



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**DISCRETE MATHEMATICS**

**B.Tech. II Year I Sem**

L	T	P	C
3	0	0	3

**Course Objectives:**

1. Introduces elementary discrete mathematics for computer science and engineering.
2. Topics include formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

**Course Outcomes:**

1. Understand and construct precise mathematical proofs
2. Apply logic and set theory to formulate precise statements
3. Analyze and solve counting problems on finite and discrete structures
4. Describe and manipulate sequences
5. Apply graph theory in solving computing problems

**UNIT - I**

Mathematical logic: Introduction, Statements and Notation, Connectives, Equivalence formulas, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

**UNIT - II**

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

**UNIT - III**

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Groups, Lattices as Partially Ordered Sets, Boolean Algebra.

**UNIT - IV**

**Elementary Combinatorics:** Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutation with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Inclusion-Exclusion.

**UNIT - V**

**Graph Theory:** Basic Concepts, Isomorphism and Subgraphs, Planar Graphs, Euler’s Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color

Problem.

**TEXT BOOKS:**

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe I. Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd ed.

**REFERENCE BOOKS:**

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph. P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata McGraw Hill Publishing co.



## DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE B.Tech Course Structure (R25 Regulations)

### COMPUTER ORGANIZATION AND ARCHITECTURE

**B.Tech. II Year, I Sem.**

L	T	P	C
3	0	0	3

**Course Objectives:**

- 1 To understand the fundamentals of Boolean algebra, data representation, and the design of combinational and sequential logic circuits used in digital systems.
- 2 To analyse the organization and operation of a basic digital computer, including register transfer language, instruction execution, and CPU control mechanisms.
- 3 To comprehend computer arithmetic operations, micro programmed control, and the organization of memory and input–output systems.

**Course Outcomes:**

1. Apply Boolean algebra and minimization techniques to analyse and design combinational logic circuits for digital applications.
2. Design and analyse sequential logic circuits using latches, flip-flops, registers, and counters with proper timing control.
3. Use Register Transfer Language and micro-operations to explain data flow and control in a basic digital computer.
4. Analyse instruction formats, addressing modes, CPU organization, and computer arithmetic algorithms including floating-point operations.
5. Explain memory hierarchy, cache organization, and input–output mechanisms including interrupts and DMA for efficient system performance.

**UNIT - I:**

**Boolean Algebra and Logic Gates:** Binary codes, Binary Storage and Registers, Binary logic, logic gates.

**Data Representation:** Data types, Complements, Fixed Point Representation, Floating Point Representation  
**Digital Computers:** Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

**UNIT - II:**

**Gate-Level Minimization Techniques:** Karnaugh Map 3&4 Variables, don't care conditions.

**Combinational Logic:** Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers.

**UNIT III:**

**Sequential Logic:** Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, Registers, shift Registers, Ripple counters, synchronous counters.

**Register Transfer Language and Micro operations:** Register Transfer language, Register Transfer,

Bus and memory transfers, Arithmetic Micro operations, logic micro-operations, shift micro-operations, Arithmetic logic shift unit.

**Basic Computer Organization and Design:** Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

#### **UNIT – IV:**

**Microprogrammed Control:** Control memory, Address sequencing, micro program example, design of control unit.

**Central Processing Unit:** General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

**Computer Arithmetic:** Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations.

#### **UNIT – V:**

**Input-Output Organization:** Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

**Memory Organization:** Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

#### **TEXT BOOKS:**

1. Digital Design – M. Morris Mano, Third Edition, Pearson/PHI.
2. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.

#### **REFERENCE BOOKS:**

1. Switching and Finite Automata Theory, ZVI. Kohavi, Tata Mc Graw Hill.
2. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th Edition, McGraw Hill.
3. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
4. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**OBJECT ORIENTED PROGRAMMING THROUGH JAVA**

**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

**Course Objectives:**

1. To Understand the basic object-oriented programming concepts and apply them in problem solving.
2. To Illustrate inheritance concepts for reusing the program
- 3 To Demonstrate multitasking by using multiple threads and event handling
- 4 To Demonstrate multitasking by using multiple threads and event handling
- 5 To Develop data-centric applications using JDBC.
- 6 To Understand the basics of java console and GUI based programming

**Course Outcomes:**

- 1 Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection.
- 2 Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords Use multithreading concepts to develop inter process communication
3. Understand the process of graphical user interface design and implementation using AWT or swings.
4. Develop applets that interact abundantly with the client environment and deploy on the server.

