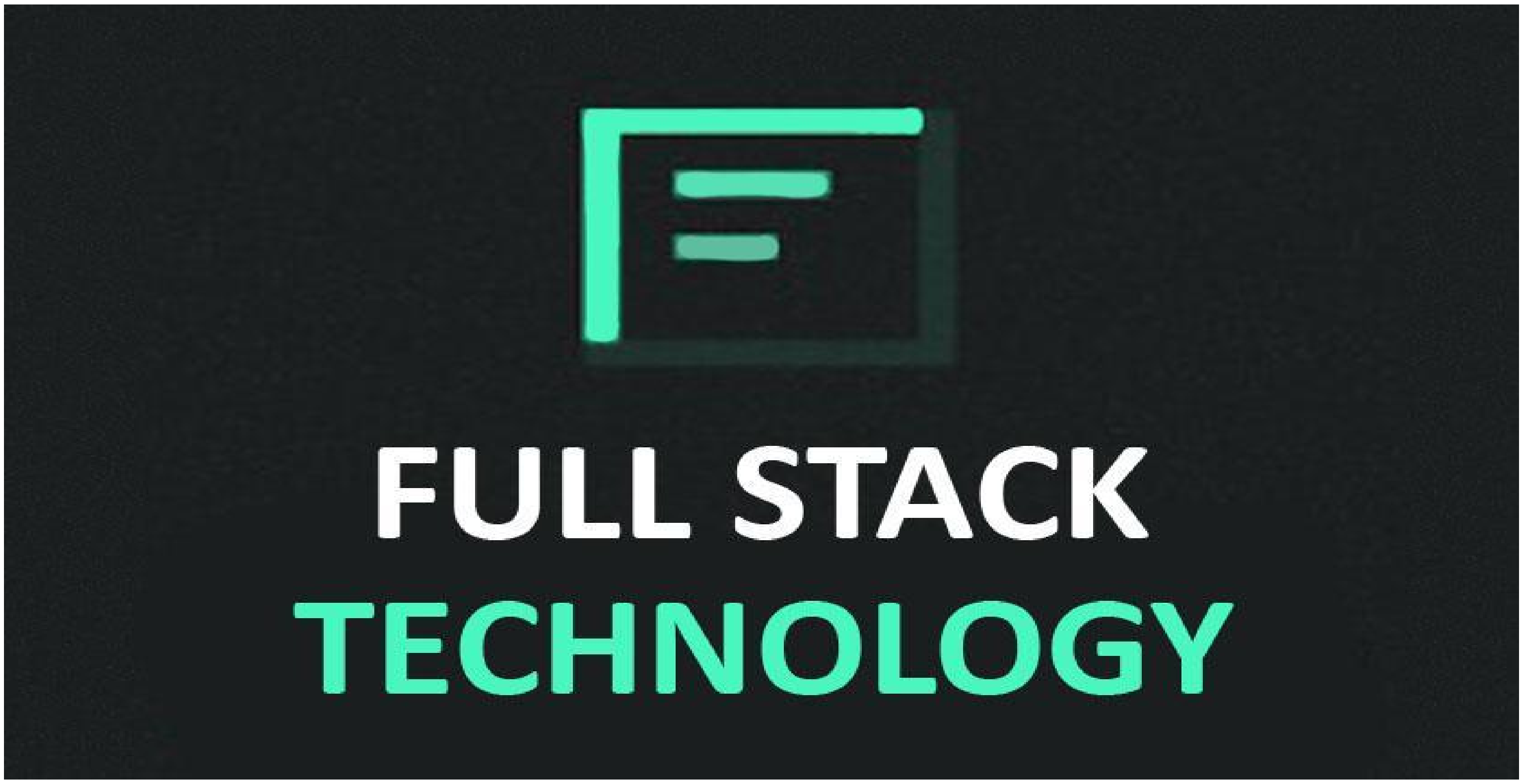
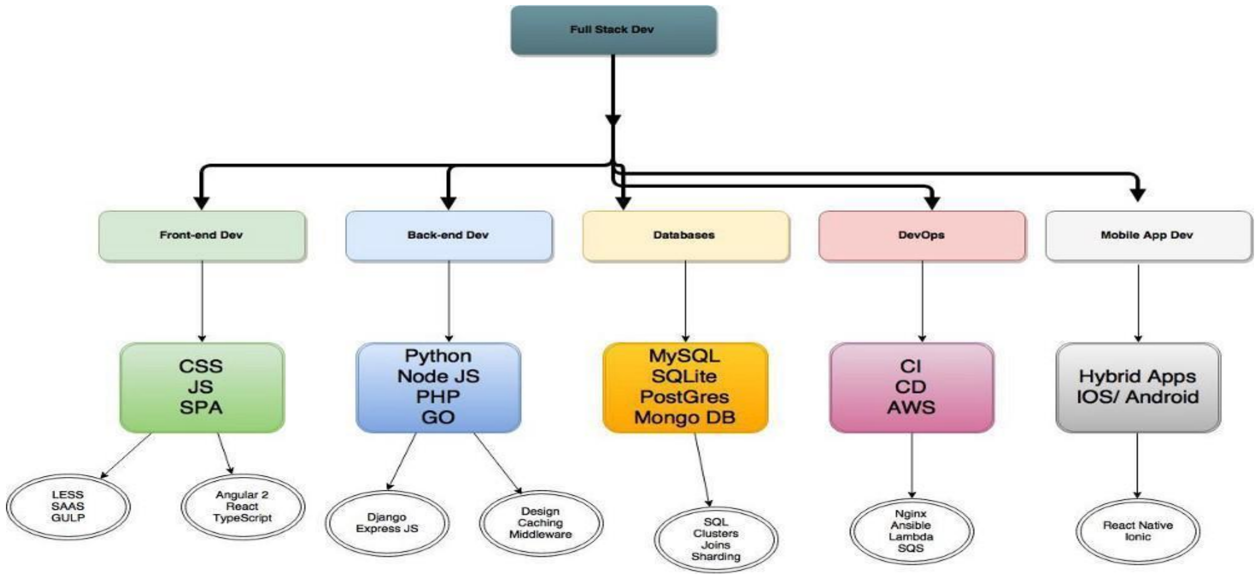
eMexo TECHNOLOGIES

