25MA301BS: MATHEMATICAL AND STATISTICAL FOUNDATIONS

Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year I Sem.

L T P C
3 0 0 3

Pre-requisites: Mathematics courses of first year of study

Objectives: To learn

1. The Number Theory basic concepts use ful for cryptography etc.
2. The theory of Probability, and probability distributions of single random variables.
3. The sampling theory and testing of hypothesis and making inferences.
4. The curve fitting, correlation and regression for the given data.

Course Outcomes: After learning the contents of this paper, the student must be able to

1. Apply the number theory concepts to cryptography domain.
2. Apply the concepts of probability and distributions to some case studies.
3. Correlate the material of one unit to the material in other units.
4. Resolve the potential misconceptions and hazards in each topic of study.
5. Fit the curve, correlation and regression for the given data.

UNIT-I: Basics of Number Theory **10L**

Greatest Common Divisors and Prime Factorization: Greatest common divisors – The Euclidean algorithm – The fundamental theorem of arithmetic – Factorization of integers and the Fermat numbers. Congruence's: Introduction to congruences –Linear congruences.

UNIT-II: Random Variables and Probability Distributions **8L**

Concept of a Random Variable-Discrete Probability Distributions-Continuous Probability Distributions- Mean of a Random Variable-Variance of a Random Variable

Discrete Probability Distributions: Binomial Distribution- Poisson distribution

UNIT-III: Continuous Distributions and Sampling **10L**

Uniform Distribution – Normal Distribution – Areas under the Normal Curve – Applications of the Normal Distribution – Normal Approximation to the Binomial Distributions. Fundamental Sampling Distributions: Random Sampling – Some Important Statistics – Sampling Distributions – Sampling Distribution of Means – Central Limit Theorem.

UNIT-IV: Tests of Hypotheses (Large and Small Samples) **10L**

Statistical Hypotheses: General Concepts-Testing a Statistical Hypothesis. Single sample: Tests concerning a single mean. Two samples: Tests on two mean (Unknown for equal variance). One sample: Test on a single proportion. Two samples: Tests on two proportions. Two- sample tests concerning variances: F-distribution

UNIT-V: Applied Statistics **10L**

Curve fitting by the method of least squares – Fitting of straight lines – Second degree parabolas and more general curves – Correlation and Regression – Rank correlation.

TEXT BOOKS:

1. Kenneth H. Rosen, Elementary Number Theory & its Applications, sixth edition, Addison Wesley, ISBN 978 0-321-50031-1.
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical Statistics, Khanna publications.

REFERENCE BOOKS:

1. T.T.Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons, Ltd, 2004.
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, academic press.
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications.

25CS302PC: COMPUTER ORGANIZATION AND ARCHITECTURE

Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year I Sem.

L T P C
3 0 0 3

Pre requisites: No prerequisites.
Co-requisite: A Course on "Digital Electronics".

Course Objectives:
After completing this course, students will be able to:

1. Understand fundamental concepts of digital systems including Boolean algebra, logic gates, Binary codes and data representation techniques.
2. Analyze and design combinational and sequential circuits using standard procedures and Hardware Description Language (HDL).
3. Explain register transfer language and micro-operations, and understand basic computer Organization and instruction execution.
4. Describe CPU organization and control mechanisms, including micro programmed control, Instruction formats, and addressing modes and arithmetic operations.
5. Understand memory and input-output organization, including data transfer methods, memory Hierarchy and interfacing techniques.

Course Outcomes:
After successful completion of the course, students will be able to:

1. Apply Boolean algebra and digital logic principles to design and simplify logic circuits.
2. Design combinational and sequential circuits.
3. Analyze computer operations at the register-transfer level.
4. Evaluate CPU architecture and control unit design.
5. Understand memory hierarchy and I/O organization.

UNIT- I:
Boolean Algebra and Logic Gates: Binary codes, Binary Storage and Registers, Binary ogic **Digital 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:
Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder- Subtractor, Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.
Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, state Reduction and Assignment, Design Procedure**.** Registers shift Registers, Ripple counters, synchronous counters, other counters.

