SêvẽnMentor PVT.LTD

JAVA FULL STACK

Code your future with Java – become a full stack pro!

www.sevenmentor.com

020 7117 3035

registration@sevenmentor.com

MEET OUR INSPIRING PILLARS

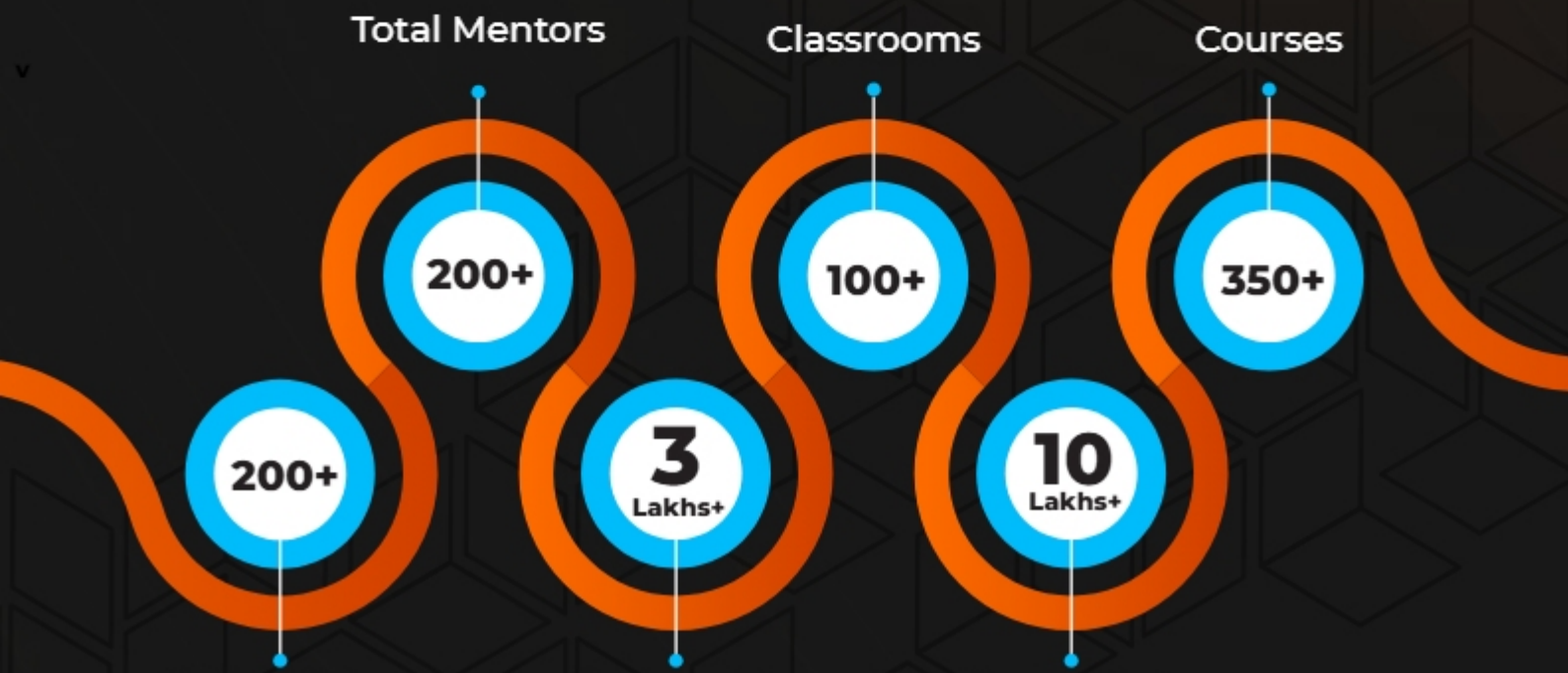
Pawan Bhosikar
(Chief Executive Officer)

Nisha Mandot
(Chief Financial Officer)

ABOUT US

"SevenMentor" has improvised a tendency to cope with all International Standards within their courses, by engaging both ends of the industry for students, professionals, and individuals to corporate clients. The organization conjointly gives chance within their educational programs to meet the needs with projected desires of quick developing networking trade. "SevenMentor" education techniques allow honing the abilities of the Networking experts from Industries which allow them to be equipped with the updated technology and standards of their operating environment. The group is a center for technical excellence with the kingdom of the art lab centers & a properly exquisite curriculum which gives them exposure in advance and enables them to be specific inside the certification enterprise.

AT A GLANCE, OUR STATISTICS SPEAK VOLUMES



COURSE HIGHLIGHTS

ONLINE AND CLASSROOM MODE

LIVE PROJECTS

PROFESSIONAL CERTIFICATES

100% JOB CALLS ASSISTANCE

MOCK INTERVIEW

MULTIPLE CAREER OPPORTUNITIES

SevenMentor PVT LTD

RESUME PRESENTATION

INTERVIEW Q&A

PLACEMENT TRAINING

ELIGIBILITY CRITERIA

MOCK INTERVIEWS

APTITUDE TEST

SCHEDULING INTERVIEWS

JOB

PLACEMENT AT YOUR DREAM JOB

CAMPUS DRIVE

SECURE A POSITION IN LEADING COMPANIES

SevenMentor PVT LTD

mantra properties
U&D Square Exploring Possibilities
BIS
Microsoft
Infosys
LARSEN & TOUBRO
BULFRO
SAR CUSTOM SOLUTIONS
ovtpl1
accenture
Capgemini
Tech Mahindra
cognizant
Simplify Healthcare
RASA
VINTECH
AI BUZZ
wipro
HITACHI Inspire the Next
Seshaasai
TATA
Integs Cloud
DUMOSH.
rubicon
Scymes
Yagna
Ashdip
NABLA
<TESTRIG>
SunONE TECHNOLOGIES
ASSIMILATE
toshvi
AURUS
NUMETRY
FABER
Manabh
Qualitas Globa
ELANSOL TECHNOLOGIES
TEKDI
appristine
GAMECLOUD
H.B. Fuller
moschip
NETSERV
GENXCEL
CodePro Technologies
Panacea INFOSEC
GROWSTATS
GEMSCAP
infogain
LEAPOT TECHNOLOGIES

ROUTE TO SUCCESS

1 CONFUSED WHAT TO DO IN CAREER..?

2 CHOOSE SEVENMENTOR

3 CAREER COUNSELING WITH EXPERTS

4 ONLINE/CLASSROOM EXPERT TRAINING

5 WHAT WILL YOU GET

- Industry-Based Projects
- Resume Preparation
- Soft Skills Development
- Mock Interview Sessions

6 CHARTING NEW PATHS: CAMPUS DRIVE

7 SECURE YOUR SPOT IN LEADING MNCS

TOOLS & TECHNOLOGIES COVERED

SevenMentor PVT.LTD.