LEARN TODAY AND LEAD TOMORROW





Overview

Let eMexo Technologies **Best Full Stack Training in Electronic City Bangalore** take you from the Full Stack to Advance Full Stack Training and make you an expert in developing real-time applications. Here are the major topics we cover under this Full Stack course Syllabus for **FRONT END - HTML5 AND CSS, BOOTSTRAP, JAVASCRIPT & JQUERY, AJAX, ANGULAR or REACT** and for **BACKEND - JAVA or PYTHON, and ORACLE**. Every topic will be covered in a most practical way with examples for our Full Stack Course in Electronic City Bangalore**.**

All the topics will be covered with Practical and hands-on training. Our trainers have industry experience with live project experience in cutting-edge technologies that they teach. We hire only Full Stack industry specialists as trainers for our **Full Stack Certification Course in Electronic City Bangalore**.

If you are looking for Full Stack Certification Training in Electronic City Bangalore, eMexo Technologies is the **Best Full Stack Training Institute in Electronic City Bangalore**. Come over to our training institute for a free demo class. Let our trainer give you a demo on Full Stack and only then you take the decision to enroll in the **Full Stack Training Course in Electronic City Bangalore**.

Training Features

Real-life Case Studies

Do a real-life case study to understand the usage in real-world scenarios.

Assignments

Each class will be followed by a practical assignment switch that can be completed before the next class.

Preparation for interview

Our trainers are professionals working in multinational corporations. They are experts in their field and know exactly what the interviewer will look for in the candidate. Experienced trainers not only share interview questions but also conduct mock interviews to help prepare for the actual interview.

Key Features

eMexo Technologies offers the **Best Full Stack Training Course in Electronic City Bangalore** with the TOP industry expert trainers.

Here are the key features.

- ★ Free Demo Class Available
- ★ Practical Approach
- ★ Expert & Certified Trainers
- ★ 100% Job Oriented Training
- ★ Real World use cases and Scenarios
- ★ Completed 500+ Batches
- ★ Certification Guidance

Front End Development

HTML

Unit 1: HTML Fundamentals

- **HTML Introduction**
    - HTML – Elements
    - HTML – Tags
    - HTML – Text
    - HTML – Formatting
    - HTML – Pre
    - HTML – Attributes
    - HTML – Font
    - HTML – Text Links
    - HTML – Comments
    - HTML – Lists
    - HTML – Images
    - HTML – Image Links
    - HTML – Tables
    - HTML – Bgcolor
    - HTML – Color Codes
    - HTML – Color Chart
    - HTML – Background
- **Web Forms**
    - HTML – Forms
    - HTML – Input

- HTML – Text Fields

- **Hidden Fields**
  - HTML – Password
  - HTML – Reset
  - HTML – Submit
  - HTML – Checkboxes
  - HTML – Radio
  - HTML – Select
  - HTML – Hidden Fields
  - HTML – Upload
  - HTML – Textarea
- **Special Tags**
  - HTML – Body
  - HTML – Meta
  - HTML – Style
  - HTML – Div
  - HTML – Layouts
  - HTML – Frames
- **Formatting Tags**
  - HTML – Bold
  - HTML – Paragraphs
  - HTML – Headings
  - HTML – Line Breaks
- **HTML - Horizontal Rule**
  - HTML – Email
  - HTML – Italic

