ACADEMIC REGULATIONS,

COURSE STRUCTURE

&

SYLLABUS

R22 REGULATIONS

CHOICE BASED CREDIT SYSTEM (CBCS)

II B.Tech.

Computer Science and Engineering

Applicable to batches admitted in the First year
from 2022-23 onwards

CVR COLLEGE OF ENGINEERING

An UGC Autonomous Institution with NAAC Grade 'A'

(Approved by AICTE & Govt. of Telangana and
Affiliated to JNT University, Hyderabad)
Vastunagar, Mangalpalli (V), Ibrahimpatan (M),
Ranga Reddy Dist., Pin – 501 510

CVR COLLEGE OF ENGINEERING

VISION

- To be a state of the art institution of engineering in pursuit of excellence, in the service of society.

MISSION

- To excel in providing quality education at undergraduate and graduate levels.
- To encourage research and innovation.
- To provide infrastructure and facilities to meet the latest technological needs.
- To establish Centres of Excellence through active interaction with industry.
- To nurture students towards holistic development with human values and ethics.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

- Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Consultancy, and Technological services to society.

MISSION

1. To produce the best quality Computer Science & Engineering professionals by imparting quality training, hands-on experience and value education.
2. To strengthen links with industry through partnerships and collaborative developmental works.
3. To attain self-sustainability and overall development through Research, Consultancy, and Development activities.
4. To extend technical expertise to other technical institutions of the region and play a lead role in imparting technical education.
5. To inculcate work ethics and commitment in students for their future endeavors to serve society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

PEO 1 : Employability: Computer Science & Engineering graduates will acquire the capability to apply their knowledge and skills to solve various kinds of computational engineering problems.

PEO 2 : Professionalism: Graduates will inculcate a professional attitude, interdisciplinary approach, ethics, and ability to relate computer engineering issues with social awareness.

PEO 3 : Managerial skills: Graduates will possess managerial skills to face challenges in the profession by working harmoniously in a team with effective communication skills.

PEO 4 : Continuous learning: Graduates will continue to learn and adapt in a world of constantly evolving technologies and pursue research towards academic excellence.

PEO 5 : Adaptability: Graduates of Computer Science & Engineering will have soft skills to adapt to the diverse global environment.

PROGRAM OUTCOMES (POs):

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Successfully design and implement algorithmic paradigms by using efficient programming language constructs, formal tools, and frameworks.

PSO2: Develop scalable and reliable distributed applications and data analytics pipelines by employing industry-agnostic technologies and secure software engineering models.

PSO3: Adapt cloud computing ecosystems and machine learning algorithms to develop smart and sustainable solutions complying the ethics of society and eventually emerge as entrepreneurs.

CVR COLLEGE OF ENGINEERING

II B.Tech. Computer Science and Engineering

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	22HS203	Computer Oriented Statistical Methods	BS	3	1	4	40	60	100	3
2	22IT201	Digital Electronics	ES	3	0	3	40	60	100	5
3	22EE205	Basic Electrical & Electronics Engineering	ES	3	0	3	40	60	100	7
4	22CS201	Discrete Mathematics	PC	3	0	3	40	60	100	9
5	22CS202	Object Oriented Programming through Java	PC	3	0	3	40	60	100	11
Practicals										
6	22EE233	Electrical & Electronics Engineering Lab	ES	0	2	1	40	60	100	15
7	22CS231	Object Oriented Programming through Java Lab	PC	0	3	1.5	40	60	100	16
8	22DT231	Data Visualization Lab	PC	0	3	1.5	40	60	100	18
		Total		**15**	**9**	**20**	**320**	**480**	**800**	
		Total Hours		**24**						
9	22HS231/281	Gender Sensitization Lab	MC	0	2	0	100	0	100	22

Service Courses of II B.Tech. 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	22CS201	Discrete Mathematics (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	3	0	3	40	60	100	9
2	22CS202	Object Oriented Programming through Java (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	3	0	3	40	60	100	11
3	22CS203/253	Database Management Systems (CSE-AI&ML, CSE-CS and CSE-DS)	PC	3	0	3	40	60	100	13
Practicals										
1	22CS231	Object Oriented Programming through Java Lab (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	0	3	1.5	40	60	100	16
2	22CS232/282	Database Management Systems Lab (CSE-AI&ML, CSE-CS and CSE-DS)	PC	0	2	1	40	60	100	19