OBJECT ORIENTED PROGRAMMING

INHERITANCE

POLYMORPHISM

EXCEPTION HANDLING

XML

JDBC

SPRING WEB

JSP

OBJECT RELATIONSHIP MAPPING

WEB SERVICE

COMPONENT MAPPING

ENTITY LIFE CYCLE

JAVA FULL STACK COURSE CURRICULUM

FRONTEND DEVELOPMENT

1. HTML4 and HTML 5

- **Introduction to HTML**
  - What is HTML, and why is it essential for web development?
  - Differences between HTML4 and HTML5
  - New web standards and evolution of HTML
- **Tags, Elements, and Attributes**
  - Inline vs. Block-level elements
  - Custom data attributes (data-*)
  - Global attributes in HTML5
- **Basic Syntax**
  - HTML document structure: <html>, <head>, <body>, and the importance of the DOCTYPE declaration
  - Valid and invalid HTML syntax examples
- **Tables**
  - Advanced table styling using CSS
  - Table layout techniques
  - Responsive tables
- **Lists**
  - Customizing lists with CSS
  - Nesting lists within other lists
- **Forms**
  - Form validation techniques in HTML5
  - New input elements in HTML5: date, number, email, tel, etc.
- **Semantic and Non-Semantic Tags**
  - The transition from HTML4 to HTML5
  - HTML5's new structural elements: <header>, <footer>, <article>, <section>, <nav>
  - Deprecation and removal of tags in HTML5
  - List of non-semantic tags and their usage (,)
- **HTML5 Features**
  - Web Storage (localStorage, sessionStorage)
  - Geolocation API
  - HTML5 Forms and Validation
- **Forms Attributes**
  - Formaction , formmethod(get,post)
  - HTML5 form validation attributes: pattern, required, minlength
- **Audio and Video Tags**
  - Implementing HTML5 audio and video
  - Handling media events and controls with JavaScript

2. CSS

- **CSS Selectors and Attributes (Style, Title)**
  - Overview of common CSS attributes
  - Types of Selectors
  - Differences between id and class selectors
- **CSS Types (Inline, Internal, External)**
  - Advantages and disadvantages of each CSS method
  - Linking external stylesheets and using inline styles
- **Box-Model**
  - Understanding the CSS box model (Content, Padding, Border, Margin)
  - Modifying box model properties to affect layout
- **Display Property (Block, Inline, None)**
  - The difference between block, inline, and inline-block elements
  - display: none vs visibility: hidden
- **Visibility-Hidden**
  - Practical usage of visibility property
  - Difference between visibility: hidden and display: none
- **Position Property (Static, Relative, Absolute, Fixed)**
  - Positioning elements: static, relative, absolute, fixed, and sticky
  - Practical examples of element positioning
- **Z-Index Property**
  - Layering elements using z-index
  - Importance of stacking context
  - Combinators
  - Descendant Selector (space), Child Selector (>), Adjacent Sibling Selector (+), General Sibling Selector (~)
  - Practical examples for combining selectors
- **CSS Pseudo-Classes**
  - Pseudo-classes like :hover, :focus, :nth-child, and :not
  - Styling interactive elements with pseudo-classes
- **CSS Pseudo-Elements**
  - Styling the first letter, first line, and content insertion using ::before, ::after
  - Selection and ::selection pseudo-element
- **Static Web Page**
  - Overview of static web pages and their applications
  - Building a simple static web page
- **Viewport Meta Tag**
  - Making websites mobile-friendly with the viewport meta tag
  - How the viewport affects layout on different screen sizes

3. Advanced CSS

- **Background and Multiple Backgrounds**
  Applying background colors, images, and gradients
  Layering multiple backgrounds
- **Font-Related Features**
  Using custom web fonts via @font-face and Google Fonts
  Font properties: font-family, font-size, font-weight, font-style, etc.
- **Text-Effect and Box-Effect**
  Text shadow, box shadow, text transform
  Adding 3D effects using CSS properties
- **Gradients (Linear and Radial)**
  Creating linear and radial gradients with CSS
  Practical uses for gradient backgrounds
- **Transition**
  Introduction to CSS transitions for smooth state changes
  Common transition properties: transition-property, transition-duration, transition-timing-function
- **Transformation**
  Using transform to rotate, scale, skew, and translate elements
  Combining transformations for advanced effects
- **Media Queries**
  Creating responsive layouts using @media queries
  Customizing designs for various screen sizes and devices

4. Bootstrap

- **Introduction to Bootstrap (Responsive)**
  The power of Bootstrap's grid system
  Importance of mobile-first design
- **Typography**
  Font styles, sizes, and alignments in Bootstrap
  Bootstrap typography classes
- **Tables**
  Advanced table designs using Bootstrap
  Responsive tables with .table-responsive
- **Images, Buttons**
  Using Bootstrap classes for styling images and buttons
  Responsive image classes: .img-fluid
- **Grid Structure and Types of Columns**
  Understanding the Bootstrap grid system: Rows, Columns, and Breakpoints
  Creating flexible layouts with col-xs, col-sm, col-md, and col-lg
- **Forms**
  Styling forms with Bootstrap's built-in form controls
  Using custom input groups and validation styles
- **Jumbotron**
  Using .jumbotron for prominent page sections
  Customizing jumbotrons for different screen sizes
- **Navbar and Nav Tabs**
  Building responsive navigation bars using Bootstrap's Navbar component
  Styling navigation tabs and pills
- **Carousel**
  Creating Bootstrap Carousels for image sliders
- **Responsive Web Page**
  Best practices for responsive design with Bootstrap
  How to make websites mobile-friendly using the grid system