HTML5

Unit 2: HTML5 Fundamentals

- **Semantic Elements**
  - <article>
  - <aside>

- <figcaption>
- <figure>
- <footer>
- <header>
- <mark>
- <nav>
- <progress>
- <section>
- <summary>
- <time>

➢ **Form Elements**

- <datalist>
- <keygen>
- <output>

➢ **Form Input Types**

- Color
- Date
- Datetime
- Datetime-local
- Email
- Month
- Number
- Range
- Search
- Tel
- Url
- Time
- Week

➢ **Form Attributes**

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod

- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)

CSS

Unit 3: CSS Fundamentals & Frameworks

- ➢ **CSS Introduction**
  - CSS Syntax
  - CSS Selectors (ID, Class, Tags, Attributes)
  - CSS Styling
- ➢ **Styling Backgrounds**
  - Styling Text
  - Styling Fonts
  - Styling Links
  - Styling Lists
  - Styling Tables
- ➢ **CSS Box Model**
  - CSS Border
  - CSS Outline
  - CSS Margin
  - CSS Padding
  - CSS Dimension
  - CSS Display
  - CSS Positioning
  - CSS Floating
  - CSS Navigation Bar
  - CSS Image Gallery
  - CSS Image Opacity
  - CSS Align

- **CSS3 Introduction**
  - Borders
  - border-radius
  - Border Images
  - Backgrounds
  - Background Size
  - background-origin
  - Text Effects
  - text-shadow
  - box-shadow
  - Text
  - text-overflow
  - word-wrap
  - word-break
  - Fonts
- **Transforms**
  - 2D Transforms
  - 3D Transforms
- **Transitions**
  - transition-delay
  - transition-duration
  - transition-property
  - transition-timing-function
- **BootStrap**
  - Overview of Bootstrap
  - Grid System
  - Typography
  - Code
  - Tables
  - Forms
  - Buttons
  - Images
  - Helper classes
  - Responsive utilities

JAVA SCRIPT

Unit 4: JAVASCRIPT Basics & Concepts

- **Introduction**
  - What is JavaSript?
  - What is AJAX?
- **Developer Essentials**
  - The development workflow
  - Selecting the right tools for the job
  - Just enough HTML and CSS
  - Understanding objects
  - Understanding variables
  - Making comparisons
  - Understanding events
- **Starting to Code**
  - Writing your first script
  - Internal vs. external scripts
  - Using comments in scripts
  - Using the NoScript tag in HTML
- **Interacting with Users**
  - Creating alert dialogs
  - Understanding conditional statements
  - Getting confirmations from users
  - Creating prompts for users
  - Understanding functions
  - Making links smarter
  - Using switch/case statements
  - Handling errors
- **JavaScript Language Essentials**
  - Getting started
  - Creating loops
  - Passing values to functions
  - Detecting objects
  - Reading arrays

  - Returning values from functions
  - Writing arrays
  - Building do and while loops
  - Re-using functions

- **Creating Rollover and More**
  - Creating a basic image rollover
  - How to write a better rollover
  - Creating a three-state rollover
  - Making rollovers accessible and 508 compliant
  - Making disjointed rollovers
  - Creating slideshows
  - Displaying random images

- **JavaScript Functions**
  - Definitions
  - Parameters
  - Invocation
  - Call
  - Apply
  - Closures

- **Building Smarter Forms**
  - Getting started
  - Creating jump menus
  - Creating dynamic menus
  - Requiring fields
  - Cross-checking fields
  - Displaying more informative errors
  - Verifying radio button selections
  - Setting one field with another field
  - Verifying email addresses

- **Handling Events**
  - Responding to window events
  - Responding to mouse movements
  - Responding to mouse clicks
  - Responding to onBlur form events
  - Responding to onFocus form events

- ○ Responding to keyboard events

➢ **Working with Cookies**
- ○ Demystifying cookies
- ○ Writing a cookie
- ○ Reading a cookie
- ○ Displaying a cookie
- ○ Counting with cookies
- ○ Deleting cookies
- ○ Handling multiple cookies
- ○ Cookies in action

➢ **The DOM, Nodes & Objects**
- ○ Understanding the DOM
- ○ Adding nodes to the DOM
- ○ Deleting nodes from the DOM
- ○ Deleting specific nodes
- ○ Inserting nodes into the DOM
- ○ Replacing nodes in the DOM

