CORE JAVA
COURSE SYLLABUS

From Java Fundamentals to Java 8 Features

Course Overview
This course is designed for beginners and aspiring software developers who want to build a strong foundation in Java programming. Students progress from basic syntax and programming logic to object-oriented programming, core Java APIs, multithreading, file handling, and important Java 8 features. The course emphasizes concept clarity, hands-on coding, assignments, and practical problem solving.

1. Introduction to Java

- Introduction to programming and Java
- History, features, and applications of Java
- Java platform and architecture
- JDK, JRE, and JVM
- Installing Java and setting up an IDE
- Writing, compiling, and executing the first Java program
- Structure of a Java program and main() method

2. Java Basics and Programming Fundamentals

- Tokens, keywords, identifiers, and literals
- Variables, constants, and data types
- Primitive and non-primitive data types
- Type casting and conversion
- Operators and operator precedence
- Input using Scanner
- if, if-else, nested if, and switch
- for, while, and do-while loops
- break, continue, and return
- Basic number, pattern, and logic-building programs

3. Arrays

- Array declaration, creation, and initialization
- One-dimensional arrays
- Two-dimensional and multidimensional arrays
- Array traversal
- Searching and sorting basics
- Common array programs and exercises

4. Strings

- String class and object creation
- String immutability
- Common String methods
- String comparison: == and equals()
- StringBuilder and StringBuffer
- String manipulation programs

5. Object-Oriented Programming (OOP)

- Classes and objects
- Fields, methods, and object initialization
- Constructors: default, no-argument, and parameterized
- this and super keywords
- Encapsulation and data hiding
- Inheritance and its types supported in Java
- Method overloading and overriding
- Polymorphism: compile-time and runtime
- Abstraction
- Abstract classes and interfaces
- static and final keywords
- Association, aggregation, and composition (overview)

6. Packages and Access Control

- Introduction to packages
- Creating and using user-defined packages
- Import statements
- Access modifiers: private, default, protected, and public
- Classpath basics

7. Exception Handling

- Errors and exceptions
- Exception hierarchy
- Checked and unchecked exceptions
- try, catch, and finally
- Multiple catch blocks and nested try
- throw and throws
- Creating custom exceptions
- Exception-handling best practices

8. Wrapper Classes and Utility Concepts

- Primitive types and wrapper classes
- Autoboxing and unboxing
- Parsing and conversion methods
- enum types
- Introduction to annotations

9. Collections Framework

• Introduction to the Java Collections Framework

• Collection hierarchy and Iterable

• List: ArrayList and LinkedList

• Set: HashSet, LinkedHashSet, and TreeSet

• Queue and Deque fundamentals

• Map: HashMap, LinkedHashMap, and TreeMap

• Iterator and ListIterator

• Comparable and Comparator

• Generics: generic classes and methods

• Sorting and traversing collections

• Fail-fast behavior (introduction)

10. Multithreading and Concurrency Basics

• Process and thread concepts

• Creating threads using Thread and Runnable

• Thread life cycle and states

• Common thread methods

• Thread priorities and scheduling overview

• Synchronization and race conditions

• Inter-thread communication basics

• ExecutorService and thread pools (introduction)

11. File Handling and Java I/O

• Introduction to Java I/O streams

• Byte streams and character streams

• File and directory handling

• Reading and writing text files

• BufferedReader and BufferedWriter

• FileInputStream and FileOutputStream

• Serialization and deserialization basics

• Try-with-resources

12. Java 8 Features

• Overview and importance of Java 8

• Functional interfaces

• Lambda expressions and syntax

• Predicate, Function, Consumer, and Supplier

• Method references

• Stream API fundamentals

• Intermediate and terminal stream operations

• filter(), map(), sorted(), distinct(), and collect()

• forEach(), reduce(), count(), and grouping basics

- Optional class
- Default and static methods in interfaces
- Date and Time API: LocalDate, LocalTime, and LocalDateTime

Practical Learning & Assessment

- Topic-wise coding exercises and assignments
- Logic-building problems and regular coding practice
- Practical demonstrations of important concepts
- Mini tasks combining multiple Java topics
- Revision, doubt-solving, and topic-wise assessments
- A final Core Java practice project

Learning Outcomes

By the end of this course, students should be able to understand Java fundamentals, write structured programs, apply object-oriented principles, use core Java APIs, work with collections and exceptions, create multithreaded programs, handle files, and use key Java 8 features such as lambda expressions and the Stream API. Students will have the foundation needed to continue to Advanced Java and Java-based application development.

Learn the concepts. Practice the code. Build your confidence.