**UNIT - I**

**Object oriented thinking and Java Basics-** Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring String class.

**UNIT - II**

**Inheritance, Packages and Interfaces** – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super keyword uses, using final keyword with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

### **UNIT - III**

**Exception handling and Multithreading**-- Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads.

### **UNIT - IV**

Exploring String class, Object class, Exploring java.util package, Exploring java.io package Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

### **UNIT - V**

**Swing** – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- J Frame and J Component, J Label, Image Icon, J Text Field, J Button, J Check Box, J Radio Button, J List, J Combo Box, Tabbed Panes, Scroll Panes, Trees, and Tables. Menu Basics, Menu related classes – J Menu Bar, J Menu, J Menu Item, J Check Box Menu Item, J Radio Button Menu Item, JSeparator. creating a popup menu

### **TEXT BOOKS:**

1. Java the complete reference, 13th edition, Herbert schildt, Dr. Denny Coward, Mc Graw Hill.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

### **REFERENCE BOOKS:**

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John Wiley & sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A.
5. Johnson- Thomson.
6. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition,
7. Pearson Education.
8. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.
9. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
10. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
11. Maurach's Beginning Java2 JDK 5, SPD.



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**SOFTWARE ENGINEERING**

**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

**Course Objectives**

1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

**Course Outcomes**

1. Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

**UNIT - I**

**Introduction to Software Engineering:** The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). Process models: The waterfall model, Spiral model, Incremental Process Models, Concurrent Models, Component based development and Agile Development.

**UNIT - II**

**Software Requirements:** Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

**UNIT - III**

**Design Engineering:** Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, activity diagrams and component diagrams.

## **UNIT - IV**

**Testing Strategies:** A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Metrics for Process and Products: Software measurement, metrics for software quality.

## **UNIT - V**

**Risk management:** Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

### **TEXT BOOKS:**

- a. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
- b. Software Engineering- Sommerville, 7th edition, Pearson Education.
- c. The unified modeling language user guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

### **REFERENCE BOOKS:**

- 1 Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
- 2 Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
- 3 Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.
- 4 Fundamentals of Software Engineering-Rajib Mall, PHI.



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**DATABASE MANAGEMENT SYSTEMS**

**B.Tech. II Year I Sem.**

**Prerequisites:** A course on “Data Structures”.

L	T	P	C
3	0	0	3

**Course Objectives:**

- 1 To understand the basic concepts and the applications of database systems.
- 2 To master the basics of SQL and construct queries using SQL.
- 3 Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

**Course Outcomes:**

- 1 Gain knowledge of fundamentals of DBMS, database design and normal forms
- 2 Master the basics of SQL for retrieval and management of data.
- 3 Be acquainted with the basics of transaction processing and concurrency control.
- 4 Familiarity with database storage structures and access techniques

**UNIT - I**

**Database System Applications:** A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

**UNIT - II**

**Introduction to the Relational Model:** Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views. Relational Algebra, Tuple relational Calculus, Domain relational calculus.

**UNIT - III**

**SQL: QUERIES, CONSTRAINTS, TRIGGERS:** form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases. Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

**UNIT - IV**

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log-Based Recovery, Recovery with Concurrent Transactions.

## **UNIT - V**

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

### **TEXT BOOKS:**

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition.3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill

### **REFERENCE BOOKS:**

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**INNOVATION AND ENTREPRENEURSHIP**

**B.Tech. II Year I Sem**

L	T	P	C
2	0	0	2

**Course Objectives:**

1. To familiarize on the basic concepts of innovation, entrepreneurship and its importance.
2. To Identify and analyze the process of problem-opportunity identification, market segmentation, and idea generation techniques.
3. To initiate prototype development and understand minimum viable product.
4. To develop initial Business and financial planning and Go-to-Market strategies
5. To impart knowledge on establishing startups, venture pitching and IPR

**Course Outcomes:**

- 1 Understand the entrepreneurship and the entrepreneurial process and its significance in economic development.
- 2 Assess the problem from an industry perspective and generate solutions using the design thinking principles.
- 3 Assess market competition, estimate market size, and develop a prototype.
- 4 Analyze Business and financial planning models and Go-to-Market strategies.
- 5 Able to build a start-up, register IP and identify funding opportunities.

**UNIT I:**

**Fundamentals of Innovation and Entrepreneurship Innovation:** Introduction, need for innovation, Features, Types of innovations, innovations in manufacturing and service sectors, fostering a culture of innovation, planning for innovation.

**Entrepreneurship:** Introduction, types of entrepreneurship attributes, mindset of entrepreneurial and intrapreneurial leadership, Role of entrepreneurs in economic development. Woman Entrepreneurship, Importance of on-campus startups. Understanding to build entrepreneurial mindset, attributes and networks individuals while on campus.

**Core Teaching Tool:** Simulation, Game, Industry Case Studies (Personalized for students – 16 industries to choose from), Venture Activity.

**UNIT II:**

Problem and Customer Identification of gap, problem, analyzing the problem from a industry perspective, real-world problems, market and customer segmentation, validation of customer problem fit, Iterating problem-customer fit, Competition and Industry trends mapping and assessing initial opportunity, Porter’s Five Force Model.