UNITIII
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

Micro programmed 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. Decimal Arithmetic unit, Decimal 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,3rd Edition, Pearson/PHI.
2. Computer System Architecture-M. Morris Mano, 3rdEdition, Pearson/PHI.

REFERENCE BOOKS:

1. Switching and Finite AutomataTheory, ZVI. Kohavi, 2nd Edition, Tata McGraw Hill.
2. Computer Organization – Carl Hamacher, Zvonks Vranesic, Safea Zaky, 5thEdition, McGraw Hill.
3. Computer Organization and Architecture-William Stallings 6thEdition, Pearson/PHI.
4. Structured Computer Organization-Andrew S. Tanenbaum, 4thEdition, PHI/Pearson.

25CS303PC: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

1. Provides understanding of fundamental object-oriented programming concepts and their application in problem solving.
2. Emphasizes the use of inheritance for code reusability.
3. Demonstrates multitasking through multithreading and event handling mechanisms.
4. Introduce event-driven programming using the delegation event model.
5. Covers the Swing framework for building flexible and scalable GUI applications in Java.

Course Outcomes: After successful completion of the course, students will be able to:

1. Identify fundamental behavior of object-oriented programming concepts and their application.
2. Apply the inheritance concept for code reusability.
3. Apply multithreading concepts for inter-process communication mechanisms.
4. Apply GUI concepts for user-friendly applications.
5. Make use of Swing concepts for integrating event handling mechanisms.

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 CLASS PATH, 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 sub classes. 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, mutlipackage, Exploring java. i/o 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 gridbag.

UNIT-V

Swing–Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JFrame and JComponent, JLabel, ImageIcon, JTextField, JButton, JCheckBox, JRadioButton, JList, JCombo Box, Tabbed Panes, Scroll Panes, Trees, and Tables. Menu Basics, Menu related classes - Jmenu Bar,JMenu, JMenuItem, JCheck Box Menu Item, Radio Button MenuItem, JSeperator, creatinga popup menu

TEXT BOOKS:

1. Java the complete reference, Herbert schildt, Dr. Denny Coward,13th Edition, McGraw Hill.
2. Understanding OOP with Java, T. Budd,3rd edition, Pearson education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J. Ninoand F. A. Hosch, 2nd edition, John Wiley & sons.
2. An Introduction to OOP, 3rd Edition, T. Budd, Pearson Education.
3. Introduction to Java programming, Y. Daniel Liang, 2nd Edition, Pearson Education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson-Thomson, 1st Edition.
5. CoreJava2, Vol1, Fundamentals, Cay. S. Horstmann and Gary Cornell, 8thEdition, Pearson Education.
6. Core Java 2, Vol 2, Advanced Features, Cay. S. Horstmann and Gary Cornell, 8thEdition, Pearson Education
7. Object Oriented Programming with Java, R. Buyya, S. T. Selvi, X. Chu,1st Edition, TMH.
8. Java and Object Orientation, an introduction, John Hunt, 2nd Edition, Springer.
9. Maurach's Beginning Java2, JDK5, 5th Edition, SPD.

25CS304PC: SOFTWARE ENGINEERING

Common to (CSE, CSE (AI&ML), & CSE (DS))

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.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams.
3. To provide knowledge of various software process models (like Waterfall, Agile) and their application in real-world software development.
4. To build skills in software requirement analysis, system design, and use of UML diagrams for building efficient and scalable software systems.
5. To understand testing methods, risk management, and quality assurance techniques for delivering reliable and high-quality software projects.

Course Outcomes: After successful completion of the course, students will be able to:

1. Analyze and evaluate software process models and frameworks.
2. Create structured Software Requirements Specifications (SRS) documents for real-time problems.
3. Design and develop software architecture to choose an appropriate design model for appropriate applications.
4. Evaluate different testing strategies.
5. Classify different risks and quality metrics.

