introduction to cyber laws, e-commerce, e-governance, certifying authority and controller, computer offences and its penalty under IT act, intellectual property rights in cyberspace.

References:

1. **Charles J. Brooks, Christopher Grow, Philip Craig, and Donald Short Cybersecurity Essentials (1e), Wiley, 2018.**
2. **Nina Godbole and Sunit Belpure Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley, 2019.**
3. **B. B. Gupta, D. P. Agrawal, Haoxiang Wang Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, 2018.**
4. **William Stallings, and Lawrie Brown Computer Security: Principles and Practice (4e), Pearson Education, 2018.**
5. **Pavan Duggal Cybersecurity and Cyberlaw, Notion Press, 2021.**

CSE3142: Android App Development [PE3] [3 0 0 3]

Course Contents

Introduction to Android Development: Overview of Mobile App Development, Introduction to Android OS and its Architecture, Setting up the Android Development Environment (Android Studio),Understanding Android Project Structure, Basics of XML for UI Design **Android UI Components and Layouts :**View and View Groups, Common UI Widgets (Text View, Edit Text, Button, Image View, etc.),Layouts: Linear Layout, Relative Layout, Constraint Layout, Recycler View and List View, Handling User Input and Event Listeners **Activities, Intents, and Navigation:** Activity Lifecycle and State Management, Implicit and Explicit Intents, Passing Data Between Activities, Fragments and Fragment Management, Navigation Components and Multi-Screen Applications **Data Persistence and**

Storage: Shared Preferences for Data Storage, SQLite Databases in Android, Room Persistence Library, File Storage, Basics of Cloud Integration (Firebase Realtime Database)**Advanced Topics in Android Development:** Networking: Using APIs and JSON Parsing, Background Tasks: Async Task, Work Manager, and Services, Notifications and Broadcast Receivers, Integrating Sensors (e.g., GPS, Accelerometer),Basics of Jetpack Components

References:

1. **Android Developers Official Documentation (developer.android.com).**
2. **Joseph Annuzzi Jr., Lauren Darcey, and Shane Conder, Android Application Development, Pearson Education, 2014.**
3. **B. Phillips, C. Stewart, B. Hardy, and K. Marsicano, Android Programming: The Big Nerd Ranch Guide, (4e), Big Nerd Ranch LLC, 2019.**
4. **Phil Dutson, Android Development Patterns: Best Practices for Professional Developers, (1e), Addison Wesley,2016.**
5. **Dawn Griffiths and David Griffiths , Head First Android Development, (2e), Shroff/O'Reilly, 2017.**

CSE3143: Advanced Java [PE3] [3 0 0 3]

Course Contents

Java Fundamentals Recap: Core Java Overview, OOP Concepts (Inheritance, Polymorphism, Abstraction, Encapsulation), Package, Exception Handling, Collections Framework, Multithreading and Concurrency. **Java Database Programming:** Basics of Java database, Java Database Connectivity (JDBC), Different Types of Drivers of JDBC. **Introduction to Swing:** Swing Basics, Key Swing Features, Components and Containers, Event Handling, Various Swing Components, Writing Swing Application, Database access. **Web Technologies**: Servlets, JavaServer Pages (JSP), Session Management. **Java Web Framework:** Spring: Overview of Spring, Spring Architecture, bean basics and life cycle, Dependency Injection, XML Configuration on Spring, Event handling in Spring, Aspect-oriented Spring, Managing Database, Managing Transaction Introduction to Distributed Applications, Spring Boot, Spring Cloud.

Reference Book:

1. **Bryan Basham, Kathy Sierra, Bert Bates, Headfirst Servlets and JSPs, O'Rilley Media.**
2. **Jim Keogh, Complete Reference J2EE, Tata McGraw Hill.**
3. **Core and Advanced Java, BlackBook, Dreamtech Press.**
4. **Herbert Schildt, Java: The Complete Reference, (13e), McGraw Hill, 2024.**
5. **Bruce Eckel, Thinking in Java (4th), Pearson 2006.**

CSE3144: Advanced Data Structures [PE3] [3 0 0 3]

Course Contents:

Foundations of Advanced Data Structures: Introduction, Amortized Analysis, External Sorting (Tournament trees, buffering, run generation, and Huffman trees). **Advanced Tree Data Structures**: AVL Trees, Red-Black Trees, Splay Trees, B-Trees, B+-Trees, and B*-Trees, Segment Trees and Interval Trees, Tries and Digital Search Trees, Suffix Trees. **Advanced Heap and Priority Queue Structures**: Binary Heaps, Binomial Heaps, and Fibonacci Heaps, Pairing Heaps and Double-Ended Priority Queues. **Spatial Data Structures**: k-d Trees, Quad Trees and Oct Trees, BSP Trees (Binary Space Partitioning Trees, R-Trees. **Specialized Data Structures**: Bloom Filters, Priority Search Trees, Persistent Data Structures, Disjoint Set Union (Union-Find).

References:

1. **Mark Allen Weiss, Data Structures and Algorithm Analysis in C++ (4e), Pearson, 2014.**
2. **Thomas H. Cormen et al., Introduction to Algorithms (4e), MIT Press, 2022.**
3. **Peter Brass, Advanced Data Structures, Cambridge University Press, 2011.**
4. **Robert Sedgewick and Kevin Wayne, Algorithms (4e), Addison-Wesley, 2011.**
5. **Michael T. Goodrich and Roberto Tamassia, Data Structures and Algorithms in Java (6e), Wiley, 2014.**

CSE3145: Game Theory [PE3] [3 0 0 3]

Course Contents:

Fundamental Concepts: Introduction to strategic decision-making using Game Theory. Games and decisions, strategies, costs and payoffs, basic solution concepts. Learning in games. **Static and Dynamic Games:** Zero-sum games (maximin and minimax strategies), normal form games (dominance, Nash equilibrium, mixed strategies, iterated dominance), extensive form games (subgame perfection, sequential equilibrium). **Correlated Equilibrium:** Concept, computation, and applications. **Repeated Games and Bargaining:** Folk Theorem, tacit collusion, Nash and Rubinstein bargaining models, negotiation strategies. **Mechanism Design:** Task sharing, Pareto optimality, Groves payments, VCG mechanism. Applications in real-world scenarios like auctions, slot allocation, and Internet-based systems.