5. JavaScript

- **Introduction to JavaScript**
  JavaScript in web development
  History and evolution of JavaScript
- **Use of JavaScript**
  JavaScript in interaction and behavior of websites
  Integrating JavaScript with HTML and CSS
- **Variables**
  Declaring variables using var, let, and const
- **Keywords**
  JavaScript reserved keywords and naming conventions
- **Data Types**
  Primitive types (String, Number, Boolean, Null, Undefined)
  Non-primitive types (Objects, Arrays, Functions)
- **JS Conditions**
  Using if, else, switch, and conditional operators
- **Loops**
  Using for, while, and do-while loops
  Loop control with break and continue
- **Functions**
  Defining and calling functions
  Function parameters and return values
- **Arrays:**
  Creating and accessing arrays.
  Array methods (push(), pop(), shift(), unshift(), map(), filter(), reduce(), etc.).
  Iterating through arrays using for, forEach(), map().
  Destructing of array
- **String and their method**
  Using backticks for multi-line strings and variable interpolation.

Spread and Rest Operators:

Spread (...) for arrays and objects.

Rest (...) for function parameters.

- **Objects:**

Creating objects: object literals, new Object().

Accessing object properties: dot notation, bracket notation.

Methods inside objects.

Destructing of object

- **SetTimeout and SetInterval**

Using setTimeout() and setInterval() for time-based actions

- **HTML DOM**

Introduction to the Document Object Model (DOM)

DOM methods for interacting with HTML elements

Selecting DOM Elements:

getElementByIId (), getElementsByClassName()

Changing text content (innerText).

Changing HTML content (innerHTML).

Changing attributes (setAttribute(), getAttribute()).

Adding/removing elements (createElement(),removeChild (),appendChild())

- **Event Handling:**

Adding event listeners (addEventListener()).

- **Asynchronous JavaScript:**

Understanding synchronous vs asynchronous code.

Callback functions.

Promises

Async/Await.

Error Handling: try, catch, finally.

- **JavaScript Classes and Prototypes:**

Introduction to ES6 Classes.

Constructor function, this in classes.

Inheritance and Polymorphism

6. React

- **Introduction to React**
- **What is React?**

Introduction to React: A component-based library for building user interfaces.

Understanding the component-based architecture and the importance of reusable components.

- **Installing React using Create React App**

Setup of a development environment with create-react-app.

Folder structure and basic setup.

- **JSX Syntax and Databinding**
- **What is JSX?**

Writing HTML inside JavaScript (React's syntax extension).

- **Embedding expressions, attributes, and class names in JSX**

Expressions in JSX: {} syntax for embedding JavaScript.

Using className for class attributes in JSX.

- **Components**
- **Functional Components vs. Class Components**

Key differences and when to use each type.

- **Props and State in React**

What are props? Passing data to components.

State in React: How to use state within components (useState hook).

- **Component Life Cycle**

Understanding lifecycle methods in class components (e.g., componentDidMount).

Using useEffect hook in functional components for side effects.

- **Component API and Props Validation**

Defining custom props and validation with PropTypes

- **Event Handling in React**
- **Handling user interactions:**

Handling events like onClick, onChange, and onSubmit.

Using functions to handle event-driven logic.

- **State Management**
- **State and Lifecycle Methods**

Deep dive into state management and how to update state in functional components using useState.

Context API

Managing global state without prop drilling using React Context API.

Passing data to components without manually passing props.

- **React Router**
- **Setting up React Router for navigation**

Installing and configuring react-router-dom for routing.

- **Creating Routes and Navigation Links**

Defining routes and linking components via the <Route> and <Link> components.

- **React Forms and User Input**
- **Controlled Components and UnControlled Components**

Managing form state using React's state management.

- **Form Submission, Validation, and Reset**

Handling form submission, validation, and resetting form fields.

- **React Hooks**
- **Using Built-in Hooks**

useState, useEffect, useContext, useMemo, useReducer, useRef.

- **Custom Hooks**
  Creating reusable logic using custom hooks.
- **Understanding useReducer and useContext**
  Handling more complex state management with useReducer and global state using useContext.
- **Advanced React Concepts**
  Error Boundaries
  Handling errors in React applications
  Higher-Order Components (HOCs)
  Understanding HOCs for reusing component logic.
  Virtual DOM
  Explanation of how React's virtual DOM works and improves performance.
- **State Management Libraries**
  Redux
  Introduction to Redux (Actions, Reducers, and Store).
  Integrating Redux with React to manage global state efficiently.
  State Management with Context API
  Understanding when to use Context API vs. Redux for state management.
- **Integrating GraphQL with React**
  What is GraphQL?
  Introduction to GraphQL and how it compares to REST APIs.
- **Working with HTTP Requests in React**
  Making HTTP Requests
  Sending GET and POST requests to external APIs.
  Adjusting request headers, handling responses, and using Axios or Fetch API.
- **Bootstrap for Styling**
  Basic Setup of Bootstrap
  Integrating Bootstrap with React for responsive UI design.
- **Authentication in React Apps**
  Introduction to JWT
  How JWT (JSON Web Tokens) works for user authentication.

