CVR COLLEGE OF ENGINEERING

II B.Tech. CSE (Cyber Security)

I Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS202	Object Oriented Programming through Java	PC	3	0	3	40	60	100	2
2	22CS201	Discrete Mathematics	PC	3	0	3	40	60	100	4
3	22HS204	Mathematical and Statistical Foundations	BS	3	1	4	40	60	100	6
4	22CS203/253	Database Management Systems	PC	3	0	3	40	60	100	8
5	22DT201	Digital Electronics and Design	ES	3	0	3	40	60	100	10
Practicals										
6	22CS231	Object Oriented Programming through Java Lab	PC	0	3	1.5	40	60	100	12
7	22CS232/282	Database Management Systems Lab	PC	0	2	1	40	60	100	14
8	22DT231	Data Visualization Lab	PC	0	3	1.5	40	60	100	17
		Total		**15**	**9**	**20**	**320**	**480**	**800**	
		Total Hours		**24**						
9	22HS201/251	Constitution of India	MC	**3**	**0**	**0**	**100**	**0**	**100**	18

II Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS252	Advanced Data Structures through Java	PC	3	0	3	40	60	100	20
2	22DT251	Design and Analysis of Algorithms	PC	3	0	3	40	60	100	22
3	22CY251	Computer Organization and Architecture	PC	3	0	3	40	60	100	24
4	22CY252	Computer Networks	PC	3	0	3	40	60	100	26
5	22IT252	Operating Systems	PC	3	0	3	40	60	100	28
Practicals										
6	22CS281	Advanced Data Structures through Java Lab	PC	0	2	1	40	60	100	30
7	22CY281	Computer Networks Lab	PC	0	2	1	40	60	100	32
8	22IT282	Operating Systems Lab	PC	0	2	1	40	60	100	33
9	22CY284	Real-Time/Field-Based Research Project	PC	0	4	2	50	0	50	
		Total		**15**	**10**	**20**	**370**	**480**	**850**	
		Total Hours		**25**						
10	22HS281/231	Gender Sensitization Lab	MC	0	2	0	100	0	100	34

Service Courses of II B.Tech. II Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CY251	Computer Organization and Architecture (CSE-AI&ML, CSE-DS, AI&DS, AIM)	PC	3	0	3	40	60	100	24

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

HS: Humanities & Sciences **BS: Basic Sciences** **ES: Engineering Sciences**
PC: Professional Core **MC: Mandatory Course**

Course Code: 22CS202

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

(Common to CSE, CSE -AI&ML, CSE-CS, CSE-DS, AI&DS, AIM and IT)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand and apply various object-oriented programming features like abstraction, encapsulation, inheritance and polymorphism to solve various computing problems using Java language.
2. To identify, define and implement exception handling and multi-threading mechanisms in application domains.
3. To design and develop GUI applications using AWT & Swings and Understanding of the new features.

Unit I - Introduction to Java and Building Blocks of Java

Basics of Java- History/Background of Java, Java Buzzwords, Java Virtual Machine and Byte code, Java Environment setup, Java Program structure, Data Types, Variables- Scope and Life Time, Operators, Expressions, Type Conversions and Type casting, Conditional statements and Control statements, Simple Java Programs, javac and java command flags.

OOP Concepts –I: Encapsulation- Classes and Objects, Classes: Class structure, class components, Objects: Object declaration, Reference variables, Constructors - default Constructor, Parameterized Constructors, Constructor overloading, this keyword and its uses, arrays concept, static modifier, access modifiers, Wrapper classes.

Methods -Passing parameters to methods – Passing primitive types and Passing Objects, getters and setters, Method Overloading, Command line arguments, garbage collection- java.lang.System.gc(), finalize(). **String Handling -** String class, String APIs, String Buffer and String Builder classes.

Unit II - OOP Concepts –II

Inheritance- Inheritance concept, super class and subclass relationship, Object class, principle of substitution, effect of access modifiers on inheritance. Usage of super (field, method, constructor) and final(field, class, method) keywords.

Polymorphism- method overriding, Dynamic method dispatch, Abstract classes and Interfaces - Abstract classes - concept, usage, Interfaces – declaration, implementation, components of an interface, extending interfaces.

Packages – package access, CLASSPATH, package access rules, sealed classes, hidden classes, Introduction to Java standard library and Java documentation.

Unit III - Dealing exceptions and I/O

Exception Handling: Fundamentals of exception handling, benefits of exception handling, Exception types, Termination or presumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, exception hierarchy, throw, throws and finally, built in Exceptions, Custom exceptions, Throwable Class.

Java I/O–Byte streams, character streams, Scanner class, Console class, Serialization and Serializable interface, File class.

Unit IV – Multithreading and Modules

Multithreading-Fundamentals, Thread Life Cycle, Ways of creating threads - Thread class and Runnable interface, Thread priorities, creating multiple threads, core methods of Thread class, Thread Synchronization, inter thread communication.

Annotations- Annotation Basics, specifying a Retention Policy, the Annotated Element Interface, Using Default Values, Marker Annotations, Single – Member Annotations.

Modules: Module Basics-module, exports, require, transitive, java.base and the Platform Modules, Unnamed Module, Specific Module.

Unit V - GUI Development

AWT - Basics of GUI Programming, Event handling – Delegation event model, event sources, event listeners, event classes, adapter classes: nested classes and interfaces, anonymous inner classes handling keyboard and mouse events.

Swing- MVC Architecture, Containers, components, layout managers, frames and windows, panels, buttons, checkboxes, radio buttons, combo boxes, lists, labels, color choosers, file choosers, text fields, text areas, tool tips.

Course Outcomes:

At the end of the course, student should be able to

CO 1 **:** Design and implement object-oriented concepts like encapsulation, abstraction and data hiding using programming constructs offered by java language.
CO 2 **:** Realize the power of inheritance, interfaces, and packages.
CO 3 **:** Understand and demonstrate the concepts of exception handling and java io streams.
CO 4 **:** Demonstrate knowledge and understanding of multi-threading, annotations, and modules in Java.
CO 5 **:** Design and develop java applications using AWT & Swings and make use of the advanced features for providing solutions to real world problems.

Textbooks:

1. Java: The Complete Reference, Herbert Schildt, 11th edition, McGraw-Hill Education, Oracle Press, 2019.
2. Head First Java, Kathy Sierra and Bert Bates, 2nd edition, O`Reilly Media, 2005.

References:

1. Core Java Volume I- Fundamentals, Cay S. Horstmann and Gary Cornell, 9th edition, Prentice Hall, 2012.
2. Core Java Volume II- Advanced Features, Cay S. Horstmann and Gary Cornell, 9th edition, Prentice Hall, 2013.