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, flowcharts Vs UML, 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:

1. Software Engineering, A practitioner's Approach-Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering-Sommerville, 7thedition, Pearson Education.
3. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, 2nd Edition, Pearson Education.

REFERENCE BOOKS:

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

25CS305PC: DATA BASE MANAGEMENT SYSTEMS

Common to (CSE, CSE (AI&ML), & CSE (DS))

B.Tech. II Year I Sem

L T P C
3 0 0 3

Prerequisites: A course on "Data Structures".

Course Objectives:

1. To understand fundamental concepts of database systems and their real-world applications.
2. To learn data modeling techniques and design databases using ER and relational models.
3. To gain proficiency in SQL for data definition, manipulation, and querying.
4. To understand normalization techniques and improve database design quality.
5. To study transaction management, concurrency control, and database storage mechanisms

Course Outcomes: After successful completion of the course, students will be able to:

1. Able to understand the basic concepts of database systems, DBMS architecture, and the ER model for database design.Use the relational model to apply integrity constraints and execute relational algebra and Calculus queries.
2. Able to develop SQL queries, constraints, triggers, and apply normalization techniques to design efficient database schemas. Use transaction management, concurrency control, and recovery methods to handle database operations effectively.
3. Able to understand file organization, indexing methods, and tree-based indexing techniques such as B+ trees.

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 data base 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, multi valued 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, 5th Edition, McGraw hill.
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, 3rd Edition, Tata McGraw Hill.

REFERENCE BOOKS:

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

25MA306PC: COMPUTATIONAL MATHEMATICS LAB
(Using Python/MATLAB software)

B. Tech. II Year I Sem. L T P C
0 0 2 1

Pre-requisites: Matrices, Iterative methods and ordinary differential equations

Course Objectives: To learn
1. Solve problems of Eigen values and Eigen Vectors using Python/MATLAB.
2. Solution of Algebraic and Transcendental Equations using Python/MATLAB
3. Solve problems of Linear system of equations
4. Solve problems of First-Order ODEs Higher order linear differential equations with constant coefficients

Course outcomes: After learning the contents of this paper, the student must be able to
1. Develop the code to find the Eigen values and Eigen Vectors using Python/MATLAB.
2. Develop the code find solution of Algebraic and Transcendental Equations and Linear system of equations using Python/MATLAB
3. Write the code to solve problems of First-Order ODEs Higher order linear differential equations with constant coefficients

***Visualize all solutions Graphically through programmes**

UNIT-I: Eigen values and Eigen vectors: **6**
Programs:
- Find ingreal and complex Eigen values.
- Finding Eigen vectors.

UNIT-II: Solution of Algebraic and Transcendental Equations **6**
Bisection method, Newton Raphson Method
Programs:
- Root of a given equation using Bisection method.
- Root of a given equation Newton Raphson Method.

UNIT-III: Linear system of equations: **6**
Jacobi' site ration method and Gauss-Seidal iteration method
Programs:
- Solution of given system of linear equations using Jacobi's method
- Solution of given system of linear equations using Gauss-Seidal method

UNIT-IV: First-Order ODEs **8**
Exact and non-exact equations, Applications: exponential growth/decay, Newton's law of cooling.
Programs:
- Solving exact and non-exact equations
- Solving exponential growth/decay and Newton's law of cooling problems

UNIT-V: Higher order linear differential equations with constant coefficients **6**

Programs:

- Solving homogeneous ODEs
- Solving non-homogeneous ODEs

TEXT BOOKS:

1. MATLAB and its Applications in Engineering, Raj kumar Basal, Ashok Kumar Geo, Manoj Kumar Sharma, Pearson publication.
2. Kenneth A. Lambert, The fundamentals of Python: First Programs, 2011, Cengage Learnings.
3. Think Python First Edition, by Allen B. Downey, Orielly publishing.
4. Introduction to Python Programming, William Mitchell, PovelSolin, Martin Novak etal, NCLab Public Computing, 2012.
5. Introduction to Python Programming, ©Jacob Fredslund, 2007.