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**

CVR COLLEGE OF ENGINEERING

II B.Tech. Computer Science and Engineering

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	22CS251	Computer Organization	PC	3	0	3	40	60	100	24
2	22CS252	Advanced Data Structures through Java	PC	3	0	3	40	60	100	26
3	22CS253/203	Database Management Systems	PC	3	0	3	40	60	100	13
4	22CS254	Software Engineering	PC	3	0	3	40	60	100	28
5	22IT252	Operating Systems	PC	3	0	3	40	60	100	30
Practicals										
6	22CS281	Advanced Data Structures through Java Lab	PC	0	2	1	40	60	100	32
7	22CS282/232	Database Management Systems Lab	PC	0	2	1	40	60	100	19
8	22CS283	Operating System and Assembly Language Programming Lab	PC	0	2	1	40	60	100	34
9	22CS284	Real-Time/Field-Based Research Project	PC	0	4	2	50	0	50	
		Total		**15**	**10**	**20**	**370**	**480**	**850**	
		Total Hours		**25**						
10	22HS251/201	Constitution of India	MC	3	0	0	100	0	100	36

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

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 (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	3	0	3	40	60	100	26
Practicals										
1	22CS281	Advanced Data Structures through Java Lab (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	0	2	1	40	60	100	32

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:22HS203

COMPUTER ORIENTED STATISTICAL METHODS

(Common to CSE & IT)

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 will be able

1. To introduce the concepts of Probability and statistics.
2. To learn how to apply Probability and Statistics to solve engineering problems.
3. To keep a balance between theory and methodology.
4. To show the applications of Probability and Statistics in engineering with examples.
5. To learn the concepts of stochastic processes and Markov chains.

Unit I – Probability

Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence and the Product Rule, Bayes' Rule.

Random Variables and Probability Distributions: Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions.

Unit II – Expectation and Discrete Distributions

Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and variances of Linear combinations of Random Variables, Chebyshev's Theorem, the definition of bivariate random variables, Concepts and applications of correlation and regression.

Discrete Probability Distributions: Binomial Distribution, Poisson distribution.

Unit III – Continuous Probability Distributions and Sampling Distributions

Continuous Probability Distributions: Uniform Distribution, Normal Distribution, Areas under the Normal curve, Applications of the Normal Distribution, Normal approximation to Binomial Distribution.

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

Unit IV – Sample Estimation & Tests of Hypotheses

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

Statistical Hypotheses: General Concepts, Null Hypothesis, Alternate Hypothesis, Type-I and Type-II errors, critical region, level of Significance, Power of the Test, One-tailed and Two-tailed tests, calculation of p-value.

Single sample: Large sample tests concerning single mean and single proportion. Two samples: Large sample tests concerning two means, two proportions and large sample tests concerning variances.

Unit V – Stochastic Processes and Markov Chains

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

Course Outcomes: At the end of the course, the student acquires the ability to

CO 1 : Compute Probabilities using theorems in probability and probability distributions.
CO 2 : Formulate and solve problems involving random variables and apply statistical methods for analyzing experimental data.
CO 3 : Apply the tools in Probability and Statistics in Engineering.
CO 4 : Apply the concept of estimation of parameters and testing of hypothesis about the parameters to case studies.
CO 5 : Correlate the concepts of one unit to the concepts of other units.

Textbooks:

1. Probability & Statistics for Engineers & Scientists, Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, 9th Edition Pearson Publishers, 2011.
2. Fundamentals of Mathematical Statistics, S C Gupta and V K Kapoor, Khanna Publications, 2002.
3. Operations Research, S.D.Sharma, Kedarnath and Ramnath Publishers, Meerut, Delhi, 2002.

References:

1. Fundamentals of Probability and Statistics for Engineers, James T.T. Soong, John Wiley & Sons, Ltd, 2004.
2. Probability and Statistics for Engineers and Scientists, Sheldon M Ross, 5th edition, Academic Press, 2014.
3. Probability and Statistics for Engineers, Miller and Freund's, 8th Edition, Pearson Educations, 2015.

Course Code: 22IT201

DIGITAL ELECTRONICS

(Common to CSE 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 impart basic properties of Boolean algebra and to simplify Boolean functions.
2 : To design fundamental components of a computer such as multiplexers and registers using combinatorial and sequential circuits.
3 : To impart skill set to translate Boolean functions into modular programmable components.

