



Spring Boot



Hibernate



Java SE
Development

JAVA FullStack

Duration: 100 Days



JAVA Introduction

- ☒ History
- ☒ Features



Java Basics & Introduction to Eclipse

- ☒ Post Install Configuration
- ☒ Writing a simple program
- ☒ compiling and executing
- ☒ Data types
- ☒ Variables and Arrays
- ☒ Using Eclipse
- ☒ Create packaged classes
- ☒ Eclipse shortcuts

✉ info@codegnan.com www.codegnan.com

📍 # 40-5-19/16, Prasad Naidu Complex, P.B. Siddhartha Bustop,
Moghalrajpuram Vijayawada-520010

1st floor, Kotwal Madhava Reddy Plaza, opposite to JNTU College Metro Station,
Hyderabad, Telangana 500072



Call For More Info
63013 41478



Operators & Expressions

- ✓ Arithmetic Operators
- ✓ Boolean Operators
- ✓ Logical Operators
- ✓ Binary Operators



Control Statements

- ✓ Branching Statements
- ✓ Iterative Statements
- ✓ Break & Continue with enhancements
- ✓ While
- ✓ For
- ✓ Do..While
- ✓ Break and Continue Statement



Java Object Oriented Programming

- ✓ Basics of OOPs
- ✓ Fundamentals of class & object
- ✓ new keyword
- ✓ Reference variables
- ✓ Member methods of a class
- ✓ Constructors
- ✓ Finalize method
- ✓ Overloading member methods
- ✓ Overloading constructors
- ✓ Passing and returning objects with methods
- ✓ Access Control
- ✓ Static Methods
- ✓ Static Variables
- ✓ Static Block
- ✓ Using final keyword
- ✓ Unit Testing using Junit-5



Inheritance

- ✓ Basics of Inheritance
- ✓ Members accessibility in inheritance
- ✓ Using super keyword
- ✓ The sequence of execution of constructors in inheritance
- ✓ Method Overriding
- ✓ Dynamic Method Dispatch
- ✓ Abstract classes
- ✓ Preventing overriding
- ✓ Preventing inheritance



Exception Handling

- ✓ Fundamentals of Exceptions
- ✓ Types of exceptions
- ✓ Using try and catch keywords
- ✓ Multiple catches
- ✓ Nesting of try blocks
- ✓ Using throw keyword
- ✓ Using throws keyword
- ✓ Finally block
- ✓ Some predefined exceptions and their usage
- ✓ User defined exceptions



Interfaces

- ✓ Purpose of Interface
- ✓ Defining an interface
- ✓ Implementing interfaces
- ✓ Interface reference variables
- ✓ Interface with variables
- ✓ Extending interfaces



Multi Threaded programming

- ✓ Basics of threads
- ✓ Java threaded model
- ✓ Defining threads using Runnable interface
- ✓ Defining threads using Thread superclass
- ✓ Multiple threads
- ✓ Thread Priority values
- ✓ Thread Synchronization using synchronized methods
- ✓ Thread Synchronization using synchronized blocks



Predefined Libraries

- ✓ Using String class
- ✓ Using java.lang package
- ✓ Working with Data & Time
- ✓ Utility framework
- ✓ Collection framework
- ✓ I/O framework



RDBMS (MySQL)

- ✓ Introduction to Database & DBMS
- ✓ Data types
- ✓ Table, Record, Field
- ✓ SQL Queries,
- ✓ Database Normalization
- ✓ Joins, Sub Queries
- ✓ INSERT | UPDATE | DELETE Operations
- ✓ DDL Commands
- ✓ PL/Sql



Database Programming

- ✓ Overview of JDBC API
- ✓ Different types of JDBC Drivers
- ✓ JDBC URLs
- ✓ DriverManager
- ✓ Establishing a connection with the database
- ✓ Creating and executing SQL Statements
- ✓ Working with CallableStatement



Server side Programming with Servlets

- ✓ Definition
- ✓ Deploying a simple servlet in a Servlet Container (Tomcat)
- ✓ Life cycle of a Servlet
- ✓ Servlet interface
- ✓ ServletRequest Interface
- ✓ ServletResponse Interface
- ✓ Additional capabilities of HTTPServlet
- ✓ Servlet Programming with JDBC CRUD Operations
- ✓ Session tracking



