Node Lectureflow

Module-1) SE - Overview of IT Industry	**8**
• Introduction of students • Career in IT • Understanding Student Login of TOPS ERP • Using Lab • What is Program • What is programming? • Types of Programming Language • World Wide Web • How Internet Works • Network Layers on Client and Server • Client And Servers • Types of Internet Connections • Protocols • Application Security • Software Applications and its types • Software Architecture • Layers in Software Architecture • Software Environments • Types of Programming Languages • Source Code • Github and introductions • Student Account in Github • Types of Software • Introduction of Software • Application software • Software development process • Software Requirement • Software Analysis • System Design • Software Testing • Maintenance • Development • Web Application • Designing • moble application • DFD • Desktop Application • Flow Chart	
Module-3) SE - Fundamentals of Programming	**8**

• Basic Syntax
• Data Structures
• Variables
• Operators
• Control and looping Structures
• functions
• Arrays and strings
• Introduction to C
• What is Language?
• What is programming and program?
• Fundamental of Algorithms and Flowchart
• Real world problems - get solution via programs
• Practical Example: 1. Write a Flow chart of real problems - Days to month conversion system.
• Data Types and Variables - Data Types, Void Data Types,
• History of C
• Compiler and interpreter
• environment setup
• Type Modifiers,
• Basic Structure of C Programs
• Importance of C
• Fundamentals of C
• Difference between turbo C and Dev C/C++
• Practical Example : 1. Write a program of scanf 2. Write a program to demonstrate escape sequence 3. Write a program to demonstrate comments
• Comments
• Keywords
• Escape Sequence
• Practical Example: 1. Write a program to print (Hello World). 2. Write a program to print the sum of two numbers. 3. Write a program to exchange values of two variables using the 3rd variable. 4. Write a program to convert days into years and years into days.

Module-3.2) SE- C Language Programing with C	**12**

- Operators and Expression
- Control statements : Conditional statements 1) If statement 2) If..else statement 3) Else if statement 4) Nested if statement 5) Switch statement
- Practical Example : 1. Write a program to find user eligible for vote or not 2. Write a program to find user entered number is even or odd 3. write a program to find user entered numbers between 1 and 100 4. write a program to calculate percentage and grade 5. Write a program to check vowel using switch case
- Looping statements 1) For loop 2) While loop 3) Do.. while loop
- Jumping Statements 1) Break statement 2) Continue statement 3) Goto statement
- Practical Example: 1. Write a program to print 1 to 50 using for loop 2. Write a program to print sum of odd and even numbers between 1 to N 3. Write a program to print 1 to 5 using while loop and do.. While loop 4. Write a program to print half pyramid using * , numbers and characters 5. Write a program to demonstrate break, continue statement with for loop 6. Write a program to demonstrate goto statement
- Functions
- User Defined Functions
- Function categories
- Function Variable scopes
- local variables
- Formal parameters
- Practical Example: 1. Write a program to demonstrate four categories of function 2. Write a program to demonstrate menu driven calculator using function 3. Write a program to create Quiz application
- Practical Example: 1. Write a program to find factorial using function 2. Write a program to find Fibonacci series using function
- Arrays
- Types of array
- Practical Example : 1. Write a program to print static array 2. Write a program to take 10 values from the user and store them in an array 3. Write a program to demonstrate multi dimensional array
- Arrays and functions
- Practical Example : 1. Write a c program to arrange accepted numbers in ascending and descending order using array. 2. Write a c program for multiplication of matrix 3. Write a c program for addition of two matrix.
- Strings
- String Functions : Theoretical Explanation
- Strlen
- Strrev
- Strcat
- Strcmp
- Practical Example: 1. Find the frequency of a character in a string 2. Find the length of a string 3. Find the number of vowels, consonants, digits and white spaces 4. Write a program to copy string 5. Write a program to concatenate two strings 6. Write a program to remove all characters in a string except alphabets 7. Find the length of a string without using inbuilt function 8. Write a program to reverse string without using inbuilt function

