

Java/Spring boot/Microservice with MySQL

Duration 12 Days

Module	Coverage (Day 1)
Introduction to Java	Installation and Environment Setup Features of Java 8-16 Hands on
	Day 2
Java Inbuilt Packages	java.lang Wrapper classes Autoboxing and Unboxing Object class Shallow Cloning – overview and example Use of equals and hashCode Understanding toString() method String, String Buffer and String Builder java.io Overview of Byte Streams and Character Streams Serialization java.util Collections Framework List – ArrayList, LinkedList, Set – HashSet, LinkedHashSet, TreeSet Map – TreeMap, ConcurrencyMap Iterator, ListIterator Sorting using Comparator and Comparable Method References – overview and example Properties, UUID, Random – overview and example Calendar, Locale – overview and example
	Day 3
Java.nio, Java.time Channels & Selectors	Executor Interfaces Thread Pools Using Callable and Future Overview of Java NIO Channels, Buffers & Selectors Channel Implementations FileChannel Usage of Buffer Buffer Types Allocating a buffer Writing data to a buffer Reading data from a buffer Scattered Reads & Gathering Writes Channel to Channel Transfers Usage of Selectors Creating a selector Registering channels with the selector Selecting Channels via a selector Using AsynchronousFileChannel

	Reading data through Future and Completion Handler Writing data through Future and Completion Handler java.time Why java Date/Time API LocalDate and LocalTime Time zones and time stamps Temporal Adjustments Period and Duration Formatting and parsing Conversion between different date/time objects
	Day 4
J2SE + Additional Frameworks - LOMBOK, SL4J, Log4J	What is logging? Introduction to SL4J When to use SL4J? Binding with different logging frameworks Using SL4J with logback logger What is Logback? Architecture & Configuration Logger, Appenders and Layouts Parameterized logging Logging Separation Design Patterns & Principles Overview of Design Pattern Overview of creational, structural, behavioral, concurrency design pattern Example for Singleton Design Pattern only SOLID Design Principles Introduction to the SOLID design principles Violations of SOLID design principles Introduction to Code Smells Refactoring techniques Introduction to DRY, YAGNI
	Day 5
Database	Concepts of Database RDBMS Concepts (MySQL) Introduction to RDBMS Overview of Database Models Overview of ER Diagram and Normalization Introduction to SQL Data types Introduction to DQL, DDL, DML, DCL CRUD operations with database INSERT/UPDATE/DELETE/RETRIEVE Using functions and ordering the result Average(), Count(), Maximum(), Median(), Minimum(), Mode(), Sum() Group By, Order By, Having Use of LIKE & WHERE clause Dropping / truncating a table SQL Joins - Inner, Outer and Self joins Sub queries – single row, multiple row, correlated Database Views

	<p>Introduction to views</p> <p>Scenarios when views are used</p> <p>Variance and standard deviation</p>
	Day 6
Expressions, Collections & Streams	<p>Regular Expressions</p> <p>java.util.stream</p> <p>What is a stream?</p> <p>Collections vs Streams</p> <p>Creating Streams</p> <p>Different types of streams</p> <p>Intermediate Operations on Streams</p> <ul style="list-style-type: none"> o map(), filter(), sorted(), flatMap() <p>Terminal Operations on Streams</p> <ul style="list-style-type: none"> o reduce(), forEach(), findFirst(), match(), collect(), count() <p>Converting a Stream to a collection or Array</p> <p>Numeric Streams – IntStream, DoubleStream, LongStream</p> <p>java.util.concurrent</p> <p>Overview of concurrency API</p> <p>Executors</p>
	Day 7
Hibernate basics, ORM framework, CRUD examples	<p>Hibernate Basics</p> <p>Introduction to ORM framework</p> <p>ORM & ORM Frameworks</p> <p>Features of Hibernate</p> <p>Installation and Environment Setup</p> <p>Hibernate Architecture Overview</p> <p>Configuration and Session Factory</p> <p>Session</p> <p>Transaction</p> <p>CRUD Examples</p> <p>Persisting data</p> <p>Loading data into an object</p> <p>get Vs load</p> <p>Deleting, updating & finding objects</p> <p>Lifecycle of Hibernate</p> <p>Mapping Association (using annotations only)</p> <p>Mapping OneToOne</p> <p>Mapping OneToMany</p>
	Day 8
Spring IOC with Springboot	<p>Spring IOC with SpringBoot</p> <p>Introduction to Spring Framework</p> <p>Overview of Spring framework</p> <p>Dependency Injection (DI)</p> <p>What is Spring?</p> <p>Spring Architecture</p> <p>Spring Container, IoC, DI</p> <p>Dependency Injection (DI) in Spring, DI Configuration</p> <p>BeanFactory Interface and ApplicationContext</p> <p>Spring Beans</p> <p>Creating a Spring Application using Spring Boot</p>

	<p>Overview of Spring Boot Installation and System Requirements</p> <p>Spring Boot Starter packages</p> <p>Using spring boot to create ready to run spring application</p> <p>Structuring the code</p> <p>Using @EnableAutoConfiguration, @ComponentScan, @Configuration</p>
	Day 9
BEAN lifecycle	<p>Use of @SpringBootApplication annotation</p> <p>Packaging as a Jar</p> <p>What are Setter Based and Constructor Based DI?</p> <p>Annotation Based Configuration</p> <p>Setter Based and Constructor Based</p> <p>Factory Methods, Definition Inheritance (Parent Beans)</p> <p>Bean Inheritance</p> <p>Use of @Component, @Value</p> <p>Collection Valued Properties</p> <ul style="list-style-type: none"> Configuring and using Array, List, Map, Set and Properties <p>Inner Bean</p> <p>Bean Scopes</p> <p>Autowiring of Dependencies using byName, byType, constructor</p> <ul style="list-style-type: none"> Autowiring using @Autowired, @Qualifier <p>Use of @PostConstruct and @PreDestroy</p> <p>Using PropertyPlaceholder external property file</p> <p>Bean LifeCycle</p> <p>InitializingBean and DisposableBean</p> <p>Creating custom lifecycle methods</p> <p>Java based Configuration</p> <p>Understanding @Configuration</p> <p>Using @Bean</p> <p>Autowiring</p> <p>Database Access with Spring</p> <p>Introduction to Spring DAO Support</p> <p>Queries and Inserts</p> <p>Additional API Capabilities</p> <p>Create a REST Application using SpringBoot</p>
	Day 10
Spring MVC	<p>Introduction to Spring Web MVC</p> <p>Spring MVC Architecture</p> <p>MVC Architecture</p> <p>DispatcherServlet</p> <p>Creating a spring application using Spring Boot</p> <p>Use of Controllers, View Resolvers</p> <p>Stereotypes: @Component, @Service, @Controller, @Repository</p> <p>Handling Exceptions</p> <ul style="list-style-type: none"> HandlerExceptionResolver Controller based ExceptionHandler Global Exception Handler <p>Handling Sessions</p> <p>Validation Framework</p> <p>Spring Validation API</p>

	JSR 303 Validation API Integration of Spring MVC with Hibernate Spring with JPA Repositories Consuming a REST Service Consume Spring REST Service from Spring MVC
	Day 11
Microservices	Spring Microservices Introduction to Microservices What is Microservices? Why Microservices? Characteristics of Microservices Patterns in Microservices Architecture
	Day 12
Spring Cloud	Microservices using Spring Cloud Components Eureka Service Discovery API Gateway - Routing, Filtering Circuit Breaker - Resilience4j