Programme on '**Java Full Stack with Core-Java-Refresher**' Skillset

Programme Schedule Model:

- Total Mentor-led VILT topics-session Duration : 118 Hours
- Practice Duration: 250 hours
- Total Duration for self-paced Course(s): 20 hours

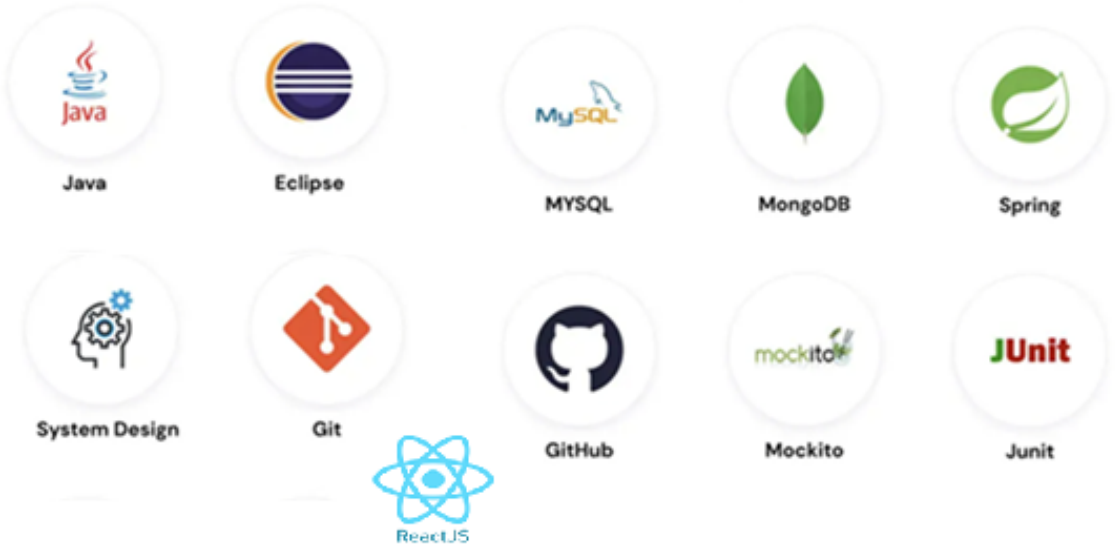
Instructor: Industry Mentors

- 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)

Technologies Covered In This Course



GUVI's Exclusive Practice Platforms:

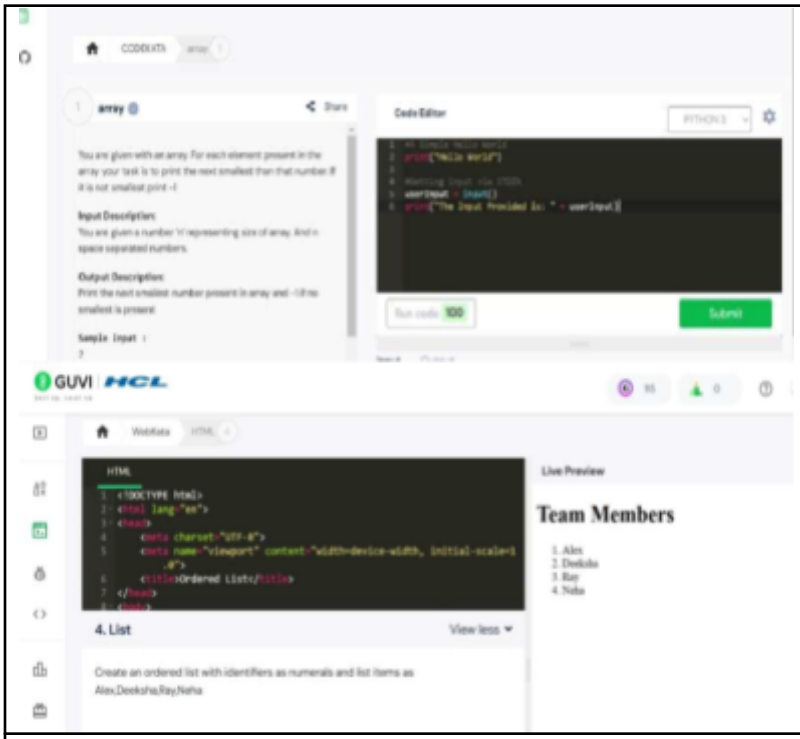
	**Codekata** : https://www.guvi.in/code-kata.html A tool-kit specifically developed to boost the coding skills and makes you ever-ready to crack interviews.
	Webkata: https://www.guvi.in/webkata.html A cloud-based module to hone your front end skills without any hassle of local environment setup.
	**IDE :** https://www.guvi.in/ide/ GUVI IDE is an Integrated Development Environment that lets you write, edit, run, test & debug your code.

