



Best Software Training and Placement Institute

- + 91 90000 08814
- www.livetechn.in
- info@livetechn.co.in
- livetechnqa/ livetechnqa/
- livetechnweet

418,4th floor, Nilgiri Block, Adithya Enclave, Ameerpet Hyderabad-500008

Full Stack JAVA



Modules

- Core JAVA 8
- My SQL Database
- Advance JAVA
- Spring Boot
- Angular 16
- HTML 5 CSS 3 with Bootstrap, Java Script
- Project Development

Core Java 8 Content

♣ Declarations and Access Control

- Identifiers & JavaBeans
- Legal Identifiers
- Sun's Java Code Conventions
- JavaBeans Standards
- Declare Classes
- Source File Declaration Rules
- Class Declarations and Modifiers
- Concrete Subclass
- Declaring an Interface
- Declaring Interface Constants
- Declare Class Members
- Access Modifiers
- Nonaccess Member Modifiers
- Constructor Declarations
- Variable Declarations
- Declaring Enums

♣ Object Orientation

- Encapsulation
- Inheritance, Is-A, Has-A
- Polymorphism
- Overridden Methods
- overloaded Methods
- Reference Variable Casting
- Implementing an Interface
- Legal Return Types
- Return Type Declarations

- Returning a Value
- Constructors and Instantiation
- Default Constructor
- Overloaded Constructors
- Statics
- Static Variables and Methods
- Coupling and Cohesion

♣ Assignments

- Stack and Heap—Quick Review
- Literals, Assignments, and Variables
- Literal Values for All Primitive Types
- Assignment Operators
- Casting Primitives
- Using a Variable or Array Element That Is Uninitialized and Unassigned
- Local (Stack, Automatic) Primitives and Objects
- Passing Variables into Methods
- Passing Object Reference Variables
- Does Java Use Pass-By-Value Semantics?
- Passing Primitive Variables
- Array Declaration, Construction, and Initialization
- Declaring an Array
- Constructing an Array
- Initializing an Array
- Initialization Blocks
- Using Wrapper Classes and Boxing
- An Overview of the Wrapper Classes
- Creating Wrapper Objects

- Using Wrapper Conversion Utilities
- Autoboxing
- Overloading
- Garbage Collection
- Overview of Memory Management and Garbage Collection
- Overview of Java's Garbage Collector
- Writing Code That Explicitly Makes Objects Eligible for Garbage Collection

♣ **Operators**

- Java Operators
- Assignment Operators
- Relational Operators
- instance of Comparison
- Arithmetic Operators
- Conditional Operator
- Logical Operators

♣ **Flow Control, Exceptions**

- if and switch Statements
- if-else Branching
- switch Statements
- Loops and Iterators
- Using while Loops
- Using do Loops
- Using for Loops
- Using break and continue
- Unlabeled Statements
- Labeled Statements
- Handling Exceptions
- Catching an Exception Using try and catch
- Using finally
- Propagating Uncaught Exceptions
- Defining Exceptions
- Exception Hierarchy
- Handling an Entire Class Hierarchy of Exceptions
- Exception Matching
- Exception Declaration and the Public Interface
- Rethrowing the Same Exception
- Common Exceptions and Errors

♣ **Strings, I/O, Formatting, and Parsing**

- String, StringBuilder, and StringBuffer
- The String Class
- Important Facts About Strings and Memory
- Important Methods in the String Class

- The StringBuffer and StringBuilder Classes
- Important Methods in the StringBuffer and StringBuilder Classes
- File Navigation and I/O
- Types of Streams
- The Byte-stream I/O hierarchy
- Character Stream Hierarchy
- RandomAccessFile class
- The java.io.Console Class
- Serialization
- Dates, Numbers, and Currency
- Working with Dates, Numbers, and Currencies
- Parsing, Tokenizing, and Formatting
- Locating Data via Pattern Matching
- Tokenizing

♣ **Generics and Collections**

- Overriding hashCode() and equals()
- Overriding equals()
- Overriding hashCode()
- Collections
- So What Do You Do with a Collection?
- List Interface
- Set Interface
- Map Interface
- Queue Interface
- Using the Collections Framework
- ArrayList Basics
- Autoboxing with Collections
- Sorting Collections and Arrays
- Navigating (Searching) TreeSets and TreeMaps
- Other Navigation Methods
- Backed Collections
- Generic Types
- Generics and Legacy Code
- Mixing Generic and Non-generic Collections
- Polymorphism and Generics