➢ **JavaScript Browser BOM**
- ○ Window
- ○ Screen
- ○ Location
- ○ History
- ○ Navigator
- ○ Popup Alert
- ○ Timing
- ○ Cookies

➢ **Working with Dates & Times**
- ○ Displaying dates
- ○ Displaying times
- ○ Creating a countdown

➢ **JavaScript JSON**
- ○ Intro
- ○ Syntax
- ○ Json vs XML
- ○ Data Types

- Parse
- Stringify
- Objects
- Arrays
- PHP
- HTML
- JSONP

➢ **Real-World Applications of JavaScript**

- Creating sliding menus
- Creating pop-up menus
- Creating slideshows with captions
- Creating a stylesheet switcher

➢ **AJAX**

- Introduction
- XMLHttp
- Request
- Response
- XML File
- PHP
- ASP
- Database
- Applications
- Examples

JQUERY

Unit 5: JQuery Basics & Concepts

➢ **Introduction to jQuery**

- What is JQuery?
- First jquery code
- Separating scripts
- Selectors
- Replacing content

- Handling events

- **Animation**
  - Show and hide elements
  - Fading
  - Hover effects
  - Toggle
  - Sliding
  - Custom animations
- **Extracting Content**
  - Overview on extracting data
  - Basic selectors
  - Basic filters
  - Advance selectors
  - Advance filters
- **Creating Content Dynamically**
  - Creating content
  - Inserting content
  - Modifying content
  - CSS Manipulation
  - Navigating the DOM and using statement chaining
  - Wrapping it up

ANGULAR JS

Unit 6: Angular Basics & Concepts

- **GETTING STARTED WITH ANGULAR**
  - Building Blocks of Web Application Development
  - Web Application Architecture
  - Introduction to Angular
  - Angular Architecture
  - Building blocks of Angular
  - Angular Installation

- ○ Angular CLI
- ○ Angular CLI commands
- ○ Understanding files in Angular
- ○ Hands-On

➢ **ANGULAR COMPONENTS AND DATA BINDING**

- ○ Working of Angular Applications
- ○ Angular App Bootstrapping
- ○ Angular Modules
- ○ Decorators and its types
- ○ Angular Components
- ○ Creating A Component Through Angular CLI
- ○ Ways to specify selectors
- ○ Template and styles
- ○ Installing bootstrap to design application
- ○ Hands-On

➢ **DATABINDING AND ANIMATIONS**

- ○ Databinding
- ○ Types of Databinding
- ○ Component Interaction using @Input and @Output decorator
- ○ Angular Animations
- ○ Component Life-cycle Hooks
- ○ Hands-On

➢ **ANGULAR DIRECTIVES AND PIPES**

- ○ Understanding Angular Directives
- ○ @Component Directive
- ○ Structural Directives
- ○ Attribute Directives
- ○ Custom Directives
- ○ Pipes
- ○ Built-in Pipes
- ○ Chaining pipes
- ○ Custom pipes
- ○ PipeTransform Interface & Transform Function
- ○ Hands-On

- ➢ **ANGULAR SERVICES AND DEPENDENCY INJECTION**
  - Angular service
  - Need for a service
  - Dependency Injection
  - Creating a service
  - Hierarchical Injector
  - Injecting A Service into Another Service
  - Observables
  - Hands-On
- ➢ **RXJS AND HTTPCLIENT**
  - RxJS Library
  - Angular’s Interaction with Backend
  - Parts of an HTTP Request
  - HttpClient
  - Hands-On
- ➢ **ANGULAR ROUTES AND NAVIGATION**
  - Angular Router
  - Setting Up Routes
  - Adding Routes Using RouterLink
  - Wildcard and Redirecting Routes
  - Adding Navigation Programmatically
  - Passing Route Parameters
  - Extracting Parameters Using ActivatedRoute
  - Optional Route Parameters
  - Child Routes
  - Route Guards
  - Location Strategies
  - Hands-On
- ➢ **HANDLING FORMS IN ANGULAR**
  - Angular forms
  - Types of forms
  - Underlying building blocks of the form model
  - Template-driven vs Reactive forms
  - Template-driven forms
  - Reactive Forms

- Dynamically adding data to a form
- Hands-On

➢ **VALIDATING ANGULAR FORMS**

- What is Form Validation?
- Types of Form Validation
- Built-in Validators
- Form control's status and validity
- Form Validation methods
- CSS classes for Form control
- Custom validators in Template Driven Forms
- Hands-On

➢ **AUTHENTICATION WITH JWT AND SECURITY**

- What is Authentication?
- Authentication and authorization
- Types of Authentication
- Where to store tokens?
- JSON Web Tokens (JWT)
- Authentication in Angular application
- Security threats in web application
- Hands-On