**Idea generation, Ideation techniques:** Brainstorming, Brain writing, Round robin, and SCAMPER, Design thinking principles, Mapping of solution to problem.

**Core Teaching Tool:** Several types of activities including: Class, game, Gen AI, ‘Get out of the Building and Venture Activity.

### **UNIT III:**

**Opportunity assessment and Prototype development Identify and map global competitors, review industry trends, and understand market sizing:** TAM, SAM, and SOM. Assessing scope and potential scale for the opportunity.

**Understanding prototyping and Minimum Viable Product (MVP). Developing a prototype:** Testing, and validation.

**Core Teaching Tool:** Venture Activity, no-code Innovation tools, Class activity

### **UNIT IV:**

**Business & Financial Models Introduction to Business Model and types, Lean Canvas Approach:** 9-block lean canvas model, building lean canvas for your startup. Business planning: components of Business plan- Sales plan, People plan and financial plan, Financial Planning: Types of costs, preparing a financial plan for profitability using a financial template, understanding the basics of Unit economics, Economies of Scale and analyzing financial performance. Go-To-Market (GTM) approach – Selecting the Right Channel, creating digital presence, and building customer acquisition strategy.

**Core Teaching Tool:** Founder Case Studies – Sama and Securely Share; Class activity and discussions; Venture Activities.

### **UNIT V:**

Startups and IPR Startup requirements, building founding team members and mentors, pitch preparation, start-up registration process, funding opportunities and schemes, institutional support to entrepreneurs, startup lifecycle, documentation, legal aspects in startup, venture pitching readiness, National Innovation Startup Policy (NISP) and its features.

Patents, Designs, Patentability, Procedure for grants of patents. Indian Scenario of Patenting, International Scenario: International cooperation on Intellectual Property. Patent Rights: Scope of Patent Rights. Copyright, trademark, and GI. Licensing and transfer of technology.

**Core Teaching Tool:** Expert talks; Cases; Class activity and discussions; Venture Activities.

### **Suggested Readings:**

1. John R Bessant, Joe Tidd, Innovation and Entrepreneurship, 4E, Wiley, Latest Edition.
2. Ajay Batra, The Startup Launch Book- A Practical Guide for Launching Customer Centric Ventures, Wiley, 2020. (For Core Teaching Tool).
3. Entrepreneurship Development and Small Business Enterprises, Poornima M Charantimath, 3E, Pearson, 2018.
4. D.F. Kuratko and T.V. Rao, Entrepreneurship: A South-Asian Perspective, Cengage Learning, 2013.
5. Robert D. Hisrich, Michael P. Peters, Dean A. Shepherd, Sabyasachi Sinha (2020). Entrepreneurship, McGrawHill, 11th Edition.
6. NISP -Brochure inside pages - startup\_policy\_2019.pdf



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**OPERATING SYSTEMS**

**B.Tech. II Year II Sem**

**Prerequisites:**

1. A course on “Computer Programming and Data Structures”.
2. A course on “Computer Organization and Architecture”.

L	T	P	C
3	0	0	3

**Course Objectives:**

1. Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
2. Introduce the issues to be considered in the design and development of operating system
3. Introduce basic Unix commands, system call interface for process management, interposes communication and I/O in Unix

**Course Outcomes:**

1. Will be able to control access to a computer and the files that may be shared
2. Demonstrate the knowledge of the components of computers and their respective roles in computing.
3. Ability to recognize and resolve user problems with standard operating environments.
4. Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

**UNIT - I**

**Operating System** - Introduction, Structures - Simple Batch, Multi programmed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

**UNIT - II**

**CPU Scheduling** - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec  
**Deadlocks** - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

**UNIT - III**

**Process Management and Synchronization** - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors  
**Interprocess Communication Mechanisms:** IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

## **UNIT - IV**

**Memory Management and Virtual Memory** - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

## **UNIT - V**

**File System Interface and Operations** -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

### **TEXT BOOKS:**

- 1 Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
- 2 Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

### **REFERENCE BOOKS:**

1. Operating Systems- Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Crowley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.



## DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE B.Tech Course Structure (R25 Regulations)

### ALGORITHMS DESIGN AND ANALYSIS

**B.Tech. II Year II Sem**

L	T	P	C
3	0	0	3

**Prerequisites: Programming for problem solving and Data Structures**

#### Course Objectives

1. Develop proficiency in evaluating algorithms using asymptotic notations, including best-, average-, and worst-case time/space complexities, and solving related recurrence relations.
2. Master various algorithmic strategies—divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound—identifying suitable use cases and demonstrating their application.
3. Critically assess and contrast different algorithms in terms of efficiency, scalability, and correctness through rigorous analytical reasoning and empirical evaluation.
4. Differentiate between tractable (polynomial-time) and intractable (super-polynomial or exponential-time) problems; Identify and classify problems as P, NP, NP-hard, or NP-complete, and assess their relationships through polynomial-time reductions and Cook's theorem.