7. TypeScript and Angular

- **Introduction to TypeScript**
  Overview of TypeScript and its benefits over JavaScript.
  Installation and setup of TypeScript.
  Configuring TypeScript with tsconfig.json.
- **Basic Syntax and Types**
  Variables, constants, and data types in TypeScript (string, number, boolean, array, tuple).
  Type inference vs. explicit typing.
- **TypeScript Functions**
  Defining functions and understanding return types.
  Function overloading and default parameters.
  Arrow functions and anonymous functions.
- **Classes and Objects**
  Classes, constructors, properties, and methods.
  Access modifiers (public, private, protected).
  Inheritance and method overriding.
  Static members and getter/setter.
- **Interfaces and Types**
  Defining and using interfaces in TypeScript.
  Optional properties, read-only properties, and method signatures in interfaces.
  Type aliases and differences with interfaces.
- **Overview of Angular**
  What is Angular? Why use it?
  Key features of Angular (dependency injection, modularity, directives, etc.).
  Installing Angular CLI and creating your first Angular project.
- **Angular Architecture and Structure**
  Understanding Angular application structure (modules, components, services).
  Bootstrapping Angular apps with main.ts and app.module.ts.
  Angular Lifecycle hooks.
- **Components in Angular**
  Creating components using ng generate component.
  Understanding the component lifecycle.
  ngOnInit, ngOnChanges , ngOnDestroy
- **Data Binding in Angular**
  Introduction to Data Binding
  Definition and Importance
  Types of Data Binding in Angular
  Interpolation Binding
  Property Binding
  Event Binding
  Two-way Binding
- **Directives in Angular**
  Introduction to Directives
  Definition and Purpose
  Types of Directives
  Structural Directives
  ngIf
  ngFor
  ngSwitch

ngContainer
Attribute Directives
ngClass
ngStyle
Custom Attribute Directives
Creating Custom Directives
Directive Definition and Lifecycle
ng-template

- **Creating and Communicating Between Components**

Creating your first data-bound component.
Using external templates.
Communicating with child components using @Input.
Communicating with parent components using @Output.
Using template variables to interact with child components.

- **Styling Components**

Exploring Angular's CSS encapsulation.
ng-bootstrap (add bootstrap in angular)
Angular material

- **Pipes**

Built-in pipes: date, currency, json, uppercase, etc.
Creating custom pipes.
Pipe chaining and pure vs impure pipes.

- **Angular Forms**

Creating template-driven forms in Angular.
Handling user input with form controls.
Form validation using built-in validators (required, minlength, etc.).
Introduction to reactive forms (model-driven forms).
Creating forms using FormGroup, FormControl, and FormArray.
Form validation with reactive forms.

- **Routing in Angular**

Introduction to Angular Routing.
Setting up routing in Angular apps.
Defining routes using RouterModule and app-routing.module.ts.
Route parameters and handling dynamic routes.

- **Creating and Using Services with HttpClientModule**

Creating Angular services with ng generate service.
Injecting services in components and other services.
Singleton services and scope of services.
HTTP Client in Angular.
Making GET, POST, PUT, DELETE requests.
Handling HTTP responses and errors.

- **Communicating with the Server Using HTTP, Observables, and RxJS**

Moving data storage to the server.
Listening to resolved data changes.
Using query string parameters.
Using query string parameters.

BACKEND DEVELOPMENT

1. Core Java

- **Object Oriented Programming Concepts**

Object
Characteristics of an Object
State
Behavior
Identity
Responsibility
Major Pillars
Abstraction
Encapsulation
'IS A' relationship – Inheritance
'HAS A' relationship – Containment
Polymorphism

- **Introduction to Java**

Features of Java
Java Compiler
Java Runtime Environment
Bytecode Verifier
Class Loader
JiT Compiler
JDK, JRE, JVM

- **Java Language Fundamentals**

Classes and Object
Class Syntax
Primitive Data Types
Implicit and Explicit Conversion
Constructor
Creating object
Memory Allocation
Garbage Collection
Method Overloading
Parameterized Constructor
this keyword
static keyword – variable & method

main method
Instance init block & Static init block
Array – 1D, 2D
Enhanced for loop - forEach loop
Variable Argument
Package
Importing from packages
static import

- **Containment, Inheritance and Polymorphism**

Implementing 'HAS A' relationship
Container Object and Contained Object
Implementing 'IS A' relationship
extends keyword to achieve Inheritance
Super class and Sub class
super keyword
protected keyword
Polymorphism
Method Overriding
Super class reference to create Sub class object
Reference Type Casting
Static Type and Dynamic Type of reference
Covariant return type
java.lang.Object class and it's methods – toString()

- **Abstract Class and Interface**

abstract keyword
Abstract class vs Concrete class
Abstract method vs Concrete method
final keyword – variable, method, class
final method vs abstract method
final class vs abstract class
Interface
Role based inheritance
Multiple Inheritance
implements keyword
Implementing interface in a class
Interface extends interfaces

- **Annotations and Enum**

Annotation
Creating user defined annotation
Use of annotation at different levels
Meta-annotation
java.lang.annotation.Annotation interface
Built-in Annotations - @Override, @SupressWarnings, etc
Enum
Creating user-defined Enum
Enum Constants
java.lang.Enum class

- **Functional-Style Programming**

Functional Interface
Lambda Expression
@FunctionalInterface
Method References
Built-in Functional interface
java.util.function package
Predicate, Consumer, Supplier, Function

- **Utility Classes (java.lang) & Inner classes**

Wrapper classes
Autoboxing & Autounboxing
String, StringBuffer, StringBuilder
String constant pool
Inner classes
Simple Inner class
Static nested class
Method Local Inner class
Regex

- **Exception Handling**

What is exception?
Type of Errors
Exception class hierarchy
Exception Handling Mechanism
try keyword
catch keyword
Checked and unchecked exceptions
throw vs throws keyword
finally block
multicatch
final re-throw
try-with-resources (ARM)
User-defined exception

- **Multithreading & Thread Synchronization**

Multitasking
Multiprocessing vs Multithreading
What is thread?
Thread Life Cycle
Creating thread using java.lang.Thread class
Creating thread using java.lang.Runnable interface
Thread class and it's methods
Inter thread communication
Thread Synchronization
synchronized keyword
synchronized method & synchronized block

- **Collection Framework & Generics**

Collection and it's types
java.util package

Generics Syntax
List – ArrayList, Vector
Set – HashSet, SortedSet, NavigableSet – TreeSet
Map – HashMap, SortedMap, NavigableMap – TreeMap
Hashtable and Properties class
Importance of equals() & hashCode() methods
Searching & Sorting Algorithm – Arrays & Collections classes
Comparable vs Comparator interfaces
Generic Collections

- **Stream API**

What is Stream?
Types of Stream – Sequential & Parallel Stream
java.util.stream package
Stream operations – intermediate and terminal operations
map(), reduce(), filter(), forEach(), limit(), skip() methods
Collectors – toCollection(), toList(), toSet(), toMap() methods