References:

1. **Michael Maschler, Eilon Solan, Shmuel Zamir, Game Theory, Cambridge University Press, 2013.**
2. **Y. Shoham and K. Leyton-Brown, Multiagent Systems, Cambridge University Press, 2008.**
3. **Y. Narahari, Game Theory and Mechanism Design, World Scientific and IISc Press, 2014.**

CSE3146: Software Testing and Automation [PE3] [3 0 0 3]

Course Contents

Introduction and concept of Software testing: in SDLC, Purpose, Principles, The Tester's Role in a Software Development Organization, Software Testing Life Cycle, Basic definitions: Error, bug, fault, mistake, failure and defects. Economics of Software Review. Test cases- Creation, design, execution, Test Bed. **Unit Testing**: Static and Dynamic Testing, mutation testing, debugging. White Box Testing-Control flow testing, control flow graph, path selection criterion, Code complexity Testing. Black Box Testing- Boundary Value, Equivalence Partitioning, Decision table-based testing, State Based Testing, Domain Testing. **Integration and System Testing:** Concept of Integration Testing, Different Types of Interfaces and Interface Errors, Granularity of System Integration Testing**,** Functionality Tests and its types, Regression Testing; Selection, Minimization and Prioritization of Test Cases for Regression Testing, Acceptance Testing. **Test Automation:** skill needed, scope of automation, design and architecture for automation, challenges in automation. Test metrics and measurements; project, progress and productivity metrics. Introduction to automation tools like Selenium, Postman, Katalon Studio.

References:

1. **Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black, Foundations Of Software Testing, ISTQB Certification, Cengage Learning India Pvt. Ltd., 2020.**
2. **Paul C. Jorgensen, Software Testing- A Craftsman's Approach Fourth Edition, CRC Press, Taylor and Francis Group, 2013.**
3. **Gerald D. Everett, Raymond McLeod, Jr., Software Testing: Testing Across the Entire Software Development Life Cycle, IEEE Press, John Wiley, and Sons, 2007.**

VI Semester

CSE3201: Machine Learning [3 1 0 4]

Machine Learning: Introduction and need; history; applications; advantages and disadvantages; limitations of machine learning; types of machine learning problems; classification and regression; supervised learning versus unsupervised learning; reinforcement learning; **Supervised Learning**: linear regression, logistic regression, cost function, activation and loss function, backpropagation, overfitting; underfitting and regularization (L1/L2), optimization techniques, hyperparameter tuning, performance metrics, validation methods, naïve Bayes, decision tree, SVM, KNN; **Ensemble Methods**: bagging, boosting, and stacking; **Unsupervised Learning**: K-means clustering, hierarchical clustering, DBSCAN, K-self organizing map; **Dimensionality Reduction**: linear discriminant analysis, principal component analysis (PCA), independent component analysis; **Deep Learning**: distributed representation, embedding, convolutional neural networks, recurrent neural networks, reinforcement learning.

References:

1. **Stephen Marsland, Machine Learning – An Algorithmic Perspective, (2e), Chapman and Hall / CRC, 2014.**
2. **Tom Mitchell, Machine Learning, (1e), McGraw-Hill, 2017.**
3. **Christopher Bishop, Pattern Recognition and Machine Learning, (1e), Springer, 2009.**
4. **Ethem Alpaydin, Introduction to Machine Learning, The MIT Press, 2014, ISBN 978-0-262-02818-9.**
5. **Ian GoodFellow, Yoshua, Bengio, Aaron Courville, Deep Learning, MIT Press Book,2016.**
6. **Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar, Foundations of Machine Learning, MIT Press (MA), 2012.**

CSE3240: Cloud Applications [PE4] [3 0 0 3]

Course Contents

Compute & Serverless Architectures: Virtual Machines, Serverless functions - AWS Lambda, Azure Functions, GCP Cloud Functions, Event-driven design principles & microservices component delivery; **Storage, Database & Data Transfer**: Object, block & file storage, Relational vs NoSQL database services, Migration tools & data transfer strategies - DataSync, Database Migration Service, Storage Transfer Service, content-delivery networks; **Resiliency, High Performance & Disaster Recovery**: Scaling patterns for high throughput & low latency, Load balancing, caching strategies, Designing for fault-tolerance & automated fail-over, Backup, replication, disaster-recovery planning; **Security, Governance & Cost Management**: IAM, Policy-driven governance & monitoring - CloudWatch, Azure Monitor, Stackdriver, Billing models, budgeting & cost optimization techniques; **Management, Networking & Component Delivery**: Virtual networking, hybrid-cloud connectivity - VPN, Direct Connect, ExpressRoute, Infrastructure as Code & automated deployments - Terraform, CloudFormation, ARM templates, Service mesh & API gateway patterns.

References:

1. **Simy Joy,Payal Anand, Priya Nair Rajeev, Cloud Computing: Concepts, Technology & Architecture, Pearson Education, 2014**
2. **Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017.**
3. **Sam Newman, Building Microservices: Designing Fine-Grained Systems, (2e), O'Reilly Media, 2021.**

CSE3242: Web Framework [PE4] [3 0 0 3]

Course Contents

Introduction: Frontend, Backend, and Full Stack development; Introduction to web servers, browsers, and the client-server model; web development tools and environments; **Web Development**: HTML, Semantic Tags Forms and Validation; CSS, uses of CSS in web pages, Layout Design, CSS Framework- Bootstrap, Tailwind CSS, Pure CSS, Materialize CSS; JavaScript, Operators, Loops, Functions, Array; XML-DTD, DOM; **Front End Framework**: JS Framework- React, Next, Angular, NodeJS; Version Control System- Git Control, VCS Hosting-GitHub; **Backend Framework**: Overview of Backend Frameworks : Django, Express, Laravel, Understanding Full-stack frameworks and their applications; Model-View-Controller (MVC) Architecture, Routing and URL Mapping; **Database**: Database Connectivity-SQL, NOSQL, **Security:** HTTPS, SSL/TSL, CORS; **Deploymen**t: AWS, Oracle Cloud, MS Azure, Performance, Testing.