#### Course Outcomes

1. Able to Apply space and time complexity analysis using asymptotic notations.
2. Able to Design divide-and-conquer algorithms and critically assess their runtime and space trade-offs.
3. Able to Device backtracking and dynamic programming solutions.
4. Able to Apply greedy methods and graph traversal algorithms
5. Able to Analyse and Design branch-and-bound algorithms for NP-hard problems

#### UNIT - I

**Introduction:** Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation, and Little oh notation.

**Divide and conquer:** General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

#### UNIT - II

**Disjoint Sets:** Disjoint set operations, union and find algorithms, Priority Queue- Heaps, Heapsort Backtracking: General method, applications, n-queens problem, sum of subsets, problem, graph coloring.

### **UNIT - III**

**Dynamic Programming:** General method, applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling salesperson problem, Reliability design.

### **UNIT - IV**

**Greedy method:** General method, applications- Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.  
Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs.

### **UNIT - V**

**Branch and Bound:** General method, applications - Travelling salesperson problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.  
NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP-Complete classes.

### **TEXT BOOK:**

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni, and Rajasekaran, University Press.

### **REFERENCE BOOKS:**

1. Design and Analysis of algorithms, Aho, Ullman, and Hopcroft, Pearson education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and Sons.



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE & DATA SCIENCE)**  
**B.Tech Course Structure (R25 Regulations)**

**COMPUTER NETWORKS**

**B.Tech. II Year II Sem**

**Prerequisites**

- 1 A course on “Programming for problem solving”.
- 2 A course on “Data Structures”.

L	T	P	C
3	0	0	3

**Course Objectives**

1. Equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.
3. Elucidate the students about working and implementation of protocols at various layers in protocols stack.
4. Appreciating the protocol working by observing and analyzing outputs of the packet sniffer

**Course Outcomes**

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the ISO-OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.
5. Understanding working of the protocols through traces captured by a packet sniffer.

**UNIT - I**

**Introduction:** The Internet, Protocol, Network Edge, Access Networks, Network Core, Packet Switching, Circuit Switching, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol **reference models:** ISO-OSI, TCP/IP, Types of Network attacks, History of Computer Networking and the Internet.

**UNIT-II**

**Application Layer:** Principles of Network Applications, Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, SMTP, DNS, Peer-to-Peer Applications, Socket Programming: Creating Network Applications.

### **UNIT - III**

**Transport Layer:** Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N (GBN), Selective Repeat (SR), Connection-Oriented Transport: TCP, The TCP Connection, Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management, Principles of Congestion Control, TCP Congestion Control, Fairness.

### **UNIT - IV**

**Network Layer:** Data and Control plane, Forwarding and Routing 308, Network Service Models, Virtual Circuit and Datagram Networks, Router working, The Internet Protocol (IP): Forwarding and Addressing in the Internet, Datagram Format, IPv4 Addressing, Internet Control Message Protocol (ICMP), IPv6, IP Security, Routing Algorithms- The Link-State (LS) Routing Algorithm, The Distance Vector (DV) Routing Algorithm, Hierarchical Routing, Routing in the Internet-Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter-AS Routing: BGP, Broadcast and Multicast Routing, Broadcast Routing Algorithms, Multicasting.

### **UNIT - V**

**The Link Layer:** The Services Provided by the Link Layer, Error-Detection and -Correction Techniques- Parity Checks, Checksum Methods, Cyclic Redundancy Check (CRC), Hamming code, Multiple Access Links and Protocols, Channel Partitioning Protocols, Random Access Protocols, Taking-Turns Protocols, DOCSIS: The Link-Layer Protocol for Cable Internet Access, Switched Local Area Networks, Link-Layer Addressing and ARP, Ethernet, Link-Layer Switches, Virtual Local Area Networks (VLANs), Link Virtualization-Multiprotocol Label Switching (MPLS), Data Center Networking, A Day in the Life of a Web Page Request. Wireless network characteristics, Wireless LAN.

### **TEXT BOOKS:**

1. Computer Networking: A Top-Down Approach – James F.Kurose, Keith W. Ross, Pearson
2. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson/PHI

### **REFERENCE BOOK:**

1. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.



