CYBORG
ROBOTICS ACADEMY PVT. LTD.

JAVA

TEACHER: CERTIFIED TRAINER	DURATION: 16 CLASSES(x3 LEVELS) (1 HOUR PER CLASS)	MODE: ONLINE AND OFFLINE

Java is a popular programming language, created in 1995.It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications
- Desktop applications
- Web application
- Web servers
- Games
- Database connections

Level 1: Java Basics

Java Home

- Introduction to Java: Overview, platform independence, uses.
- Setting up JDK: Installing Java Development Kit on your machine.
- Configuring Environment Variables: Setting PATH and JAVA_HOME for Java programs.

Java Introduction

- History of Java: Java's development and evolution.
- Features of Java: Key features like OOP, portability, security.
- JVM & JRE: Role of JVM in executing Java bytecode and JRE as the runtime environment.

Getting Started with Java

- First Java Program: Write a simple "Hello, World!" program.
- Compiling & Running: Using javac to compile and java to run programs.

Java Syntax

- Program Structure: Class, methods, and main method.
- Keywords & Identifiers: Reserved words and naming rules.
- Semicolons & Braces: Java's syntax rules.

Java Output

- System.out.println(): Printing to console.
- Formatting Output: Use of escape sequences and formatting.

Java Comments

- Single-line & Multi-line: Adding comments to explain code.
- Documentation Comments: Writing comments for API documentation.

Java Variables

- Declaring & Initializing Variables: How to declare and assign values.
- Variable Scope: Local, instance, and class-level scope.

Java Data Types

- Primitive Data Types: int, char, float, etc.
- Non-primitive Data Types: Arrays, strings, classes.

- Type Inference: Using var to declare variables.

Java Type Casting

- Implicit & Explicit Casting: Converting between compatible data types.
- Type Conversion: Between int, float, double, etc.

Java Operators

- Arithmetic & Relational Operators: +, -, *, /, >, <, ==.
- Logical Operators: AND, OR, NOT.

Java Strings

- String Declaration & Initialization: Creating string objects.
- String Methods: length(), substring(), indexOf(), etc.

Java Math

- Math Class: Methods like Math.sqrt(), Math.abs().
- Basic Math Operations: Addition, subtraction, multiplication, division.

Java Booleans

- Boolean Data Type: true or false.
- Logical Expressions: Combining conditions using logical operators.

Java Control Flow Statements

- If...Else & Switch: Conditional branching.
- Nested Control Statements: Combining conditions and loops.

Java Loops

- While, For & Do-While: Looping through code multiple times.

Java Break & Continue

- Break Statement: Exiting a loop prematurely.
- Continue Statement: Skipping iterations in a loop.

Java Arrays

- Array Declaration & Initialization: Creating and filling arrays.
- Multidimensional Arrays: 2D and 3D arrays.

Level 2: Java Intermediate Concepts

Java Methods

- Defining & Calling Methods: Modularizing code into reusable blocks.
- Return Types & Parameters: Using void or returning values.

Java Method Parameters

- Passing Parameters: Sending values to methods.
- Varargs: Handling variable numbers of arguments.

Java Method Overloading

- Overloading by Parameters: Defining multiple methods with the same name.
- Overloading with Different Data Types: Same method name, different parameter types.

Java Scope

- Method, Class & Block Scope: Where variables are visible.

Java Recursion

- Recursive Methods: A method that calls itself.
- Recursion vs Iteration: Different problem-solving approaches.

Java Classes

- Defining Classes & Objects: Creating blueprints for objects.
- Access Modifiers: Controlling visibility (public, private).

Java OOP (Object-Oriented Programming)

- Principles of OOP: Encapsulation, Inheritance, Polymorphism, and Abstraction.
- OOP vs Procedural Programming: Modular vs linear programming.

Java Class Attributes

- Instance & Static Variables: Variables at object vs class level.
- Final Variables: Constants that can’t be changed.

Java Class Methods

- Instance & Static Methods: Methods called on objects vs the class.
- Overriding Methods: Changing behavior in subclasses.

Java Constructors

- Default & Parameterized Constructors: Creating objects with specific values.
- Constructor Overloading: Defining multiple constructors.

Java Modifiers

- Access Modifiers: public, private, protected for variables and methods.
- Non-access Modifiers: static, final, abstract for method and class behavior.

Java Encapsulation

- Getters & Setters: Accessing private fields through public methods.
- Benefits of Encapsulation: Data hiding and security.

Java Packages & APIs

- Creating Packages: Grouping related classes together.
- Importing Packages: Using Java's built-in libraries.

Java Inheritance

- Types of Inheritance: Single, multilevel, hierarchical.
- super Keyword: Accessing parent class methods and constructors.

Java Polymorphism

- Compile-time vs Runtime Polymorphism: Method overloading and overriding.
- Method Overloading vs Overriding: Different forms of method behavior.

Java Inner Classes

- Member & Static Nested Classes: Inner classes within another class.
- Anonymous Classes: Creating classes on the fly.

Java Abstraction

- Abstract Classes & Methods: Defining classes with incomplete methods.
- Interfaces: Contracts for class behavior.

Java Interfaces

- Defining Interfaces: Abstract methods for class implementation.
- Implementing Multiple Interfaces: Achieving multiple inheritance in Java.

Level 3: Java Advanced Features

Java Enums

- Defining Enums: Special data types for a fixed set of constants.
- Using Enums in Switch Statements: Simplifying decision logic.

Java User Input

- Scanner Class: Reading input from the console.
- Handling Input Exceptions: Error handling for invalid input.

Java Date and Time

- Date & Calendar Classes: Working with dates.
- LocalDate & LocalDateTime: Modern date and time API.

Java ArrayList

- Creating ArrayLists: Flexible lists.
- Common ArrayList Methods: add(), remove(), get().

Java LinkedList

- Creating LinkedLists: Dynamic memory allocation.
- ArrayList vs LinkedList: Comparing performance and use cases.

Java List Sorting

- Sorting with Comparable & Comparator: Custom sorting logic.

Java HashMap

- Creating HashMaps: Key-value storage.
- Common Methods: put(), get(), remove().

Java HashSet

- Creating HashSets: Unique elements collection.
- HashSet vs List: Comparing uses and differences.

Java Iterator

- Using Iterators: Looping through collections safely.
- Fail-fast vs Fail-safe Iterators: Handling concurrent modifications.

Java Wrapper Classes

- Autoboxing & Unboxing: Converting between primitives and objects.

Java Exceptions

- Try-Catch Blocks: Handling runtime errors.
- Custom Exceptions: Creating your own exceptions.

Java Regular Expressions (RegEx)

- Pattern & Matcher Classes: Matching patterns in strings.

Java Threads

- Creating Threads: Multithreading using Runnable.
- Thread Synchronization: Managing concurrent access to resources.

Java Lambda Expressions

- Lambda Syntax: Simplifying code with functional interfaces.
- Using Lambdas with Collections: Streamlining operations on lists.

Java Advanced Sorting

- Sorting with Streams: Functional sorting techniques.

Java File Handling

- File Class: Working with file attributes.
- File Paths: Managing files and directories.

Java Files

- Reading & Writing with NIO: Modern file I/O techniques.

Java File Operations

- Creating, Reading, and Deleting Files: Manipulating file content and structure.