- **File IO & Object Serialization**

java.io package
File class & it's method
File Reading, Writing and Appending Operation
Byte Stream – FileInputStream & FileOutputStream
Character Stream – FileReader & FileWriter
BufferedReader & BufferedWriter, PrintWriter
Autocloseable
Using try-with-resources
Non-blocking IO (nio)
java.nio.file package
Path interface
Files class, Paths class
FileSystem
Object Serialization
ObjectInputStream & readObject() method
ObjectOutputStream & writeObject() method
transient keyword

- **Unit Testing**

Introduction
What is Testing?
Why Unit Testing?
Testing Terminologies
Junit Framework
Junit Annotations
Creating Test Cases

- **Design Pattern**

What is a Design Pattern?
Type of Design Patterns – Creational, Structural and Behavioral
Singleton Design Pattern
Factory Design Pattern
Facade Design Pattern

- **Java9 version features**
- **Java11 version features**

2. SQL

- **Introduction to SQL**

What is SQL?
RDBMS
Installation of MySQL Database
SQL Data Types
SQL Constraints
Connecting to MySQL
Creating Database
Creating Table in Database
Performing CRUD operations
INSERT Query
SELECT Query with WHERE clause
UPDATE Query
DELETE Query
Sorting rows using ORDER BY clause
Grouping rows using GROUP BY & HAVING clause
Performing Joins
Inner Join
Outer Join
Self Join
Cross Join

3. Adv Java

- **Maven**

What is Maven?
Features of Maven
Environment Setup
Build Life Cycle
Phases
Goals
Plugins
POM xml file
Dependency Management
Maven Dependencies
Maven Repositories
Maven Archetypes
Creating Maven Project using Eclipse

- **JDBC**

Java Database Connectivity
java.sql package

Connection interface
Driver interface
Types of JDBC Drivers
DriverManager class
Connecting to MySQL using JDBC API
Statement
PreparedStatement
CallableStatement
ResultSet
Types of ResultSet – Forward Only, Scrollable
ResultSetMetaData
DatabaseMetaData

- **Introduction to Web Technologies**

Distributed Architecture
Two-tier Architecture
Three tier (N-tier) Architecture
Web Application
Web Application Architecture
Web Container
Web Server
Web Browser
HTTP – Stateless Protocol
HTTP Methods – GET, POST, PUT, DELETE, etc
HTTP Message Format
Request Message Format – Request Header, Request Body
Response Message Format – Response Header , Response Body
Client Side Programming
Server Side Programming

- **Servlet**

Introduction to Servlet
Servlet API – Servlet, ServletConfig interfaces
GenericServlet
HttpServlet
Servlet Life Cycle
ServletContext vs ServletConfig
Servlet Configuration - web.xml
@WebServlet Annotation
Creating Servlet
Configuring Tomcat Server in Eclipse IDE
Deploying and Running Servlet on Tomcat via Eclipse

- **JSP**

Introduction to Java Server Pages
JSP API
JSP Life Cycle
Creating and Running JSP Page on Tomcat Server
Building Blocks of JSP
JSP Directives
JSP Implicit Objects
JSP Scripting Elements
Scriptlet
JSP Expression

- **Web Service**

What is Web Service?
Web Service vs Web Application
SOA – Service Oriented Architecture
Service Provider
Service Consumer
Types of Web Service
SOAP Web Service
RESTful Web Service
ROA – Resource, URI and Representation
Uniform Interfaces
Creating REST API
Running RESTful Web Service on Tomcat
Consuming REST API

4. Java Framework - Hibernate JPA

- **Object Relationship Mapping**

ORM Principle
Drawbacks of JDBC
Benefits of ORM
Paradigm Mismatch
Types of ORM Frameworks

- **Hibernate JPA Introduction**

What is Hibernate?
Features of Hibernate
Advantages of Hibernate
Hibernate in Layered Application Design
Hibernate Architecture
Components of Hibernate
Configuration
SessionFactory
Session and it's methods – get(), load(), save(), update(), delete()
What is JPA?
Hibernate and JPA

- **Simple Class Mapping**

Entity Class
POJO class and it's advantages
Hibernate JPA Mapping
Mapping POJO class with Database Table
JPA Annotations - @Entity, @Table, @Id, @Column

Hibernate Configuration
hibernate.cfg.xml file
Configuring JDBC Driver, JDBC URL
Hibernate Dialect
Creating Hibernate Application
Performing CRUD operations

- **Hibernate JPA Entity Life Cycle**

Transient State
Persistent State
Detached State

- **JPQL**

Java Persistent Query Language
From clause
Where clause
Order By
Select clause
Using Criteria

- **Component Mapping**

Mapping 'HAS A' relationship
Creating Contained class
Creating Container class
Using JPA Annotations - @Embedded, @Embeddable

- **Inheritance Mapping**

Types of Inheritance Mapping
Table Per Class Hierarchy
JPA Annotations - @Inheritance
Inheritance Strategy – InheritanceType.SINGLE_TABLE
@DiscriminatorColumn, @DiscriminatorValue
Table Per Concrete Class
Inheritance Strategy – InheritanceType.TABLE_PER_CLASS
Table Per Joined Subclass
Inheritance Strategy – InheritanceType.JOINED
@PrimaryKeyJoinColumn

- **Relationship Mapping**

One-to-One Mapping
@OneToOne, mappedBy
@JoinColumn
One-to-Many Mapping
@OneToMany
Many-to-One Mapping
@ManyToOne
Many-to-Many Mapping
@ManyToMany
@JoinTable

- **Hibernate Caching Mechanism**

First-Level Cache
Second Level Cache

5. Java Framework – Spring & Spring Boot

- **Introduction to Spring**

What is Spring?
Features of Spring Framework
IoC – Inversion of Control
DI – Dependency Injection
AOP – Aspect Oriented Programming

- **Spring Architecture**

Types of Spring Module
Spring IoC Container
Spring Core - BeanFactory
Spring Context - ApplicationContext
Spring AOP
Spring DAO - JdbcTemplate
Spring ORM - HibernateTemplate
Spring Web
Spring Web MVC

