Java Tutorial

Log in

Create Account

Basics

OOPs & Interfaces

Collections

Exception Handling

Java Advanced

Practice Java

Why Learn Java?

Java Hello World Program

Java Basics

Java Methods

Java Arrays

Java Strings

Java OOPs Concepts

Java Interfaces

Java Collections

Java Exception Handling

Java Multithreading

Java File Handling

Java Streams and Lambda Expressions

Java IO

Java Synchronization

Java Regex

Java Networking

Java Database Connectivity(JDBC)

Java Memory Allocation

Java Interview Questions

Best Approach to Learn Java

Java Program Examples

Java Coding Practice Problems

Java Quizzes and MCQs

Important Java Projects

Career & Jobs in Java 2025

List of Companies Using Java

Introduction to Java

Writing First Program in Java

Variables and Primitive DataTypes

Non Primitive DataTypes

Type Conversion in Java

Input in Java

Thank You!

What kind of Experience do you want to share?

Javais a high-level,object-oriented programming languageused to build web apps, mobile applications, and enterprise software systems. It is known for itsWrite Once, Run Anywherecapability, which means code written in Java can run on any device that supports the Java Virtual Machine (JVM).

Java syntax and structure is similar to C-based languages likeC++and C#. Its robustness, platform-independent compatibility, and strong memory management have made it a go-to language for developers worldwide.

ThisJava tutorialis designed for bothbeginnersandexperienced professionalsand it covers basic to advanced Java topics.

Do you wish to learn Java in a scheduled manner ?Try our ongoing free courseJava Skillupwith weekly topic coverage, notes, daily quizzes and coding problems.

Do you wish to learn Advance Java in scheduled manner?Try our ongoing free courseAdvance Java Skillupwith weekly topic coverage, notes, daily quizzes and coding problems.

Here is a simple Java program that prints "Hello World".

Java basics form the foundation of your programming journey, covering essential concepts like syntax, data types, variables, loops, and conditionals. Mastering these fundamentals is key to building strong, error-free Java applications:

Java methods are reusable blocks of code that perform specific tasks and help organize your program. They improve code readability, reduce repetition, and make debugging easier:

Java arrays are containers that store multiple values of the same data type in a single variable. They provide an efficient way to manage and access collections of data using index-based positions:

Java Strings represent sequences of characters and are widely used in text processing. They are immutable, meaning once created, their values cannot be changed:

Java follows the Object-Oriented Programming (OOP) paradigm, which organizes code into classes and objects. Core OOP principles like inheritance, encapsulation, polymorphism, and abstraction make Java modular and scalable:

Java interfaces define a contract that classes must follow, specifying method signatures without implementations. They enable abstraction and support multiple inheritance in Java through a clean, structured approach:

Java Collections provide a framework for storing and manipulating groups of objects efficiently. It includes interfaces like List, Set, and Map, along with classes like ArrayList, HashSet, and HashMap:

ava Exception Handling is a mechanism to handle runtime errors, ensuring the program runs smoothly without crashing. It uses keywords like try, catch, throw, throws, and finally to manage exceptions:

Java Multithreading allows concurrent execution of two or more threads, enabling efficient CPU utilization and faster program performance. It is commonly used for tasks like parallel processing and responsive applications:

Java File Handling enables programs to create, read, write, and manipulate files stored on the system. It uses classes from thejava.ioandjava.niopackages for efficient file operations:

Java Streams and Lambda Expressions simplify data processing by enabling functional-style operations on collections. Lambdas provide concise syntax for anonymous functions, while Streams allow efficient filtering, mapping, and reduction of data:

Java IO (Input/Output) provides a set of classes and streams to read and write data from various sources like files, consoles, and network connections. It is part of thejava.iopackage and supports both byte and character streams:

Java Synchronization is used to control access to shared resources in multithreaded environments. It ensures that only one thread can access a critical section at a time, preventing data inconsistency:

Java Regex (Regular Expressions) allows pattern matching and text manipulation using thejava.util.regexpackage. It is powerful for validating, searching, and replacing strings based on specific patterns:

Java Networking enables communication between devices over a network using classes from thejava.netpackage. It supports protocols like TCP and UDP for building client-server applications and data exchange:

Java Memory Allocation refers to how memory is assigned to variables, objects, and classes during program execution. It involves stack and heap memory, with the JVM managing allocation and garbage collection automatically:

Prepare for Java interviews with these commonly asked questions, covering core concepts, OOP, collections, multithreading, exception handling, and frameworks like Spring and Hibernate:

Here’s the step-by-step approach to learn and master Java efficiently and effectively:

1.Understand Java Fundamentals:Start with the basics: what Java is, how it works (JVM, JRE, JDK), and why it’s platform-independent. Learn about Java’s role in building desktop, web, and mobile applications.

2.Set Up the Java Environment:Install the latest JDK and configure your development environment. Use a beginner-friendly IDE like IntelliJ IDEA, Eclipse, or VS Code for writing and testing your code.

3.Learn Core Java Concepts:Focus on essential topics like variables, data types, operators, control structures (if-else, switch, loops), and input/output. Practice writing simple programs to strengthen your understanding.

4.Master Object-Oriented Programming (OOP):Dive into the pillars of OOP—classes, objects, inheritance, polymorphism, encapsulation, and abstraction. Understand how Java handles these concepts and how to apply them effectively in real projects.

5.Explore Collections and Data Structures:Learn about Java's Collection Framework—ArrayList, HashMap, Set, LinkedList, and more. Understand how to use them and when to choose one over the other.

6.Work with Exception Handling:Understand the importance of error handling in Java using try-catch blocks, finally, throw, and custom exceptions. Learn best practices for writing clean, safe code.

7.Get Comfortable with File I/O:Learn how Java reads from and writes to files usingFile,FileReader,BufferedReader,FileWriter, andStreams. Practice creating programs that manage and manipulate file data.

8.Understand Multithreading and Concurrency:Grasp the basics of threads, synchronization, and concurrent programming. Learn how Java handles multiple tasks simultaneously using theThreadclass andExecutorService.

9.Learn Java Libraries and APIs:Familiarize yourself with essential APIs like Java Math, Date & Time (java.time), and String manipulation. Explore popular libraries like Apache Commons, Gson, or Jackson for real-world development.

10.Build Real-World Projects:Apply your knowledge by building console apps, file management systems, or basic desktop GUIs using JavaFX or Swing. Gradually move into web development with Java Servlets, JSP, or Spring Boot.

Java opens doors to a wide range of tech careers across industries—from enterprise software to mobile apps and cloud systems. Here’s a list of top career options for professionals with Java expertise, along with their average salary ranges:

These are some popular companies that use Java in thier workflow:

Already have an account ?Log in

\*Please enter your email address or userHandle.