Table of Contents: Java Full-Stack Development

Frontend Development

JavaScript (JS) (Mentor-led session Duration: 12 Hours)

- Basics of JavaScript (Syntax, Variables, Operators)
- DOM Manipulation and Events
- ES6+ Features (Arrow Functions, Promises, Modules)
- Fetch API and Asynchronous Programming
- Introduction to JavaScript Frameworks and Libraries
- React JS (Frontend Framework)

Introduction to ReactJS and JSX (Mentor-led session Duration: 18 Hours)

- Components, Props, and State
- React Lifecycle Methods and Hooks
- State Management (Context API, Redux)
- Routing with React Router
- Integrating APIs with Axios or Fetch
- Advanced Topics (Performance Optimization, Error Boundaries)
- Mini Project Development (FrontEnd)

Backend Development

Core Java Programming
– Refresher Sessions
(Mentor-led session
Duration: 12 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 (try-catch, throws, custom exceptions)

- Exception Handling
- Collections
- Comparable & Comparator
- File Handling
- Memory management & Garbage collection
- Collections Framework (List, Set, Map, Queue)
- Generics

Java 8 & latest
(Mentor-led session
Duration: 14 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: 16 Hours)

- Spring MVC
- Intro to Spring Boot Web (Servlet) stack
- Spring Core (Inversion Of Control & Dependency Injection)
- ORM Concepts
- 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

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

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

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

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

Full-Stack Integration (Mentor-led session Duration: 8 Hours)

- Connecting Frontend (ReactJS) to Backend (Spring Boot)
- REST API Consumption in React

- State Management Across Full Stack
- Data Flow and Error Handling
- Real-Time Data with WebSockets

Security Tools: (1 Hour)
- Trivy & OWASP Dependency Check
- Prowler – Cloud Platform Security Tool
- Dockle

Deployment (Mentor-led session Duration: 2 Hours)
- Version Control with Git and GitHub , Netlify or vercel

Harnessing Generative Artificial Intelligence (GenAI)

GenAI Fundamentals (2 Hours)
- Introduction to GenAI, LLMs, NLP
- GenAI solutions for Application Development Life Cycle
- GenAI vs Agents vs Agentic AI

Prompt Engineering (2 Hours)
- Prompt Fundamentals
- Prompt Techniques – Few Shots, Persona(role) based, Chain-of-Thought, etc.,
- Prompt Best Practices - Be Specific and Remove Needless Words, Delete irrelevant history, Start new threads, Reference previous responses, etc.,

Responsible & Ethical AI (1 Hour)
- AI Risks, Secure Prompting, Mitigation

GitHub CoPilot – Utilization (5 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
- GitHub Copilot - Prompts, Instructions, Chat Modes.
- Code Documentation – Java Docs, Functional Spec, Functional Test Case, etc.,
- Copilot in IDEs.
- Code Optimization with Copilot.
- Refactoring Strategies.
- Code Migration (Java: Spring Boot Migration)
- Setup Tests, Fix Test Failures, Generate Tests using copilot (Improve Test Coverage)

- NLP and GitHub Copilot
- Debugging with Copilot.
- Copilot Best Practices.
- Usage of CoPilot in the existing/legacy code
- MCPs, Coding Agents, Copilot code review

Mini Project Development

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

Project variations - Sample Applications:

MealDB App

Use Postman to hit the meals db api to query the data from:

https://www.themealdb.com/ An API that fetches a list of meals from

TheMealDb.com and returns a meal that

requires the least number of ingredients:
https://www.themealdb.com/api/json/v1/1/search.php?s=Arrabiata

Rest Countries API

Build a website using HTML, CSS (or bootstrap), JS, and rest countries API.Users are able to view the list of countries. search the countries with criteria (Name, Country code,continent, and capital)
Reference: http://restcountries.eu/

Spotify App

Clone the Spotify app via Angular(angular modules, angular services, and observables).Users can able to listen to the songs, search songs, search album, search artists, follow artist, unfollow artists, create a playlist, update the playlist, add the song to playlist, remove the song from playlist and updateuser profile info.
Reference: https://developer.spotify.com/documentation/web-api/

FreshDesk App-

Clone the Freshdesk product. User can able to view a ticket, search the ticket, create a newticket, update the ticket, delete ticket, view a contact, search the contacts, create a new contact, update the contact, and delete a contact. Reference: https://developers.freshdesk.com/api/