REFERENCE BOOKS:

1. An Introduction to Python, John C. Lusth, The University of Alabama, 2011.
2. Introduction to Python, ©DaveKuhlman, 2008.

25CS307PC: OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB
(Common to CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year I Sem.

L T P C
0 0 2 1

Course Objectives:

1. Understand Java IDEs, debugging, coding, and refactoring techniques.
2. Apply Swing concepts to build GUI-based applications.
3. Apply event-driven programming and exception handling mechanisms.
4. Analyze problems using core Java and object-oriented principles.
5. Create complete Java applications integrating GUI and advanced features.

Course Outcomes: After successful completion of the course, students will be able to:

1. Implement the Java Collection Framework to solve real-world problems.
2. Implement programs using abstract classes and object-oriented principles.
3. Implement inter process communication using multithreading concepts.
4. Develop a Java application with exception handling.
5. Design and build GUI applications.

Note:

1. Use LINUX and My SQL 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. Trycode formatter and code refactoring like renaming variables, methods, and classes. Trydebug step bystepwithasmallprogramofabout10to15lineswhichcontainsatleastoneifelsecondition 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 simple message.
   B) DevelopanappletinJavathatreceivesanintegerinonetextfield,andcomputesits factorial

4. Value and returns it in another text field, when the button named "Compute" is clicked.

5. Write a Java program that create sauser interface to perform integer divisions.Theuserenters twonumbersinthetextfields,Num1andNum2.Thedivisionof Num1andNum 2isdisplayed 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.

6. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every1second 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.
7. 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.

8. 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.

9. 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.

10. Suppose that at able named Table.txt iss to red 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.
11. Write a java program to display the table using Labels in Grid Layout.

12. 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).

13. 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).

14. Write a Java program that correctly implements the producer-consumer problem using the Concept of inter thread communication.

15. 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, 1st Edition, Pearson Education.

REFERENCE BOOKS

1. Java Programming,D.S. Malik and P.S.Nair,1st Edition, Cengage Learning.
2. Core Java, Volume1,9th Edition, Cay S.Horstmann and GCornell, Pearson.

25CS308PC: SOFTWARE ENGINEERING LAB
(Common to CSE, CSE (AI&ML), & CSE (DS))

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:

1. 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.
2. Gain practical exposure to different SDLC models like Waterfall and Agile during project development.
3. Develop skills in creating use case, class, sequence, and activity diagrams for real-time projects.
4. Learn to design test cases, execute testing methods, and identify as well as fix defects in software.
5. Acquire hand-on experience with tools for version control (like Git), project management, and documentation.

Course Outcomes: After successful completion of the course, students will be able to:

1. Apply software engineering concepts to formulate Software Requirement Specification (SRS) documents for real-world applications.
2. Design software system models using appropriate design techniques and CASE tools.
3. Prepare software configuration management and risk management plans for project execution.
4. Awareness of testing problems and will be able to develop a simple testing report.
5. Evaluate software quality by designing and applying white-box and black-box testing techniques.

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. Pass port 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 Modelling Language User Guide, Grady Booch, James Rambaugh, Ivar Jacobson, 1st Edition, Pearson Education.

REFERENCE BOOKS:

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

25CS309PC: DATABASE MANAGEMENT SYSTEMS LAB
(Common to CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year I Sem.

L T P C
0 0 2 1

Course Objectives:

1. Understand ER modeling and schema design.
2. Practice normalization techniques.
3. Develop SQL skills.
4. Implement advanced queries.
5. Build applications using procedures and triggers.

Course Outcomes: After successful completion of the course, students will be able to:

1. Apply E-R modeling and relational design techniques to construct database schemas.
2. Apply normalization and SQL (DDL, DML) commands to create and manage database structures and data.
3. Apply SQL queries including joins, subqueries, and aggregate functions to retrieve and manipulate data effectively.
4. Analyze database queries using nested queries, correlated subqueries, and views for efficient data retrieval.
5. Analyze and implement 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 sub queries
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, Raghu rama Krishnan, Johannes Gehrke,3rd Edition, Tata McGraw Hill.
2. Database System Concepts, Silberschatz, Korth,5th Edition, McGraw Hill.