- **Spring Bean and Configuration**

Creating Spring Bean
Spring Bean Configuration
Xml Based Configuration
Annotation Based Configuration
Stereotype Annotation - @Component
@Configuration
@ComponentScan
JavaConfig - @Bean
Bean Scoping
Bean Scopes – singleton, prototype, request, session
@Scope Annotation

- **Spring Bean Life Cycle**

Spring Aware Interfaces
BeanNameAware
BeanFactoryAware
ApplicationContextAware
InitializingBean – afterPropertiesSet()
@PostConstruct
Custom Init Method
DisposableBean
Custom Destroy Method

- **Dependency Injection**

Constructor Injection
constructor-arg
Setter Injection
property
Bean Autowiring
@Autowired
@Qualifier

Method Injection
Collection Injection
List Injection
Set Injection
Map Injection
Properties Injection

- **Spring Boot**

Introduction to Spring Boot
Features of Spring Boot
Spring Boot Starter Dependencies
@SpringBootApplication
AutoConfiguration
Creating Spring Boot Application
Spring Initializr – https://start.spring.io/
STS - Spring Tool Suite
Selecting starter dependencies

- **Spring Data**

What is Spring Data?
Features of Spring Data
Spring Data JDBC
Repository interface and it's hierarchy
Creating custom Data Repository interface
Performing CRUD Operations
Custom methods
@Query, @Param
Spring Data JPA
JpaRepository interface
Creating custom Data JPA Repository interface
Performning CRUD Operations
Customer methods
JPQL Queries
Transaction Configuration

- **Spring Web**

Spring Web MVC
MVC Architecture
Spring Web Architecture
DispatcherServlet
HandlerMapping
ViewResolver
Controller - Request Handler
@Controller Stereotype Annotation
@RequestMapping

- **RESTful Web Service**

@RestController
@GetMapping
@PostMapping
@PutMapping
@DeleteMapping
Creating REST API

- **Spring Security**

Authentication
Authorization
Roles
Permissions
Features of Spring Security
Authentication Models
Password Encoders
User Store - Types
OAuth2

- **Spring Microservices:**

Overview of Microservices Architecture
Definition of Microservices
Benefits and challenges
Monolithic vs. Microservices architecture
API Gateway Pattren
Service discovery tools (e.g., Eureka)
Resilience and Fault Tolerance
Circuit Breaker Pattern

6. DevOps

- **Introduction to DevOps**

Git and Github
CI/CD tools with jenkins
Docker and kubernetes
Sonarcube

7. AWS Cloud

- **Introduction to Cloud Computing**

What is Cloud computing?
Characteristics of Cloud Computing
Cloud Service Models
Saas (Software as a Service)
Paas (Platform as a Service)
Iaas (Infrastructure as a Service)
Cloud Deployment Models – Private Cloud, Public Cloud, Hybrid & Community
Cloud Service Providers – AWS, Microsoft Azure, GCP (Google Cloud Platform)

- **AWS Cloud**

Overview of AWS
Features of AWS
Advantages of using AWS
Brief Introduction of AWS Service
AWS IAM
AWS S3
AWS EC2

8. Flutter

- **Introduction**
- What is Flutter?
- Mobile App Development
- Features of Flutter
- Benefits of Flutter in Android App Development
- Advantages of Flutter
- What is Dart Programming?
- **Brief Introduction of Flutter Programming Elements**
- Widgets
- UI Components
- Animations

9. React Native

- **Introduction**
  - What is React Native?
  - Features of React Native in Mobile App Development
  - Importance of React Native in both Android and iOS development
- **React Native Components**
  - React Native View
  - State
  - Props
  - Style

10. Kotlin

- **What is Kotlin?**
- **Role of Kotlin in Android App Development**
- **Important Features of Kotlin**
- **Advantages of using Kotlin in Android App Development**
- **Kotlin Programming Elements – Functions, Classes, etc**

PROJECT LISTS

- **E-commerce Web Application**
- **Online Learning Platform**
- **Social Media Platform**
- **Job Portal**
- **Booking System (Hotel, Flights, or Events)**
- **Fitness Tracker Web App**