➢ **TESTING AND APPLICATION DEPLOYMENT IN ANGULAR**

- Testing
- Why should we perform testing?
- Types of testing
- Testing Angular application using Jasmine and Karma
- Maintaining application code using Git
- Version control system
- Why should we use Git?
- Git file workflow
- Running application on the production server: Nginx
- Architecture of Nginx
- How to configure Nginx?
- Deployment of an application using Docker
- Problems before containers
- How containers solve the problems

- What is Docker?
- Docker file
- Docker image
- Docker containers
- Docker hub
- Basic Docker commands
- Hands-On

Back End Development

JAVA

Unit 7: JAVA Basics & Concepts

- **Basics of Java Programming**
  - Java – What, Where, and Why?
  - History and Features of Java
  - Internals of Java Program
  - Difference between JDK, JRE, and JVM
  - Internal Details of JVM
  - Variable and Data Type
  - Unicode System
  - Naming Convention
  - To run the first program in CMD
- **Java Packages**
  - What are Packages in java?
  - Needs of Packages in Java
- **Variables, Data Types, and Operators**
  - Types of Variables and their uses
  - Primitive and Non-primitive Data Type
  - Numeric and Character values
  - Keywords
  - Types of Operators in Java

- **Control Flow statements and Methods in Java**
  - What is a function?
  - if, if-else, if-else-if methods
  - Switch case statement
  - For loop statement
  - While and do-while loop implementation
  - Break statement syntax
  - return statements
  - Continue statement
  - Java Comments
- **OOPS Concepts and its applications**
  - Java OOPs Concepts Introduction
  - Naming Conventions
  - Object and Class
  - Creating Object outside of a class
  - Method Overloading
  - Constructor
  - Abstraction Implementation
  - Concept of Inheritance
  - Polymorphism in Java
  - Java Encapsulation methods
  - Java Array
  - Association-Composition & Aggregation
- **JAVA Constructors**
  - What are constructors and how to implement it?
  - Methods for Constructors Enumerated Data Types
- **String Handling**
  - String: What and Why?
  - Immutable String
  - String Comparison
  - String Concatenation
  - Substring
  - Methods of String class
  - StringBuffer class
  - StringBuilder class

- Creating Immutable class
- toString method
- StringTokenizer class

➢ **Exception Handling**

- Exception Handling: What and Why?
- Types of Java Exceptions
- Checked and Unchecked Exceptions
- Java Exception keyword
- Throw & throws
- Finalize
- Try with Resource
- Exception Handling with Method Overriding
- Java Custom Exception

➢ **Java Inner or Nested Classes**

- Nested Class: What and Why?
- Member Inner class
- Annonymous Inner class
- Local Inner class
- static nested class
- Nested Interface

➢ **Multithreading**

- Multithreading: What and Why?
- Synchronized Block and Method
- Life Cycle of a Thread
- Thread API in Java
- Creating Thread
- States of Tread
- Advance Thread concepts
- Thread Pooling - Executor, Callable, Future
- Queues, Worker thread model - Executer Service, etc.
- Fork Join Framework
- CountDownLarch
- CyclicBarrier

- Semaphore
- Mutex
- ThreadLocal
- ReentrantLock
- ShutdownHook
- Performing multiple tasks by multiple threads
- Garbage Collection

➢ **Generics**

- Generics for Collections
- Non-Generics in Collections
- Generics for class
- Generics Method
- Bounded Types
- Advantages of Java Generics

➢ **Runnable class**

- Synchronization: What and Why?
- synchronized method
- synchronized block
- static synchronization
- Deadlock
- Inter-thread Communication
- Interrupting Thread

➢ **Input and output**

- What is Stream?
- Input and Output Streams
- Types of Streams
- java.io package
- The Byte-stream I/O hierarchy
- Character Stream Hierarchy
- FileWriter & FileReader
- File I/O and Object Serialization
- CharArrayWriter
- Console
- Compressing and Uncompressing File

- Reading and Writing data simultaneously
- DataInputStream and DataOutputStream
- Object Stream
- Buffered Stream
- StreamTokenizer class

➢ **Serialization & Externalization**

- Serialization & Deserialization
- Serialization with IS-A and Has-A
- transient keyword
- Object Serialization
- Externalizable

➢ **Java Collections**

- What is a framework in Java?
- Collection Framework
- Collection Interfaces and Implementor classes
- Types of Interfaces
- Types of Classes
- List, Set, Map
- Comparable and Comparator
- Collections and Arrays Classes
- Enhanced ForEach Loop
- Java Vector and Stack
- Enumeration
- List Interface
- ArrayList
- LinkedList
- Set Interface
- HashSet, LinkedHasSet, TreeSet
- Map Interface
- HashMap, LinkedHashMap, TreeMap
- Collection Sort
- Collection Shuffle