References:

1. **T. Felke-Morris, Web Development and Design Foundations with HTML5, (9e), Pearson Publication, 2018.**
2. **A. K. Verma, Full Stack Development: Frontend, Backend, Database & Deployment, (1e), Khanna Publishing, 2017.**
3. **Xavier, Web Technology and Design, (3e), New Delhi: New Age International Pvt. Ltd., 2024.**
4. **C. Bates, Web Programming: Building Internet Applications, (2e), Wiley Dreamtech Publication, 2002.**
5. **R. Kamal, Internet and Web Technologies, (2e), McGraw Hill Education Publication, 2017.**
6. **J. Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, (5e), O'Reilly Media Publication, 2018.**
7. **I. Bayross, Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP, (4e), New Delhi: BPB Publications, 2019.**
8. **J. Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, (5e), Sebastopol, CA: O'Reilly Media, 2018.**

CSE3241: Computer Vision [PE4] [3 0 0 3]

Course Contents

Fundamentals of Image Formation and Processing: Image formation principles: camera models, light, shading, and color; Image sampling, interpolation, and transformations; Linear filtering, convolution, and Fourier analysis; Histogram processing, enhancement, and restoration techniques; **Feature Detection and Description**: Edge detection: Canny, LoG, DoG methods; Corner detection: Harris, Hessian-Affine; Keypoint descriptors: SIFT, SURF, HOG; **Image Segmentation and**

Grouping: Segmentation techniques: region growing, edge-based; Texture segmentation and analysis using Gabor filters; Grouping methods: least squares fitting, RANSAC; Image alignment and stitching; **Geometric Vision and 3D Reconstruction**: Camera calibration and epipolar geometry; Two-view and multi-view stereo vision; Structure from motion and 3D reconstruction frameworks; Shape-from-X techniques: shading, texture, motion, and edges; **Recognition and Learning in Vision**: Statistical learning frameworks: supervised, unsupervised, semi-supervised; Classification techniques: Bayes, KNN, ANN models; Dimensionality reduction: PCA, LDA, ICA; Object detection and image classification methodologies; **Advanced Topics in Computer Vision**: Motion analysis: optical flow, KLT tracking, dynamic stereo; Deep learning in vision: CNNs, object detection, segmentation; Vision and language integration, video analysis.

References:

1. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, (2e), Pearson Education, 2011.
2. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited, 2011.
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, (4e),Pearson Education, 2018.

CSE3243: Ethical Hacking [PE4] [3 0 0 3]

Course Contents

Introduction to Ethical Hacking: scope, legality, hacker types, professional ethics; **Hacking Methodology & Foot printing**: reconnaissance, Google dorking, WHOIS, NSlookup; **Scanning and Enumeration**: Nmap, Hping, Netcat, SMB & SNMP enumeration; **System Hacking**: password cracking, privilege escalation, keylogging, backdoors; **Network Hacking**: ARP spoofing, DNS poisoning, packet sniffing (Wireshark), MITM attacks; **Web App Hacking**: SQL injection, XSS, CSRF, command injection, file inclusion; **Wireless and Mobile Hacking**: WPA/WPA2 cracking, Android attack vectors; **Penetration Testing & Reporting**: PT phases, report writing, risk rating (CVSS); **Hands-on Labs**: Metasploit, Burp Suite, Kali Linux tools, HackTheBox/CTF practice.

References:

1. **Michael T. Simpson, Kent Backman, and James E. Corley** ***Hands-on Ethical Hacking and Network Defense*** **(2e), Delmar Cengage Learning, 2010.**
2. **Patrick Engebretson The** ***Basics of Hacking and Penetration Testing*** **(3e), Elsevier, 2013.**
3. **Dafydd Stuttard, and Marcus Pinto** ***The Web Application Hacker's Handbook Finding and Exploiting Security Flaws*** **(1e), Wiley, 2013.**
4. **Wil Allsopp,** ***Advanced Penetration Testing: Hacking the World's Most Secure Networks*** **(1e), Wiley, 2017.**
5. **Justin Seitz,** ***Black Hat Python: Python Programming for Hackers and Pentesters*** **(1e), No Starch Press, 2014.**
6. **https://www.coursera.org/learn/certified-ethical-hacking-v12-ethical-hacking-fundamentals#modules**

CSE3245: Cloud Security and Privacy [PE5] [3 0 0 3]

Course Contents

Cloud Security Fundamentals: Overview of cloud security and privacy challenges, Shared responsibility model (AWS, Azure, Google Cloud), Common threats and vulnerabilities in cloud environments, AWS IAM, Azure Active Directory, Google Cloud IAM, Multi-Factor Authentication (MFA), Role-Based Access Control (RBAC) implementation; **Cloud Infrastructure Security**: Securing virtual machines and containers, Network security: AWS VPC, Azure Virtual Network, Google Cloud VPC, Firewalls, load balancers, security groups, Cloud security architecture design principles; **Data Protection and Privacy**: Data security and encryption methods (AWS KMS, Azure Key Vault, Google Cloud KMS), Secure data storage solutions (AWS S3, Azure Blob Storage, Google Cloud Storage), Privacy protection regulations (GDPR, HIPAA), Techniques for data anonymization and privacy preservation; **Security Monitoring and Incident Response**: Cloud security monitoring tools (AWS CloudWatch, Azure Security Center, Google Security Command Center), Logging, auditing, and alerting mechanisms, Incident response and disaster recovery plans, Security automation and threat detection; **Compliance, Audit, and Risk Management**: Understanding cloud compliance frameworks (ISO 27001, SOC 2, PCI DSS), Cloud audit methodologies, Risk assessment and management strategies, Best practices for ensuring continuous compliance.

References:

1. **Tim Mather, Subra Kumaraswamy and Shahed Latif, Cloud Security and Privacy, O'Reilly, 2011.**
2. **Raj Kumar Buyya , James Broberg, andrzejGoscinski, Cloud Computing, Wiley, 2013.**
3. **Dave shackleford, Virtualization Security, SYBEX a wiley Brand, 2013.**
4. **Simon Gallagher and Aidan Dalgleish, VMware Private Cloud Computing with vCloud Director, SYBEX, A Wiley Brand, 2013.**
5. **NIST SP 800-144: Guidelines on Security and Privacy in Public Cloud Computing, 2011.**

CSE3247: Agile Methodology [PE5] [3 0 0 3]

Course Contents

Introduction to Agile: Comparison of Agile vs traditional methods, process tailoring, principles, values, roles, challenges, and business benefits of software agility; **Project Planning**: Structure of an agile team (Programmers, Managers, Customers), User Stories – definition, characteristics, and content; **Software Process Models**: Overview of Agile methodologies – Scrum, XP, Lean, Kanban; Agile Manifesto, Scrum process, roles (Product Owner, ScrumMaster, Team, Project Manager, Product Manager, Architect), events and artifacts, product vision, stakeholders, backlog creation, and grooming; **Agility And Knowledge Management**: Agile information systems, Agile decision making, Earl's schools of KM, KM in software engineering – managing software knowledge – challenges of migrating to agile methodologies, Story-card Maturity Model (SMM); Agile requirements – User personas, story mapping, 3Cs, INVEST, acceptance criteria; Tools: Agile tracking tools (e.g., JIRA), Scaled agile frameworks (SAFe, Scrum@Scale, Disciplined Agile); **Agile Forecasting and Project Management**: Information radiators, velocity, progress tracking, project forecasting, product backlog, reviews, retrospectives, and iteration planning; **Sprints**: Overview of iterations/sprints, velocity determination, sprint planning, development, testing, daily stand-ups, progress tracking, burn-down charts; **Testing**: Impact of Agile lifecycle on testing, Test-Driven Development (TDD), acceptance tests, continuous integration, code refactoring, regression tests, and security testing, test automation, and Agile testing principles (Selenium); **Agility And Quality Assurance**: Agile Interaction Design, Agile product development, Agile Metrics, Feature Driven Development (FDD) - Financial and Production Metrics in FDD, Agile approach to Quality Assurance, Issues and Challenges - Agile approach to Global Software Development.

