Java from beginner to Professional

**Beginner**

Duration: 42 Hrs

# Environment setup (2 Hrs)

1. Installing and configuring Java 11
2. Practicing code without an IDE
3. Installing and IDE and using it
4. **Practice.**

# Reviewing basic programming concepts (8 Hrs)

1. Data types, variables and assignments
2. Operators and keywords
3. Inputs and outputs
4. Conditions
5. Iterations
6. Arrays
7. Methods
8. **Practice.**

# Understanding Object Oriented Concepts and (Inheritance + Encapsulation) (4 Hrs)

1. Theory of thinking in Object Oriented approach
2. Basic discussion of the OOP conceptual terminologies
3. Inheritance (IS-A relationship) & Aggregation (HAS-A relationship)
4. Encapsulation
5. **Practice.**

# Polymorphism (4 Hrs)

1. Method Overloading / Static binding / Compile time polymorphism
2. Method Overriding / Dynamic Binding / Runtime polymorphism
3. Overriding cases while implementing interfaces
4. **Practice.**

# All about Interfaces and Abstraction (4 Hrs)

1. Understanding Abstraction
2. Use cases of Abstraction
3. When to use Interface vs Abstract Class
4. **Practice**

# String Classes (2 Hrs)

1. String Literal vs using new keyword
2. Immutable Strings
3. Using the methods of String class
4. String Buffer & String Builder
5. Comparison of String classes
6. **Practice**

# Exception Handling (2 Hrs)

1. Exception vs Error
2. Checked Exception / Compile time exception
3. Unchecked Exception / Runtime Exception
4. Creating own exception classes

# Collection Framework (8 Hrs)

1. Knowing the classes & interfaces of collection framework
2. Using the most common collection frameworks
3. Knowing the best practices and use cases of several collection framework classes

# Enums and Annotations (4 Hrs)

1. Why enums
2. Using predefined enums
3. Creating own enums and observing use cases
4. WHY annotations
5. Use predefined annotations
6. Declare and use your own annotation

# JDBC API (4 Hrs)

1. Conceptual understanding and knowing the necessity of JDBC API
2. Installing a database server
3. Downloading and using driver class for the database server
4. Writing and executing database SQL queries with the JDBC API
5. Working with result set

# **Advanced**

Duration: 44 Hrs

# Generics (4 Hrs)

1. Why Generics: Benefits
2. Components of Generics
3. Use cases
4. Where generics are not possible

# Reviewing Annotations and learning Reflection API (4 Hrs)

1. Write custom annotations and access metadata
2. Reflection API understanding and some examples

# Creating a basic CDI tool with Annotations and Reflections (4 Hrs)

1. Building a basic CDI tool with Annotation and Reflection API by setting some problem statements and solving them

# Solid Principles (2 Hrs)

1. Understanding and code demonstrations

# Design Patterns: learning the common ones (10 Hrs)

1. General discussion and categorization
2. Singleton
3. Builder
4. Decorator
5. Factory and Abstract factory
6. Prototype
7. Proxy
8. Mediator
9. Façade
10. Observer
11. MVC

# Concurrency and multithreading (20 Hrs)

1. Understanding of Thread
2. Knowing and using thread classes and interface to create a simple thread
3. Creating, starting and stopping a thread, Thread lifecycle
4. Java memory model
5. Discussing visibility problem and its solution, discussing volatile keyword
6. Discussing race condition and its solution, discussing synchronized block
7. Discussing about ThreadLocal
8. Discussing concurrency and parallelism
9. Thread Pools in Java
10. Discussing about locks – deadlocks and live locks, prevention techniques
11. Java ExecutorService
12. BlockingQueue
13. Producer and consumer pattern in java
14. Compare and Swap for designing concurrent algorithms
15. False Sharing in Java
16. Single threaded and same threaded designs
17. Thread Congestion in Java

# **Spring By Practical Examples**

Duration: 86 Hrs

# Introduction to Servlet and JSP (4 Hrs)

1. Introduction to Servlet and Servlet Container
2. Configuring Servlet Application
3. Implementing HttpServlet and working with different HTTP Methods
4. View rendering with Servlet and JSP
5. Simple form data handling with Servlet and JSP
6. Making WAR file and deploying to tomcat
7. **Practice.**