♣ **Threads**

- Defining, Instantiating, and Starting Threads
- Defining a Thread
- Instantiating a Thread
- Starting a Thread
- Thread States and Transitions
- Thread States
- Preventing Thread Execution

- Sleeping
- Thread Priorities and yield()
- Synchronizing Code
- Synchronization and Locks
- Thread Deadlock
- Thread Interaction
- Using notifyAll() When Many Threads May Be Waiting

♣ Lambda Expressions

- Introduction
- Writing Lambda Expressions
- Functional Interfaces
- Types of Functional Interfaces
- Method reference

♣ Stream API

- Introduction
- Stream API with Collections
- Stream Operations

MySQL Database:

♣ Introduction

- The Relational Model
- What is MySQL?
- SQL – Data Types

♣ Understanding Basic MySQL Syntax

- The Relational Model
- Basic SQL Commands - SELECT
- Basic SQL Commands - INSERT
- Basic SQL Commands - UPDATE
- Basic SQL Commands – DELETE

♣ Querying Data with the SELECT

Statement

- Wildcards (% , _)
- The SELECT List
- SELECT List Wildcard (*)
- The FROM Clause
- How to Constrain the Result Set
- DISTINCT and NOT DISTINCT

♣ Row functions

- String
- Number
- Date and time
- Olap

♣ Filtering Results with the Where Clause

- WHERE Clause
- Boolean Operators
- The AND Keyword
- The OR Keyword
- Other Boolean Operators between, like, in, is, is not

♣ Shaping Results with ORDER BY and GROUP BY

- ORDER BY
- Set Functions
- Set Function And Qualifiers
- GROUP BY
- HAVING clause

♣ Matching Different Data Tables with JOINS

- Table Aliases
- CROSS JOIN
- INNER JOIN
- OUTER JOINS
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN
- SELF JOIN
- Natural Join

♣ Creating Database Tables

- CREATE DATABASE
- CREATE TABLE
- NULL Values
- PRIMARY KEY
- CONSTRAINT
- ALTER TABLE
- DROP TABLE

♣ MySQL Transactions

- BEGIN, COMMIT, ROLLBACK

♣ MySQL Constraints

- CHECK, UNIQUE, NOT NULL

♣ MySQL Procedure, Functions

Advanced Java:

♣ JDBC:

- Introduction to JDBC
- JDBC Architecture
- Types of JDBC Drivers
- Establishing a JDBC Connection
- Integration Java to Database using JDBC API
- Using Statement
- Using PreparedStatement
- Using CallableStatement
- Using ResultSetMetaData
- Using DataBaseMetaData
- Establish ResultSet
- ScrollableResultSet
- Batch Updates
- Inserting BLOB&CLOB type of columns
- Managing Transaction

♣ JPA with Hibernate 3.0

♣ Introduction

- Introduction & overview of data persistence
- Overview of ORM tools
- Understanding JPA
- JPA Specifications

♣ Entities

- Requirements for Entity Classes
- Persistent Fields and Properties in Entity Classes
- Persistent Fields
- Persistent Properties
- Using Collections in Entity Fields and Properties
- Validating Persistent Fields and Properties
- Primary Keys in Entities

♣ Managing Entities

- The EntityManager Interface
- Container-Managed Entity Managers
- Application-Managed Entity Managers
- Finding Entities Using the EntityManager
- Removing Entity Instances
- Synchronizing Entity Data to the Database
- Persistence Units

♣ Querying Entities

- Java Persistence query language (JPQL)
- Criteria API

♣ Entity Relationships

- Direction in Entity Relationships
- Bidirectional Relationships
- Unidirectional Relationships
- Queries and Relationship Direction
- Cascade Operations and Relationships

Spring Boot

♣ Spring Boot Introduction

- Spring Boot starters, CLI, Gradle plugin
- Application class
- @SpringBootApplication
- Dependency injection, component scans, Configuration
- Externalize your configuration using application. Properties
- Context Root and Management ports
- Logging

♣ Using Spring Boot

- Build Systems, Structuring Your Code, Configuration, Spring Beans and Dependency Injection, and more.

♣ Spring Boot Essentials

- Application Development, Configuration, Embedded Servers, Data Access, and many more
- Common application properties
- Auto-configuration classes
- Spring Boot Dependencies

♣ Spring Data JPA

- Spring Data JPA Intro & Overview
- Core Concepts, @RepositoryRestResource
- Defining Query methods
- Query Creation
- Using JPA Named Queries
- Defining Repository Interfaces
- Creating Repository instances
- JPA Repositories
- Persisting Entities
- Transactions

♣ Spring Data REST

- Introduction & Overview
- Adding Spring Data REST to a Spring Boot Project
- Configuring Spring Data REST

- Repository resources, Default Status Codes, Http methods
- Spring Data REST Associations
- Define Query methods

♣ Git, Maven, Junit

♣ Project1 : Using Springboot and Hibernate

HTML 5, CSS 3 with Bootstrap, JavaScript

♣ Contents: HTML 5:

♣ HTML Basics

- Understand the structure of an HTML page.
- New Semantic Elements in HTML 5
- Learn to apply physical/logical character effects.
- Learn to manage document spacing.