Java Server Pages (JSPs)

- ✓ JSP Basics
- ✓ Differences between Servlets and JSPs
- ✓ Running a simple JSP
- ✓ The JSP generated Servlet code
- ✓ JSP Implicit Objects
- ✓ JSP Syntax for Different JSP Elements
- ✓ Developing JSP Beans



JPA-Hibernate (ORM Framework)

- ✓ Introduction
- ✓ BeanFactory and application Context
- ✓ Container Concepts
- ✓ DAO Support and JDBC framework
- ✓ Transaction and Resource Management
- ✓ O/R Mapping



Spring

- ✓ Introduction
- ✓ BeanFactory and application Context
- ✓ Container Concepts
- ✓ Spring Data JPA Template
- ✓ AOP
- ✓ MVC



Spring Boot

- ✓ Introduction to STS
- ✓ DI with STS
- ✓ MVC, AOP
- ✓ Security, Role based Authentication, OAuth2, Token based authentication



Spring WebServices

- ✓ Introduction to Web Service
- ✓ Basics of REST APIs
- ✓ Spring REST



Microservices with Spring Boot, Spring Cloud

- ✓ Introduction to MicroService architecture
- ✓ Advantages with MicroService over Monolithinc architecture
- ✓ Develop and Deploy MicroService application in localhost
- ✓ Introduction to Service Discovery
- ✓ Client side Discovery pattern
- ✓ Server side Discovery pattern
- ✓ Load Balancing configuration



DevOps Tools

- ✓ Introduction to DevOps and advantages
- ✓ Git
- ✓ Maven
- ✓ Jenkins
- ✓ Docker
- ✓ Unit Testing with JUnit



Design Patterns

- ✓ Creational Design Patterns
- ✓ Structural Design Patterns
- ✓ Behavioral Design Patterns
- ✓ Miscellaneous Design Patterns

Projects:-



Life Line - A Health Assistance Web Application

Requirement description:

- ✓ Authentication and Authorization
- ✓ Patient portfolio service- Health records (medical checkup history), book a bed (hospital), compare the cost of beds, compare Test reports, order medicines
- ✓ Doctor (service) -find doctor, book an appointment, doctor details
- ✓ Hospital (service) - list of doctors, specialty, available beds, book outdoor appointment, treatment package(pregnancy, heart surgery etc.)
- ✓ Pathology (service) - list of tests, details about each test, book an appointment, test results(current and history)
- ✓ Medical Store (Service) - search Medicines, order medicines, upload prescriptions, view order history
- ✓ Ambulance service – Find Ambulance, Get travel cost, Book a travel, View travel history



Online Medicine Shopping System

Requirement description:

- ☒ To provide computerised data storage facilities.
- ☒ Users can search easily for any record.
- ☒ The new system requires less time for completion of any work.
- ☒ All the stock of medicine is updated automatically in the new system.
- ☒ Integration of Payment Gateway
- ☒ The system is user friendly and anyone having computer knowledge can handle it easily.
- ☒ Maintaining stock, Supplier information, Customer information & bill information are easy.



Online Auction Application (e-Auction)

Requirement description:

- ☒ Develop a web based application auctions. This application allows the users to sell something in auctions.
- ☒ Users of the application bid on the products they are interested in buying.
- ☒ The main objective of the e-Auction must be to obtain best value and the highest price.
- ☒ It cannot be possible to achieve best value outcomes whilst the focus remains on price.



Online Banking Application:-

Requirement description:

An online-banking application allows you to perform day to day banking activities.