UNIT - I: Boolean Algebra and Logic Gates

Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic**.**

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, Integrated circuits.

UNIT - II: Gate – Level Minimization

The map method, Four-variable map, Five-Variable map, product of sums simplification, Don't-care conditions, NAND and NOR implementation using Two-level implementations, Exclusive – Or function, Quine-McClusky Method – Row and Colum Dominance.

UNIT - III: Combinational Logic

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

UNIT - IV: 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.

UNIT – V: Memories and Asynchronous Sequential Logic

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

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

CO 1 : Understand and master different number systems and realize the binary operations of Boolean algebra using logic gates.
CO 2 : Solve gate-level minimization problems using K-map and Quine-Mc Cluskey methods.
CO 3 : Analyze a given combinational circuit and design a new optimized circuit for a given specification.
CO 4 : Analyze a given sequential circuit and design an optimal circuit to implement a memory element or a counter.
CO 5 : Realize Programmable logic elements used in the design of processors and embedded systems.

Textbooks:

1. Digital Design – Third Edition, M. Morris Mano, Pearson Education/PHI.
2. Digital Principles and Applications Albert Paul Malvino Donald P. Leach TATA McGraw HillEdition.

References:

1. Fundamentals of Logic Design, Roth, 5th edition, Thomson.
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 JohnWiley.

Course Code: 22EE205

BASIC ELECTRICAL AND ELECTRONICS ENGINEERING

(Common to CSE & 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 : Learn basics of various fundamental laws, analysis of electrical circuits, and study nature of ac quantities.
2 : To study the construction, functioning of different types of electrical machines and their performance.
3 : To study working and applications of various electronic devices.

Unit I - Network analysis

Ohm's law, basic circuit components, power and energy calculations, types of elements, Kirchhoff's laws. Resistive, inductive andcapacitive networks, series and parallel circuits, star delta and delta star transformation. Mesh and Nodal Analysis. Principle of superposition, Simple problems.

Alternating Quantities

Basic definitions: frequency, average values and RMS values of alternating currents and voltage, form factor and peak factor.

Unit II - DC Machines

Construction of dc machines, DC Generator- working and principle of operation, EMF equation, types of DC Generators. DC Motor - working and principle of operation, types of dc motor, torque equation, losses and efficiency calculations. Simple problems.

Unit III - Transformers

Principles of operation, Constructional Details, Ideal Transformer andPractical Transformer, Losses, Transformer Test, Efficiency and Regulation Calculations

3-Phase Induction Machines: Principle of operation of induction motor, Torque, Slip, Slip – torque characteristics – applications.

Unit IV - Diode Applications

Rectifiers – Half wave, Full wave and Bridge rectifiers, introduction to filters, Capacitor filter, Zener Diode characteristics Voltage regulationusing Zener Diode, Varactor Diode.

Unit V - Transistor Biasing and Amplifiers

Need for Biasing, operating point, Bias stability, DC load line, Fixed Bias, Voltage divider Bias, Principal of operation of CE Amplifier.

Components of LT Switchgear:

Switch Fuse Unit (SFU), MCB, Earthing, Types of Batteries.

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

CO1 : Identify basic circuit components and solve basic electrical and electronic problems using different principles.
CO2 : Understand the construction and working of different types of DC machines and calculate the losses and efficiency.
CO3 : Understand the Construction and working principle of AC machines and their applications in real time.
CO4 : Analyze and design different types of diodes and rectifiers.
CO5 : Study different protection devices and basics transistor circuits.

Textbooks:

1. Principles of Electrical Engineering and Electronics byV.K.Mehta, S.Chand & Co, 3rd edition, 2014.
2. Electronics Devices and Circuits, S Salivahanan & N SureshKumar, McGraw-Hill, 4th edition, 2017.

References:

1. Basic Electrical and Electronics Engineering- S. K. Bhattacharya, Pearson Education India, 2nd edition, 2017.
2. Basic Electrical Engineering, D Kothari, I Nagrath, 3rd edition, McGraw-Hill Education, 2009.
3. Basic Electrical & Electronics Engineering, J. B. Gupta, S.K.Kataria & Sons, 2013.
4. Electronics Devices and Circuits, R.L. Boylestand and Louis Nashelsky, 9th edition, Pearson/Prentice Hall, 2006.