Course Code: 22CS201

DISCRETE MATHEMATICS

(Common to CSE, CSE-AI&ML, CSE-CS, CSE-DS and IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To inculcate mathematical thinking and problem-solving skills in Logic, Relations, and Inferences.
2. To expose students to a wide variety of mathematical concepts that are used in Computer Science based on Number Theory and Combinatorics.
3. To represent real-world problems over Graphs and solve similarity and traversal related problems.

Unit I - Mathematical Logic

Statements and notations, connectives, Well Formed Formulas, Truth tables, tautology, equivalence implication, Normal forms, Predicative logic, Quantifiers, universal quantifiers, Free & Bound variables.

Unit II – Inference and Relations

Rules of inference, Consistency, Proof by contradiction, Automatic Theorem proving, and Applications.

Properties of binary Relations, Equivalence, Transitive closure, Compatibility & Partial ordering Relations, Lattice and its properties, Hasse Diagram. Recursive functions, and Applications.

Unit III - Algebraic structures

Algebraic systems Examples and general properties, semi-groups and Monoids, Groups, subgroups, Homomorphism & Isomorphism, and Applications.

Unit IV - Elementary Combinatorics and Recurrence Relations

The principle of inclusion and exclusion, Binomial Coefficients, Binomial & Multinomial theorems, Pigeon hole principle, and its applications.

Generating Functions-Generating Functions of sequences, calculating the coefficient of generating function and applications. Recurrence Relations- Homogenous and non-homogeneous, and their solutions.

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.

Course Outcomes:

CO 1 **:** Apply formal logic proofs and/or informal, but rigorous, logical reasoning to evolve theoretical proofs to real problems, such as predicting the behavior of software or solving problems such as puzzles.

CO 2 **:** Apply the logical notations to define and reason about fundamental mathematical concepts such as sets, and relations and exercise the guidelines for constructing valid arguments. A representation of a partially ordered set such as a lattice as a directed graph.

CO 3 **:** Define Group properties and construct simple functions that preserve the algebraic structures over groups.

CO 4 **:** Solve counting problems efficiently by applying the principle of inclusion and exclusion and solve recurrence relations.

CO 5 **:** Characterize edge preserving similarity between two graphs and verify the Eulerian property of graphs.

Textbooks:

1. Discrete Mathematical Structures with Applications to Computer Science, J.P. Tremblay, and R. Manohar, Tata McGraw-Hill Publishing Company, 2008.
2. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott., A. Kandel and T.P. Baker, 2nd edition, Prentice Hall, 2009.

References:

1. Discrete Mathematics and its Applications, Kenneth H.Rosen, 7th edition, TMH, 2015.
2. Discrete and Combinatorial Mathematics- An Applied Introduction, Ralph P. Grimaldi, 5th edition, Pearson Education, 2008.
3. Elements of Discrete Mathematics – A computer Oriented Approach, C L Liu, and D P Mohapatra, 3rd edition, Tata McGraw-Hill, 2008.

Course Code:22HS204

MATHEMATICAL AND STATISTICAL FOUNDATIONS

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

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	1	Semester End Examination	:	60 Marks
Credits	:	4	Semester End Exam Duration	:	3 Hours

Course Objectives: By studying this course students can:

1. To understand number theory basic concepts are useful for cryptography etc.
2. To learn how to apply Probability and Statistics to solve engineering problems
3. To show the applications of Probability and Statistics in engineering with examples
4. To learn how to apply testing of hypothesis techniques to make decisions in real-time problems.
5. Stochastic process and Markov chains

UNIT I - Greatest Common Divisors, Prime Factorization, and Congruences

Greatest Common Divisors and Prime Factorization: Greatest common divisors, The Euclidean algorithm, The fundamental theorem of arithmetic, Factorization of integers, and the Fermat numbers.

Congruences: Introduction to congruences, Linear congruences, The Chinese remainder theorem, and System of linear congruences.

UNIT II – Random Variables, Discrete and Continuous Probability Distributions

Random Variables: Concept of Random Variables, Univariate random variable, Probability mass and density function of the random variables, Mathematical Expectations.

Discrete Probability Distributions: Discrete Probability Distributions, Binomial Distribution, Poisson distribution.

Continuous Probability Distributions: Continuous Probability Distributions, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial.

UNIT III – Fundamental Sampling Distributions and Estimation

Fundamental Sampling Distributions: Random Sampling, Sampling Distributions, Sampling Distribution of Mean, variance (Chi-square, t, F) (without proof), Central Limit Theorem.

Estimation: Introduction, Statistical Inference, Classical Methods of Estimation, Point Estimate, Prediction Intervals.

UNIT IV – Testing of Hypothesis, Simple Linear Regression, and Correlation

Testing of Hypothesis: Null hypothesis, Alternate hypothesis, Type I & Type II errors, Critical region, Level of significance, Power of the test, One-tailed and Two-tailed Tests, Large sample Tests: Test for a Single proportion, Single mean, Difference between two Proportions, Difference between two Means.

Simple Linear Regression and Correlation: Introduction to Linear Regression, The Simple Linear Regression Model, Inferences Concerning the Regression Coefficients, Simple Linear Regression Case Study, Correlation coefficient, Rank correlation coefficient.

UNIT V - Stochastic Processes and Markov Chains

Introduction to Stochastic processes-Markov process. Transition Probability, Transition Probability Matrix, First order and Higher-order Markov Process, n-step transition probabilities, Markov chain, Steady state condition, Markov analysis.

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

CO 1 : Apply the Number Theory concepts to the cryptography domain
CO 2 : Compute probabilities using theorems in probability and probability distributions
CO 3 : Find estimates of parameters and test hypothesis about parameters
CO 4 : Establish a relationship between variables using correlation and regression
CO 5 : Take decisions using testing of hypothesis techniques & Analyze live data
CO 6 : Resolve the potential misconceptions and hazards in each topic of study

Textbooks:

1. Elementary number theory & its applications, Kenneth H. Rosen, 6th edition, Addison-Wesley, 2011.
2. Probability & Statistics for Engineers & Scientists, Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, 9th edition Pearson Publishers, 2011.
3. Operations Research, S.D. Sharma, Kedarnath and Ramnath Publishers, Meerut, Delhi, 2002.

References:

1. Fundamentals of Mathematical Statistics, S C Gupta and VK Kapoor, Sultan Chand & Sons, 2014.
2. Fundamentals of Probability and Statistics for Engineers, T.T. Soong, John Wiley & Sons Ltd, 2004.
3. Probability and Statistics for Engineers and Scientists, Sheldon M Ross, 5th edition, Academic Press, 2014.

