Programme on ‘**Core Java with SpringBoot- Backend framework’** Skillset

# Programme Schedule Model:

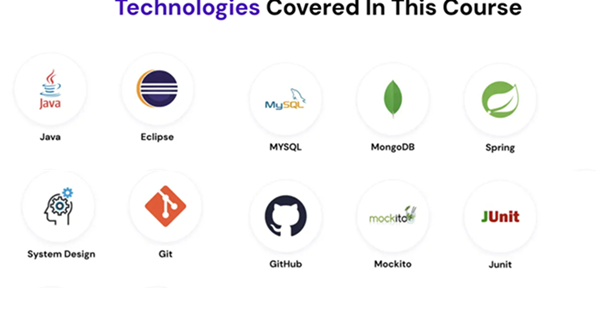
* Total Mentor-led VILT topics-session Duration : 105 Hours
* Practice Duration: 180 hours
* Total Duration for self-paced Course(s): 18 hours

# Instructor: Industry Mentors

* Pre & Post Assessment (via AI-powered Assessment platform)
* Mock Assessments (1 each in the last 3 weeks of the programme)
* Session-wise Quiz challenges (for better interactions & knowledge sharing)
* Hands-on practice with use cases

# Programme Deliverables:

* Consolidated Learners’ performance report on each skillset
* Certificates for completed Participants
* ZEN Query portal for ad hoc doubt clarification by the participants, out of the session (TAT: same day; Clarification types: email/chat/gmeet)

****

**GUVI’s Exclusive Practice Platforms:**

|  |  |
| --- | --- |
|  | **CodeKata :** [**https://www.guvi.in/code-kata/**](https://www.guvi.in/code-kata/)  A comprehensive toolkit designed to enhance coding skills and thoroughly prepare you for success in technical interviews. |
|  | **Webkata**: <https://www.guvi.in/webkata.html>  A cloud-based module designed to refine your front-end skills, eliminating the need for complex local environment setup. |
|  | **SQLKata:** <https://www.guvi.in/sqlkata/sql/>  an interactive environment that enables you to write, execute, and test SQL queries, helping you sharpen your database skills without the need for complex setups. |
|  | **IDE :** <https://www.guvi.in/ide/>  GUVI IDE is an Integrated Development Environment (IDE) that allows you to write, edit, run, test, and debug your code seamlessly in one platform. |

# Table of Contents: Java Full-Stack Development

# Core Java Programming (Mentor-led session Duration: 24 Hours)

* Intro to Java Platform & Language
* JVM, JRE, JDK
* JVM Architecture
* Data types - Primitive, Arrays
* Operators
* Branching (if, switch)
* Looping (while, for)
* OOPs in Java
* Classes, fields, methods, constructors
* Keywords this, super & Modifiers
* Interfaces & Inheritance
* Method overloading & Overriding
* Abstract classes
* Packages
* Access modifiers
* Exception Handling
* Collections
* Comparable & Comparator
* File Handling
* Memory management & Garbage collection
* Collections Framework (List, Set, Map, Queue)
* Generics
* MultiThreading

# Java 8 & latest (Mentor-led session Duration: 18 Hours)

* Intro to Functional programming
* lambda expressions
* Functional interfaces
* Method references
* Optional Class
* Streams API
* Data & Time API
* Callable & Future Interfaces
* Completable Future & Completion Stage
* Brief intro Java 9-17 features
* Collectors API & Immutable Collections
* Var keyword
* File APIs
* Switch expressions
* Text blocks
* Records
* Logging: Log4j framework utilization
* Debugging: How to Debug in IDE, Launch/Attach Breakpoints/Conditional Breakpoints/Logpoints Exceptions Pause & Continue Step In/Out/Over Variables Callstacks Threads Debug console Evaluation Hot Code Replace

# Spring IOC & Beans (Mentor-led session Duration: 21 Hours)

* Spring MVC
* Intro to Spring Boot Web (Servlet) stack
* Simple REST service using Spring Web
* RESTful API Development with Spring Boot
* Swagger API
* Exception Handling
* Data Access Layer with Spring Data JPA
* Optimistic & Pessimistic locking
* Security with Spring Security (Authentication and Authorization)
* Intro to Spring Cloud
* Declarative Service to service communication
* Spring Security & JWT
* Intro to Spring Boot Reactive stack
* Reactive Streams & Reactor
* Data access with R2DBC
* Spring Messaging (JMS)
* Unit testing & Remote Debugging
* Spring Core (Inversion Of Control & Dependency Injection)
* ORM Concepts
* AOP (Aspect-Oriented Programming)