Module-4) OOP Concept	**8**

- Procedure Oriented And object Oriented Programming
- Basic Concepts of OOP
- OOP - Objects and Classes
- Constructors and Destructors
- Data Abstraction and Encapsulation
- inheritance
- Encapsulation
- Types of polymorphism
- Dynamic Binding
- Array
- Types of constructors
- Compile time
- Types of Array
- Class and arrays : 1) Array within class 2) Array of objects
- Run time
- String
- Practical Example: 1. Write a program to print the score card of two students using an array of objects.
- Difference between constructor and destructor
- Practical Example: 1. Write a program to demonstrate difference between constructor and destructor 2. Write a program to demonstrate copy constructor
- Abstract class
- Practical Examples: 1. Write a program to check whether entered number is even or not using if..else statement in C++ 2. Write a menu - driven program to calculate the area of the circle,rectangle and triangle. 3. Write a program to calculate factorial of given number using for loop 4. Write a program to print the fibonacci series using while loop 5. Write a program to check whether the given number is palindrome using do..while loop. 6. Write a program to demonstrate jumping statements
- Practical Example: 4. Write a program to demonstrate pass object to a function 5. Write a program to demonstrate return object from function
- Class and pointer
- Aggregation
- Class and objects
- Practical Example: 1. Write a program to demonstrate pointer with class 2. Write a program to demonstrate dynamic object using new keyword
- Access modifiers
- Practical Example: 1. Write a program to demonstrate function overloading with different types of arguments 2. Write a program to demonstrate function overloading with default arguments 3. Write a program to show the constructor function overloading
- Member Function
- Types of inheritance 1 - Single level 2 - Multi-level 3 - Multiple 4- Hierarchical 5- Hybrid
- Comparisons of class and object
- Practical Example : Write a program to implement single level inheritance 2. Write a program to demonstrate single level inheritance in private mode 3. Write a program to demonstrate the ambiguity in single level inheritance 4. Write a program to demonstrate multilevel inheritance 5. Write a program to demonstrate multiple inheritance 6. Write a program to demonstrate the hierarchical inheritance 7. Write a program to demonstrate the hybrid inheritance
- Namespace
- Static Keyword
- Practical Example: 1) Write a program to demonstrate constructor invocation in inheritance
- Scope resolution operator

Module-5) SE - Database	**9**

• What is Database • DBMS and RDBMS • Types of Database • Normalization • algebra • Primary key • foreign key • unique key • Database Programming Language SQL • SQL Statements Types • DDL • DML • TCL • TQL • Database backup and Restore • What are Joins • Types of Joins • Function • Procedure • Trigger • Curser • Transaction concepts • properties of transactions • rollback and commit savepoint • ER database schema	
Module 6) WD - HTML	**10**

• Student Intro , Career Center Login ,What is Internet, HTTP/HTTPS, WWW, Domain name and Top Domain name • SEO, What is HTML, What is Text Editor, Web Browser, Downloading Text Editor , HTML Structure, First Program in HTML • 1) HTML Introduction 2) HTML Getting Started 3) HTML Elements 4) HTML Attributes 5) HTML Basic Tags • 1) HTML Doctypes 2) HTML Layout 3) HTML Head 4) HTML Meta 5) HTML Scripts • Practical Examples: 1) Create any simple web page to display your name. 2) Importance of meta tag and Doctypes • Tags and self Closing Tags, Basic Tag , Attribute and Events, Marquee Tag • HTML - Meta Tags, HTML - Comments, HTML - Images, HTML - Tables, HTML - Lists, HTML - Text Links, HTML - Image Links • HTML Headings HTML Paragraphs HTML Links HTML Text Formatting HTML Styles HTML Images • HTML - Frames, HTML - Iframes, HTML - Blocks, HTML - Backgrounds, HTML - Colors, HTML - Fonts • Anchor Tag, Img Tag, Image Mapping • HTML - Fonts, HTML - Forms, HTML - Embed Multimedia ,HTML - Marquees, HTML - Header, HTML - Style Sheet, HTML - Javascript ,HTML - Layouts • List Tag, Tables, Forms • HTML - Tags Reference, HTML - Attributes Reference, HTML - Events Reference, HTML - Fonts Reference, HTML - ASCII Codes, ASCII Table, Lookup, HTML - Color Names, HTML - Entities, HTML - Fonts, Ref HTML - Events, Ref MIME Media Types, HTML - URL Encoding Language, ISO Codes HTML - Character Encodings, HTML - Deprecated Tags • PRactical Examples: 1) Create simple Doc and display your name using different heading tag 2) Create link for open google. 3) Create document using all text formatting tags • HTML online editor • HTML Tables HTML Lists HTML Forms HTML Iframes • Practical Examples: 1) Create simple table 2) Create time table for your school 3) Create table with colspanrowspan example 4) Create invoice using table 5) Create hotel menu. 6) Create index page for your book. 7) Create list with different categories. • PRactical Examples: Create registration form with all fields and validation	
Module 7) WD - CSS and CSS 3	**16**

