



JAVA FULLSTACK CONTENT

Duration: 120 Hours

0731-4069788,7805063968

Java Full-Stack

- ❖ Explain JAVA Technology, Java Short History, Comparison of JAVA from C/C++
- ❖ Type of Application of JAVA
- ❖ JAVA 10 Features ,How to download JAVA Software Tools
- ❖ SCOPE of JAVA
- ❖ Set Environment of JAVA(Temp Path | Permanent Path)
- ❖ First Hello Word Program of JAVA
- ❖ Data type using Primitive type and Wrapper class
- ❖ Operator and Conditional Statements (if, if—else, nested, else if, multiple if)
- ❖ Control Structure and Switch Statement, Nested Switch Statement
- ❖ Array (One dimension, Multidimensional, Jagged, Object)
- ❖ String class, String Buffer Class, String Builder Class
- ❖ OOP's Introduction, Class and Object, Type of Class, Inner class, Anonymous object, Generic Class, Data abstraction, Encapsulation, Polymorphism, Constructor, Garbage collector, Inheritance, Abstract class, Interface, Super Keyword, this Keyword
- ❖ Command Line input
- ❖ Input By System.in.read ()
- ❖ Exception Handling (Checked , Unchecked) Explain throw, throwable
- ❖ File Handling (File, File Reader, File Writer, Buffered Reader, Buffered Writer, FileInputStream, FileOutputStream)
- ❖ Serialization and Deserialization
- ❖ Multithreading, explain Thread, Runnable, Sleep, Wait, notify, and notify all, Thread Synchronization
- ❖ Create JAR file by Command
- ❖ Create Package by Command

JAVA 10 Features Implementation

- ❖ Time Based Release Versioning
- ❖ Local-Variable Type Inference
- ❖ Experimental JAVA-Based JIT Compiler
- ❖ Application Class-Data Sharing
- ❖ Parallel Full GC
- ❖ Garbage Collector Interface
- ❖ Heap Allocation



HIBERNATE Framework

- ❖ Hibernate Introduction
- ❖ Mapping and configuration files
- ❖ Create database and tables using hibernate.
- ❖ Add record into database using Hibernate
- ❖ POJO Class life cycle on hibernate
- ❖ Hibernate Select Query, Delete Query, Update Query
- ❖ Versioning on Hibernate
- ❖ Mapping on Hibernate
- ❖ Generators in Hibernate
- ❖ HQL in hibernate
- ❖ Criteria in Hibernate
- ❖ Projection concept in hibernate
- ❖ How to write native SQL query in hibernate
- ❖ Annotation in hibernate, relationship in depth
- ❖ Join in Hibernate
- ❖ Cache in hibernate, first level and second level cache

SPRING BOOT WITH JPA, WEB,API

- ❖ What is MICROSERVICES
- ❖ What is Spring boot, difference between Spring core and Spring MVC
- ❖ Spring boot starters,
- ❖ Spring boot annotations
- ❖ What is API, Restful API, how to create using Spring boot]
- ❖ What is application.proprties file
- ❖ How to change server port of tomcat
- ❖ How to reload changes without restarting the server
- ❖ JDBC + MYSQL configuration using SPRING Boot
- ❖ How to configure multiple data source
- ❖ How to deploy Spring boot on external server
- ❖ Spring boot + Spring Security
- ❖ How to configure cache in Spring Boot
- ❖ CREATE REST API and CHECK WITH POSTMAN
- ❖ LIVE PROJECT IMPLEMENTATION

Angular Framework

- ❖ Explain Angular, Angular Short History, Comparison of Angular JS and Angular Framework
- ❖ Angular JS 1.0 Overview
- ❖ Angular architecture, Angular application flow
- ❖ Comparison of Different Version of Angular
- ❖ Angular CLI to create project ,component and module
- ❖ Angular Architecture,MVC Design pattern
- ❖ What is Type Script, Data type in Angular Typescript
- ❖ Operator and Conditional Statements
- ❖ Control Structure or Loop, Array ,String Concept
- ❖ Angular Routing
- ❖ Create multiple Components and develop single page application
- ❖ Data binding in Angular
- ❖ Directives in Angular ,Pipe in Angular
- ❖ Dependency Injection and services in Angular
- ❖ File System in Angular,Read,Write.Open other operation
- ❖ Forms in angular, Template driven, reactive
- ❖ Http Request using GET,POST,PUT,DELETE
- ❖ Authentication, Create Login Form
- ❖ Modules, Predefine modules, Custom Module
- ❖ Animation,HTML Template Integration, query integration
- ❖ Validation using Angular Directive
- ❖ Web API implementation in Angular ,Create Login,Registration,CRUD
- ❖ State management in Angular
- ❖ Create Admin dashboard using Angular