# Web Applications & Services (Mentor-led session Duration: 6 Hours)

* Web application architecture
* Monolithic Vs Microservices architecture
* Microservices with Spring Boot - Service Registry and Discovery
* Microservices with Spring Boot - API Gateway
* Microservices with Spring Boot - Load Balancing
* Introduction to Microservice Communication in Spring Boot
* Intro to REST
* Intro to Java Containers & Servlets
* IOC & Dependency Injection
* Blocking & Non-blocking web stacks
* Spring 5

# Version Control – Git (Mentor-led session Duration: 3 Hours)

* Introduction
* Versioning, staging & un-staging
* Branching, Merging, and rebase
* Rollback, reset
* Git ssh login

# Database Management

# Databases (Mentor-led session Duration: 10 Hr)

* Introduction to Relational Databases & SQL
* What is MySQL? & it’s engines
* Basic queries - create DB, table and insert, update, alter of tables
* Select query & it’s operations
* Count & Sum
* Update & Delete
* Order By and Group By
* AND OR Between In Like
* Joins
* Working with Dates
* Auto Increment
* Triggers
* Index & Views
* Commit & Rollback
* Functions – MySQL & User Defined
* SQL Queries (CRUD Operations, Joins, Aggregations)
* Database Indexing
* Integration with Spring Boot

# Unit Testing (Mentor-led session Duration: 5 Hours)

* JUNIT Introduction
* Configuring unit tests in IDE/Java project
* Writing and executing unit tests
* Mockito Framework-Handson: Maven Dependencies, Mock creation, Mockito Behavior Verification, Mockito Verify Interaction, Stub Concrete Class, Mockito Spy

# Code Coverage Techniques and Tools (Mentor-led session Duration: 2 Hours)

* What is Code Coverage?
* How is Code Coverage measured?
* Code coverage vs Test coverage
* Code Coverage Techniques
* Code Coverage Tools

**Basic of DevOps** (Mentor-led session Duration: 4 Hours)

* Maven/Gradle (Build Automation)
* Docker & Containerization
* Kubernetes (Introduction)
* CI/CD Pipelines (Jenkins, GitHub Actions, GitLab CI/CD)

# Cloud Computing Services (Mentor-led session Duration: 3 Hours)

* Introduction to basic Cloud services (API gateway, file storage, RDS, Compute engine, serverless)
* Types of Cloud Deployment
* PaaS - Introduction
* Architecting for scalability and reliability on PaaS
  + Design principles for scalable applications
  + Ensuring reliability and availability
* Data services and management
* Security and compliance

# SonarQube Implementation (Mentor-led session Duration: 2 Hours)

* Introduction to SonarQube
* SonarQube Architecture Overview
* Dockerized environment setup
* Maven project scanning with SonarQube
* SonarQube Functionality and Tricks
* SonarQube Analysis & Code Coverage on Node.Js Apps
* SonarQube Setup with SSL Certificates & HTTPS

# Security Tools: (1 Hour)

* Trivy & OWASP Dependency Check
* Prowler – Cloud Platform Security Tool
* Dockle

# Deployment (Mentor-led session Duration: 2 Hours)

* Deployment with Netlify or vercel

# Harnessing Generative Artificial Intelligence (GenAI)

# GitHub CoPilot – Utilization (3 Hours)

* GitHub Copilot Overview.
* Basic Copilot Usage.
* Understanding Copilot Suggestions.
* Generative Commit Messages.
* Understanding Copilot Chat.
* Copilot for the CLI.
* Copilot’s Limitations and Strengths.
* Troubleshooting and Common Issues.
* GitHub Copilot Language Support
* Code Documentation.
* Copilot in IDEs.
* Code Optimization with Copilot.
* Refactoring Strategies.
* NLP and GitHub Copilot
* Debugging with Copilot.
* Copilot Best Practices.
* Usage of CoPilot in the existing/legacy code

# Mini Project Development

* Business Use cases as Project Problem Statement.
* With Mentor-led doubts clarification support.