# Reviewing the basics of networking (2 Hrs)

1. Understanding OSI layers
2. Accessing the application from different devices under same router

# Understanding Docker and dockizing the Hello Servlet app (6 Hrs)

1. Understanding docker concepts and why we need it
2. Introduction to common commands
3. Building docker image for the developed application
4. Running application from inside docker container and accessing from localhost

# Spring HelloWorld Application (4 Hrs)

1. Downloading application server (Tomcat)
2. Configuring tomcat server with Intellij Idea
3. MVC architecture in Spring
4. Understanding IoC container
5. What is a Bean, how to work with it
6. View Resolver Bean definition, working with different view resolvers.
7. @Component and its stereotype annotations and @Configuration annotations
8. **Practice.**

# Accessing Form Data with Spring (4 Hrs)

1. Introduction to Taglibs.
2. Form Taglib, core Taglib, and other taglib examples
3. Accessing the form data and rendering those data to page
4. Storing the form data in a list-based storage, rendering them in the page
5. **Practice.**

# Validating Form Data with JSR 380 Specification (2 Hrs)

1. Understanding Java validation framework and its needs.
2. Implementing validation for the form data processing.
3. **Practice.**

# Understanding JavaScript and JQuery (8 Hrs)

1. Understanding Client-side business logic rendering and manipulation with JavaScript.
2. Practicing JavaScript with real world examples.
3. Converting javascript examples with JQuery.
4. Handling different types of events and event listeners.
5. Learning Simple Ajax requests.
6. **Practice.**

# Making a simple CRUD (4 Hrs)

1. Installing PostgreSQL locally
2. Integrating JDBC Template to make a CRUD (Connecting with a PostgreSQL DB)
3. Setting up different profiles (Dev, QA, Prod etc)
4. Setting up 3 database instances for different profiles on docker
5. Accessing application data from docker hosted database from Java application
6. **Practice.**

# Learning and using docker compose (2 Hrs)

1. **Practice.**

# Making advanced CRUD (8 Hrs)

1. Integrating and configuring Hibernate Session Factory, Transaction Managers etc.
2. Performing the CRUD with Hibernate
3. Using Criteria API
4. **Practice.**

# Integrating an UI Template (2 Hrs)

1. Configuration to render Static contents
2. Adding CSS, JS, images and other static contents
3. Collecting data from Form and Rendering them into a page
4. **Practice.**

Understanding AoP (4 Hrs)

1. Understanding concepts of AoP (Aspect, JoinPoint, PointCut, Advice)
2. Implementing AoP with demonstration of a simple security checker

# Applying basic security (4 Hrs)

1. Integrating and Configuring Spring security
2. Discussing Role-Based Access Control feature (RBAC) in Spring framework
3. Configuring Login, Log Out, default error pages
4. Integrating custom filter and custom interceptor with spring security
5. **Practice.**

# Applying advanced security (4 Hrs)

1. Defining Custom Authentication Provider
2. Defining Custom Authentication Success handler, Custom Logout handler
3. Using Password encryption algorithms
4. Role Based Menu access handling
5. **Practice.**

# Project deployment to Heroku (4 Hrs)

* Configuring heroku deployment environment

# A template integrated web project development by participants (1) (8 Hrs)

* Grooming, development and evaluation

# Understanding and creating RESTful API (2 Hrs)

1. Understanding GET, POST, PUT, DELETE, FETCH methods
2. Understanding HATEOS
3. Understanding the benefits and drawbacks of making API call over browser loading
4. Developing APIs and testing with Postman
5. Making HTTP requests with Ajax and/or fetch API
6. **Practice.**

# Exception Handling and implementing Exception Handler to handle exceptions before Requests & Responses (2 Hrs)

1. Implementing Global Exception Handler to prettify and well format the exception messages.
2. **Practice.**

# Understanding the security measures of RESTful API (4 Hrs)

1. Understanding the differences between Web and API security
2. Different approaches to secure an API
3. Discussing the JWT and OAuth 2 Security principles

# Converting old project concept/developing a new project with API and Angular (8 Hrs, Depending on Participants)

* Redesigning and redeveloping old project concept to Api driven single page application