**AVN** INSTITUTE OF  
ENGINEERING & TECHNOLOGY  
Accredited by NAAC & NBA  
An Autonomous Institute Affiliated to JNTU Hyderabad



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**  
**B.Tech Course Structure (R25 Regulations)**

**MACHINE LEARNING**

**B.Tech. II Year II Sem**

L	T	P	C
3	0	0	3

**Course Objectives:**

1. To introduce students to the basic concepts and techniques of Machine Learning.
2. To have a thorough understanding of the Supervised and Unsupervised learning techniques
3. To study the various probability-based learning techniques

**Course Outcomes:**

1. Distinguish between, supervised, unsupervised and semi-supervised learning.
2. Understand algorithms for building classifiers applied on datasets of non-linearly separable classes
3. Design an ensemble model to increase the classification accuracy
4. Understand the principles of RL evolutionary computing algorithms

**UNIT – I**

Introduction to Machine Learning: Types of Human learning, machine learning process, Well-posed learning problem, Types of machine learning and comparison, applications of machine learning. Model Preparation, Evaluation and feature engineering: Machine learning activities, Types of data in machine learning, dataset understanding, plotting and exploration, checking data quality, remediation, data pre-processing, selecting a model, predictive and descriptive models, supervised learning model training, cross-validation and boot strapping, lazy vs eager learner, interpreting the model- underfitting, overfitting, bias-variance trade-off. Parameter for evaluating performance of classification, regression, and clustering model. Improving performance of a model.

**UNIT - II**

Feature Engineering: Feature transformation - feature construction, feature extraction by PCA, SVD, LDA. Feature subset selection – feature relevancy and redundancy measures. Feature selection process and approaches.

Review of Probability concepts: joint probability, conditional probability, bayes rule, Common discrete and continuous distributions, dealing with multiple random variables, central limit theorem. Bayes classifier, Multi-class Classification, Naïve Bayes classifier, Bayesian belief network.

### **UNIT - III**

**Supervised Learning** - Introduction to supervised learning, Regression: Introduction of regression, Regression algorithms: Simple linear regression, Multiple linear regression, Polynomial regression model, Logistic regression, Maximum likelihood estimation. Classification: Classification model and learning steps, Classification algorithms: Naïve Bayes classifier, Distance measures, k-Nearest Neighbor (kNN), Decision tree, Support vector machines, Kernel trick, Random Forest.

### **UNIT - IV**

**Unsupervised Learning:** Introduction to unsupervised learning, Unsupervised vs supervised learning, Application of unsupervised learning, Clustering and its types, Partitioning method: k-Means and KMedoids, Hierarchical clustering, Density-based methods – DBSCAN.

### **UNIT - V**

**Artificial Neural Network:** Biological neuron, Artificial neuron, Activation functions, neural network architecture, perceptron, learning process in ANN, Back propagation.

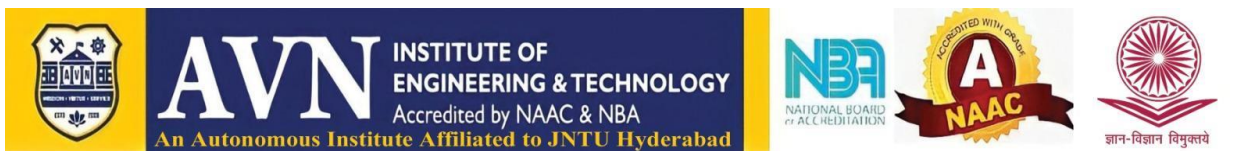
Introduction to deep learning, overview of reinforcement learning, Representation learning, Evolutionary learning. Case-study of ML applications: Image recognition, Email spam filtering, Online fraud detection.

### **TEXT BOOKS:**

- 1 Saikat Dutt, S. Chjandramouli, Das – Machine Learning, Frist Edition, Pearson
- 2 M N Murty, Anathanarayana V S – Machine Learning, First Edition, University Press
- 3 Tom M Mitchell, —Machine Learning, First Edition, McGraw Hill Education, 2013.

### **REFERENCE BOOKS:**

1. Stephen Marsland, —Machine Learning – An Algorithmic Perspective, Second Edition,
2. Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)**  
**B.Tech Course Structure (R25 Regulations)**

**INDIAN KNOWLEDGE SYSTEM**

**B.Tech. II Year, II Sem**

L	T	P	C
1	0	0	1

Bharat is considered one of the oldest civilizations of the world. Some of the archaeological evidences proved the existence of Indus Valley Civilization in 7000 B.C. Bhartiya traditions, culture, cultural activities, rituals, sacraments, painting, art of dancing, art of singing etc. is being practised till the modern times without knowing scientific approaches behind that. Eternity of Indian knowledge system proved itself that not only many rituals but also many traditions, many streams of knowledge like astrology, mathematics, physics, chemistry, biology, language studies, yoga and meditation had been following from the starting till now with some changes, in the form of traditions.

This course is for undergraduate students to inculcate Indian values. It will promote advance study and inter disciplinary research on all aspects of the Indian knowledge system.

**Course Objectives:** This course aims:

- To provide a tribune of the rich culture and traditions of Indian knowledge system to students of various disciplines.
- To introduce historical account on the education and scientific literature available in ancient Indian traditions and its connections with ancient Indian Philosophy
- To give insights about the applications of Bharatiya Jnana Parampara
- To introduce Indian approach towards health and wellbeing
- To elaborate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world.