Course Code: 22CS203/253

DATABASE MANAGEMENT SYSTEMS

(Common to CSE, CSE-AI&ML, CSE-CS, CSE-DS, AI&DS, AIM and CS&BS)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To introduce the role of database management systems in an organization.
2 : To represent real-world scenarios using E-R diagrams.
3 : To model the database using relations avoiding redundancies.
4 : To learn transaction management and concurrency protocols to ensure data consistency.
5 : To understand the database file organization system and database recovery techniques.

Unit I - Introduction to DBMS

History of DBMS, Concepts, and overview of DBMS, Data models - ERmodel, Relational model, Levels of Abstraction in DBMS, Database Languages, Architecture of DBMS, Data Base Users and Administrators.

ER-Model

Database design and ER model, ER modeling Constructs, Additional features of ER Model, Class Hierarchies, Aggregation, Conceptual Design with ER model, Case study: ER design for Large Enterprises.

Unit II - Relational Algebra and Calculus

Introduction to the relational model, Logical Database Design- ER to Relational, Relational Algebra - Selection and Projection, Set operations, Renaming, joins, Examples of Relational Algebra Relational Calculus- Tuple relational Calculus, Domain relational calculus.

Introduction to Structured Query Language

Form of Basic SQL Query, Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set Comparison Operator-Aggregate Operators, NULL values and Comparison using Null values, Logical connectivity's – AND, OR and NOT, OUTER Joins, Disallowing NULL Values.

Unit III - PL/SQL

Data Types, Declaration of Variables, Strings, Control Conditional Statements, Functions, Procedures, Cursors, and Triggers.

Schema Refinement

Introduction to schema refinement, Problems caused by decomposition, Functional dependencies (FDs) and reasoning about FDs, Normal Forms (NF) – 1NF, 2NF, 3NF and BCNF, Properties of Decomposition, Schema Refinement in Data Base Design, Case studies using Normal Forms

Unit IV - Transaction Management

Transaction concept & state, Implementation of atomicity and durability, Concurrent executions of a transaction, Serializability and Recoverability, Implementation of Isolation, Testing for serializability, Lock-Based Protocols, Graph-Based Protocol, Timestamp-Based Protocols, Validation-Based, Protocols, Multiple Granularity.

Unit V – Database File Organization and Recovery

Data Base File Organization

Data on External storage, File Organization and Indexing, Cluster Indexes, Primary, and secondary indexes, Index data structures, Hash-based indexing - Static hashing and Extensible Hashing, Tree based indexing - Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index structure.

Database Recovery
Recovery and Atomicity, Log-based Recovery, and Recovery with the concurrent transaction.

Course Outcomes:

At the end of the course, the students should be able to

CO 1 : Demonstrate an understanding of database management system components and features. Design E-R Model torepresent real-world database application scenarios.
CO 2 : Demonstrate a mathematical approach towards querying a database using relational algebra and relational calculus and implement using SQL.
CO 3 : Convert E-R Model to a relational Model and design a proper relational database while eliminating anomalies.
CO 4 : Demonstrate the role of transaction management and concurrency control protocols.
CO 5 : Demonstrate an understanding of database file organization and recovery of the database in case of crashes.

Textbooks:

1. Database System Concepts, A.Silberschatz, H.F. Korth,S.Sudarshan, 6th edition, McGraw-Hill, 2006.
2. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, TMH, 2003.

References:

1. Fundamentals of Database Systems, Ramez Elmasri, Shamkant B.Navathe, 7th edition, Pearson Education, 2008.
2. Database Systems: The Complete Book by Hector Garcia- Molina, Jeffery D.Ullman, Jennifer Widom, 2nd Edition, Pearson Education, 2008.
3. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, 2nd edition, PHI, 2013.

Course Code: 22DT201

DIGITAL ELECTRONICS AND DESIGN

(Common to CSE-AI&ML, CSE-CS, CSE-DS, AI&DS and AIM)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To impart basic properties of Various Digital Number systems, Logic Gates, Boolean algebra and to simplify Boolean functions and the circuits Implementations.

2. To make the students to understand combination logic and synchronous and asynchronous logic circuits and Basic Digital storage used in Digital Computers.

Unit I - Boolean Algebra and Logic Gates:

Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal umbers, complements, signed binary numbers, Binary codes – ASCII, Binary coded Decimal, Excess-3 Code and Gray Code.

Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, other logic operations, Digital logic gates, Universal Building Blocks.

Unit II- Gate – Level Minimization

The Karnaugh map method, Two- variable map, Three – Variable map, Four-variable map, Don't-care conditions, Sum of Products simplification, Product of Sums simplification, Basic Gates Implementations, NAND and NOR implementation, other Two-level implementations, Exclusive – Or function & -NOR function.

Unit III - Combinational Logic

Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor, Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, De-multiplexers.

Unit IV - Sequential Logic

Sequential circuits, latches, Binary Storage Devices, Flip-Flops (RS, D, JK & T) clocked sequential circuits, Edge (Positive & Negative) Triggered circuits, Registers, shift Registers, Counters - Ripple counters, synchronous counters, other counters.

Unit V - Memories and Asynchronous Sequential Logic

Introduction, Volatile & Non-Volatile Memories, Random-Access Memory, Memory Decoding, Error Detection and correction Circuits, Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.

Course Outcomes*:*

At the end of the course, the student should be able to

CO1: Master different number systems and realize the binary operations of Boolean algebra using logic gates.
CO2: Simplification of Complex problems of gate-level circuits using K-Map and simple circuit implementations.
CO3: Student has the knowledge about the design and functionalities of Combinational circuits.
CO4: Design of Synchronous Sequential circuits and itsworking principles.
CO5: Explain the concepts of basic memory and programmablearray logics used in a digital system.

Textbooks:

1. Digital Design, 3rd edition, M. Morris Mano, Pearson Education/PHI.
2. Digital Principles and Applications Albert Paul Malvino Donald P. Leach Tata McGraw-Hill Edition.
3. Fundamentals of Logic Design, Roth, 5th edition, Thomson.

References:

1. Switching and Finite Automata Theory, Zvi. Kohavi, Tata McGraw-Hill.
2. Switching and Logic Design, C.V.S. Rao, Pearson Education
3. Digital Principles and Design – Donald D.Givone, Tata McGraw-Hill, Edition.
4. Fundamentals of Digital Logic and Microcomputer Design, 5th edition, M. Rafiquzzaman John Wiley.

Course Code: 22CS231

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

(Common to CSE, CSE -AI&ML, CSE-CS, CSE-DS, AI&DS, AIM and IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To set up the necessary environment for running java applications.
2. To implement the basic concepts of object-oriented programming.
3. To implement the practical aspects of exception handling, multithreading mechanisms and Java I/O.
4. To be able to design and implement applications using GUI components.