REFERENCE BOOKS:

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

25CS310SD: SKILL DEVELOPMENT COURSE
(Data Visualization – R Programming / Power BI)
Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year I Sem.

L T P C
0 0 2 1

Course Objectives:

1. Effective use of Business Intelligence (BI) technology (Tableau) to apply data visualization
2. To discern patterns and relationships in the data.
3. To build dashboard applications.
4. To communicate the results clearly and concisely.
5. To be able to work with different formats of datasets.

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

1. Analyze how to import data into Tableau.
2. Apply tableau concepts for Dimensions and Measures.
3. Develop Programs to map Visual Layouts and Graphical Properties.
4. Create a dashboard that links multiple visualizations.
5. Apply graphical user interfaces to create Frames for 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, cyclic al data and circular area charts, Dual Axis charts.

REFERENCES:

1. Microsoft Power BIcook book, Brett Powell, 2ndEdition.
2. R Programming for Data Science by Roger D.Peng (References), 1st Edition.
3. The Art of R Programming by Norman Matloff Cengage Learning India, 1st Edition.

25CS301PC/25CS401PC: DISCRETE MATHEMATICS

Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year IISem.

L T P C
3 0 0 3

Course Objectives:

1. Understand fundamentals of mathematical logic, propositions, connectives, and inference techniques.
2. Apply set theory concepts, relations, functions, and ordering in discrete structures.
3. Analyze algebraic structures including semigroups, monoids, lattices, and Boolean algebra.
4. Solve combinatorial problems using counting principles, permutations, combinations, and binomial theorems.
5. Explore graph theory concepts including trees, circuits, colouring, and planar graphs.

Course Outcomes: After successful completion of the course, students will be able to:

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 in combinatories.
5. Apply graph theory in solving computing problems.

UNIT-I

Mathematical logic: Introduction, Statements and Notation, Connectives, Normal Forms, 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, 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 Exclusion.

UNIT-V

Graph Theory: Basic Concepts, Isomorphism and Sub graphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, 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, 1st Edition, McGraw-Hill.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joel.Mott, Abraham Kandel, Teodore P. Baker, 2nd Edition, Prentis Hall of India.

REFERENCE BOOKS:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph. P. Grimald,5th Edition, Pearson Education.
2. DiscreteMathematicalStructures:ThomasKosy,1st Edition Tata McGraw Hill Publishing co.

25CS402PC: OPERATING SYSTEMS

Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year II Sem. **L T P C**
3 0 0 3

Course Objectives:

1. Understand fundamentals of mathematical logic, propositions, connectives, and inference techniques.
2. Apply set theory concepts, relations, functions, and ordering in discrete structures.
3. Analyze algebraic structures including semigroups, monoids, lattices, and Boolean algebra.
4. Solve combinatorial problems using counting principles, permutations, combinations, and binomial theorems.
5. Explore graph theory concepts including trees, circuits, colouring, and planar graphs.

Course Outcomes: After successful completion of the course, students will be able to:

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 in combinatories.
5. Apply graph theory in solving computing problems.

UNIT-I

Mathematical logic: Introduction, Statements and Notation, Connectives, Normal Forms, 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, 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 Exclusion.

UNIT-V

Graph Theory: Basic Concepts, Isomorphism and Sub graphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

TEXT BOOKS:

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

REFERENC EBOOKS:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph. P. Grimald,5th Edition, Pearson Education.
2. DiscreteMathematicalStructures:ThomasKosy,1st Edition Tata McGraw Hill Publishing co

25CS403PC: ALGORITHMS DESIGN AND ANALYSIS

Common to (CSE, CSE (AI&ML), & CSE (DS))

B. Tech. II Year II Sem.

L T P C
3 0 0 3

Course Objectives

1. Understand the fundamental concepts of algorithms, time and space complexity, and analyze performance using asymptotic notations.
2. Apply divide-and-conquer strategies to design and solve computational problems efficiently.
3. Implement and analyze data structures and algorithms such as disjoint sets, heaps, graphs, and tree traversal techniques.
4. Utilize advanced problem-solving techniques such as backtracking, greedy methods, and dynamic programming to solve optimization problems.
5. Analyze and solve complex problems using branch-and-bound techniques and understand NP-Hard and NP-Complete problem classifications.

