**Java and Spring Boot Micro Services**

**Pre-Requisites:** Participants should be familiar with programming fundamentals.

**Duration: 18 Days**

* Course Outline: Mapped to Java 8 with Brief Intro to Java 17

**Software Requirements and Download Links**

**Java - 8**

* <https://aws.amazon.com/corretto/?filtered-posts.sort-by=item.additionalFields.createdDate&filtered-posts.sort-order=desc>
* <https://www.openlogic.com/openjdk-downloads>

**Java -17**

* https://download.oracle.com/java/17/archive/jdk-17.0.12\_windows-x64\_bin.zip

**MySQL :**

* <https://dev.mysql.com/downloads/>

**Maven :**

* https://maven.apache.org/download.cgi

**Postman :**

* https://www.postman.com/downloads/

**Spring Tool Suite 4.x :**

* https://spring.io/tools

**Apache Kafka :**

* https://kafka.apache.org/downloads

**GIT :**

* <https://git-scm.com/download/win>

**Core Java – 5 Days**

**Introduction to Java**

* Class Structure
* Primitives
* Constructors
* Local variables
* Instance,static and local variables
* Garbage collection
* Method overloading
* Arrays
* Packages and libraries in Java
* JavaDocs

**Controlling Program Flow**

* If-else for loop
* While loop
* Do-while loop
* Switch statement

**Reusing Classes**

* Composition
* Inheritance
* Final

**Polymorphism**

* Abstract classes and methods
* Interfaces
* Polymorphism

**Error/Exception Handling**

* Introduction to Exception and Errors
* Create a try Block and catch Block
* Catch Multiple Exception Errors
* Create a finally Block
* Runtime exceptions and non-Runtime Exceptions implication on overriding

**Generics**

* Why use generics
* Generic Types, raw types
* Generic methods
* Collections,List,Set,Map,Arrays
* Iterators
* Sorting using Comparable and Comparator
* Choosing best collection
* Equals method

**Working with Databases**

* Introduction to JDBC
* Types of Drivers in JDBC
* Connect to Database
* Create a prepared Statement
* Using a Parameters in a Prepared Statement
* Creating and Retrieving Data from a Result Set
* Format Data for Display
* Position the Cursor in the Result Set
* CRUD Operations
* Make a Batch Update
* Connection Pooling

**Java 8 – 3 Days**

**What's new in Interfaces**

* Static Methods
* Default Methods
* Functional Interfaces

**Introducing Lambda Expressions**

* Motivation for Lambdas
* Lambda Expression Overview
* Lambda Expressions and Functional Interfaces
* Using Lambda Expressions
* Working with Method References

**The Stream API**

* Method references
* java.util.Function
* The Stream API
* Chaining - Intermediate and Terminal Operations
* Important Functional Interfaces: Predicate, Comparator, Function
* Stream Processing
* Using filter(),map()
* Sorting / Comparator.comparing()
* Result producing terminal operations - collect, min/max ...
* Terminal Operations - forEach(),toArray() and Collect
* Existence and Finder Operations
* Collectors.toList(), Collectors.toSet()
* Supplier and Collectors.toCollection()
* Partitioning and Grouping Collectors

**Parallel Processing and Concurrency**

* Java 8 Parallel Processing using Streams
* Multithreading / Concurrency Issues
* Race Conditions, Data Synchronization, Locks
* Performance Issues and Considerations

**Date/Time API**

* Overview and Limitations of Previous API
* The Date/Time API (JSR 310)
* Formatting Date/Time

**Introduction to Java 17**

* Introduction to Java 17
* Sealed Classes
* Records in Java
* Text Blocks

**Introduction to Build Tools**

* Maven Basics
* Maven Dependency Management
* Maven Lifecycle
* Maven Packaging and Distribution

**Spring – 5 Days**

**Spring Framework**

* Spring In Context – Core Concepts
* Bean management through IOC
* Bean Creation
* Constructor Injection
* Setter Injection

**Spring Boot Internals**

* Using @EnableAutoConfiguration, @ComponentScan, @Configuration
* Auto-Configuration
* Use of @SpringBootAppilcation annotation
* External Configuration
* Profiles and Logging
* Packaging Spring Application

**Data Access with Spring Boot**

* Spring Data JPA & Rest
* Spring Data JPA: The Data Tier
* @Repository Annotation
* Introduction to JPA
* Adding Spring Data JPA
* Creating a Spring Data JPA Repository
* Making Crud Operations with Repository
* Adding Entity Relationship and Extending Repository
* Service Layer and DTOs
* Exception Handling
* Using RestTemplate and Spring's WebClient for making API calls

**Spring MVC**

* Dispatcher Servlet
* MVC Controllers
* Model Interface
* Request Param
* Form Tag Library
* Thyme leaf Template
* Validation

**Microservices – 5 Days**

**Introduction to Microservices**

* Problems with Monolithic Application
* What and Why Microservices?
* Advantages and disadvantages of microservices
* Key characteristics: loose coupling, independent deployment, and domain-driven design.
* Overview of Spring cloud and its modules

**Spring Cloud Eureka**

* Introduction to Discovery Server
* How to Create a Eureka Server
* How to register Eureka Client
* How to look up a service from Eureka Server
* Configuring a cluster of eureka servers

**Spring Cloud Config**

* Configuring Server and client pointing to configuration file in GitHub
* @EnableConfigServer
* HTTP, resource-based API for external configuration
* @RefereshScope

**Spring Cloud Load balancer**

* Need for Client-Side Load Balancing
* The @LoadBalanced Annotation
* ReactiveLoadBalancer

**Spring Cloud Gateway**

* **API Gateway with Spring Cloud Gateway**
  + The role of an API Gateway in a microservices architecture.
  + Setting up a gateway to handle routing, authentication, and other cross-cutting concerns.
  + Configuring routes to our microservices.
* **Resilience and Fault Tolerance**
  + Understanding the Circuit Breaker pattern.
  + Implementing a Circuit Breaker using a library like Resilience4j to prevent

**Spring Security and Oauth**

* **Introduction to Spring Security**
  + Review of core security concepts: Authentication and Authorization.
* **Securing Microservices with OAuth2 and JWT**
  + Setting up a simple OAuth2 authorization server.

**Spring Cloud Stream**

* Need for a Spring Cloud Stream
* Introduction to Message Broker
* Apache Kafka
* **Asynchronous Communication with Messaging**
  + Introduction to event-driven architecture and messaging queues.
  + Using Spring Cloud Stream to build event-driven microservices