- **Concurrent Collection Framework**
  - CopyOnWriteArrayList
  - ConcurrentHashMap
  - SynchronizedList
  - SynchronizedSet
  - SynchronizedMap
- **JDBC**
  - JDBC Drivers
  - Steps to connect to the database
  - JDBC API Interfaces
  - JDBC API Classes
  - Stored procedures and functions
  - Transaction Management
  - Batch Processing
  - JDBC New Features
  - Mini Project
- **Reflection API**
  - Instantiating classes
  - Exploring Methods
  - Calling methods
  - Creating Object
  - Exploring Constructors
- **Java 8 Features**
  - Lambda Expressions
  - Pipelines and Streams
  - Date and Time API
  - Default Methods
  - Type Annotations
  - Nashhorn JavaScript Engine
  - Concurrent Accumulators
  - Parallel operations
  - PermGen Error Removed
  - TLS SNI
- **Java 9 Features**
  - Platform Module System (Project Jigsaw)

- Interface Private Methods
- Try-With Resources
- Anonymous Classes
- @SafeVarargs Annotation
- Collection Factory Methods
- Process API Improvement
- New Version-String Scheme
- JShell: The Java Shell (REPL)
- Process API Improvement
- Control Panel
- Stream API Improvement
- Installer Enhancement for Microsoft windows and many more

➢ **Java 10 Features**

- Local-Variable Type Inference
- Consolidate the JDK Forest into a Single Repository
- Garbage-Collector Interface
- Parallel Full GC for G1
- Application Class-Data Sharing
- Thread-Local Handshakes
- Remove the Native-Header Generation Tool (javah)
- Additional Unicode Language-Tag Extensions
- Heap Allocation on Alternative Memory Devices
- Experimental Java-Based JIT Compiler
- Root Certificates
- Time-Based Release Versioning

➢ **Java 11 Features**

- Running Java File with single command
- New utility methods in String class
- Local-Variable Syntax for Lambda Parameters
- Nested Based Access Control
- JEP 321: HTTP Client
- Reading/Writing Strings to and from the Files
- JEP 328: Flight Recorder.

➢ **Java 12 Features**

- VM Changes - JEP 189, JEP 346, JEP 344, and JEP 230.

- Switch Expressions
- File mismatch() Method
- Compact Number Formatting
- Teeing Collectors in Stream API
- Java Strings New Methods - indent(), transform(), describeConstable(), and resolveConstantDesc()
- JEP 334: JVM Constants API
- JEP 305: Pattern Matching for instanceof
- Raw String Literals is Removed From JDK 12

➢ **Java 13 Features**

- Text Blocks - JEP 355
- New Methods in String Class for Text Blocks
- Switch Expressions Enhancements - JEP 354
- Reimplement the Legacy Socket API - JEP 353
- Dynamic CDS Archive - JEP 350
- ZGC: Uncommit Unused Memory - JEP 351
- FileSystems.newFileSystem() Method
- Support for Unicode 12.1
- DOM and SAX Factories with Namespace Support

➢ **Java 14 Features**

- Switch Expressions (Standard) - JEP 361
- Pattern Matching for instanceof (Preview) - JEP 305
- Helpful NullPointerExceptions - JEP 358
- Records (Preview) - JEP 359
- Text Blocks (Second Preview) - JEP 368
- Packaging Tool (Incubator) - JEP 343
- NUMA-Aware Memory Allocation for G1 - JEP 345
- JFR Event Streaming - JEP 349
- Non-Volatile Mapped Byte Buffers - JEP 352
- ZGC on macOS - JEP 364
- ZGC on Windows - JEP 365
- Foreign-Memory Access API (Incubator) - JEP 370

➢ **Java 15 Features**

- Sealed Classes (Preview) – JEP 360
- Pattern Matching for instanceof (Second Preview) – JEP 375

- Records (Second Preview) – JEP 359
- Text Blocks (Standard) – JEP 378
- Hidden Classes – JEP 371
- Remove the Nashorn JavaScript Engine – JEP 372
- Reimplement the Legacy DatagramSocket API – JEP 373
- Disable and Deprecate Biased Locking – JEP 374
- Shenandoah: A Low-Pause-Time Garbage Collector – JEP 379
- Remove the Solaris and SPARC Ports – JEP 381
- Foreign-Memory Access API (Second Incubator) – JEP 383
- Deprecate RMI Activation for Removal – JEP 385