Lab Problems:

1. a. Write a program to implement the different types of operators, to perform the following tasks: comparison of values, simple arithmetic, bit–wise operations.
   b. Write a program to check and print the grade of a student when the score is given as an integer. Use a switch statement. Rewrite the program to use a sequence of if-else statements.
   c. Write a program to demonstrate the command-line arguments.
2. a. Write a program to demonstrate the task of overloading of constructors and methods.
   b. Write a program to understand the concept of type casting.
3. a. Use an array of integers and find the sum and average of the elements of that array.
   b. Practice further programs on the usage of arrays.
4. a. Write a program to utilize both standard and custom packages. The program should reflect the usage of packages in a correct manner, along with the purpose of access modifiers.
   b. Write a program to use gc() method of both System and Runtime classes. Experiment with other methods of those classes.
5. a. write a program using the hierarchy of employees in a university.
   b. Write a program to understand polymorphic invocation of methods, while overriding the methods. Use an employee base class and manager sub class; override the computeSalary() method to illustrate the concept.
   c. Develop an application that uses inheritance. Use the class Account and then subclass it into different account types. Then making use of Customer and Employee classes to develop the application to reflect the nature of banking operations. Use minimum operational sequence.
6. a. Demonstrate the use of abstract classes. Write a Person abstract class and then subclass that into Student and Faculty classes. Use appropriate fields and methods.
   b. Write a program to demonstrate the usage of interfaces.
7. a. Write a program to understand the full capability of String class. Implement as many methods as required. Consult API documentation to read through the methods.
   b. Write programs using StringBuffer and StringBuilder library classes.
8. a. Write a program to demonstrate the usage of try and associated keywords. Introduce bugs into the program to raise exceptions and then catch and process them.
   b. Learn how to create and use custom exceptions.
   c. Experiment on using various methods of Throwable, Exception classes and Practice on chaining the exceptions.
9. a. Using byte streams, write a program to both read from and write to files.
   b. Using FileReader and FileWriter, write a program to perform file copying and any other suitable operations.
   c. Write a Java Program that displays the number of characters, lines and words in a text file.
10. a. Use the classes StringTokenizer, StringReader and StringWriterto write a program to find the capabilities of these classes.
    b. Write a program to demonstrate enumerations and usage of Assertions.
    c. Demonstrate assertions through simple programs.
11. a. Write programs to illustrate the use of Thread class and Runnable interface.

b. Write a program to show the assignment of thread priorities.
c. Write a program to synchronize threads. Use Producer and Consumer problem to illustrate the concept.

12. a. Create simple advanced calculator, which checks whether a number is prime, calculates the sum of 'N' prime numbers, checks whether a number is even, and calculates the sum of 'N' even and odd numbers using modules.
    b. Write a java program to perform the operations: sort and search on an array of integers and define the following: i. Simple Junit testcases ii. Multiple testcases iii. Suite test
13. a. Write a program to design a frame and control its various display properties.
    b. Write a program to understand the Keyboard and Mouse Events using adapter classes.
14. a. Write a program to demonstrate any layout manager. Use a suitable application.
    b. Write a GUI based application to demonstrate the usage of various javax.swing components and the corresponding event handling techniques.

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

CO 1 **:** Implement object-oriented concepts like encapsulation, data hiding, and abstraction using programming constructs offered by java language.
CO 2 **:** Develop java programs to realize the power of inheritance, interfaces, and packages.
CO 3 **:** Develop java programs to demonstrate the concepts of exception handling and I/O streams.
CO 4 **:** Implement java applications using a multithreading mechanism and understand the power of modules.
CO 5 **:** Use graphical user interfaces to create Frames for providing solutions to real-world problems.

References:

1. Java: The Complete Reference, Herbert Schildt, 10th edition, McGraw-Hill Education, Oracle Press, 2017.
2. Head First Java, Kathy Sierra and Bert Bates, 2nd edition, O'Reilly Media, 2005.
3. https://junit.org/junit5/docs/current/user-guide/#running-tests-junit-platform-runner.

Course Code: 22CS232/282

DATABASE MANAGEMENT SYSTEMS LAB

(Common to CSE, CSE -AI&ML, CSE-DS, CSE-CS, AI&DS, AIM and CS&BS)

Instruction	:	2 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	--	Semester End Examination	:	60 Marks
Credits	:	1.0	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand the relational model.
2. Analyze database requirements and determine the entities involved in the system and their relationship to each other.
3. Understand logical design of the database modeling concepts such as E-R diagrams.
4. Demonstrat SQL DML/DDL commands to insert and manipulate the database.
5. Understand procedures, functions and triggers in PL/SQL.

Database Description: This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example - **Boat reservation by the sailor** and -**employee data maintenance in an organization** whose description is as given below. The student is expected to practice the designing, developing and querying a database in the context of reserving a boat and employee data maintenance. Students are expected to use - MySq database.

"Boat reservation by the sailor" is a schema with several boats which could be reserved depending on color and availability on a particular day. The sailor reserves the boat on a particular day y registering himself with a rating. The sailor is identified by sailor id, boats are identified by boat id and reservation is uniquely identified by sailor id, boat id and day.

"Employee data maintenance in an organization": In any organization, we need to maintain the data of employees categorized into department as per the salary. The scheme contains employee, department and sal grade tables which are identified by employee id,department id and range of salary respectively.

1. E-R Model

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Boat reservation by the sailor:

Entities:

1. SAILORS
2. BOATS
3. RESERVES

PRIMARY KEY ATTRIBUTES:

1. SID (SAILOR ENTITY)
2. BID (BOATS Entity)
3. SID,BID,DAY (RESERVES ENTITY)

Employee data maintenance in an organization Entities:

1. EMPLOYEE
2. DEPT
3. SALGRADE

PRIMARY KEY ATTRIBUTES:

1. EID (EMPLOYEE ENTITY)
2. DID (DEPT Entity)
3. LOWSAL AND HIGHSAL (SALGRADE ENTITY)

2. Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc. wherever required for

1) Boat reservation by the sailor

2) Employee data maintenance in an organization

3. Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality.Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi valued, and Derived) have different way of representation.

SAILORS

SID	SNAME	RATING	AGE

EMPLOYEE

EID	ENAME	DID	SAL	DESIGNATION	MGRNUM	DOJ	AGE

4. Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical orstructural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

Perform do the second and third normal forms for sailors and Employee databases if required.

5. Installation of Mysql and practicing DDL commands

Installation of MySql. In this week student will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. Students will also try truncate, rename commands etc.

6. Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

1) SELECT - retrieve data from the a database
2) INSERT - insert data into table
3) UPDATE - updates existing data within a table
4) DELETE - deletes all records from a table, the space for the records remain

7. Querying - I

In this week students are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

8. Querying - II

Students are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

9. Triggers

In this week students are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

10. Procedures

In this session students will learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

11. Cursors

In this week students will learn to declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done.