- ☒ New User Registration
- ☒ Login
- ☒ Deposit
- ☒ Withdrawal
- ☒ Fund Transfer

FrontEnd Development



Introduction

- ✓ History and Importance of HTML
- ✓ Basic HTML syntax
- ✓ The current state of HTML
- ✓ HTML resources
- ✓ Choosing a code editor
- ✓ Exploring an HTML document
- ✓ DOCTYPE declarations
- ✓ Document head
- ✓ Understanding content models



Formatting Page Content

- ✓ Formatting content with HTML
- ✓ Headings
- ✓ Paragraphs
- ✓ Whitespaces
- ✓ Images



Creating Lists

- ✓ Unordered lists
- ✓ Ordered lists
- ✓ Definition lists



Creating Links

- ✓ Anchor element
- ✓ Page internal linking
- ✓ Linking to external pages
- ✓ Linking to downloadable
- ✓ Linking to page regions

HTML Forms

- ✓ Introduction to forms and their importance
- ✓ Form structure and basic form elements
- ✓ Text inputs, textareas, and buttons

Form Controls, Attributes and Elements

- ✓ Radio buttons and checkboxes
- ✓ Select dropdowns and file uploads
- ✓ Hidden inputs and labels

Projects:-

- ✓ 1. Favourite Blog
- ✓ 2. Resume building with HTML
- ✓ 3. Registration & Login page



CSS3



Getting started

- ✓ HTML overview
- ✓ Default browser styles
- ✓ Browser support and inconsistencies
- ✓ Inline, internal and external CSS
- ✓ Naming conventions



CSS Core

- ✓ Syntax, terminology, and naming conventions
- ✓ Type, class, and id selectors
- ✓ Combinator selectors
- ✓ Selectors: Best practices



Typography

- ✓ Web-safe fonts and the font-family property
- ✓ Web fonts and Google fonts
- ✓ The font-size property
- ✓ The font-style and font-weight properties
- ✓ The color, line height, and text properties
- ✓ Adjusting the font-weight property



Layouts

- ✓ Block vs. inline display
- ✓ The box model
- ✓ Margin and page layout
- ✓ Padding
- ✓ Floats
- ✓ The box model fix
- ✓ Box model review
- ✓ Float and display review
- ✓ Horizontal navs with the display property
- ✓ Horizontal navs with the float property
- ✓ Positioning
- ✓ Float, display, and position
- ✓ Layers and the z-index property



Advanced layout

- ✓ CSS grid
- ✓ CSS Flexbox



Projects

- ✓ 1. Building Paytm clone Page
- ✓ 2. Building Portfolio page



Bootstrap - CSS Framework



Introduction

- ☒ What is Bootstrap? Why do we use it?
- ☒ Initial environment setup



Typography and Utilities

- ☒ Section introduction
- ☒ Headings and basic typography
- ☒ Text alignment and Display
- ☒ Floats and Fixed positions
- ☒ Colors and Backgrounds
- ☒ Margin and Padding spacing
- ☒ Sizing and Borders
- ☒ CSS Breakpoints



CSS Components

- ☒ Section Intro
- ☒ Buttons and Button groups
- ☒ Navbar and Navs
- ☒ List groups and Badges
- ☒ Forms & input
- ☒ Input groups
- ☒ Alerts and Progress Bars
- ☒ Tables and Pagination
- ☒ Working with cards
- ☒ Media objects
- ☒ Jumbotron



Grid system and Flex-box

- ✓ Section Intro
- ✓ Grid system
- ✓ Grid alignment
- ✓ Flexbox classes
- ✓ Auto margin, wrap, and ordering



JavaScript



Introduction

- ✓ What is JavaScript?
- ✓ How does JavaScript relate to Java?
- ✓ History of JavaScript
- ✓ Navigating the JavaScript landscape
- ✓ Tools for JavaScript development
- ✓ Introducing the browser console
- ✓ Add inline JavaScript to an HTML document
- ✓ Add JavaScript in an external file



Writing JavaScript

- ✓ Basic syntax & comments
- ✓ Declarations
- ✓ Arithmetic operators and math operators
- ✓ Working with strings and numbers
- ✓ Literals



Control Flow

- ✓ Conditional statements and logic
- ✓ Looping Statements



Arrays

- ☒ Arrays
- ☒ Predefined functions of an Array



Functions

- ☒ Defining functions
- ☒ Calling functions
- ☒ Closures
- ☒ Arguments & parameters
- ☒ Arrow functions