➢ **Java 16 Features**

- Vector API (Incubator) - JEP 338
- Enable C++14 Language Features - JEP 347
- Migrate from Mercurial to Git - JEP 357
- Migrate to GitHub - JEP 369
- ZGC: Concurrent Thread-Stack Processing - JEP 376
- Unix-Domain Socket Channels - JEP 380
- Alpine Linux Port - JEP 386
- Elastic Metaspace - JEP 387
- Windows/AArch64 Port - JEP 388
- Foreign Linker API (Incubator) - JEP 389
- Warnings for Value-Based Classes - JEP 390
- Packaging Tool - JEP 392
- Foreign-Memory Access API (Third Incubator) - JEP 393
- Pattern Matching for instanceof - JEP 394
- Records - JEP 395
- Strongly Encapsulate JDK Internals by Default - JEP 396
- Sealed Classes (Second Preview) - JEP 397

➢ **Java 17 Features**

- Context-Specific Deserialization Filters - JEP 415
- Vector API (Second Incubator) - JEP 414
- Foreign Function & Memory API (Incubator) - JEP 412
- Deprecate the Security Manager for Removal - JEP 411
- Remove the Experimental AOT and JIT Compiler - JEP 410
- Sealed Classes - JEP 409

- Remove RMI Activation - JEP 407
- Pattern Matching for switch (Preview) - JEP 406
- Strongly Encapsulate JDK Internals - JEP 403
- Deprecate the Applet API for Removal - JEP 398
- macOS/AArch64 Port - JEP 391
- New macOS Rendering Pipeline - JEP 382
- Enhanced Pseudo-Random Number Generators - JEP 356
- Restore Always-Strict Floating-Point Semantics - JEP 306

➢ **Java 18 Features**

- UTF-8 by Default - JEP 400
- Simple Web Server - JEP 408
- Code Snippets in Java API Documentation - JEP 413
- Reimplement Core Reflection with Method Handles - JEP 416
- Vector API (Third Incubator) - JEP 417
- Internet-Address Resolution SPI - JEP 418
- Foreign Function & Memory API (Second Incubator) - JEP 419
- Pattern Matching for switch (Second Preview) - JEP 420
- Deprecate Finalization for Removal - JEP 421

Spring Boot

Unit 8: Spring Boot Basics & Concepts

➢ **Introduction to spring boot**

- Types of software architectures
- SOA and Monolith Architecture
- Why Microservices
- Detailed MicroService Architecture
- App Layer
- Business Layer
- Enterprise Layer
- Infra Layer
- Need of Spring Boot
- Difference between Spring & Spring Boot
- Advantages with Micro Services

- **Building SPRING BOOT Application**
  - Normal Spring Manual Approach
  - Maven Overview
  - Spring Initializer
  - STS
  - Eclipse with STS Plugin
  - Understanding the Spring Boot auto-configuration
- **Rest Annotation with In-Memory Database & CRUD Operations**
  - H2
  - Derby
  - HSQL
  - Redis Cache
  - PostMan or Swagger Overview
- **Rest Annotation with Relation DB**
  - MySql
  - PostGresSQL
- **JPA Repository Concepts**
  - Crud Repository
  - JPA Query Concepts
  - NamedQueries
  - QueryAnnotation
  - AsyncResults
  - Pagination and Sorting
- **Actuator Concepts**
  - Production Monitoring
  - Health Check Concepts
  - Security Measurements
- **Spring Boot Custom Logging**
  - Logging Level
  - Patterns Changes
  - Rolling Logs
- **Spring Boot Profile Components**
  - Introduction
  - Multiple Properties

- YML File
- Command Line Runner Example
- Real time scenarios of components

➢ **Auto Configuration**

- Introduction
- Conditional Flow
- Customize conditional annotations
- Spring Boot built-in conditional annotations

➢ **Thymleaf Concepts**

- Introduction
- Example on Web Application
- Validatins on Web Applications
- Internalization i18n Concepts

➢ **Integration with Spring Web**

- Using Spring Web MVC
- Using Spring Restful
- Need of embedded servers & customization

➢ **Spring Boot Security**

- Spring JDBC
- Database to CSV
- Spring Batch
- Flyway Database Migration
- Liquid Database Migration
- Flyway vs Liquid
- Hikari Connection Pool

➢ **Core Concepts**

- Spring Boot AOP
- Spring Boot Cache
- Guava Cache integration
- Caffenine Cache
- EH Cache
- MultiResourceItemReader
- Spring MVC vs JAX-RS

- SprinBoot with Jersey
- Junit Integration
- Rest Integration Test Cases

➢ **Micro Services**

- Micro Services Introduction
- Principle and Characteristics
- Use cases and Benefits
- Challenges
- Design standards
- Micro Services Communication
- Synchronous
- Asynchronous
- Pitfalls