• 1) CSS 2) In-line CSS Internal Style External Style Sheet @import Style Sheet 3) CSS Class CSS ID • What is CSS How to Implement CSS Class and ID Width and Height Css Unit Box Model (Margin,padding,Border) and create basic template design • Practical example : Create page with difference color text • CSS Selectors , Pseudo Classes and Elements , Float and Clear and Alignment , Font Styling , Opacity and Visibility , Line Height • 1) CSS Text 2)CSS Font 3) CSS Background 4) CSS Links 5) CSS Lists 6) CSS Display 7) CSS Visibility • Creating Header of Website , Outline , Background , Counter increment , Counter reset ,Cursor , Overflow • PRactical Example : Create layout for your project • Position , Creating Submenu , Border Radius, Transform , Animation , Font Awesome Icons • 1) CSS Layout Model 2) CSS Border 3) CSS Margin 4) CSS Padding 5) CSS Outline • Font Family Through Google Font , import fontface rule ,FlexBox • 1) CSS Float 2) CSS Align 3) CSS Position 4) CSS Element Size 5) CSS Layer • Practical Example : Create image gallery • 1) CSS Pseudo Class Selector 2) CSS Pseudo Element Selector • CSS Properties 1) Background, 2) border 3) bottom 4) caption-side 5) clear 6) clip 7) color 8) content • Practical Example: Create Menu with logo at left side and contact info at right side using clear effect • 1) counter-increment 2) counter-reset 3) cursor 4) direction 5) display 6) empty-cells • Practical Example: 1) Create submenu list using counter • 1) float 2) font 3) height 4) left 5) letter-spacing 6) line [height, style, style-7) image, style-position, 8) style-type] 9) margin 10) outline 11) overflow 12) padding • 1) page-break 2) position 3) quotes 4) right 5) table-layout 6) text 7) top 8) vertical-align 9) visibility 10) white-space 11) width 12) word-spacing 13) z-index • Practical Example: wireframe layout for your template using div • Media Query (For Responsive Website) , Creating a Responsive Website • Validate a Website, Hosting a website with free domain name, Column , Clippath , Gradient Color , Filter, Border Image	
Module 8) Website Designing - HTML5	**5**
• HTML5 Tags, HTML5 Input and Attribute • Audio and Video, Semantic Element in HTML5 • Canvas, Svg • Display Grid	
Module 9) WD - JAVASCRIPT BASIC & DOM	**10**

• 1) JS Introduction 2) JS Getting Started 3) JS Syntax 4) JS Variables 5) JS Generating Output 6) JS Data Types 7) JS Operators 8) JS Events 9) JS Strings 10) JS Numbers 11) JS If, Else 12) JS Switch Case • What is Javascript, Creating First Javascript Program, Way to apply Javascript , Event in Javascript , How to select Tag Classes and Id • Practical Example: 1) Create program for input color and output that code 2) Create program for pattern using loop • Functions, Alert ,Confirm , Prompt , Addition of Two Number , Hide and Show Password • a) JS Arrays b) JS Sorting Arrays c) JS Loops d) JS Functions e) JS Objects • How to apply CSS using Javascript, Image Changing • Practical Example: 1) Create function Finding the Maximum and Minimum Value in an Array 2) Create pyramid pattern program • If Else Statement, Javascript Form Validation • JAVASCRIPT & DOM 1) JS DOM Nodes 2) JS DOM Selectors 3) JS DOM Styling 4) JS DOM Get Set Attributes 5) JS DOM Manipulation 6) JS DOM Navigation • How to see Trafic in Our Website • Practical Examples: 1) Get input data and perform different operations 2) Make dynamic CSS by click • JAVASCRIPT ADVANCED 1) JS Date and Time 2) JS Math Operations 3) JS Type Conversions 4) JS Event Listener 5) JS Regular Expressions 6) JS Error Handling • Practical Example: Create custom Validation	
Module 10) WD - JQuery Basic, Effects & Advanced	**6**
• jQuery Basic a) jQuery Introduction b) jQuery Getting Started c) jQuery Syntax d) jQuery Selectors e) jQuery Events • What is JQuery , Downloading JQuery File , First Program in JQuery • Practical Example: Change CSS • JQuery Syntax , Query Selector, Hide , Slide , Fade Effect in JQuery • JQuery Effects 1) jQuery Show/Hide 2) jQuery Fade 3) jQuery Slide 4) jQuery Animation 5) jQuery Stop 6) jQuery Chaining 7) jQuery Callback • How to Apply CSS Using JQuery, How to Add Class and Remove Class in Jquery , JQuery Animation • Practical Example: Create slider with animation • Filter using JQuery , JQuery Slider Plugin , Validation Plugin • JQuery Advanced 1) jQuery Traversing 2) jQuery Ancestors 3) jQuery Descendants 4) jQuery Siblings 5) jQuery Filtering 6) jQuery Load 7) jQuery No-Conflict • Zoom Plugin, Now Make Your Existing Website Dynamic with Javascript and JQuery	
Module 11) WD - Bootstrap Basic & Advanced	**9**

