



*LEARN TODAY AND LEAD TOMORROW*



## Overview

Let eMexo Technologies Best **Core Java Training in Electronic City Bangalore** take you from the fundamentals of Core Java to Advance Java and make you an expert in developing real-time Java applications. Here are the major topics we cover under this **Core Java course in Electronic City Bangalore** Syllabus, [Basics of Java Programming](#), [Java Packages](#), [Variables](#), [Data Types](#), and [Operators](#), [Control Flow statements and Methods in Java](#), [OOPS Concepts and its applications](#), [JAVA Constructors](#), [String Handling](#), [Exception Handling](#), [Java Inner or Nested Classes](#), [Multithreading](#), [Generics](#), [Runnable class](#), [Input and output](#), [Serialization](#), [Java Collections](#), [Concurrent Collection Framework](#), [JDBC](#), [Reflection API](#), and [Java Latest Features](#). Every topic will be covered in the most practical way with examples.

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 which they teach. We hire only the Best Core Java industry specialists as trainers for our **Core Java Certification Training in Electronic City Bangalore** through online or classroom classes.

Come over to our training institute for a free demo class. Let our trainer give you a demo on Core Java and only then will you decide to enroll in the training program.

Here is the full information of the **Java Certification Course Training in Electronic City Bangalore** syllabus and content for making you an expert in web development.

## 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 which 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 AWS Development 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

## Unit 1: 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
- Hands-On

## Unit 2: Java Packages

- What are Packages in java?
- Needs of Packages in Java
- Hands-On

### **Unit 3: Variables, Data Types, and Operators**

- Types of Variables and its uses
- Primitive and Non-primitive Data Type
- Numeric and Character values
- Keywords
- Types of Operators in Java
- Hands-On

### **Unit 4: 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
- Hands-On

### **Unit 5: 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
- Hands-On

## **Unit 6: JAVA Constructors**

- What are constructors and how to implement it?
- Methods for Constructors Enumerated Data Types
- Hands-On

## **Unit 7: 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
- Hands-On

## **Unit 8: 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
- Hands-On

## Unit 9: Java Inner or Nested Classes

- Nested Class: What and Why?
- Member Inner class
- Anonymous Inner class
- Local Inner class
- static nested class
- Nested Interface
- Hands-On

## Unit 10: Multithreading

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

- Performing multiple tasks by multiple threads
- Garbage Collection
- Hands-On

## Unit 11: Generics

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

## Unit 12: Runnable class

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

## Unit 13: 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
- Hands-On

## **Unit 14: Serialization & Externalization**

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

## **Unit 15: 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, LinkedHashSet, TreeSet
- Map Interface
- HashMap, LinkedHashMap, TreeMap
- Collection Sort
- Collection Shuffle
- Hands-On

## Unit 16: Concurrent Collection Framework

- CopyOnWriteArrayList
- ConcurrentHashMap
- SynchronizedList
- SynchronizedSet
- SynchronizedMap
- Hands-On

## Unit 17: 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
- Hands-On

## Unit 18: Reflection API

- Instantiating classes
- Exploring Methods
- Calling methods
- Creating Object

- Exploring Constructors
- Hands-On

## Unit 19: Java 8 Features

- Lambda Expressions
- Pipelines and Streams
- Date and Time API
- Default Methods
- Type Annotations
- Nashorn JavaScript Engine
- Concurrent Accumulators
- Parallel operations
- PermGen Error Removed
- TLS SNI
- Hands-On

## Unit 20: 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
- Hands-On

## Unit 21: 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
- Hands-On

## Unit 22: 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
- Hands-On

## Unit 23: Java 12 Features

- JVM 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
- Hands-On

## Unit 24: 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
- Hands-On

## Unit 25: 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
- Hands-On

## Unit 26: 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
- Hands-On

## Unit 27: 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

- Hands-On

## Unit 28: 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
- Hands-On

## Unit 29: 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
- Hands-On

## 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 60 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 48 hours or 60 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 +91-9513216462 or email us at [info@emexotechnologies.com](mailto: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.