Course Outcomes

After Completion of the course, students are able to:

1. Analyze algorithms by measuring time and space complexity.
2. Device divide-and-conquer, greedy method, backtracking and dynamic programming solutions.
3. Apply disjoint sets and heaps data structures in different algorithmic strategies.
4. Solve real-world problems by identifying appropriate algorithmic techniques.
5. Analyze and classify problems into P and NP classes.

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, Heap sort **Back tracking**: General method, applications, n-queens problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

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, Connected components, Biconnected components.

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, Cook's theorem.

TEXT BOOK:

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

REFERENCE BOOKS:

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

25CS405PC: MACHINE LEARNING

Common to (CSE, CSE (AI&ML), & CSE (DS))

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

1. Understand basic concepts of machine learning.
2. Gain the knowledge of feature engineering and model evaluation techniques.
3. Identify an appropriate model for the real-world problem.
4. Distinguish between supervised, unsupervised and semi-supervised learning.
5. Understand neural networks, deep learning, and reinforcement learning.

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 bootstrapping, 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 line arregression, 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 K-Medoids, 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,1st Edition, Pearson
2. M N Murty, Anatha narayana V S- Machine Learning,1st Edition, University Press
3. Tom M Mitchell, Machine Learning,1st Edition, McGraw Hill Education, 2013.

REFERENCE BOOKS:

1. Stephen Mars land, Machine Learning-An Algorithmic Perspective, Second Edition.
Chapman and Hall/ CRC Machine Learning and Pattern Recognition Series ,2014, Student Edition.

25MS406HS: INNOVATION AND ENTREPRENEURSHIP

Common to (CSE, CSE (AI&ML), & CSE (DS))

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 one stabilizing 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

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, GenAI, '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 Stratup 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, McGraw Hill, 11th Edition.
6. NISP-Brochureinsidepages- startup_policy_2019.pdf

25CS407PC: OPERATING SYSTEMS LAB
(Common to CSE, CSE (AI&ML), & CSE (DS))

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 understand and implement various CPU scheduling algorithms such as FCFS, SJF, Round Robin, and Priority scheduling.
2. To gain hands-on experience with UNIX/Linux system calls for file and process management.
3. To analyse and implement deadlock avoidance and synchronization techniques, including Banker's Algorithm and Producer-Consumer problem.
4. To explore and implement different Inter-Process Communication (IPC) mechanisms such as Pipes, FIFOs, Message Queues, and Shared Memory.
5. To understand and simulate memory management techniques including Paging, Segmentation, and Page Replacement algorithms.

Course Outcomes:

1. Implement different CPU scheduling algorithms.
2. Develop programs using UNIX/Linux system calls.
3. Implement deadlock avoidance algorithm.
4. Apply Interprocess communication and synchronization mechanisms to solve real-world problems.
5. Implement memory management techniques and page replacement algorithms.

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

25CS408PC: COMPUTER NETWORKS LAB

(Common to CSE, CSE (AI&ML), & CSE (DS))

B.Tech. II Year II Sem

L T P C
0 0 2 1

Course Objectives:

1. Understand and implement data link layer protocols such as framing methods and flow control techniques.
2. Develop skills in applying error detection and correction techniques like CRC and encoding/decoding methods.
3. Understand and implement routing algorithms such as Dijkstra's algorithm and Distance Vector Routing.
4. Analyze and simulate network performance and congestion control mechanisms using tools like NS2 simulator.
5. Familiarize students with network analysis tools such as Wireshark and Nmap for monitoring and analyzing network traffic.