Reference Books

1. **Jim Highsmith, Agile Project Management: Creating Innovative Products, (2e), Addison-Wesley Professional, 2009.**
2. **James A. Crowder, Shelli Friess, Agile Project Management: Managing for Success, Springer, 2014.**

3. **Janet Gregory, Lisa Crispin, More Agile Testing: Learning Journeys for the Whole Team, (1e), Addison Wesley, 2015.**
4. **Sricharan Vadapalli, DevOps: Continuous Delivery, Integration, and Deployment with DevOps, (1e), Packt, 2018.**
5. **Thomas Uphill, John Arundel, Neependra Khare, Hideto Saito, Hui-Chuan Chloe Lee, Ke-Jou Carol Hsu, DevOps: Puppet, Docker, and Kubernetes, (1e), Packt, 2017.**
6. **Robin Nixon, Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5, (5e), O'Reilly Publications, 2018.**

CSE3246: Natural Language Processing [PE5] [3 0 0 3]

Course Contents

Introduction to NLP: Fundamentals of human language processing, semantics and pragmatics in NLP, challenges in NLP (ambiguity and context), stages of NLP, parts of speech (POS), phrase structure, statistical foundations, entropy, perplexity, cross-entropy, character encoding, corpora and morphological analysis (inflectional and derivational), text preprocessing (tokenization, sentence segmentation, etc.); **Corpus Linguistics and Language Modeling**: Corpus design and annotation, statistical approaches to language modeling, n-gram models, collocations, word sense disambiguation (WSD), supervised and unsupervised disambiguation, dictionary and thesaurus-based methods, second-language corpus-based WSD; **POS Tagging and Sequence Labeling**: Markov Models and Hidden Markov Models (HMM), parameter estimation, Viterbi algorithm, multiple observation HMMs, information sources in tagging, applications of POS tagging; **Parsing and Syntax Processing**: Probabilistic Context-Free Grammars (PCFGs), parsing algorithms, inside-outside algorithm, disambiguation using parsing, Treebanks, dependency parsing, phrase structure grammar, lexicalized and derivational parsing models; **Advanced NLP and Deep Learning**: Syntax and semantics in real-world applications, machine translation (word/sentence alignment), information extraction, text mining, question answering systems, sentiment analysis, social network analysis, introduction to deep learning models in NLP – CNN, RNN, LSTM, encoders, and decoders.

References:

1. **Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 2003.**
2. **D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, (3e), Upper Saddle River, NJ: Pearson Prentice Hall, 2025.**

CSE3244: Secure Programming [PE5] [3 0 0 3]

Course Contents

Motivation for Secure Programming: Software Security – Why? Example Vulnerabilities; **Introduction**: Security goals, Secure system design, Secure design principles, Worms, and other malware; **Secure Programming Techniques**: Anatomy of a buffer overflow, Safe string libraries, Stack guard, Static analysis tools, Heap-based overflow, other memory corruption vulnerabilities, SQL Injection attack scenario and solutions, Password security; **Cross-Domain**

Security in Web Applications: Interaction between webpages from different domains, Attack patterns, preventing cross-site request forgery, Cross-site script inclusion, Cross-site scripting; **Other Web Vulnerabilities**: Cookie protocol problems, SSL/TLS vulnerabilities, Session hijacking, Guninski attack, Defenses; **Trusted Execution Environment**: Case study on Trust Zone, Security vulnerability tools, Exploit development with metasploit.

References:

1. **N. Daswani, C. Kern, A. Kesavan, Foundations of Security: What Every Programmer Needs to Know, (1e), Berkeley, CA: Apress, 2007.**
2. **J. C. Foster, V. T. Liu, Writing Security Tools and Exploits, (1e) Syngress Publishing, 2006.**
3. **J. Ericson, Hacking: The Art of Exploitation, (2e), No Starch Press, 2008.**
4. **C. Anley, J. Heasman, F. Linder, G. Richarte, The Shellcoder's Handbook: Discovering and Exploiting Security Holes, (2e), Addison-Wiley, 2011.**
5. **Coursera course: https://www.coursera.org/specializations/intro-cyber-security.**

CSE3249: Digital Forensics and Cyber Crimes[PE6] [3 0 0 3]

Course Contents:

Introduction to Digital Forensics: Cybercrimes, Overview of hardware and operating systems, types of digital evidence, Logical structure of storage media/devices, Branches of Digital Forensics, Phases of Digital forensics; Seizure, Chain of custody, Various File Systems, Type of Acquisition live vs standalone machine, memory volatility and precautions, Hidden/deleted Data recovery HPA, DCO; Computer forensics tools (Encase, FTK, Autopsy, ProDiscover, Sleuth Kit, X-Ways Forensics, Volatility, WinHex); **Network Forensics**: Cyber-attacks on network, network-based digital evidence, Acquisition, live acquisitions, Traffic Analysis, wireless network forensics, Event log analysis, Intrusion detection, Reconstructing web browsing, email investigation; Network forensics tools (tcpdump, Wireshark, Network Miner, Snort, Metasploit, SQLMap); **Mobile Forensics**: Different OS and Memory in Mobile phones, collecting evidence, preservation methods, interpretation of digital evidence on mobile networks; Mobile Forensics Tools (Cellebrite UFED, Oxygen Forensics, Magnet AXIOM, XRY, MOBILedit Forensic, MailXaminer); **IoT Forensics**: Analyzing evidence from IoT devices and smart systems, Forensic Toolkit for IoT, Cellebrite IoT Module, IoT-Analyzer; **Cyber Laws and Ethics**: Indian IT Act, Intellectual property right, Criminal Justice system for forensics, Audit/investigative, Investigative procedures/standards for extraction, preservation, and deposition of legal evidence in a court of law, challenges in court of law.