MASTER IN JAVA FULL STACK

SevenMentor PVT LTD

YOUR DREAM JOB IS AHEAD

COURSE EXAM

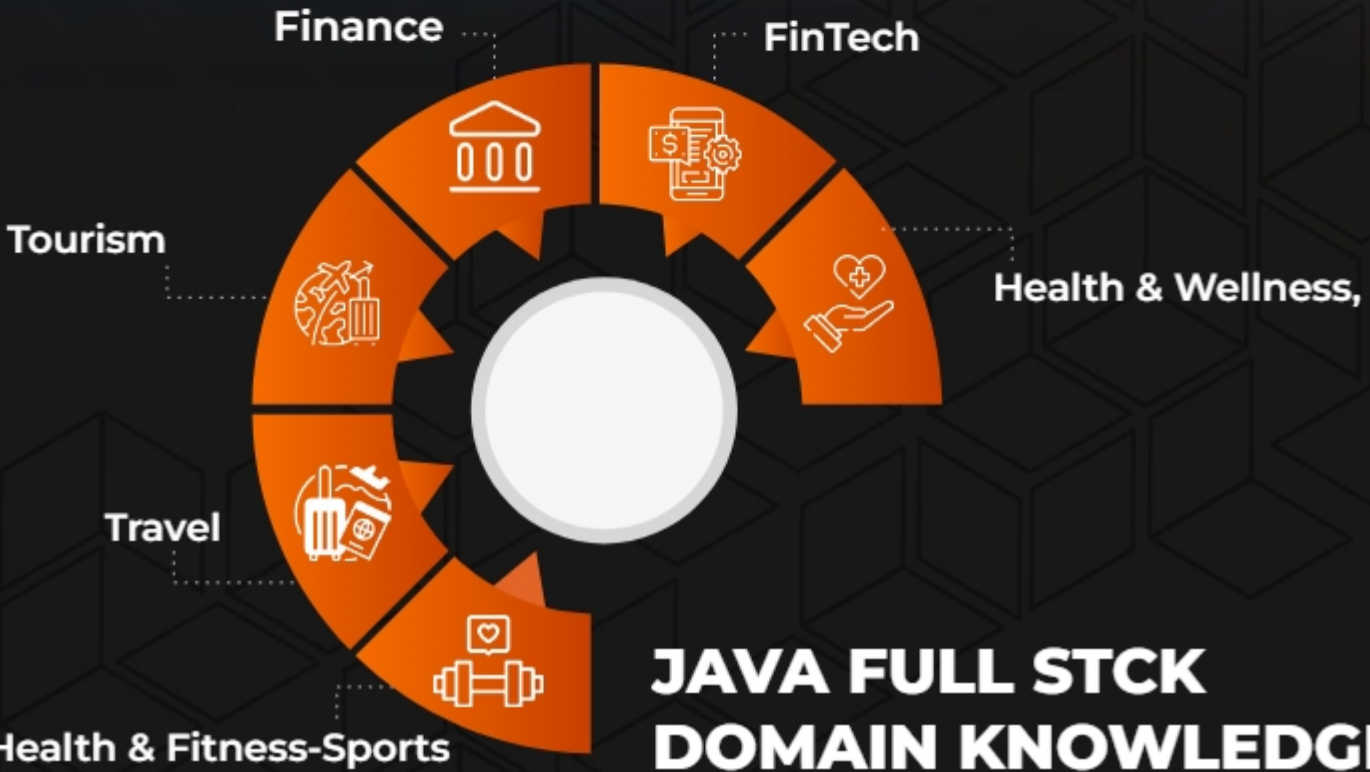
CASE STUDY

ADVANCE SKILLS

INDUSTRY EXPERT TRAINING

PROFILE BUILDING

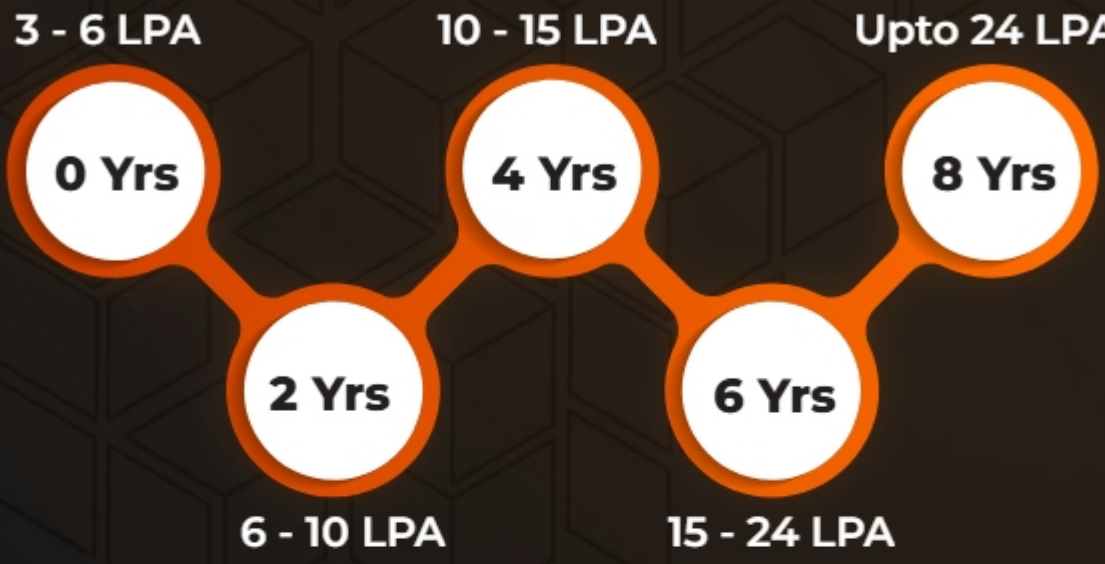
ENROLL IN SEVENMENTOR



JAVA FULL STCK DOMAIN KNOWLEDGE

YOUR FUTURE AS A JAVA FULL STACK

SALARY RANGE



JOB METER

Full Stack Java Developer 1,22,911 Jobs	Senior Java Web Full Stack Developer 1,10,239 Jobs	Java Full Stack Developer 2,31,300 Jobs	Backend Java Developer 2,12,462 Jobs
Software Developer 71,499 Jobs	Application Developer (Java Full Stack) 2,18,644 Jobs	Java Cloud Full Stack Developer 59,914 Jobs	Full Stack Lead 80,039 Jobs
	Solution Architect (Full Stack Java) 1,12,399 Jobs	SDET – Java Stack 40,953 Job	

JOB ROLES FOR JAVA FULL STACK

SENIOR FULL STACK ENGINEER (JAVA)	JUNIOR FULL STACK SOFTWARE DEVELOPER (JAVA)	FULL STCK DEVELOPER	FULL STACK JAVA WEB APPLICATION ENGINEER
JAVA FULL STACK DEVELOPER	JAVA UI DEVELOPER	JAVA FULL STACK ENGINEER	JAVA ENTERPRISE FULL STACK DEVELOPER

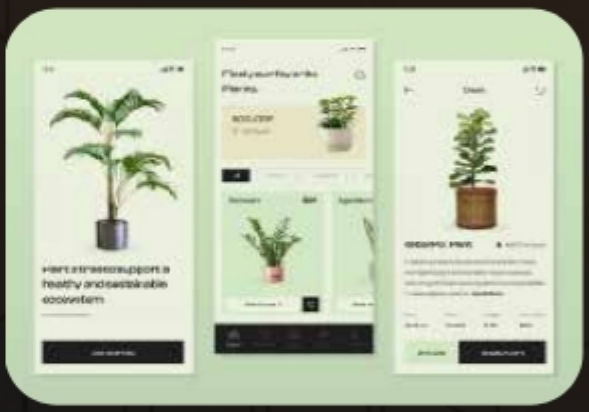
INDUSTRY BASED PROJECT

Hands-on industry-based project by SevenMentor & Training to enhance practical skills and real-world expertise