Course Outcomes:

At the end of the course, student should be able to:

CO1 : Analyze database requirements and determine the entities involved in the system and their relationship to each other.
CO2 : Design E-R Model to represent database application scenarios.
CO3 : Convert/transform the E-R Model to relational tables, populate relational database and formulate SQL queries on data.
CO4 : Improve the database design by normalization.
CO5 : Implement PL/SQL procedures, function, triggers and cursors.

References:

1. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, TMH, 2003.
2. Introduction to SQL, Rick F.VanderLans, 4th edition, Pearson education, 2007.
3. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, 2nd edition, Pearson education, 2002.

Course Code: 22DT231

DATA VISUALIZATION LAB

(Common to CSE, CSE-CS, CSE-DS, AI&DS & IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. Understand the various types of data, apply and evaluate the principles of data visualization.
2. Acquire skills to apply visualization techniques to a problem and its associated dataset.

List of Experiments:

1. Understanding Data types & creating respective charts at Univariate, Bivariate and Multivariate.
2. Creating dashboards for effective data visualization.
3. Acquiring and plotting data.
4. Statistical Analysis – such as Multivariate Analysis, PCA, LDA, Correlation regression and analysis of variance.
5. Financial analysis using Clustering, Histogram and HeatMap.
6. Time-series analysis – stock market.
7. Visualization of various massive dataset - Finance - Healthcare - Census – Geospatial.
8. Visualization on streaming dataset (Stock market dataset, weather forecasting).
9. Market-Basket Data analysis-visualization.
10. Text visualization using web analytics

Course Outcomes:

CO 1: Identify the different data types, visualization types to bring out the insight.
CO 2: Relate the visualization towards the problem based on the dataset to analyze and bring outvaluable insight on a large dataset.
CO 3: Demonstrate the analysis of a large dataset using various visualization techniques and tools.
CO 4: Identify the different attributes and showcasing them in plots. Identify and create various visualizations for geospatial and table data.
CO 5: Ability to create and interpret plots using R/Python.

References:

1. Robert Spence "Information visualization – Design for interaction", Pearson Education, 2nd edition, 2007.
2. Alexandru C. Telea, "Data Visualization: Principles and Practice," A. K. Peters Ltd, 2008.

Course Code: 22HS201/251

CONSTITUTION OF INDIA

(Mandatory Course) (Common to all Branches except CS&BS)

Instruction	:	3 Periods/week	Sessional Marks	:	100
Credits	:	0			

Course Objectives: Students will be able to

1. : Understand the history and making of the Indian Constitution.
2. : Recognize the Philosophy of the Indian Constitution and Preamble
3. : Identify the importance of fundamental rights as well as fundamental duties.
4. : Understand the functioning of organs of governance and local administration
5. : Learn composition and activities of Election Commission and institutional bodies.

Unit I

History of Making of the Indian Constitution: The meaning of constitutional Government, the roots of the constituent Assembly of India, Composition of the proposed constituent Ass embly. History of Drafting Committee.

Unit II

Philosophy of the Indian Constitution: Salient features of Indian Constitution, Preamble of the Constitution. Contours of Constitutional Rights & Duties - Fundamental Rights-Right to Equality, Right to Freedom, Right against Exploitation, Right to Freedom of Religion, Cultural and Educational Rights, Right to Constitutional Remedies, Directive Principles of State Policy and Fundamental Duties.

Unit III

Organs of Governance: Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions.

Unit IV

Local Administration: District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation. Panchayat raj: Introduction, PRI: Zila Panchayat. Elected officials and their roles, CEO Zila Panchayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy.

Unit V

Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners. State Election Commission: Role and Functioning. Institute and Bodies for the welfare of SC/ST/OBC and women.

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

CO 1 : Understand and explain the significance of Indian Constitution as the fundamental law of the land.

CO 2 : Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India. Exercise his fundamental rights in proper sense at the same time identifies his responsibilities in national building.

CO 3 : Analyze the organs of governance and District's Administration head
CO 4 : Analyse the Local Administration: District's and Village Administration
CO 5 : Understand Election Commission Process and Institutional Bodies for the welfare of SC/ST/OBC and women.

Textbooks:

1. The Constitution of India, 1950 (Bare Act), Government Publication, 2015.
2. Dr. S. N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st edition, 2015.
3. M. P. Jain, Indian Constitution Law, 7th edition, Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

Course Code: 22CS252

ADVANCED DATA STRUCTURES THROUGH JAVA

(Common to CSE, CSE -AI&ML, CSE-CS, CSE-DS, AI&DS, AIM and IT)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	--	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand the importance of generic programming, Java's collection framework and functional programming.
2. To implement various basic data structures like stacks, queues, linked lists etc. using user defined generic classes and java's collection classes.
3. To learn various data structures for implementing dictionaries.

Unit I - Generics and Functional Programming

Generics: Introduction to Generics, simple Generics examples, Generic Types, Generic methods, Bounded Type Parameters and Wild cards, Inheritance & Sub Types, Generic super class and sub class, Type Inference, Restrictions on Generics.

Functional Programming: Functional Interfaces – Function, BiFunction, Predicate, and Supplier, Lambda Expression Fundamentals, Block Lambda Expressions, Passing Lambda Expressions as Arguments, Lambda Expressions and Exceptions, Variable Capture, Method References.

Unit II - 1D and 2D Collections & Stream API

1D Collection: 1D Collection Interfaces: Collection, Set, List, NavigableSet, SortedSet, Queue, Deque. 1D Collection Classes-Hash Set, Linked HashSet, TreeSet, ArrayList, LinkedList.

2D Collection:2D Collection Interfaces-Map, NavigableMap, SortedMap, 2D Collection Classes-HashMap, LinkedHashMap, TreeMap.

Stream API: Stream basics, Stream Interface, Intermediate operations – map(), filter(), distinct(), sorted(), limit(), skip(), Terminal operations – forEach(), reduce(), collect(), min(), max(), count().

Unit III - Dictionaries

Introduction: Dictionary definition, Dictionary ADT.

Dictionaries Implementation-I:

Linear List Representation: Basics of linear list, implementation of sorted list using user defined generic classes and, LinkedList Collections class.

Hashing: basics, closed hashing – linear probing, quadratic probing, double hashing, rehashing, extendible hashing and their implementation, open hashing-separate chaining and its implementation using user defined generic classes.

Binary Search Trees: definition and basics, implementation of operations-searching, non-recursive traversals, insertion and deletion using user defined generic classes.

Unit IV - Dictionaries Implementation-II

AVL Tree: definition, the height of an AVL tree, representation, operations-rotations, insertion, searching, deletion and, their implementation using Java's Collection framework.