**Course Outcomes:** Students will be able to:

- Understand nature, scope and related fields of Indian knowledge system.
- Demonstrate the scientific literature available in ancient Indian traditions
- Understanding the application of Bharatiya Jnana Parampara
- Understand Indian approach towards Wellbeing
- Appreciate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world

**Unit 1: Introduction to Indian Knowledge Systems**

Meaning, Nature, Scope and Salient Aspects of Bharatiya Jnana Parampara - Introduction to Vedas, Upanishads, Vidya, Kala, Jnana, Shastra - Practices and Continuity of Tradition

**Unit 2: Overview of History of Indian Education and Scientific Literature**

Gurukul System - Role of Sanskrit in Natural Language Processing - Scientific Literature - Vedic Literature - Available Scientific Treatises - Interlinkings

### **Unit 3: Introduction to Scientific Theories from Pure Sciences from Ancient Indian Knowledge Systems**

Overview of theories from available ancient Indian Literature about Physics, Chemistry and Mathematics - Interlinkings and applications

### **Unit 4: Introduction to Ancient Indian Wellness Systems**

Concept of Wellness – Yoga System - Ayurveda System - Ancient Indian Aesthetics

### **Unit 5: Development of Engineering, Science, Technology & Fine Arts in India**

Various Industries - Silk, Cotton and Ship Building - Evolution of Indian Fine Arts – Cave and Temple

Architecture, Vastu - Vidya, Sculpture, Forts and Stepwells, Observatories and Paintings - Music and Natyakala - Cultural Traditions & Folk Arts

**Pedagogy for Teachers: Apart from Class Room Instruction, the following Methods are Suggested.**

1. Project based activities and learning.
2. Presentation and case studies.
3. Film screening and book reviews.
4. Visit to historical places, archives centre, research centre or library nearby.

**Note:** Activities mentioned above are only suggestive. Teacher-educators should encourage students to be innovative.

#### **Suggested Readings:**

- i. B. Mahadevan, Bhat Vinayak and Nagendra Pavan R.N., (2022) 'Introduction to Indian Knowledge Systems: Concepts and Applications' PHI learning PVT, New Delhi ISBN [9789391818203]
- ii. Dharmapal (1971) 'Indian Science and Technology in the Eighteenth Century'. Other India Press, Goa
5. Kapil Kapoor, Singh Avdhesh Kumar, (2005) 'Indian Knowledge Systems' D.K. Printworld (P) Ltd. ISBN 10: 8124603367 / ISBN 13: 9788124603369
6. Chakradeo, Ujwala, Temples of Bharat, Aayu Publications, New Delhi, 2024.
7. D.N. Bose, S.N. Sen and B. V. Subbarayappa, A Concise History of Science in India, Indian National Science Academy, New Delhi, 2009.
8. Datta B. and A. N. Singh, History of Hindu Mathematics: Parts I and II, Asia Publishing House, Bombay, 1962.
9. Kapoor, K. (2021), Indian Knowledge System: Nature, Philosophy, Character in Indian Knowledge System, vol. 1, Pub. Indian Institute of Advanced Studies, Shimla
10. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Philosophical Systems, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.
11. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Knowledge: Framework and Classification, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.

**Video Resources:**

- i. Introductory lectures by Prof. Gauri Mahulikar
- ii. Introductory lectures by Prof. Kapil Kapoor

**Websites:**

- <https://iksin dia.org/index.php>
- Official Website of IKS- Indian Knowledge System
- <https://www.youtube.com/watch?v=uKcf-hSlcUE>
- Address by Prof Kapil Kapoor | Indian Institute of Advanced Study (FDP 2021)
- [https://www.youtube.com/watch?v=MDJTXNiH2\\_A](https://www.youtube.com/watch?v=MDJTXNiH2_A)
- Mukul Kanitkar on Bharatiya Knowledge System
- <https://www.youtube.com/watch?v=uARMhv97pjk>
- <https://www.youtube.com/watch?v=oTwgf56GbsA>
- Scientific History of India | Mukul Kanitkar Lecture in DTU
- <https://youtu.be/gNJNmPJqXJc?si=WFBbuUT65mLZzpOW>
- Ancient India's Scientific Achievements & Contribution in Mathematics, Astronomy, Science & Medicine



**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

**OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB**

**B.Tech. II Year I Sem.**

**L T P C**

**0 0 2 1**

**Course Objectives:**

1. To write programs using abstract classes.
2. To write programs for solving real world problems using the java collection framework.
3. To write multithreaded programs.
4. To write GUI programs using swing controls in Java.
5. To introduce java compiler and eclipse platform.
6. To impart hands-on experience with java programming.

**Course Outcomes:**

1. Able to write programs for solving real world problems using the java collection framework.
2. Able to write programs using abstract classes.
3. Able to write multithreaded programs.
4. Able to write GUI programs using swing controls in Java.

