14 days plan for Java/Spring/Angular Training

Pre-requisites for attending this training:

- Basic knowledge of any programming language (like C or C++)
- Good understanding of how to use SQL commands on RDBMS tables
- During the training, the students will be given online resources for self learning, and students are expected to learn few topics by making use of the given resources.

Software setup required in participant's PC/Laptop:

- Java development kit Version 8 (JDK 1.8) 64 bit
- MySQL (preferred) Version 8
- Eclipse for Java EE developers
- Microsoft Visual Studio Code (https://code.visualstudio.com)
- Node JS (latest version LTS download, must include additional developer tools)
- Apache Tomcat 9 (ZIP file)
- Open internet connection is required for downloading project dependencies during training via Maven

Detailed day wise breakup:

Day 1:

- Introduction to Core Java Programming and Concepts
  - A First Look
  - A Simple Java Class
  - Java's "Hello World" Program
  - Java Basics
  - Language and Platform Features
  - Program Life Cycle
  - The Java SE Development Kit (JDK)
  - Working with the Development Environment
- Class and Object Basics
  - The Object Model and Object-Oriented Programming
  - Classes, References, and Instantiation
  - Garbage Collection
  - Adding Data to a Class Definition
  - Adding Methods (Behavior)
- Packages
  - Package Overview - Using Packages to Organize Code
  - import statements
  - Creating Packages, package Statement, Required Directory Structure
  - Finding Classes, Packages and Classpath
- More on Classes and Objects
  - Accessing data, the "this" variable
  - Encapsulation and Access Control, public and private Access
  - Constructors and Initialization
  - static Members of a Class
  - Scopes, Blocks, References to Objects

Day 2:

- Composition and Inheritance
  - Using Composition to Deal With Complexity
  - Composition/HAS-A, Delegation
  - Using Inheritance and Polymorphism to share commonality
  - IS-A, extends, Inheriting Features, Overriding Methods, Using Polymorphism
  - Class Object
  - Abstract Classes
- Interfaces
  - Using Interfaces to Define Types
  - Interfaces and Abstract Classes
  - Default Methods and static Methods (Java 1.8 or later only)
  - Using Interfaces to Remove Implementation Dependencies

Day 3:

- Exceptions
  - Exceptions and the Exception Hierarchy
  - try, catch and finally
  - Handling Exceptions
  - Program Flow with Exceptions
  - Creating user defined exceptions and exception funneling
- I/O Streams
  - Readers and Writers
  - Filter Streams
  - Byte Streams
  - Formatted Output

Day 4:

- Java Collections and Generics
  - The Collections Framework and its API
  - Collections and Java Generics
  - Collection, Set, List, Map, Iterator
  - Auto boxing
  - Collections of Object (non-generic)
  - Using ArrayList, HashSet, and HashMap
  - Processing items with an Iterator
  - More about generics

Day 5:

- Database Access with JDBC
  - JDBC Overview
  - JDBC Architecture
  - Drivers and types of drivers
  - DriverManager,
  - Connection,
  - Statement, PreparedStatement, CallableStatement
  - ResultSet

Day 6:

- Web Application Basics
  - How the Web works, Thin Clients, TCP/IP
  - HTTP overview, Brief HTML review
  - Overview of Java EE, servlets & Web applications.
- Servlet Basics

    - Servlet API
    - HTML Forms
    - HTTP: Request-response, headers, GET, POST
- Overview: How Servlets Work
    - Servlet Lifecycle: init(), service(), destroy()
    - Requests and responses
- Core Servlet API
    - GenericServlet
    - ServletRequest, and ServletResponse
- HTTP Servlets
    - HttpServlet
    - HttpServletRequest and HttpServletResponse
    - Accessing Parameters
- JSP Overview
    - JSP architecture
    - JSP tags and JSP expressions
    - Fixed Template Data
    - Lifecycle of a JSP
    - JSP Expression Language (EL)
    - The JSTL

Day 7:

