

VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN
B. Tech. in COMPUTER SCIENCE AND ENGINEERING
COURSE STRUCTURE – VR25

II B. Tech. – I Semester

S.No	Course Code	Course	L	T	P	Credits
1	25CS301PC	Discrete Mathematics	3	0	0	3
2	25EC302PC	Computer Organization and Architecture	3	0	0	3
3	25CS303PC	Object Oriented Programming through Java	3	0	0	3
4	25IT304PC	Software Engineering	3	0	0	3
5	25CS305PC	Database Management Systems	3	0	0	3
6	25MS306BS	Innovation and Entrepreneurship	2	0	0	2
7	25CS307PC	Object Oriented Programming through Java Lab	0	0	2	1
8	25IT308PC	Software Engineering Lab	0	0	2	1
9	25CS309PC	Database Management Systems Lab	0	0	2	1
10	25ML310SD	Node Js/React JS/ Django	0	0	2	1
11	25VA311BS	Environmental Science	1	0	0	1
		Total	18	0	08	22

II B. Tech. - II Semester

S.No	Course Code	Course	L	T	P	Credits
1	25MA401BS	Computer Oriented Statistical Methods	3	0	0	3
2	25CS402PC	Operating Systems	3	0	0	3
3	25DS403PC	Algorithm Design and Analysis	3	0	0	3
4	25CS404PC	Computer Networks	3	0	0	3
5	25ML405PC	Machine Learning	3	0	0	3
6	25MA406BS	Computational Mathematics Lab	0	0	2	1
7	25CS407PC	Operating Systems Lab	0	0	2	1
8	25CS408PC	Computer Networks Lab	0	0	2	1
9	25ML409PC	Machine Learning Lab	0	0	2	1
10	25DS410SD	Data Visualization- R/ Python/ Power BI	0	0	2	1
		Total	15	0	10	20

Note: L - Theory

T - Tutorial

P - Practical

C - Credits

B.Tech.**L T P C****II Year - I Semester****3 0 0 3****DISCRETE MATHEMATICS**

(Common for CSE, IT, CSE(AI&ML), CSE(DS))

Course Objectives:

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

Course Outcomes: The student will learn

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

UNIT - I: Mathematical logic:

Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT - II: Set theory:

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

UNIT - III: Algebraic Structures:

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

UNIT - IV: Elementary Combinatorics:

Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutation with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Exclusion.

UNIT - V: Graph Theory:

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

Text Books:

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

Reference Books:

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

B.Tech.

L T P C

II Year - I Semester

3 0 0 3

COMPUTER ORGANIZATION AND ARCHITECTURE

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

Course Objectives:

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

Course Outcomes: The student will learn

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

UNIT - I:

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

Digital logic gates. Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation

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

UNIT - II:

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

UNIT - III:

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference

Instructions, Input – Output and Interrupt.

UNIT - IV:

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

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

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

UNIT - V:

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

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

Text Books:

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

Reference Books:

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

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

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

(Common for CSE, IT, CSE(AI&ML), CSE(DS))

Course Objectives:

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

Course Outcomes: The student will learn

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

UNIT - I:

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

UNIT - II:

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

UNIT - III:

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

UNIT - IV:

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

UNIT - V:

Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring JFrame and JComponent, JLabel, ImageIcon, JTextField, JButton, JCheckBox, JRadioButton, JList, JComboBox, Tabbed Panes, Scroll Panes, Trees, and Tables. Menu Basics, Menu related classes - JMenuBar, JMenu, JMenuItem, JCheckBoxMenuItem, JRadioButtonMenuItem, JSeparator. creating a popup menu

Text Books:

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

Reference Books:

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

B.Tech. - CSE

L T P C

II Year - I Semester

3 0 0 3

SOFTWARE ENGINEERING
(Common for CSE, CSE(AI&ML), CSE(DS))**Course Objectives:**

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

Course Outcomes: The student will learn

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

UNIT - I:

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

UNIT - II:

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

UNIT - III:

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

UNIT - IV:

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

UNIT - V:

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

Text Books:

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

Reference Books:

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

B.Tech.

L T P C

II Year - I Semester

3 0 0 3

DATABASE MANAGEMENT SYSTEMS
(Common for CSE, CSE(AIML), CSE(DS))**Course Objectives:**

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

Course Outcomes: The student will learn

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

UNIT - I:

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

UNIT - II:

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

UNIT - III:

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV:

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

with Concurrent Transactions.

UNIT - V:

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

Text Books:

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

Reference Books:

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

B.Tech.