• Bootstrap Basic 1) Bootstrap Introduction 2) Bootstrap Getting Started 3) Bootstrap Grid System 4) Bootstrap Fixed Layout 5) Bootstrap Fluid Layout 6) Bootstrap Responsive Layout • Practical Example: Create Navigation Menu • 1) Bootstrap Typography 2) Bootstrap Tables 3) Bootstrap Lists 4) Bootstrap List Groups 5) Bootstrap Forms 6) Bootstrap Custom Forms 7) Bootstrap Input Groups 8) Bootstrap Buttons 9) Bootstrap Button Groups • Practical Example: Create login registration form • 1) Bootstrap Images 2) Bootstrap Cards 3) Bootstrap Media Objects 4) Bootstrap Icons 5) Bootstrap Navs 6) Bootstrap Navbar 7) Bootstrap Breadcrumbs 8) Bootstrap Pagination 9) Bootstrap Badges 10) Bootstrap Progress Bars 11) Bootstrap Spinners 12) Bootstrap Jumbotron 13) Bootstrap Helper Classes • Practical Example: 1) Create image gallery 2) Create model for login Product list page with pagination • Bootstrap Advanced 1) Bootstrap Modals 2) Bootstrap Dropdowns 3) Bootstrap Tabs 4) Bootstrap Tooltips 5) Bootstrap Popovers 6) Bootstrap Alerts 7) Bootstrap Stateful Buttons 8) Bootstrap Accordion 9) Bootstrap Carousel 10) Bootstrap Typeahead 11) Bootstrap ScrollSpy 12) Bootstrap Toasts • Practical Example: Create your project website using bootstrap	
Module 12) Node - Javascript Fundamental	**2**
• JavaScript Introduction, Difference between JavaScript and Java, Client Server Model, Objects in JavaScript • Arrow Function, Bind, Apply, Closure, Classes in JavaScript, JavaScript JSON • Practicals	
Module 13) Node - NodeJS - Introduction	**5**
• Node JS introduction, Chrome V8 engine, Core Modules, Local Modules, NPM Modules, module.exports, nodemon • Command line arguments, Process object, Args parsing with Yargs, Storing data with JSON, Filter VS Find, Reading and Writing notes • File Module example, Local Module example, Validator npm library example • Create TODO app in node js using command line arguments	
Module 14) Node - Web Development with Node	**8**
• Asynchronous Node JS , Making HTTP request • JSON Parsing, Geo-coding, Error Handling • Callback function, Callback chaining, De-structure Object • Create an app to fetch temperature of given city using geocoding and temperature API, • Web Server with Express, Install express, Create server • Serve data like HTML, JSON Object, JSON Array, Static Assets, Path Module • Template Engine, Dynamic Page, Passing data to .hbs file, Handlebars Particles, Error 404 Pages • Practical of - Returning html response, Returning json object, Returning json array, Display image from static assets • Use of CSS and JavaScript file from Assets folder, Create header, footer particles and use in different hbs files like index.hbs, about.hbs, help.hbs • Accessing API from browser, Query String, Call Weather and Geo Code API from browser • ES6 : Default Function Parameter, Default Value with De-structuring, Browser HTTP request with fetch() • Search Form with prevent form submission and fetch weather information for city input by user	
Module 15) Node - Node with Mongodb	**6**