➢ **Micro Services Design Considerations**

- Micro Services per JVM?
- Micro Services share the data stores?
- Micro Services Transaction boundaries
- User Interfaces integration with Micro Services
- Challenges in Micro Services implementation

➢ **Spring Cloud**

- Introduction
- Cloud Architecture
- Cloud application benefits

➢ **Spring Cloud Config**

- Introduction
- Setup version control repository
- Integration with repository

➢ **Fault Tolerance Concepts**

- Circuit Breaker Pattern
- Hystrics Concepts & Hystrix Dashboard

➢ **API Gateway**

- Introduction to ZUUL
- Design standards

- Integration

➢ **Messaging Queue Concepts (CloudBus)**

- Apache KAFKA
- RabbitMQ
- JMS

➢ **Oatuh2 Concepts**

- Client Types
- Protocol End Points
- Grant Types
- Implantation with Token Based
- JWT Tokens

➢ **Swagger API**

- Introduction
- Integration

➢ **Cloud Hosting**

- Pivotal Cloud Foundry Account Setup
- Hosting to Pivotal
- AWS Account Setup
- Hosting to AWS
- Enabling cloud features like load balancing, security

MYSQL

Unit 9: MYSQL ESSENTIALS

➢ **Theory, Terminology, and Concepts**

- Client/Server Concepts
- Database and Database Objects

➢ **Data Definition using SQL**

- Databases
- Data Types
- Tables
- Constraints and Indexes
- Views

- **Basic Data Manipulation using SQL**
  - Recurring SQL Constructs
  - Adding data
  - Modifying data
  - Removing data
  - Searching data
- **Advanced Data Manipulation using SQL**
  - Expressions
  - Grouping and Aggregate Functions
  - Joining Tables
- **Transactions**
  - Transaction Concepts
  - SQL for working with Transaction
- **Import/Export**
  - Tools for Import/Export
  - SQL for Import/Export

FAQs

1. How is the training organized? How much percentage is theoretical and how much is practical hands-on?

We at eMexo believe nothing beats hands-on practice when it comes to learning a concept. Our teaching methodology is 100% practical and hands-on-oriented. You learn a concept, you practice it then and there with the trainer. We also give you assignments for each topic which you can practice at home and any doubts regarding the topic can be cleared with the trainer the next day.

2. What is the course duration? How and when do you plan to complete the course?

We generally cover our courses in 30 hours, however, we are aware that we can't put a hard stop to learning with a number. Our trainer will make sure that you have learned everything that is part of the curriculum. This could mean 28 hours or 35 hours, doesn't matter.

3. What is the material provided in the training?

We have industry standard course material which is used by our trainers to train you. At the end of the training apart from the notes which you have taken during the course, we will also provide you with the training material which was used. This training material includes the training content, interview questions, etc.

4. Do you help in preparing for the interview?

Our trainers are working professionals who work in MNCs. They are the expert in their domain and know exactly what an interviewer looks into a candidate. Our expert trainers apart from sharing the interview questions will also conduct mock interviews to help you prepare for the real interview.

5. Who are your trainers?

Our trainers are industry experts who work in their respective technologies day in and day out. They work in MNCs and are technology experts within their organizations.

6. What is the total batch size per course?

We maintain a strict batch size of a maximum of 5 students. We also provide exclusive one-to-one training as well. Talk to our training partner to get more details.

7. Do you provide certification for the course?

Yes, at the end of training we provide a certification of completion.

8. Will I be joining a new batch or be merged with another batch?

You will be added to a new batch.

9. Is fast-track training available?

Yes, we also provide fast-track training for those who want to complete the course faster. The curriculum and the total hours required to complete the course will remain the same. However, the trainer will be spending more hours with you to complete the course.

10. Do you assist in job placement?

Our trainers are expert professionals in their organizations and they often act as the interviewer to hire new candidates. Our trainers will help you prepare your resume with industry standards. After all, they know exactly what to look for in a resume.

11. Timings for training - Regular training/weekend training

We provide both regular and weekend training. Talk to our training partner to learn more about the timings.

12. Will you be working on a live project during training?

Yes, apart from doing the hands-on practice our trainer will also be taking a real-world project and working with you for the implementation.

13. What happens if I miss a class?

If you miss a class the content of that class will be taught to you again. With us, you might miss a class but not the content.

14. Can I attend a demo before the actual class?

Yes, absolutely! Talk to our training counselor on phone at 9513216462 or email us at info@emexotechnologies.com to arrange a free demo. You can also fill in the contact us form below and we will call you to discuss your training requirements.