L T P C

II Year - I Semester

2 0 0 2

INNOVATION AND ENTREPRENEURSHIP
(Common for ECE, CSE, CSE(AI&ML), CSE(DS))**Course Objectives:**

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

Course Outcomes: The student will learn

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

UNIT - I: Fundamentals of Innovation and Entrepreneurship Innovation:

Introduction, need for innovation, Features, Types of innovations, innovations in manufacturing and service sectors, fostering a culture of innovation, planning for innovation. Entrepreneurship: Introduction, types of entrepreneurship attributes, mindset of entrepreneurial and intrapreneurial leadership, Role of entrepreneurs in economic development. Woman Entrepreneurship, Importance of on-campus startups. Understanding to build entrepreneurial mindset, attributes and networks individuals while on campus.

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

UNIT - II: Problem and Customer Identification

Identification of gap, problem, analyzing the problem from a industry perspective, real-world problems, market and customer segmentation, validation of customer problem fit, Iterating problem-customer fit, Competition and Industry trends mapping and assessing initial opportunity, Porter's Five Force Model. Idea generation, Ideation techniques: Brainstorming, Brain writing, Round robin, and SCAMPER, Design thinking principles, Mapping of solution to problem.

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

UNIT - III: Opportunity assessment and Prototype development

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

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

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

UNIT - IV: Business & Financial Models

Introduction to Business Model and types, Lean Canvas Approach: 9-block lean canvas model, building lean canvas for your startup. Business planning: components of Business plan- Sales plan, People plan and financial plan, Financial Planning: Types of costs, preparing a financial plan for profitability using a financial template, understanding the basics of Unit economics, Economies of Scale and analyzing financial performance. Go-To-Market (GTM) approach – Selecting the Right Channel, creating digital presence, and building customer acquisition strategy. Core Teaching Tool: Founder Case Studies – Sama and Securely Share; Class activity and discussions; Venture Activities.

UNIT - V: Startups and IPR

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

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

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

Suggested Readings:

1. John R Bessant, Joe Tidd, Innovation and Entrepreneurship, 4E, Wiley, Latest Edition.
2. Ajay Batra, The Startup Launch Book- A Practical Guide for Launching Customer Centric Ventures, Wiley, 2020. (For Core Teaching Tool).
3. Entrepreneurship Development and Small Business Enterprises, Poornima M Charantimath, 3E, Pearson, 2018.
4. D.F. Kuratko and T.V. Rao, Entrepreneurship: A South-Asian Perspective, Cengage Learning, 2013.
5. Robert D. Hisrich, Michael P. Peters, Dean A. Shepherd, Sabyasachi Sinha (2020). Entrepreneurship, McGrawHill, 11th Edition.
6. NISP -Brochure inside pages - startup_policy_2019.pdf

B.Tech.**L T P C****II Year - I Semester****0 0 2 1****OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB**

(Common for CSE, IT, CSE(AI&ML), CSE(DS))

Course Objectives:

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

Course Outcomes: The student will learn

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

List of Experiments:

1. Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
 - A) Develop an applet in Java that displays a simple message.
 - B) Develop an applet in Java that receives an integer in one text field, and computes its factorial
3. Value and returns it in another text field, when the button named "Compute" is clicked.
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
5. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is

- even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.
6. Write a Java program for the following:
 7. Create a doubly linked list of elements.
 8. Delete a given element from the above list.
 9. Display the contents of the list after deletion.
 10. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in the selected color. Initially, there is no message shown.
 11. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
 12. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas.
 13. Write a java program to display the table using Labels in Grid Layout.
 14. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
 15. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (t). It takes a
16. name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
 17. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
 18. Write a Java program to list all the files in a directory including the files present in all its
19. subdirectories.

Text Books:

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

Reference Books:

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

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

SOFTWARE ENGINEERING LAB
(Common for CSE, CSE(AI&ML), CSE(DS))

Course Objectives:

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

Course Outcomes: The student will learn

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

List of Experiments

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

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

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

Text Books:

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

Reference Books:

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

B.Tech.**L T P C****II Year - I Semester****0 0 2 1****DATABASE MANAGEMENT SYSTEMS LAB**
(Common for CSE, IT, CSE(AIML), CSE(DS))**Course Objectives:**

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

Course Outcomes: The student will learn

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

List of Experiments:

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

Text Books:

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

Reference Books:

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

B.Tech. - CSE

L T P C

II Year - I Semester

0 0 2 1

NODE JS/REACT JS/DJANGO

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

Course Objectives:

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

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

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

Exercises:

1. Build a responsive web application for shopping cart with registration, login, catalog and cart pages using CSS3 features, flex and grid.
2. Make the above web application responsive web application using Bootstrap framework.
3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1 and experiment 2.
3. Explore the features of ES6 like arrow functions, callbacks, promises, async/await.
4. Implement an application for reading the weather information from openweathermap.org and display the information in the form of a graph on the web page.
5. Develop a java stand alone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.
6. Create an xml for the bookstore. Validate the same using both DTD and XSD.
7. Design a controller with servlet that provides the interaction with application developed in experiment 1 and the database created in experiment 5.
8. Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)
9. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.
10. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)
11. For the above application create authorized end points using JWT (JSON Web Token).
12. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.

13. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using `chart.js`
14. Create a TODO application in react with necessary components and deploy it into GitHub.

Reference Books:

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

B.Tech.	L	T	P	C
II Year - I Semester	1	0	0	1

ENVIRONMENTAL SCIENCE
(Common for CSE, ECE)

Course Objectives:

- Understand the components, structure, and functions of ecosystems and their relevance to human society.
- Comprehend classification, sustainable management, and challenges of natural resources including water, minerals, land, forests, and energy.
- Grasp the significance, value, and conservation approaches for biodiversity, including threats and legislative frameworks.
- Analyze types, sources, and impacts of environmental pollution, and learn technological and policy measures for pollution prevention and control.
- Develop awareness about global environmental challenges, international agreements, and the role of policy, law, and Environmental Impact Assessment (EIA) in sustainable development.

Course Outcomes: The student will learn

- Understand the structure, function, and significance of ecosystems, including energy flow, biogeochemical cycles, and biodiversity conservation.
- Analyze the classification, utilization, and sustainable management of natural resources, along with alternative energy options.
- Evaluate biodiversity at genetic, species, and ecosystem levels, its values, threats, and conservation methods.
- Identify types, sources, and impacts of environmental pollution and apply suitable control technologies.
- Interpret environmental policies, legislation, and the EIA process to propose management plans.

UNIT - I: Ecosystems

Definition, scope, and importance of ecosystems. Classification, structure, and function of an ecosystem. Food chains, food webs, and ecological pyramids. Flow of energy, biogeochemical cycles, bioaccumulation, biomagnification, ecosystem value, services, and carrying capacity, field visits.

UNIT-II: Natural Resources

Classification of resources: living and non-living resources.

Water resources: use and overutilization of surface and groundwater, floods and droughts, dams: benefits and problems.

Mineral resources: use and exploitation, environmental effects of extracting and using

mineral resources.

Land resources: Forest resources.

Energy resources: growing energy needs, renewable and non-renewable sources, use of alternate energy source, case studies.

UNIT - III: Biodiversity and Biotic resources

Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity
consumptive use, productive use, social, ethical, aesthetic, and optional values. India as
a mega-diversity nation, hotspots of biodiversity. Field visit. Threats to biodiversity:
habitat loss, poaching of wildlife, man-wild life conflicts. Conservation of biodiversity:
in-situ and ex-situ. National Biodiversity Act.

UNIT - IV: Environmental Pollution and Control Technologies

Classification of pollution. **Air pollution:** primary and secondary pollutants, automobile
and industrial pollution, ambient air quality standards. **Water pollution:** sources and
types of pollution, drinking water quality standards. **Soil pollution:** sources and types,
impacts of modern agriculture, degradation of soil.

Noise pollution: sources and health hazards, standards, **Solid waste:** Municipal solid
waste management, e-waste management.

Pollution control technologies: wastewater treatment (primary, secondary, tertiary), air
pollution control, bioremediation.

Global issues: climate change, ozone depletion, deforestation, desertification.

International conventions: Earth Summit, Kyoto Protocol, Montreal Protocol, NAPCC
initiatives.

UNIT - V: Environmental Policy, Legislation & EIA

Environmental Protection Act, Air Act 1981, Water Act, Forest Act, Wildlife Act.

Rules for municipal solid waste, biomedical waste, and hazardous waste management.

EIA: structure, baseline data acquisition, impacts on air, water, biological, and socio-
economic aspects. Risk assessment strategies, Environmental Management Plan (EMP).

Contemporary issues: climate change, SDGs, global challenges, environmental policies,
and international agreements.

Text Books:

1. Introduction to Environmental Science – Y. Anjaneyulu.
2. Textbook of Environmental Studies – Erach Bharucha.
3. Environmental Studies – R. Rajagopalan.

Reference Books:

1. Environmental Science: Towards a Sustainable Future – Richard T. Wright.
2. Environmental Engineering and Science – Gilbert M. Masters & Wendell P. Ela.
3. Environmental Science – Daniel B. Botkin & Edward A. Keller.
4. Environmental Studies – Anubha Kaushik.
5. Environmental Science and Technology – M. Anji Reddy 2007 BS publications.