**Note:**

1. Use LINUX and MySQL for the Lab Experiments. Though not mandatory, encourage the use of the Eclipse platform.

2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

**List of Experiments:**

1. Use Eclipse or Net bean platform and acquaint yourself with the various menus.

Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, \*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
  
3.
  - A) Develop an applet in Java that displays a simple message.
  - B) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
  
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
  
5. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.
  
6. Write a Java program for the following:
  - Create a doubly linked list of elements.
  - Delete a given element from the above list.
  - Display the contents of the list after deletion.
  
7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in the selected color. Initially, there is no message shown.
  
8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
  
9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are

separated by commas. Write a java program to display the table using Labels in Grid Layout.

10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
  
11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
  
12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
  
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

#### **TEXT BOOKS:**

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.

#### **REFERENCE BOOKS**

1. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
2. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

### **SOFTWARE ENGINEERING LAB**

**B.Tech. II Year I Sem.**

**L T P C**

**0 0 2 1**

**Prerequisites:** A course on “Programming for Problem Solving”.

**Co-requisite:** A Course on “Software Engineering”.

#### **Course Objectives:**

- To have hands-on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

#### **Course Outcomes:**

- Ability to translate end-user requirements into system and software requirements
- Ability to generate a high-level design of the system from the software requirements
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

#### **List of Experiments**

**Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:**

1. Development of problem statements.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

#### **Sample Projects:**

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System

8. Credit Card Processing
9. E-book management System.
10. Recruitment system

**TEXT BOOKS:**

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

**REFERENCE BOOKS:**

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

**DATABASE MANAGEMENT SYSTEMS LAB**

**B.Tech. II Year I Sem.**

**L T P C**

**0 0 2 1**

**Course Objectives:**

- Introduce ER data model, database design and normalization
- Learn SQL basics for data definition and data manipulation

**Course Outcomes:**

- Design database schema for a given application and apply normalization
- Acquire skills in using SQL commands for data definition and data manipulation.
- Develop solutions for database applications using procedures, cursors and triggers

**List of Experiments:**

1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A) Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.)  
B) Nested, Correlated subqueries
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

**TEXT BOOKS:**

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill, 3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw Hill, V edition.

**REFERENCES BOOKS:**

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7<sup>th</sup> Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

**NODE JS/ REACT JS/ DJANGO**

**B.Tech. II Year I Sem.**

**L T P C**

**0 0 2 1**

**Prerequisites:** Object Oriented Programming through Java, HTML Basics.

**Course Objectives:**

1. To implement the static web pages using HTML and do client-side validation using JavaScript.
2. To design and work with databases using Java
3. To develop an end to end application using java full stack.
4. To introduce Node JS implementation for server-side programming.
5. To experiment with single page application development using React.

**Course Outcomes:** At the end of the course, the student will be able to,

1. Build a custom website with HTML, CSS, and Bootstrap and little JavaScript.
2. Demonstrate Advanced features of JavaScript and learn about JDBC
3. Develop Server – side implementation using Java technologies like
4. Develop the server – side implementation using Node JS.
5. Design a Single Page Application using React.

**Exercises:**

1. Build a responsive web application for shopping cart with registration, login, catalog and cart pages using CSS3 features, flex and grid.
2. Make the above web application responsive web application using Bootstrap framework.
3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1 and experiment 2.
4. Explore the features of ES6 like arrow functions, callbacks, promises, async/await. Implement an application for reading the weather information from openweathermap.org and display the information in the form of a graph on the web page.
5. Develop a java stand alone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.
6. Create an xml for the bookstore. Validate the same using both DTD and XSD.
7. Design a controller with servlet that provides the interaction with application developed in experiment 1 and the database created in experiment 5.
8. Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)
9. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.

10. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)
11. For the above application create authorized end points using JWT (JSON Web Token).
12. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.
13. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js
14. Create a TODO application in react with necessary components and deploy it into GitHub.

**REFERENCE BOOKS:**

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2010.
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node ,2nd Edition, APress.

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

**OPERATING SYSTEMS LAB**

**B.Tech. II Year II Sem.**

**L T P C**

**0 0 2 1**

**Prerequisites:**

- A course on “Programming for Problem Solving”.
- A course on “Computer Organization and Architecture”.

**Co-requisite:** A course on “Operating Systems”.

**Course Objectives:**

1. To provide an understanding of the design aspects of operating system concepts through simulation.
2. Introduce basic Unix commands, system call interface for process management, inter process communication and I/O in Unix.

**Course Outcomes:**

1. Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
2. Able to implement C programs using Unix system calls.

**List of Experiments:**

1. Write C programs to simulate the following CPU Scheduling algorithms a) FCFS b) SJF c) Round Robin d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, lseek, stat, fork, exit)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write C programs to simulate the following memory management techniques a) Paging b) Segmentation
7. Write C programs to simulate Page replacement policies a) FCFS b) LRU c) Optimal

**TEXT BOOKS:**

1. Operating System Principles- Abraham Silber chatz, Peter B. Galvin, Greg Gagne 7<sup>th</sup> Edition, John Wiley.
2. Advanced programming in the Unix environment, W. R. Stevens, Pearson education.