Essential JavaScript Built-in methods

- ☒ Number Methods
- ☒ Boolean Methods
- ☒ String Methods
- ☒ Array Methods
- ☒ Math Methods
- ☒ RegExp Methods



Objects:

- ☒ Creating objects
- ☒ Objects and properties
- ☒ Map



JavaScript modules

- ☒ Exporting
- ☒ Importing
- ☒ Default exports
- ☒ Renaming features
- ☒ Aggregating modules
- ☒ Dynamic module loading



JavaScript and the DOM

- ✓ Intro to DOM
- ✓ Target elements in the DOM with querySelector methods
- ✓ Access and change elements
- ✓ Access and change classes
- ✓ Access and change attributes
- ✓ Add DOM elements
- ✓ Apply inline CSS to an element
- ✓ DOM events



JavaScript Server Actions

- ✓ AJAX calls
- ✓ get, post, put, delete using fetch() and axios library

Projects:-

- ✓ 1. Creating a website in which we can upload files, displaying data in table format, and adding CSS to it from the front end.
- ✓ 2. Creating a simple College website using HTML, CSS, JS. In which we can move from one page to another page, building a registration page.



ReactJs



Introduction to React.js

- ✓ Setting up a React development environment (e.g., Node.js, npm, Create React App)



Creating Your First React Application

- ✓ Hello World example
- ✓ Understanding React components
- ✓ JSX syntax



Understanding Components and Props

- ✓ Functional components
- ✓ Class components
- ✓ Passing and using props



State and Lifecycle

- ✓ State in React components
- ✓ Updating state
- ✓ Component lifecycle methods



React Hooks

- ✓ useState()
- ✓ useEffect()
- ✓ useContext()



Handling Events

- ✓ Event handling in React
- ✓ Binding event handlers
- ✓ Arrow functions vs. regular functions



Working with Forms

- ✓ Controlled components
- ✓ Handling form submission
- ✓ Form validation



Conditional Rendering

- ✓ Conditional rendering with if statements
- ✓ Ternary operators and logical && in JSX



Lists and Keys

- ✓ Rendering Lists
- ✓ Using .map() to render lists of elements
- ✓ Providing a key for each item



Understanding Keys

- ✓ The importance of keys in React
- ✓ Choosing the correct key



Styling in React.js

- ✓ CSS in React
- ✓ Different approaches for styling (CSS, CSS-in-JS, CSS Modules)
- ✓ Inline styles
- ✓ Styling Libraries
- ✓ Popular CSS frameworks (e.g., Bootstrap, Material-UI)



React Router

- ✓ Introduction to React Router
- ✓ Setting up React Router
- ✓ Creating routes
- ✓ Navigating with React Router
- ✓ Using Link and NavLink
- ✓ Nested Routes and Dynamic Routing
- ✓ Nested routes
- ✓ Passing parameters to routes



State Management with Redux

- ✓ Introduction to Redux
- ✓ Understanding the need for state management
- ✓ Basic concepts: actions, reducers, store
- ✓ Setting Up Redux
- ✓ Installing Redux and setting up a store
- ✓ Creating actions and reducers
- ✓ Connecting React with Redux
- ✓ Using connect to connect components to the store
- ✓ Dispatching actions



Asynchronous Programming and API Integration

- ✓ AJAX and Fetch API
- ✓ Making HTTP requests in React
- ✓ Fetching data from an API
- ✓ Async/Await and Promises



Handling errors in React applications

- ✓ Error Handling and Debugging
- ✓ Debugging React apps
- ✓ Performance Optimization
- ✓ Memoization
- ✓ React.memo and PureComponent

Hands-on Projects

1) Hospital Management System:- It is a single Page application. In which doctors and patients are available. Using ReactJS, json-server package.

Doctors are going to be added based on their Specialisation.

Patients are going to book appointments for a Specific Doctor.

2) Online Banking Application:- A simple online-banking application allows you to perform day to day banking activities.

New User Registration

Login

Deposit

Withdrawal

Fund Transfer

Deploying a React application

- Deployment strategies (e.g., Netlify, Vercel, AWS)