Red-Black Binary search trees: definition, insertion, deletion, and search operations.

Unit V - B-Trees and Priority Queues

B-Tree: B-Tree of order m, the height of a B-Tree, searching, insertion, and deletion operations.

Priority Queue: definition, max and min heaps, realizing priority queues using heaps, operations-insertion, deletion, and their implementation using user-defined generic classes, heap sort and its implementation using user-defined generic classes.

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

CO 1 **:** Realize the power of generics and functional programming in java.
CO 2 **:** Understand Java's Collection class hierarchy and also know the power of data processing using streams.
CO 3 **:** Implement dictionaries using linear lists, hashing & binary search tree and compare their performances.
CO 4 **:** Implement dictionaries using an AVL tree and red, black tree.
CO 5 **:** Understand the advantages of B-trees and Priority Queues.

Textbooks:

1. Java: The Complete Reference, Herbert Schildt, 10th edition, McGraw-Hill Education, Oracle Press, 2017.
2. Data Structures and Problem-Solving Using Java, Mark A. Weiss, 4th edition, Pearson Education, 2009.

References:

1. Data Structures and Algorithms in Java, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 6th edition, Wiley Publications, 2014.
2. Data Structures, Algorithms, And Applications in Java, Sartaj Sahni, 2nd edition, Universities Press, 2005.
3. Data Structures: Abstraction and Design Using Java, Elliot B. Koffman, Paul A. T.Wolfgang. 2nd second Edition, Wiley publications, January 2010.
4. Head First Java, Kathy Sierra and Bert Bates, 2nd edition, OREILLY publications, 2005.

Course Code: 22DT251

DESIGN AND ANALYSIS OF ALGORITHMS

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

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To emphasize upon the demands of real-world problems in engineering solutions
2. To make students conversant with the various paradigms of algorithms
3. To handcraft the performance analysis of designed solutions
4. To take students through various optimization principles of ill-posed problems

Unit I – Fundamentals of algorithm analysis

Introduction- Definition of algorithm, algorithmic problem solving, pseudo code for expressing algorithms. Asymptotic notations- o, Ω, and θ notations. Performance analysis: Time and space complexity: count, tabular methods, examples on non-recursive, recursive algorithms. Recursive algorithms and recurrence relations - ToH problem, Amortized analysis.

UNIT II - **Algorithm paradigms: Divide and Conquer**

Control abstraction, binary search algorithm and its complexity, Merge sort, its complexity, quick sort, its complexity. Graph traversals: Depth first search (dfs), breadth first search (bfs), articulation points, bi-connected components.

UNIT III - **Greedy paradigm**

Control abstraction, fractional knapsack problem, job sequencing problem, minimum cost spanning tree: Prims algorithm, Kruskal's algorithm, Djkstra's algorithm.

UNIT IV - Dynamic programming and Backtracking

Dynamic programming - Control abstraction, Multistage Graphs, OBST, Travelling salesperson problem, reliability design, 0/1 knapsack problem, All pair shortest path algorithm.

Backtracking - n-queens problem, Graph coloring, Sum of subsets problem.

UNIT V - **Branch and Bound, Complexity Theory**

Branch and Bound: General method, Applications: Travelling sales person problem, 0/1 knapsack problem, LC branch and bound solution, FIFO branch and bound solution, Game trees.

Np-hard and NP-complete problems: basic concepts, non-deterministic algorithms, NP-hard and NP-complete classes, Cook's theorem.

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

CO 1	:	Analyze worst-case running times using asymptotic analysis of algorithms.
CO 2	:	Describe the divide-and-conquer paradigm and Synthesize divide-and-conquer algorithms.
CO 3	:	Define optimization problems and solve them through various greedy policies
CO 4	:	Describe the dynamic-programming paradigm and synthesize dynamic-programming algorithms and analyze them.
CO 5	:	Reduce the size of search space of the optimization problems by applying backtracking and branch and bound tools. Appreciate the Non-Deterministic modeling of algorithms.

Textbooks:

1. E. Horowitz and S.Sahni, *Fundamentals of algorithms*, 2nd edition, Galgotia Publications, 2010
2. T.H.Cormen, C.E.Leiserson, R.L.Rivest, and C.Stein, *Introduction to algorithms*, 2nd edition, PHI/Pearson Education, 2001.

References:

1. R C T Lee, Hang and TT Sai, Introduction to Design and Analysis of Algorithms, A strategic approach, TMH.
2. Allen Weiss, Data structures and Algorithm Analysis in C++, 2nd edition, Pearson Education
3. Aho, Ullman and Hopcroft, Design and Analysis of Algorithms, Pearson Education, 1974.
4. Richard Johnson Baugh, and Marcus Schaefer, Algorithms, Pearson Education.

Course Code: 22CY251

COMPUTER ORGANIZATION AND ARCHITECTURE

(Common to CSE-AI&ML, CSE-CS, CSE-DS, AI&DS and AIM)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Pre-requisite: A Course on "Digital Electronics and Design".

Course Objectives:

1. The purpose of the course is to introduce principles of computer organization and the basic architectural concepts, Basic levels of pipelining and vector processing Architecture.
2. It begins with basic organization, design, and programmingof a simple digital computer and introduces simple register transfer language to specify various computer operations. Control Unit, Hardwired & Micro programmed Control, CPU organization, instruction set design
3. Topics include Various Data Formats, computer arithmetic & Algorithms, memory organization and I/O systems, and Study about multiprocessors Architecture.

Unit I

Digital Computers: Introduction, Block diagram of Digital Computer, Types of Computers, Definition of Computer Organization,Computer Design and Computer Architecture, Functional Units-ALU, CPU, CU, Program, Sub-Program, Subroutine Call, Interrupts, Interrupt Service Routines.

Introduction to Pipeline and Vector Processing: Instruction Cycle, Instruction Pipeline, Arithmetic Pipeline, Parallel Processing,Vector Processing, Array Processor.

Unit II

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, Memory Reference Instructions, Input – Output and Interrupt.

Unit III

Control Unit: Hardwired Control, 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, STACK organization, Program Control, RISC, CISC.

Unit IV

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

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

Unit V

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

Input-Output Organization & Multiprocessors: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupts, Direct memory Access. Multiprocessors, Characteristics of Multiprocessors, Interconnection Structure, Inter-processor arbitration, Inter-processor communication.

Course Outcomes*:*

At the end of the course, the student should be able to

CO 1: Fundamentals of Digital Computer Organization and Architecture, its basic concepts and functions, pipeline and parallel computing.
CO 2: Understand the deep operations of Registers & Memory withnotation language using RTL and Micro operations. Instruction formats with respect to computer system.
CO 3: The execution of Micro program from control memory and how the hardwired control is designed thro combinational logic and detailed CPU organization, Basics of RISC & CISC.
CO 4: Data representations as per IEEE standards, various arithmetic operations and various algorithms for arithmetic operations multiplication and division.
CO 5: Importance of memory, different types of memory and its operations, I-O operations and Multiprocessor Architecture.