References:

1. **B. Nelson, A. Phillips, and C. Steuart, Guide to Computer Forensics and Investigations, (6e), Boston, MA: Cengage Learning, 2018.**
2. **S. H. Davidoff and J. Davidoff, Network Forensics: Tracking Hackers through Cyberspace, (1e), Boston, MA: Addison-Wesley, 2012.**
3. **R. Montasari, H. Jahankhani, R. Hill, and S. Parkinson, Digital Forensic Investigation of Internet of Things (IoT) Devices, (1e), Cham, Switzerland: Springer, 2020.**
4. **K. Barmpatsalou, T. Cruz, E. Monteiro, and P. Simoes, Mobile Device Forensics: A Practitioner's Guide to iOS and Android Investigations, (1e), Hoboken, NJ: Wiley, 2022.**
5. **M. E. Alex and R. Kishore, Forensic Investigations in Cloud Computing Environments, (1e), Hershey, PA: IGI Global, 2021.**

CSE3250: Fog and Edge Computing [PE6] [3 0 0 3]

Course Contents:

Emerging Computing Technology: Introduction: IoT, Fog Edge Cloud (FEC) Computing, MIST computing, Connectivity, communication and data protocols; **Fog Computing Conceptual Model**: Architecture, layers, Fog Node: attribute, architecture; Data management: life cycle, characteristics, storage and placement; **Management and Orchestration of Network**: Introduction of 5G network, benefits, slicing; **Software Defined Network:** introduction, operation, slicing, controller, open flow components, flowable; **Network Function Virtualization:** architecture, management and operations; **FEC Use Cases**: Health Care, Traffic Monitoring, Self-driving cars, Smart Cities.

References:

1. **Satish Narayana Srirama and Rajkumar Buyya, Fog and Edge Computing: Principles and Paradigms, Wiley, 2019.**
2. **Sudip Misra, Subhadeep Sarkar, Subarna Chatterjee, Sensors, Cloud, and Fog: The Enabling Technologies for the Internet of Things, CRC Press, 2019**
3. **K. Anitha Kumari, G. Sudha Sadasivam, D. Dharani, M. Niranjana Murthy, Edge Computing: Fundamentals, Advances and Applications, CRC Press, 20220**
4. **https://nptel.ac.in/courses/106105166, Introduction to internet of things.**

CSE3251: Deep Learning [PE6] [3 0 0 3]

Course Contents

Introduction to Deep Learning: Evolution from classical machine learning to deep learning, Perceptron, Multilayer Perceptron, Neural Networks, Impact of deep learning in computer vision and NLP; Artificial neurons, activation functions, loss functions, Feedforward neural networks, Gradient Descent, Backpropagation; **Training Deep Neural Networks**: Optimization techniques: SGD, Momentum, RMSprop, Adam, Vanishing/exploding gradients, Regularization techniques: Dropout, Weight Decay, Batch normalization, learning rate schedules, Hyperparameter tuning: manual vs. automated (grid search, random search); **Convolutional Neural Networks (CNNs)**: Motivation for CNNs in image processing, Convolution operation, feature maps, strides, padding, Pooling operations (max, average), **CNN architecture:** convolutional, pooling, and fully connected layers, Regularization in CNNs: dropout, batch normalization, CNN models: AlexNet, VGGNet, ResNet, Applications: Image classification, object detection, segmentation, transfer learning; **Recurrent Neural Networks (RNNs)**: Sequential data and challenges, Basic RNN, LSTM, GRU, Encoder-Decoder models.

References:

1. **Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.**
2. **T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, (2e), Springer,2017.**
3. **D. Koller, and N. Friedman Probabilistic Graphical Models, MIT Press,2010.**
4. **Francois Chollet, Deep Learning with Python, Manning Publications, 2017.**
5. **Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, (2e), O'Reilly Media, 2019.**
6. **Ashish Vaswani et al., Attention Is All You Need, (Research Paper), Advances in Neural Information Processing Systems (NIPS), 2017.**

CSE3252: UI/UX Design [PE6] [3 0 0 3]

Course Contents

Introduction to UX Design: Definition and importance of UX, Difference between UX and UI, Historical evolution of UX, The role of UX in software development; **Human-Centered Design & Design Thinking:** Human-centered design process, The Design Thinking model (Empathize, Define, Ideate, Prototype, Test), Personas and user scenarios, Use cases and task flows; **Information Architecture:** Organizing content and navigation, Site maps, card sorting, and user flows, Wireframes vs mockups vs prototypes; **Interaction Design Principles:** Affordances, signifiers, and feedback, Mental models and cognitive load, Consistency and usability heuristics (Jakob Nielsen's heuristics); **Usability Evaluation:** Types of evaluation: formative and summative, Usability testing methods, Heuristic evaluation, Surveys, interviews, and A/B testing**, Accessibility and Inclusive Design:** Principles of accessible design (WCAG overview), Designing for diverse users (age, disability, tech literacy), Assistive technologies (screen readers, keyboard navigation)**; UX in the Real World:** UX in web vs mobile application, UX for enterprise systems, Agile UX and integration with development processes, Ethics in UX design (dark patterns, privacy).

References:

1. **R. Hartson and P. S. Pyla, The UX Book: Agile UX Design for a Quality User Experience, (2e), Burlington, MA: Morgan Kaufmann, 2019.**
2. **B. Nunnally and D. Farkas, UX Research: Practical Techniques for Designing Better Products, (1e), Sebastopol, CA: Rosenfeld Media, 2016.**
3. **J. Tidwell, C. Brewer, and A. Valencia, Designing Interfaces, (3e), Sebastopol, CA: O'Reilly Media, 2020.**
4. **R. Unger and C. Chandler, A Project Guide to UX Design: For User Experience Designers in the Field or in the Making, (3e), Berkeley, CA: New Riders, 2023.**
5. **Link for Coursera:** https://www.coursera.org/specializations/ui-ux-design?utm_source=chatgpt.com#courses

CSE3253: DevOps [PE6] [3 0 0 3]

Course Contents

Introduction: overview of DevOps, market trends, skills, delivery pipeline, ecosystem; Version Control tools, working with remote repositories; **Agile software development**; **Continuous Integration (CI)**: branching and merging in Git, workflows, Git cheat sheet, introduction to Jenkins, Jenkins management; **Continuous Testing (CT):** need, Selenium and Webdriver; **Continuous Deployment**: introduction to container, life cycle, working with docker, publishing image; Docker ecosystem, Kubernetes; **Configuration Management (CM)**: Puppet installation and configuration, master and agent setup, puppet module, node classification; **Continuous Monitoring**: Nagios installing, Plugins(NRPE) and objects, Nagios commands and notification; Cloud platforms: AWS, Azure, Google Cloud, and their DevOps services.