• Mongodb vs MySQL, Installing MongoDB , Robo 3t GUI viewer, Connecting to Mongodb, Object ID, ES6 : Promises • Inserting Document, Insert Bulk Documents, Read one Document, Read all Documents • Count all Documents, Update One Document, Update Many Documents, Deleting Documents • • Create User Model with name (string) and age (int) field and save it. Create Task model with description (String) and completed (Boolean) fields and save it. • User Age is more 18 - Remove Space around name, Custom validator : age>0 - Custom validator : email validation • Add Password field to User with proper validation, Apply validation to Task model	
Module 16) Structuring REST API	**4**
• Request, Response, Resource Creation End Points, HTTP status, Resource Reading End Points, Promise chaining • ES6 - Async/Await, Resource Updating End Points, Deleting End points, Separating Route files for user and task • Create User & Task creation end point with success and error handling , Create User & Task update end point with success and error handling • Create User & Task delete end point with success and error handling, Create User & Task reading end point with success and error handling	
Module 17) - Node - [API Authentication & Security]	**3**
• Securely storing password, Comparing password when login, Mongoose Middleware, Unique email id in User model • JSON Web Tokens, Verify Token and Set Expiry, Express Middleware, Logging out • Login , Logout with token, Hiding private data, Encrypt password - Read Task for login user only • Api authentication with token • Image uploading in api • Payment with node • socket programming , like chat app • SMs and email sending , for OTP verification	
Adv. Javascript -> Module - 18) Introduction to Javascript	**3**
• Introduction to JavaScript, Javascript Features, Ways to write and Add JS code • Comments in JS, Variables in JS, • DataTypes - Primitive, Non Primitive • Operators - Arithmetic, Comparision, Bitwise, Logical, Assignment, Special JS operators • JS if/else, switch, do/while, for/in, • JS functions, function call, bind v/s apply	
Advance Javascript - Module 19) Javascript Object	**4**
• JS objects, JS Object methods, JS Arrays, JS array methods • JS string, JS string methods • JS numbers, and number constants, Number methods • JS data Objects, JS math Objects	
Advance JS - Module 20) JavaScript BOM and DOM	**4**
• BOM - Window object, history object, Navigtator Object, screen Object, • DOM, Model, Methods • JS Validations, Form Validations, Email Validations, OOP	
ADv. JS - Module 21) JavaScript ES 6 Object Oriented Concept	**6**

• JS classes • JS Prototypes • Construtors • Static Methods • OOP Encapsulation • Inheritance • Polymorphism • Abstraction	
Adv. JavaScript - Module 22) Advance JavaScript	**10**
• JS cookies, Cookie Attributes, Multiple Names, deleting a cookie, examples • JS events, ADDevent listner, Click Events, Double Click event, onload event, OnResize event, • Exceptional Handling, JS Throw statement • Promise in JS, Promises Chaining, Error Handling with Promises, Promise API, Async wait in JS • JS Async - Callbacks - Promises - Async/Await • JS Typed Array Methods, SETS in JS, JS Map, Weakset, WeakMap, • JS Callbacks, Closures, Date Difference, Date Formats, Date PArse, Defer, Redirect • Scope, scroll, Sleep, Void, Form events, Type conversions	
Module 23) Introduction to react	**3**
Module-24) React - Components, State, Props	**8**
• Installation - Add React to a HTML Website - Create New React App - Hello World • Getting started in React • JSX • Components • Component Composition • JSX - Why JSX? - Embedding Expressions in JSX - Attributes with JSX - Children with JSX • Props & Prop Types • Event Handlers • State • React Web App • Components, State, Props - Function Component - Class Component - Props - State - Class Component Lifecycle	
Module 25) Lists and Hooks	**6**
• Conditional Rendering - Lists and Keys - Forms - Handling Events - Lifting State up • Hooks - Introduction - Using the State hook - Using the Effect hook - Rules of Hook - Custom Hook	
Module-26) React - Styling & Advance React	**5**

• Creating the first App • Understanding the App • Styling the App • Inspecting & Debugging styles • Built-in components • Working with Images • ListViews • TextInput • Styling React Components - CSS stylesheet - Inline Styling - CSS Modules - CSS in JS Libraries (styled components) • Creating Views (Scenes) • Conditional Rendering - Lists and Keys - Forms - Handling Events - Lifting State up • Hooks - Introduction - Using the State hook - Using the Effect hook - Rules of Hook - Custom Hook • Advance Concepts - Context, useContext() - Working with Refs and useRefs() - Fragments - Performance optimization with useMemo() - Styling React Components - CSS stylesheet - Inline Styling - CSS Modules - CSS in JS Libraries (styled components) • Bootstrap with React • React Router - Browser - Router - Link - Route - Template integration - Http Request in React - Get and Post data	
Module 27) React Router	**8**
• React Router • Browser - Router - Link - Route • Template integration - Http Request in React - Get and Post data	
Module-28) React - Applying Redux	**8**
• State • State storage problem • Redux Basics • Redux Principles • Implementing Redux • React-Redux • Middleware • Counter App Demo • Redux - Complexity of Managing state - Understand the Redux Flow - Setting up Reducer and store - Dispatching Actions - Passing and Retrieving Data with Action - Combining Multiple Reducers - Adding Middleware - Redux Dev tools	