PETPAL APP

BUDGETFLOW APP

TRAVELMATE (TRAVAL PLANNING PLATFORM)



GREEN THUMB (GARDENING AND PLANT CARE APP)

PERSONAL FITNESS TRACKING APP

WHY CHOOSE SEVENMENTOR

Empowering Careers with Industry-Ready Skills

- Flexible EMI Plans
- Industry Experienced Trainers
- Live Projects With Hands-on Experience
- Real World Topics
- Flexible Scheduling
- Online, Classroom & Hybrid Batches
- 1:1 Mentorship when required
- Affordable Programs Tailored to Your Needs
- Class Recordings for Missed Classes
- Workshops & Seminars with Industry Experts
- Adaptive LMS
- Free Wifi Facilities
- Be Different With Master Certificate
- Placement Drives
- Interview Calls Assistance & Mock Sessions
- Weekday and Weekend Batches
- Ongoing Career Support
- Latest Market Tech & Practical Training
- 1 Year FREE Repeat Option
- Unlimited Interview Calls

TESTIMONIALS OF OUR STUDENTS

The software testing course at SevenMentor Training Institute, Pune, is well-structured and practical. The trainers are highly experienced and explain concepts in a simple, easy-to-understand way. Hands-on projects and real-world scenarios make the learning very effective. The institute provides excellent support and placement assistance. Overall, it's a great place to learn and grow in the field of software testing.

Nilesh Lipane

The course materials and resources provided were excellent. SevenMentor really cares about their students' learning experience and this puts their students at a tremendous advantage. I have extensive experience due to learning at SevenMentor and now I'm able to work on my own UJ designs. 👍Completing the UI/UX course at SevenMentor gave me a significant career boost. I was able to land a job shortly after finishing the UI UX certification course from SevenMentor institute. I urge everyone to join this course if there are serious about their career.🙏

Chetan Sawant

I recently joined the DevOps batch at SevenMentor, and I'm so glad I did. The trainers were amazing—they really knew their stuff and made even tough topics easy to understand. The course was hands-on, which helped me learn tools like Jenkins, Docker, and Kubernetes in a practical way.
Thanks to this training, I feel confident about applying DevOps practices in real-world scenarios. If you're looking for solid training in DevOps, I'd definitely recommend SevenMentor

Vikram Marag

Excellent Training & Placement Support
I am happy to share that I got placed successfully! The trainers were highly knowledgeable and provided in-depth explanations, making even complex topics easy to understand. The hands-on projects and real-world scenarios helped me gain confidence in my skills.
Overall, my experience with SevenMentor was excellent. I highly recommend it to anyone looking for quality training with placement support.

Shravani Pawar

I had a great experience with SevenMentor Pvt Ltd's Java, Angular2S, and MEAN Stack classes. The instructors were knowledgeable, and the courses offered a perfect blend of theory and hands-on practice. The content was comprehensive, and the real-world projects helped solidify my understanding. Overall, it's an excellent choice for anyone looking to advance their full-stack development skills

Ansh Puri

Seven mentor is the best training institute for full stack training in pune. I recommend this center for everyone who looking for practical based training. I had completed Java full stack classes from seven mentor Pune .It was such a amazing experience .I am very happy and satisfied with the training provided by trainers right now. I am getting interviews calls as well thanks team Seven mentor

Sanjyoti Choudhari

GLOBAL CERTIFICATION FROM SEVENMENTOR

SevenMentor provides globally recognized certification programs designed to help individuals enhance their knowledge, skills, and career prospects. Our certifications validate the competence and expertise in various fields, including IT, business management, digital marketing, and more. Achieving a SevenMentor certification offers a competitive edge and ensures your capability in your chosen domain.

WHY SHOULD YOU JOIN THIS PROGRAM?

Our 100% Job Assurance Program guarantees that you will secure a job as a data scientist. With access to over 500 leading partner companies, we ensure you have ample opportunities to land your ideal role.

The program is structured to provide comprehensive training, equipping you with the technical skills, real-world experience, and industry insights needed to excel in the field of data science. Additionally, you'll receive ongoing support, including job placement assistance and career coaching, to help you navigate the hiring process. With our program, you're fully prepared to succeed in the competitive job market and launch your data science career.

PLACEMENT TEAM

Our placement staff is committed to your IT career success. With strong industry connections, market insights, and specialized assistance, we prepare you for a competitive job market and assist you in achieving your objectives. We offer individual resume development, mock interviews, and hands-on career coaching to help you stand out to top recruiters. Our dedication extends beyond placements; we provide you with the confidence and skills you need to succeed in your chosen field. At SevenMentor, your success is our priority!

CRM TEAM

At SevenMentor, our CRM team is committed to ensuring that every student has a positive and enjoyable experience. We attentively listen to your issues, answer your inquiries quickly, and provide personalized service that is suited to your specific requirements. Our team takes a student-first approach, ensuring that your needs are handled with professionalism, kindness, a nd attention to detail. Your satisfaction is our first priority, and we commit ourselves to supporting you every step of the way.

GET INFO OF OUR OTHER DOMAINS

020 7117 2515
Networking Department

020 7117 3035
Development Department

020 7117 1500
Data Science And Ai Department

020 7117 1747
Softskills & Language Department

020 7117 3116
SAP Department

020 7117 1757
Interior Design & Fashion Department

020 7117 1727
HR Department

020 4855 6661
Placement Department

LOCATE US:

SHIVAJI NAGAR HEAD BRANCH
09/A Wing, Shreenath Plaza,Dnyaneshwar Paduka Chowk, F.C Road,Shivaji Nagar, Pune, Maharashtra – 411005

GET THE DIRECTION

Pimpri Chinchwad Branch
Office number 38 wing A and B, 3rd Floor, KUNAL PLAZA off Mumbai Pune Highway Chinchwad, Maharashtra 411019

GET THE DIRECTION

Hadapsar Branch
Manisha Blitz, Office 34-35, 3rd floor, Near Shankarmath, Pune Solapur Hwy, Hadapsar, Pune - 411013

GET THE DIRECTION

www.sevenmentor.com

registration@sevenmentor.com