- Spring overview and architecture
    - What is Spring framework?
    - Why Spring framework?
    - Spring framework architecture
    - Usage scenario
    - Step by step refactoring of Helloworld application using Dependency Injection
- Spring Dependency Injection Basics
    - What is and Why Dependency Injection (DI)?
    - Two DI variants
    - Reading configuration
    - Bean configuration
    - Bean parameter types
    - Auto-wiring and auto-scanning
    - Bean naming
- Spring Dependency Injection Annotation
    - Annotation-based Dependency Injection -@Autowired, @Required
      Qualifier - @Qualifier, Custom qualifier
    - @Component and further stereotyped annotations - @Service, @Repository, @Controller, @ComponentScan

Day 8:

- Spring Database Introduction
    - DAO support
    - @Repository annotation
    - Using HibernateTemplate to build a DAO layer
- Spring Transaction
    - Transaction management in Spring framework
    - Global transaction vs. local transaction
    - PlatformTransactionManager interface

- Declarative transaction management
- Transaction propagation

Day 9:

- Spring MVC
  - Role of DispatchServlet, HandlerMapping, ViewResolver
  - Simple Spring MVC application demo
  - Creating a RESTful service using Spring MVC

Day 10:

- Spring Boot
  - What is and Why Spring Boot?
  - Getting started with Spring Boot
  - Building a Web app using Spring Boot
  - Auto-configuration
  - SpringApplication class
  - External configuration
  - Actuator
  - Misc. features

Day 11:

- Basics of HTML and CSS
  - HTML elements and attributes
  - Most common tags – headings, p, div, span
  - Bullets and numbers (lists)
  - Working with tabular data
  - Forms
  - Styling basics – CSS inline, internal and external
  - Selectors, properties and values
  - Importing external CSS
  - Using Bootstrap CSS framewor

Day 12:

- Introducing Angular
  - What is Angular?
  - Angular versions
  - Advantages of Angular
  - Architecture overview
  - Introduction to key building blocks of Angular
  - Setting up Development Environment
  - Setting up Angular
  - Creating an app using Angular CLI
  - Setting up Bootstrap for styling
  - How an Angular app gets loaded and started?
- Components & Templates
  - What is a Component? What are its benefits?
  - The Root component
  - What are Decorators?
  - Understanding the component decorator
  - Creating and using components
  - Component templates
  - Component styles

    - Lifecycle Hooks
- Data Binding
    - What is Data Binding?
    - Interpolation
    - Property binding
    - Event binding
    - Passing and using event data
    - Two-way data binding
    - Component interaction
    - Parent to child interaction
    - Child to parent interaction
- Directives
    - Understanding Directives
    - ngIf – Outputting data conditionally
    - ngStyle – Styling elements dynamically
    - ngClass – Applying CSS classes dynamically
    - ngFor – Outputting lists
- Pipes
    - Introduction to Pipes
    - Using pipes
    - Parameterizing pipes
    - Chaining multiple pipes
    - Creating custom pipes

Day 13:

- Building Single Page Apps using Routing
    - Need for a Router
    - Setting up and loading routes
    - Navigating with router links
    - Styling active links
    - Navigating programmatically
    - Passing parameters to routes
    - Fetching route parameters
    - Passing query parameters
    - Retrieving query parameters
    - Setting up nested routes
- Services & Dependency Injection
    - Need for a Service
    - Creating a service
    - Understanding Dependency Injection (DI) and its benefits
    - Using a service within a component
    - Using a service within another service
    - Cross component interaction using a service
- Server Communication
    - Introduction to Angular HttpClient service
    - Sending requests to server
    - Getting data from the server
    - Sending data to the server
    - Handling Http errors
    - Adding headers

    - URL parameters

Day 14:

- Forms & Validation
    - Template-driven forms vs Reactive forms
    - Building a form
    - Registering form controls
    - Submitting the form
    - Understanding form state
    - Adding form validation
    - Outputting Validation Error messages
    - Using two-way binding
    - Grouping form controls
    - Resetting forms