Textbooks:

1. Computer System Architecture – M. Morris Mano, 3rd edition, Pearson/PHI.
2. Computer Organization and Architecture – William Stallings 6th edition, Pearson/PHI.

References:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th edition, McGraw-Hill.
2. Structured Computer Organization – Andrew S. Tanenbaum, 4th edition, PHI/Pearson.
3. Computer Organization and Architecture, V.Rajaraman, T.Radhakishnan, PHI Learning
4. Computer Organization and Design, Pal Choudary, PHI.

Course Code: 22CY252

COMPUTER NETWORKS

(For CSE-CS Only)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. How computer network hardware and software operate
2. Investigate the fundamental issues driving network design
3. Perform and understand methods for reliable data transmission
4. The data transmission through protocols across the network in wired and wireless using routing algorithms.

Unit –I

Introduction: Networks, Network Topologies, Types of Networks, Protocol Layering and Principles.

Network Models: TCP/IP, ISO-OSI Reference Model, Comparison of the OSI Reference Models and TCP/IP Reference Models.

Physical Layer: Transmission Media: Guided media and unguided media.

Unit –II

Data Link Layer: Data Link Layer Design Issues, Error detection and correction, Elementary data link protocols, sliding window protocols,

Multiple access protocols: CSMA protocols, Link Layer Addressing
Data link layer switching: Repeaters, hubs, bridges, switches, routers, and gateways.

Unit –III

Network Layer: Network layer design issues, **Routing algorithms:** Shortest path, flooding, distance vector, link state, hierarchical, broadcast, multicast, Congestion control algorithms**.**

Internet working: Tunneling, Fragmentation.

The Network Layer in the Internet: IP Addresses, Internet control protocols, OSPF, BGP, IPv4, and IPv6.

Unit–IV

Transport Layer: The transport service: Service provided to the upper layer, Transport service primitives, Elements of Transport Protocols

The Internet Transport Protocols: UDP - Remote Procedure Call, Real time transport protocol, TCP - segment header, connection establishment, connection release, connection management, Transmission policy, congestion control.

Unit-V

Application Layer: DNS (Domain Naming System), E-mail, FTP Protocol, SNMP Protocol, Sockets.

Course Outcomes:

After completion of the course, students would be able to:

CO 1: Outline the basic concepts of data communications including the key aspects of networking and their interrelationship, packet and circuit switching as internal and external operations, physical structures, types, models, and internetworking
CO 2: Make use of different types of bit errors and the concept of bit redundancy for error detection and error correction.
CO 3: Identify the suitable design parameters and algorithms for assuring quality of service and internetworking in various internet protocols
CO 4: Interpret transport protocols (TCP, UDP) services and functionalities for measuring the network performance
CO 5: Illustrate the various application layer protocols (FTP, POP, and SMNP), E-Mail, WWW and standards (DNS) in data communications among network.

Textbooks:

1. Andrew S. Tanenbaum, David. J. Wetherall, "Computer Networks", Prentice-Hall, 4th edition, 2010.
2. Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw-Hill, 5th edition, 2012.

References:

1. S. Keshav, "An Engineering Approach to Computer Networks", 2nd edition, Pearson Education.
2. Youlu Zheng, Shakil Akthar, "Networks for Computer Scientists and Engineers", Oxford Publishers, 2016.
3. L.L Peterson and B.S. Davie, "Computer Networks", 4th edition, ELSEVIER.
4. Nader. F. Mir, "Computer and Communication Networks", Pearson Prentice Hall Publishers, 2010.

Course Code: 22IT252

OPERATING SYSTEMS

(Common to CSE, CSE -AI&ML, CSE-CS, CSE-DS, CS&BS and IT)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites:

1. A course on "Computer Programming and Data Structures".

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, inter-process communication and I/O in Unix.

Unit I – Operating System Introduction

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, 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. Process related system calls – fork, exit, wait and exec.

Unit II- CPU Scheduling, Process Management and Synchronization

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling.

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware and Software, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors.

Unit III- Interprocess Communication Mechanisms and Deadlocks

Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

Unit IV- Memory Management and Virtual Memory

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

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

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

CO1 : Understand the role of Operating System with its function and services.
CO2 : Compare various algorithms used for CPU scheduling and apply various concepts related to concurrency and synchronization to solve problems.
CO3 : Understand the inter process communication mechanism and resolve deadlock in a multi-programmed environment.
CO4 : Understand the concepts of virtual memory and how it is realized in systems
CO5 : Differentiate and Demonstrate file systems, directory structures and their implementation issues.

Textbooks:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th edition, John Wiley, 2017.
2. Advanced programming in the UNIX environment, W.R. Stevens, 3rd edition, Pearson education, 2013.

References:

1. Operating Systems- Internals and Design Principles, William Stallings, 5th edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Croley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education

Course Code: 22CS281

ADVANCED DATA STRUCTURES THROUGH JAVA LAB

(Common to CSE, CSE -AI&ML, CSE-CS, CSE-DS, AI&DS, AIM and IT)

Instruction	:	2 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To implement generic programming and Java's collection framework.
2. To apply Java's collection framework for implementing basic data structures like stacks, queues, linked lists, etc.
3. To understand the concepts of functional programming, lambda expressions and streams.
4. To implement dictionaries using advanced data structures like Binary search trees, and AVL trees.

Lab problems:

1. Write a java program to demonstrate the use of bounded type parameters and wild card arguments.
2. Write a java program that returns the value of pi using the lambda expression.
3. Write a java program that takes a string as parameter and calculates the reverse of the string using lambda expression.
4. Write a java program to implement iterators on Array List and LinkedList.
5. a) Implement a Generic stack to deal with Integer, Double and String data using user-defined arrays and linked lists.
   b) Implement a Generic queue to deal with Integer, Double and String data user-defined arrays and linked lists.
6. a) Write a Java program to implement Generic stack using Array List Collection class.
   b) Write a Java program to implement Generic stack using LinkedList Collection class.
7. a) Write a Java program to implement Generic queue using ArrayList Collection class.
   b) Write a Java program to implement Generic queue using LinkedList Collection class.
8. Write a Java program to demonstrate the use of the following Collection classes.
   a. HashSet b. LinkedHashSet c. TreeSet
9. Write a java program to create a class called Person with income, age, and name as its members. Read set A of persons from a user and compute the following sets:
   i) Set B of persons whose age > 60
   ii) Set C of persons whose income < 10000 and iii) B ∩ C
10. Write a Java program to demonstrate the use of the following Collection classes.
    a. HashMap b. LinkedHashMap c. TreeMap
11. Create a class Product(id, name, price, type, rating) and perform the following operations using stream:
    i) Find all the products having rating between 4 and 5.
    ii) Find first n products having price > 10000.
    iii) Find the number of products under each type(map containing type and count).
    iv) Find average rating of products with type = "Electronics".
12. Write a Java program to implement Sorted Chain.
13. Write a Java program to implement Separate Chaining
14. Write a Java program to implement Linear Probing.
15. Implement BST using Collection API, and use recursive procedures to implement inOrder, preOrder and postOrder traversals.
16. Implement AVL tree using Collection API.
17. Implement priority queues with max Heap tree using Collection API.
18. Implement heap sort with max Heap tree using Collection API.