**REFERENCE BOOKS:**

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI.
2. Operating System - A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI.
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education.
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education.

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

## **COMPUTER NETWORKS LAB**

**B.Tech. II Year II Sem.**

**L T P C**

**0 0 2 1**

### **Course Objectives:**

- To understand the working principle of various communication protocols.
- To understand the network simulator environment and visualize a network topology and observe its performance
- To analyze the traffic flow and the contents of protocol frames

### **Course Outcomes:**

- Implement data link layer framing methods
- Analyze error detection and error correction codes.
- Implement and analyze routing and congestion issues in network design.
- Implement Encoding and Decoding techniques used in presentation layer
- To be able to work with different network tools

### **List of Experiments**

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijkstra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting techniques used in buffers.
10. **Wireshark**
  - i. Packet Capture Using Wire shark
  - ii. Starting Wire shark
  - iii. Viewing Captured Traffic
  - iv. Analysis and Statistics & Filters.
1. How to run Nmap scan
2. Operating System Detection using Nmap
3. Do the following using NS2 Simulator
  - I. NS2 Simulator-Introduction
  - II. Simulate to Find the Number of Packets Dropped
  - III. Simulate to Find the Number of Packets Dropped by TCP/UDP

- IV. Simulate to Find the Number of Packets Dropped due to Congestion
- V. Simulate to Compare Data Rate & Throughput.
  
- VI. Simulate to Plot Congestion for Different Source/Destination
- VII. Simulate to Determine the Performance with respect to Transmission of Packets

**TEXT BOOK:**

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5<sup>th</sup> Edition.  
Pearson Education/PHI

**REFERENCES:**

1. An Engineering Approach to Computer Networks, S.Keshav, 2<sup>nd</sup> Edition, Pearson Education
2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

### **MACHINE LEARNING LAB**

**B.Tech. II Year II Sem.**

**L T P C**

**0 0 2 1**

#### **Course Objective:**

- The objective of this lab is to get an overview of the various machine learning techniques and can demonstrate them using python.

#### **Course Outcomes:**

- Understand modern notions in predictive data analysis
- Select data, model selection, model complexity and identify the trends
- Understand a range of machine learning algorithms along with their strengths and weaknesses
- Build predictive models from data and analyze their performance

#### **List of Experiments:**

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode, Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for ML application such as Pandas and Matplotlib
4. Write a Python program to implement Simple Linear Regression
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn
6. Implementation of Decision tree using sklearn and its parameter tuning
7. Implementation of KNN using sklearn
8. Implementation of Logistic Regression using sklearn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

#### **TEXT BOOK:**

1. Machine Learning – Tom M. Mitchell, - MGH.

#### **REFERENCE BOOK:**

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.

**R25 B.Tech. AI&DS Syllabus**

**AVNIET**

**DATA VISUALIZATION - R PROGRAMMING/ POWER BI**

**B.Tech. II Year II Sem.**

**L T P C**

**0 0 2 1**

**Course Objectives:**

- Effective use of Business Intelligence (BI) technology (Tableau) to apply data visualization
- To discern patterns and relationships in the data.
- To build Dashboard applications.
- To communicate the results clearly and concisely.
- To be able to work with different formats of data sets.

**Course Outcomes:** At the end of the course a student should be able to

- Understand How to import data into Tableau.
- Understand Tableau concepts of Dimensions and Measures.
- Develop Programs and understand how to map Visual Layouts and Graphical Properties.
- Create a Dashboard that links multiple visualizations.
- Use graphical user interfaces to create Frames for providing solutions to real world problems.

**Lab Problems:**

1. Understanding Data, what is data, where to find data, Foundations for building Data Visualizations, Creating Your First visualization?
2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts (line, bar charts, Tree maps), Using the Show me panel.
3. Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.
4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, Formatting specific parts of the view.
5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.
6. Structuring your data, Sorting and filtering Tableau data, Pivoting Tableau data.
7. Advanced Visualization Tools: Using Filters, Using the Detail panel, using the Size panels, customizing filters, Using and Customizing tooltips, Formatting your data with colors.

8. Creating Dashboards & Storytelling, creating your first dashboard and Story, Design for different displays, adding interactivity to your Dashboard, Distributing & Publishing your Visualization.
9. Tableau file types, publishing to Tableau Online, Sharing your visualizations, printing, and Exporting.
10. Creating custom charts, cyclical data and circular area charts, Dual Axis charts.

**REFERENCES:**

1. Microsoft Power BI cookbook, Brett Powell, 2nd edition.
2. R Programming for Data Science by Roger D. Peng (References)
3. The Art of R Programming by Norman Matloff Cengage Learning India.