Course Outcomes: After Completion of the course, students are able to:

1. Implement data link layer protocols functionalities.
2. Implement error detection techniques.
3. Implement and analyze routing and congestion issues in network design.
4. Implement encoding and decoding techniques used in the presentation layer.
5. Simulate 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 CRCCCIP
3. Develop as imple datalink layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra'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 Wireshark
    ii. Starting Wireshark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.

1. How to run N map scan
2. Operating System Detection using N map
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, 5th Edition. Pearson Education/PHI.

REFERENCES:

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

25CS409PC: MACHINE LEARNING LAB

(Common to CSE, CSE (AI&ML), & CSE (DS))

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

1. Implement statistical measures on various data sets.
2. Apply basic Python libraries on data analysis.
3. Implement and analyze different machine learning algorithms
4. Build predictive models from data sets.
5. Apply machine learning techniques for real-world applications.

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 sk learn
6. Implementation of Decision tree using sk learn and its parameter tuning
7. Implementation of KNN using sk learn
8. Implementation of Logistic Regression using sk learn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

TEXT BOOK:

1. MachineLearning-TomM.Mitchell,1st Edition, MGH.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, 2nd Edition, Taylor & Francis.

25CS410SD: SKILL DEVELOPMENT COURSE
(Node JS / React JS / Django)
Common to (CSE, CSE (AI&ML), & CSE (DS))

B.Tech. II Year II Sem.

L T P C
0 0 2 1

Prerequisites: Object Oriented Programming through Java, HTML Basics.

Course Objectives:

1. Implement static web pages using HTML and perform client-side validation using JavaScript.
2. Design and work with databases using Java.
3. Develop an end-to-end application using Java Full Stack technologies.
4. Understand Node.js implementation for server-side programming.
5. Develop with Single Page Application development using React.

Course Outcomes: After successful completion of the course, students will be able to:

1. Build a custom website using HTML, CSS, Bootstrap, and basic JavaScript.
2. Implement advanced features of JavaScript and JDBC.
3. Develop server-side applications using Java technologies.
4. Develop 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 RESTAPI to perform CRUD operations on student data. (Use Postman)
11. For the above application create authorized endpoints using JWT (JSON WebToken).
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 To-Do application in React with necessary components and deploy it into GitHub.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2nd Edition, 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, A Press.

NALLA NARASIMHA REDDY
Education Society's Group of Institutions - Integrated Campus
(UGC AUTONOMOUS INSTITUTION)
EAMCET/ECET/ICET/PGECET Code NNRG

25VA400HS: INDIAN KNOWLEDGE SYSTEM

Common to (CSE, CSE (AI&ML), & CSE (DS))

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:

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

Course Outcomes: Students will be able to:

1. Understand nature, scope and related fields of Indian knowledge system.
2. DemonstratethescientificliteratureavailableinancientIndiantraditions
3. Understanding the application of Bharatiya Jnana Parampara
4. Understand Indian approach towards Wellbeing
5. Appreciate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world

Unit1: 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 Sanskritin Natural Language Processing-Scientific Literature - Vedic Literature - Available Scientific Treatises - Interlinkings

Unit3: 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

Unit4: Introduction to Ancient Indian Wellness Systems Concept of Wellness-Yoga System - Ayurveda System-Ancient Indian Aesthetics

Unit5: 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:

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

Video Resources:

1. Introductory lectures by Prof. Gauri Mahulikar
2. Introductory lectures by Prof. Kapil Kapoor

Websites:

1. https://iksindia.org/index.php
   a. 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)

b. https://www.youtube.com/watch?v=MDJTXNiH2_AMukul Kanitkar on Bharatiya Knowledge System https://www.youtube.com/watch?v=uARMhv97pjk https://www.youtube.com/watch?v=oTwgf56GbsA

c. Scientific History of India | Mukul Kanitkar Lecture in DTU https://youtu.be/gNJNmPJqXJc?si=WFBbuUT65mLZzpOW

2. Ancient India's Scientific Achievements & Contribution in Mathematics, Astronomy, Science & Medicine