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

CO 1 **:** Understand the power of generics and functional programming.
CO 2 **:** Implement hashing, sets, stacks and queues using collection classes in java.util package and process the data using streams.
CO 3 **:** Implement dictionaries using various data structures like sorted list, and hashing.
CO 4 **:** Implement dictionaries using various height-balanced trees and also analyze the advantages and disadvantages of height-balanced trees.
CO 5 **:** Understand the importance of Priority Queues and their applications.

References:

1. Java: The Complete Reference, Herbert Schildt, 10th edition, McGraw-Hill Education, Oracle Press, 2017.
2. Data Structures and Problem Solving Using Java, Mark A. Weiss, 4th edition, Pearson Education, 2009.
3. Data Structures and Algorithms in Java, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 6th edition, Wiley Publications, 2014.

Course Code: 22CY281

COMPUTER NETWORKS LAB

(for CSE–CS Only)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

Course Objectives:

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

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 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 Wire shark
    ii. Starting Wireshark
    iii. Packet Capturing for plain and secure communication protocols (http/https)
    iv. Analysis and Statistics & Filters.
11. How to run Nmap scan
12. Operating System Detection using Nmap
13. TCP socket programming using C or LINUX (iterative and concurrent server)

Course Outcomes:

CO 1: Implement data link layer farming methods

CO 2: Analyze error detection and error correction codes.

CO 3: Implement and analyze routing and congestion issues in network design.

CO 4: Implement Encoding and Decoding techniques used in presentation layer

CO 5: To be able to work with different network tools

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.
3. Computer Networks, Andrew S Tanenbaum, David. J. Wetherall, 5th edition, PearsonEducation/PHI.

Course Code: 22IT282

OPERATING SYSTEMS LAB

(Common to CSE-AI&ML, CSE-DS and CSE-CS, IT)

Instruction	:	2 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

Prerequisites: A course on "Programming for Problem Solving", A course on "Computer Organizationand Architecture".

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, interprocess communication and I/O in Unix.

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms
   a) FCFS b) SJF c) RoundRobin d) priority.
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
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 memory management techniques Paging and Segmentation
7. Write C programs to simulate Page replacement policies
   a) FCFS b) LRU c) Optimal

Course Outcomes: At the end of the course, the student will beable to

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

Textbooks:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th edition, John Wiley, 2017.
2. Advanced programming in the Unix environment, W.R.Stevens, 3rd edition, Pearson education, 2013.

References:

1. Operating Systems – Internals and Design Principles, William Stallings, 5th edition–2005,Pearson Education/PHI.
2. Operating System - A Design Approach-Crowley, TMH.
3. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education
4. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education.
5. Design of the UNIX Operating System, Maurice Bach, 1st edition, Pearson Education.

Course Code: 22HS281/231

GENDER SENSITIZATION LAB

(Mandatory Course) (Common to all Branches except CS&BS)

Instruction	:	2 Periods/week	Sessional Marks	:	100
Credits	:	0			

Course Description:

This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines – such as literature, history, economics, psychology, sociology, philosophy, political science, anthropology and media studies – to examine cultural assumptions about sex, gender, and sexuality.

This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women, and of the multiple ways that sex andgender interact with race, class, caste, nationality and other social identities. This course also seeks to build an understanding and initiate and strengthen programmes combating gender-based violence anddiscrimination. The course also features several exercises and reflective activities designed to examinethe concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development.

Course Objectives:

1. : To develop students' sensibility with regard to issues of gender in contemporary India.
2. : To provide a critical perspective on the socialization of men and women
3. : To introduce students to information about some key biological aspects of genders
4. : To expose the students to debates on the politics and economics of work
5. : To help students reflect critically on gender violence and to support a sustainable gender-equal society.

Unit I - Understanding Gender

Introduction: Definition of Gender-Basic Gender Concepts and Terminology-Exploring Attitudes towards Gender-Construction of Gender-Socialization: Making Women, Making Men -Preparing for Womanhood. Growing up Male. First lessons in Caste.

Unit II Gender Roles and Relations

Two or Many? -Struggles with Discrimination-Gender Roles and Relations-Types of Gender Roles-Gender Roles and Relationships Matrix-Missing Women-Sex Selection and Its Consequences-Declining Sex Ratio. Demographic Consequences-Gender Spectrum: Beyond the Binary

Unit III - Gender and Labour

Division and Valuation of Labour-Housework: The Invisible Labor- "My Mother doesn't Work." "Share the Load."-Work: Its Politics and Economics -Fact and Fiction. Unrecognized and Unaccounted work.

-Gender Development Issues-Gender, Governance, and Sustainable Development- Gender and Human Rights-Gender and Mainstreaming

Unit IV - Gender-Based Violence

The Concept of Violence- Types of Gender-based Violence-Gender-based Violence from a Human Rights Perspective-Sexual Harassment: Say No!-Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: *"Chupulu".*

Domestic Violence: Speaking Out: Is Home a Safe Place? -When Women Unite [Film]. Rebuilding Lives.Thinking about Sexual Violence Blaming the Victim-"I Fought for my Life…."

Unit V Gender and Culture

Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular Literature- Gender Development Issues-Gender Sensitive Language-Gender and Popular Literature - Just Relationships: Being Together as Equals

Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Mothers and Fathers. Rosa Parks-The Brave Heart.

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

CO 1 : Students will have developed a better understanding of important issues related to gender in contemporary India.
CO 2 : Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender.
CO 3 : Students will attain a finer grasp of the biological spheres of gender in our society and how to counter it.
CO 4 : Students will acquire insight into the gendered division of labor and its relation to politics and Economics.
CO 5 : Students will develop a sense of appreciation for women in all walks of life and contribute to establish an egalitarian society.

Textbook:

1. *Towards a World of Equals: A Bilingual Textbook on Gender,* A. Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu, Telugu Academy, Telangana Government, 2015.