♣ Tables

- Understand the structure of an HTML table.
- Learn to control table format like cell spanning, cell spacing, border

♣ List

- Numbered List
- Bulleted List

♣ Working with Links

- Understand the working of hyperlinks in web pages.
- Learn to create hyperlinks in web pages.
- Add hyperlinks to list items and table contents.

♣ Image Handling

- Understand the role of images in web pages
- Learn to add images to web pages
- Learn to use images as hyperlinks

♣ Frames

- Understand the need for frames in web pages.
- Learn to create and work with frames.

♣ HTML Forms for User Input

- Understand the role of forms in web pages
- Understand various HTML elements used in forms.
- Single line text field

- Text area
- Check box
- Radio buttons
- Password fields
- Pull-down menus
- File selector dialog box

♣ New Form Elements

- Understand the new HTML form elements such as date, number, range, email, search and datalist
- Understand audio, video, article tags

♣ CSS3 :

♣ Introduction to CSS 3.0

- What CSS can do
- CSS Syntax
- Types of CSS

♣ Working with Text and Fonts

- Text Formatting
- Text Effects
- Fonts

♣ CSS Selectors

- Type Selector
- Universal Selector
- ID Selector
- Class selector

♣ Colors and Borders

- Background
- Multiple Background
- Colors RGB and RGBA
- HSL and HSLA
- Borders
- Rounded Corners
- Applying Shadows in border

♣ BootStrap

♣ Introduction to Bootstrap

- Introduction
- Getting Started with Bootstrap

♣ Bootstrap Basics

- Bootstrap grid system
- Bootstrap Basic Components

♣ Bootstrap Components

- Page Header
- Breadcrumb
- Button Groups
- Dropdown
- Nav & Navbars

♣ JavaScript Essentials

♣ ES6 & Typescript

- Var, Let and Const keyword
- Arrow functions, default arguments
- Template Strings, String methods
- Object de-structuring
- Spread and Rest operator
- Typescript Fundamentals
- Types & type assertions, Creating custom object types, function types
- Typescript OOPS - Classes, Interfaces, Constructor, etc

Angular 16

♣ Introduction to Angular Framework

- Introduction to Angular Framework, History & Overview
- Environment Setup, Angular CLI, Installing Angular CLI
- NPM commands & package.json
- Bootstrapping Angular App, Components, AppModule
- Project Setup, Editor Environments
- First Angular App & Directory Structure
- Angular Fundamentals, Building Blocks
- MetaData

♣ Essentials of Angular

- Component Basics
- Setting up the templates
- Creating Components using CLI
- Nesting Components
- Data Binding - Property & Event Binding, String Interpolation, Style binding

- Two-way data binding
- Input Properties, Output Properties, Passing Event Data

♣ Templates, Styles & Directives

- Template, Styles, View Encapsulation, adding bootstrap to angular app
- Built-in Directives, Creating Attribute Directive
- Using Renderer to build attribute directive
- Host Listener to listen to Host Events
- Using Host Binding to bind to Host Properties

♣ Pipes, Services & Dependency

Injection

- In-built Pipes, Creating a Custom Pipes
- Services & Dependency Injections
- Creating Data Service
- Understanding Hierarchical Injector

♣ Components Deep Dive / Routing

- Component Life Cycle Hooks
- Reusable components in angular using <ng-content>
- Navigating with Router links
- Understanding Navigation Paths
- Navigating Programmatically
- Passing Parameters to Routes
- Passing Query Parameters and Fragments
- Setting up Child (Nested) Routes
- Outsourcing Route Configuration (create custom module)

♣ Http Requests / Observables

- HTTP Requests
- Sending GET Requests
- Sending a PUT Request
- Using the Returned Data
- Catching Http Errors
- Basics of Observables & Promises

Final Project

ADDRESS

#418, 4th Floor, Nilgiri Block, Aditya Enclave, Beside Metro Station,
Ameerpet, Hyderabad-500038, Mobile: +91- 9000008814

<https://www.livetechnology.com>

<https://www.facebook.com/livetechnology>