References:

1. **K. Hightower, B. Burns, and J. Beda, Kubernetes Up & Running: Dive into the Future of Infrastructure, (1e), O'Reilly Media, 2018.**

2. **J. Geerling, Ansible for DevOps: Server and Configuration Management for Humans, (1e), Leanpub, 2019.**
3. **J. McAllister, Mastering Jenkins, (1e), Birmingham, UK: Packt Publishing, 2018.**
4. **G. Kim, J. Humble, P. Debois, and J. Willis, The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, (2e), Portland, OR: IT Revolution Press, 2021.**
5. **M. Krief, Learning DevOps: Continuously Deliver Better Software, (1e), Birmingham, UK: Packt Publishing, 2019.**
6. **N. Poulton, Docker Deep Dive, (1e), Leanpub, 2023.**
6. **https://www.coursera.org/learn/intro-to-devops?specialization=devops-and-software-engineering. (Coursera Link)**

CSE3254: Analytics Insights [PE6] [3 0 0 3]

Course Contents

DBMS: RDBMS Concepts, Data Definition Language (DDL), Data Manipulation Language (DML), Select Statements, Single-Row Functions, Group Functions, Joins, Subqueries, Constraints; **Programming Fundamentals Python:** Introduction to Python, Conditional Statements, Iterative statements, Strings, Data Structures in Python, Functions, Modules and Packages, Working with files, Classes and objects, Errors and Exception Handling, Regular expression; **Data Science & Data Engineering:** Data Processing using Pandas, Introduction to Pandas Data Structures, Indexing and Selecting, Column and Row Operations, Descriptive Statistics, Exploratory Data Analysis, Introduction to EDA, Univariate Analysis, Bivariate Analysis, Multivariate Analysis, Web Scraping using Python, Machine Learning Fundamentals, General Concept of Data Warehouse, Dimensional Modeling, Online Analytical Processing (OLAP), ETL, Data Mining, Business Intelligence.

References:

1. **Martin Martin C. Brown, Python: The Complete Reference, (4e), New York, NY: McGraw Hill Education, 2018.**
2. **Joe Reis and Matt Housley, Fundamentals of Data Engineering: Plan and Build Robust Data Systems, (1e), Sebastopol, CA: O'Reilly Media, 2022.**
3. **Abraham Silberschatz, Henry F. Korth & S. Sudarshan, Database System Concepts, (7e), New York, NY: McGraw Hill Education, 2020.**
4. **https://pwc.tekstac.com/login/index.php**

VII Semester

CSE4141 : Cloud Automation Tools [PE7] [3 0 0 3]

Course Contents

Fundamentals of DevOps: Introduction to DevOps: principles and practices, Importance of DevOps in modern cloud applications, Cultural and technical aspects of DevOps, Agile methodologies and DevOps; **Continuous Integration and Delivery:** CI/CD Concepts, Version Control Systems: Git, GitHub, GitLab, **CI/CD Tools:** Jenkins, GitLab CI, Azure DevOps, Practical implementation of CI/CD pipelines; **Containerization and Orchestration:** Container fundamentals: Docker, Dockerfile, Docker

Compose, Kubernetes: concepts, components, deployment, Managed Kubernetes services (EKS, AKS, GKE); **Infrastructure as Code and Automation:** Infrastructure as Code (IaC) concepts, Terraform and CloudFormation, Configuration Management: Ansible, Puppet, Automation best practices; **Monitoring and Logging:**Cloud monitoring essentials: AWS CloudWatch, Azure Monitor, Application performance monitoring: Prometheus, Grafana, **Logging tools:** ELK Stack (Elasticsearch, Logstash, Kibana), Alerting and incident management; **Security, Compliance, and Optimization:** DevSecOps: integrating security into DevOps, Cloud security tools and compliance frameworks, Optimization of cloud resources, Disaster recovery and business continuity strategies.

References:

1. **G. Kim, J. Humble, P. Debois, and J. Willis, The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, (2e), Portland, OR: IT Revolution Press, 2021.**
2. **L. Bass, I. Weber, and L. Zhu, DevOps: A Software Architect's Perspective, (1e), Boston, MA: Addison-Wesley Professional, 2015.**
3. **K. Morris, Infrastructure as Code: Managing Servers in the Cloud, (1e), Sebastopol, CA: O'Reilly Media, 2016.**

CSE4149: Software Verification and Testing [PE7] [3 0 0 3]

Course Contents

Introduction to Software Verification: Software Quality and its Importance, Definitions: Verification vs. Validation, V&V in the Software Development Life Cycle (SDLC), V&V Planning and Documentation, Overview of Testing Levels: Unit, Integration, System, Acceptance; **Static Verification Techniques:** Walkthroughs, Inspections, and Code Reviews, Formal Methods: Model Checking, Hoare Logic, Symbolic Execution, Static Analysis Tools, Types of Static Code Metrics, Control Flow and Data Flow Analysis, Identifying Bugs and Vulnerabilities Using Static Verification, Formal Verification Techniques, Limitations of Static Verification Methods; **Dynamic Testing and Test Design:** Testing Techniques: Black Box and White Box, Structural Testing: Control Flow, Data Flow, Functional Testing: Equivalence Class Partitioning, Boundary Value Analysis, Non-Functional Testing: Performance, Usability, Security, Test Case Design and Coverage Criteria, Tools for Formal Verification (SPARK, Z3), Frameworks for Verification, Integration of Verification Tools in Build and Deployment Pipelines; **Test Process and Management:** Test Planning, Estimation, and Scheduling, Test Strategies and Test Plan Development, Traceability Matrix, Test Metrics, Defect Life Cycle, Defect Tracking Tools (JIRA, Bugzilla), Risk-Based Testing and Prioritization; **V&V in Agile and DevOps:** Agile Testing Life Cycle, Role of QA in Agile Teams, Test-Driven Development (TDD) and Behaviour-Driven Development (BDD), Exploratory and Ad-Hoc Testing, DevOps and Continuous Validation; **Software Quality and Standards:** Software Quality Assurance (SQA), Standards for Testing and Quality (ISO/IEC 25010, IEEE 829), Software Metrics: Defect Density, Test Effectiveness, Code Coverage, Case Studies on Quality Failures.