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, Pigeonhole principles, 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: 22CS202

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

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

Instruction	:	3Periods / 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: 22CS203/253

DATABASE MANAGEMENT SYSTEMS

(Common to CSE, CSE-AI&ML, CSE-CS and 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 introduce the role of database management systems in an organization.
2. To represent real-world scenarios using E-R diagrams.
3. To modal 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

CO1 : Demonstrate an understanding of database management system components and features. Design E-R Model torepresent real-world database application scenarios.
CO2 : Demonstrate a mathematical approach towards querying a database using relational algebra and relational calculus and implement using SQL.
CO3 : Convert E-R Model to a relational Model and design a proper relational database while eliminating anomalies.
CO4 : Demonstrate the role of transaction management and concurrency control protocols.
CO5 : 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: 22EE233

ELECTRICAL AND ELECTRONICS ENGINEERING LAB

(Common to CSE & 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. : Students analyze various electrical parameters and different theorems.
2. : Students can understand the working and operation of different electrical machines.
3. : Students learn the working and applications of various semiconductor devices.

List of the Experiments:

Part-A

(Any 5 experiments from the below given list)

1. Verification of Superposition principle with Resistive load
2. Verification of voltage division and current division in series & parallel circuits
3. Magnetization characteristics of D. C Shunt generator.
4. Swinburne's Test on DC shunt machine (Predetermination of efficiency of a given DC shunt machine working as motor and generator).
5. O. C & S. C tests on Single-phase transformer. (Predetermination of efficiency and regulation at given power factors and determination of Equivalent circuit)
6. Measurement of RMS and average values of AC quantities.

Part-B

(Any 5 experiments from the below given list)

1. Zener diode characteristics
2. Rectifier without filters (HWR and FWR)
3. Rectifier with filters (HWR and FWR)
4. Bridge rectifier
5. Design of Voltage divider bias Circuit
6. Frequency response of Common Emitter Amplifier

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

CO1 : Understand and verify the superposition principle.

CO2 : Determine currents and voltages in parallel and series circuits.

CO3 : Analyze the performance characteristics of various electrical machines.

CO4 : Design Rectifier circuits.

CO5 : Understand the basic transistor biasing techniques.

Course Code: 22CS231

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

(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	:	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: 22DT231

DATA VISUALIZATION LAB

(Common to CSE, CSE-CS, CSE-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: 22CS282/232

DATABASE MANAGEMENT SYSTEMS LAB

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

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 entitiesinvolved in the system and their relationship to each other.
3. Understand logical design of the database modeling conceptssuch as E-R diagrams.
4. Demonstrated SQL DML/DDL commands to insert andmanipulate the database.
5. Understand procedures, functions and triggers in PL/SQL.

Database Description: This lab enables the students to practice theconcepts learnt in the subject DBMS by developing a database for an example -**Boat reservation by the sailor** land -**employee data maintenance in an organization**l 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 - MySql 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 categorizedinto 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 organizationEntities:

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 waysof 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 relationaldatabase 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 schemaobjects. 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. Creationof insert trigger, delete trigger, update trigger. Practice triggersusing 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. Fetchthe 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, Pearsoneducation, 2007.
3. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, 2nd edition, Pearson education, 2002.

Course Code: 22HS231/281

GENDER SENSITIZATION LAB

(Mandatory Course) (Common to all Branches)

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 questionsabout 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 aboutsex, 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 withEveryday 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.

Course Code: 22CS251

COMPUTER ORGANIZATION

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**".

Course Objectives:

1. The purpose of the course is to introduce principles of computer organization and basic architectural concepts.
2. It begins with the basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
3. Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.

Unit I

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design, and Computer Architecture.

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 II

Microprogrammed Control: Control memory, Address sequencing, microprogram example, design of control unit.

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

UNIT III

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 IV

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.

UNIT V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor arbitration, Inter-processor communication and synchronization, Cache Coherence.

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

CO1 : Understand the basics of instruction sets and their impact on processor design.
CO2 : Demonstrate an understanding of the design of the functional units of a digital computer system.
CO3 : Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory
CO4 : Design a pipeline for consistent execution of instructions with minimum hazards.
CO5 : Recognize and manipulate representations of numbers stored in digital computers.

Textbooks:

1. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI, 2014.

References:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th edition, Mc Graw Hill, 2011.
2. Computer Organization and Architecture – William Stallings 6th edition, Pearson/PHI, 2002.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4th edition, PHI/Pearson, 2013.

Course Code: 22CS252

ADVANCED DATA STRUCTURES THROUGH JAVA

(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 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: 22CS254

SOFTWARE ENGINEERING

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

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 and Agile methodology.

UNIT - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, 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, softwarereliability, the ISO 9000 quality standards.

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

CO 1	:	Understand the need for software engineering and use of different software process models for different types of projects.
CO 2	:	Translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
CO 3	:	Identify and apply appropriate software architectures to carry out high level design of a system and be able to carry out detailed design using different UML diagrams.
CO4	:	Develop a strategic approach to testing and debugging. Will have experience and/or awareness of testing and debugging problems. Be able to apply metrics to assess software quality.
CO5	:	Identify software risks and apply RMMM. Be able to conduct formal technical reviews on artifacts during different stages of software development to improve quality of the artifacts.

Textbooks:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 8th edition, McGraw-Hill International Edition, 2014.
2. Software Engineering- Sommerville, 10th edition, Pearson Education, 2017.

References:

1. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson,Pearson Education, 2005.
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley, 2019.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies, 2004.
4. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education, 1999.

Course Code: 22IT252

OPERATING SYSTEMS

(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

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, weight 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.
2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

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 and IT)

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 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: 22CS283

OPERATING SYSTEM AND ASSEMBLY LANGUAGE PROGRAMMING LAB

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 provide an understanding of the design aspects of operating system concepts through simulation
2. To introduce system call interface for process management, inter-process communication and I/O in Unix
3. To enable students gain hands on experience on Assembly Language Programming on 8086

List of Experiments:

1. Implement shell commands such as cp, ls, chmod, ls -ls using the I/O system calls of UNIX/LINUX operating system.
   (open, read, write, close, fcntl, seek, stat, opendir, readdir)
2. Write C programs to simulate the following CPU Scheduling algorithms:
   a) FCFS b) SJF.
3. Write C programs to simulate the following CPU Scheduling algorithms:
   a) Round Robin b) priority.
4. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
5. Write C programs to illustrate the following IPC mechanisms.
   a) pipes b) FIFOs
6. Write C program to illustrate the Message Queues IPC mechanism.
7. Write C program to illustrate the Shared Memory IPC mechanism.
8. Write a C program to implement the Producer–Consumer problem using semaphores using UNIX/LINUX system calls (pthread library API is optional).
9. Write an ALP in 8086 add, subtract and multiply two 16-bit unsigned numbers.
10. Write an ALP in 8086 to implement ASCII Adjust and decimal adjust instructions.
11. Write an ALP to pack two digits into a Byte.
12. Write an ALP to Count number of 1’s and number of 0’s present in the binary representation of a given number.
13. Implement the following string manipulation functions using appropriate registers.
    a) Copy a string b) Lower to upper case c) Reverse a string d) Palindrome.
14. Write an ALP to Count no of even and odd numbers from the given array of numbers.
15. Write a program to check whether a given number is Positive or Negative number.
16. Write an ALP to sort the given array of numbers.
17. Write C programs to simulate Paging memory management techniques.
18. Write C programs to simulate Segmentation memory management techniques.

Note: Programs 1 to 16 are mandatory and 17, and 18 are optional.

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

CO1 : Write programs using I/O System calls for implementing file operations
CO2 : Simulate and implement operating system concepts such as scheduling, deadlock management, and memory management.
CO3 : Implement and realize the semantics of synchronous and asynchronous Inter - Process communication models.
CO4 : Demonstrate the memory segmentation and implement the programming model of Intel 8086 processor
CO5 : Realise the data and string manipulation instructions

References:

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, 2nd edition, 2015, *Pearson* education.
3. Operating Systems – Internals and Design Principles, William Stallings, 5th edition–2005, Pearson Education/PHI.
4. Advance Microprocessors and Peripherals – A.K.Ray and K.M.Bhurchandani, TMH, 3rd edition, 2013.
5. Microprocessors and Interfacing – D.V.Hall, TMGH, 2nd edition, 2006.

Course Code: 22HS251/201

CONSTITUTION OF INDIA

(Mandatory Course) (Common to all Branches)

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 Assemble of India, Composition of the proposed constituent Assembly. 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 Edn., Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.