References:

1. **M. S. Fisher, Software Verification and Validation: An Engineering and Scientific Approach, (1e), Berlin, Germany: Springer Science & Business Media, 2007.**
2. **P. Ammann and J. Offutt, Introduction to Software Testing, (2e), Cambridge, UK: Cambridge University Press, 2016.**
3. **A. Dasso, Verification, Validation and Testing in Software Engineering, (1e), Hershey, PA: IGI Global, 2006.**

4. **M. Haug, E. W. Olsen, and L. Consolini, Software Quality Approaches: Testing, Verification, and Validation: Software Best Practice 1 (Vol. 1), (1e), Berlin, Germany: Springer Science & Business Media, 2001.**
5. **C. Baier and J. P. Katoen, Principles of Model Checking, (1e), Cambridge, MA: MIT Press, 2008.**
6. **P. Boca, J. P. Bowen, and J. I. Siddiqi, Formal Methods: State of the Art and New Directions (Vol. 1), (1e), London, UK: Springer, 2010.**
7. **N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, (1e), Cambridge, MA: The MIT Press, 2016.**

CSE4143: Recommender System [PE7] [3 0 0 3]

Course Contents

Introduction: Basic taxonomy of recommender systems – Traditional and non-personalized Recommender Systems – Overview of data mining methods for recommender systems; **Content-based Recommendation:** High-level architecture of content-based systems, Advantages and drawbacks of content-based filtering, Item profiles, Representing item profiles, Methods for learning user profiles, Similarity-based retrieval, Classification algorithms; **Collaborative Filtering:** User-based Nearest-neighbor collaborative filtering (CF), and item-based CF, Attacks on collaborative recommender systems; **Attack Resistant Recommender System:** Introduction – Types of Attacks – Detecting attacks on recommender systems - Individual attack - Group attack - Strategies for robust recommender design - Robust recommendation algorithms; **Evaluating Recommender System:** Evaluating paradigms, General properties of evaluation research, Online and Offline evaluation, Evaluation design, Evaluation on historical datasets, Error and accuracy metrics, Limitations of evaluation measures; **Case Based Recommender System:** Constraint-based recommenders, Hybrid approaches: Opportunities for hybridization, **Monolithic hybridization design:** Feature combination, Feature augmentation, Parallelized and pipelined hybridization design, Limitations of hybridization strategies.

References:

1. C.C. Aggarwal, Recommender Systems: The Textbook (1e), Springer, 2016.
2. Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, Mining of massive datasets, (3e), Cambridge University Press, 2020.
3. N Manouselis, H. Drachsler, K. Verbert and E. Duval., Recommender Systems for Learning (1e), Springer 2013.
4. K.Falk, Practical Recommender Systems, (1e), Manning Publication, 2017.

CSE4140: Blockchain Technology [PE7] [3 0 0 3]

Course Contents

Basic Cryptographic Primitives: encryption and decryption, public-key cryptography, RSA, ECC, digital signature, secure hash algorithms, SHA-256, Keccak, properties of hash algorithms, Merkle tree, Patricia tree, distributed hash table; **Introduction to Blockchain Technology:** history of blockchain, elements of blockchain, features of blockchain, tiers of blockchain technology, types of blockchain, consensus in blockchain, applications of blockchain, benefits and limitations of blockchain; **Bitcoin:** overview of bitcoin and its architecture, bitcoin mining and proof-of-work, bitcoin transactions and scripts, bitcoin wallets and security; **Ethereum:** introduction to Ethereum and smart contracts, Ethereum virtual machine, solidity programming language, decentralized applications on Ethereum; **Blockchain Applications:** blockchain applications in finance, supply chain, identity, and voting, case studies of successful blockchain applications, regulatory and legal challenges for blockchain applications; **Future of Blockchain Technology:** emerging trends in blockchain technology, decentralized finance, non-fungible tokens, potential impact of blockchain technology on the global economy and society.

Reference Books:

1. **N. Bonneau, A. Miller, and S. Goldfeder, Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction, (1e), Princeton, NJ: Princeton University Press, 2016.**
2. **I. Bashir, Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained, (2e), Birmingham, UK: Packt Publishing, 2018.**
3. **R. Modi, Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain, (1e), Birmingham, UK: Packt Publishing, 2018.**

CSE4144: INFORMATION RETRIEVAL [PE8] [3 0 0 3]

Course Contents

Introduction to IR: IR Concepts; Boolean Retrieval Model; Inverted Index; Processing Boolean Queries; **Term Vocabulary and Postings Lists:** Document Delineation and Character Sequence Decoding; Term Vocabulary; **Dictionaries and Tolerant Retrieval:** Search Structures for Dictionaries; Wildcard Queries; Spelling Correction; Phonetic Correction; **Index Construction:** Hardware Basics; Blocked Sort-Based Indexing; **Vector Space Model:** Parametric and Zone Indexes; Term Frequency and Weighting; Vector Space Model for Scoring; **Introduction to Probabilistic Models; Evaluation in Information Retrieval:** Information Retrieval System Evaluation; Standard Test Collections; Evaluation of Ranked and Unranked Retrieval Sets; **Web Search Basics:** Web Characteristics; Advertising as the Economic Model; Search User Experience; Index Size and Estimation; Near-Duplicates and Shingling; **Web Crawling and Indexes:** Overview; Crawling; Distributing Indexes; Connectivity Servers; **Link Analysis:** The Web as a Graph; Page Rank; Hubs and Authorities.

Reference Books

1. **S. Büttcher, C. L. A. Clarke, and G. V. Cormack, Information Retrieval: Implementing and Evaluating Search Engines, (1e), Cambridge, MA: MIT Press, 2016.**
2. **W. Hersh, Information Retrieval: A Biomedical and Health Perspective, (4e), Berlin, Germany: Springer, 2020.**
3. **C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, (1e), Cambridge, UK: Cambridge University Press, 2009.**
4. **B. Croft, D. Metzler, and T. Strohman, Search Engines: Information Retrieval in Practice, (2e), Boston, MA: Pearson, 202.**

CSE4145: Virtual & Augmented Reality [PE8] [3 0 0 3]

Course Contents

Introduction to AR/VR: Evolution and history of AR and VR; Definitions and differences: AR vs. VR vs. MR; Immersive technologies and Extended Reality; Applications; Overview of major AR/VR devices; **Hardware and Software Components:** Input devices; Output devices; Tracking systems; Software development kits: ARCore, ARKit, Unity; **3D Graphics and Interaction:** Basics of 3D graphics: coordinate systems, modeling, textures, rendering; Scene creation and environment setup;

Interaction design in VR: gesture, gaze, speech, controllers; UX design principles for immersive experiences; Depth perception and spatial audio; **Augmented and Virtual Reality Systems:** Types of AR; SLAM (Simultaneous Localization and Mapping) in AR; Mobile AR development with ARCore/ARKit; Use of Unity for marker-based AR; VR content creation pipelines; VR development using Unity; Navigation and locomotion in virtual spaces; Immersion and presence in VR; Motion sickness; **Advanced Topics and Applications:** Mixed Reality and Extended Reality; WebAR and WebVR; Ethical and privacy issues in AR/VR; **Future trends:** AI in AR/VR, Metaverse, digital twins; Case studies: real-world AR/VR applications.

References:

1. **Alan A. B. Craig, Understanding Augmented Reality, (1e), Burlington, MA: Morgan Kaufmann, 2013.**
2. **G. C. Burdea and P. Coiffet, Virtual Reality Technology, (3e), Hoboken, NJ: Wiley-IEEE Press, 2024.**
3. **D. Schmalstieg and T. Hollerer, Augmented Reality: Principles and Practice, (1e), Boston, MA: Addison-Wesley, 2016.**
4. **J. Jerald, The VR Book: Human-Centered Design for Virtual Reality, (1e), New York, NY: ACM Books, 2015.**

CSE4146: Quantum Computing [PE8] [3 0 0 3]

Course Contents

Quantum Measurements and Density Matrices: Positive-Operator Valued Measure (POVM), Quantum superposition, entanglement, decoherence, Bell's inequalities; **Quantum Gates and Circuits:** Quantum teleportation, no-cloning theorem, Deutsch and Deutsch–Jozsa algorithms, Grover's search algorithm, Quantum Fourier Transform, Shor's factoring algorithm; **Quantum Error Correction:** Fault tolerance; **Quantum Cryptography:** Quantum Key Distribution protocols; **Quantum Technologies and Platforms:** Fidelity, scalability, NMR quantum computing, Spintronics and QED approaches, Linear and nonlinear optical methods, Quantum hardware platforms: superconducting qubits, trapped ions, quantum dots, photonic qubits; **Advanced Applications:** Quantum sensing, quantum networks, quantum internet; **Future Outlook:** Current limits and future prospects of quantum technology.

References:

1. **M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, (2e), Cambridge, UK: Cambridge University Press, 2010.**
2. **G. Benenti, G. Casati, and G. Strini, Principles of Quantum Computation and Information, (2e), Singapore: World Scientific, 2018.**
3. **C. Gerry and P. Knight, Introductory Quantum Optics, (2e), Cambridge, UK: Cambridge University Press, 2023.**
4. **D. Bouwmeester, A. Ekert, and A. Zeilinger, The Physics of Quantum Information, (1e), Berlin, Germany: Springer, 2000.**
5. **H. M. Wiseman and G. J. Milburn, Quantum Measurement and Control, (1e), Cambridge, UK: Cambridge University Press, 2009.**

CSE4147: Cognitive Computing [PE8] [3 0 0 3]

Course Contents

Foundations of Cognitive Computing: Understanding cognition, interdisciplinary perspectives, Symbolic vs. Sub-symbolic approaches, Intelligent Systems inspired by human cognition, Human-computer interaction principles, IBM's Watson and its cognitive capabilities, Design for human cognition, Augmented Intelligence vs. Artificial Intelligence; **Cognitive Modelling Paradigms**: Declarative/logic-based computational cognitive modelling, Connectionist models (Neural Networks), Bayesian models of cognition, Dynamical systems approach to cognition, Hybrid models of reasoning and decision-making, Cognitive simulation techniques, Symbolic reasoning; **Cognitive Architecture and Decision Making:** Modelling interaction of language, memory, and learning, Classical models of rationality and bounded rationality, Artificial cognitive architectures: ACT-R, SOAR, OpenCog, CopyCat, Memory Networks; **Applications and Modern Cognitive Technologies:** inductive generalization, categorization, causality, analogy in problem-solving, Deep QA architecture, Unstructured Information Management Architecture (UIMA),Building applications using Watson APIs, OpenAI APIs, Applications in business, education, robotics, and healthcare; **Inference Models of Cognition:** Generative Models – Conditioning – Causal and statistical dependence; **Learning Models of Cognition** Learning as Conditional Inference – Learning with a Language of Thought – Hierarchical Models-Learning (Deep) Continuous Functions – Mixture Models.

References:

1. **J. H. Hurwitz, M. Kaufman, and A. Bowles, Cognitive Computing and Big Data Analytics, (1e), Hoboken, NJ: Wiley, 2015.**
2. **V. V. Raghavan, V. N. Gudivada, V. Govindaraju, and C. R. Rao, Cognitive Computing: Theory and Applications (Handbook of Statistics 35), (1e), Amsterdam, Netherlands: Elsevier, 2016.**
3. **J. Hurwitz, M. Kaufman, and A. Bowles, Cognitive Computing and Big Data Analytics, (1e), Hoboken, NJ: Wiley, 2015.**
4. **N. D. Goodman, J. B. Tenenbaum, and The ProbMods Contributors, Probabilistic Models of Cognition, (2e), (Open Access), 2016.**

CSE4148: Reinforcement Learning [PE8] [3 0 0 3]

Course Contents

Foundations of Reinforcement Learning: Introduction to agent-environment interaction, learning task - Q learning - The Q function, Markov Decision Process (MDP) – Markov Process, Markov Reward Process, Markov Decision Process and Bellman Equations; **Classical Solution Methods:** Dynamic programming; **Deep Reinforcement Learning:** Introduction to function approximation in RL, Deep Q-Networks (DQNs) and challenges in high-dimensional state spaces, Policy gradient methods and actor-critic architectures; **Human Feedback in Reinforcement Learning (RLHF)** Overview of RLHF: Concepts, motivation, and differences from standard RL, Data Collection: Methods for gathering human preference data (pairwise comparisons, rankings, numerical and natural language feedback), Reward Model Training: Techniques for training reward models from human judgments; **Applications and Project Work:** Real-world application projects in robotics, natural language processing, or gaming, Project presentations and discussions.

References:

1. **R. S. Sutton and A. G. Barto, Introduction to Reinforcement Learning, (2e), Cambridge, MA: MIT Press, 2017.**
2. **D. P. Bertsekas, Reinforcement Learning and Optimal Control, (1e), Belmont, MA: Athena Scientific, 2019.**

3. R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, (2e), Cambridge, MA: MIT Press, 2018.
4. V. S. Borkar, Stochastic Approximation: A Dynamical Systems Viewpoint, (1e), New Delhi, India: Hindustan Book Agency, 2009.