Java Interview Preparation Roadmap

Java Basics

Variables

Variables store data. Java has different types of variables: int (integer), float (decimal values), double (higher precision decimal), char (single character), boolean (true/false), and String (text).

Data Types

Primitive types (int, float, boolean, etc.) store values directly, while Non-primitive types (String, Arrays, Classes) store references to objects.

Operators

Operators perform operations on variables. Arithmetic (+, -, *, /), Logical (&&, ||, !), Relational (>, <, ==), and Bitwise (&, |, ^) are commonly used.

Conditionals

Conditional statements (if, else, else if, switch) are used for decision-making in Java programs.

Loops

Loops (for, while, do-while, for-each) allow repeated execution of a block of code.

Methods

Methods define reusable blocks of code. Static methods belong to a class, while non-static methods belong to an instance.

Arrays and Collections

Arrays store multiple values of the same type, while collections (ArrayList, HashSet, etc.) provide dynamic data structures.

Object-Oriented Programming (OOP)

Classes & Objects

A class is a blueprint for creating objects. An object is an instance of a class.

Encapsulation

Encapsulation hides data using private access modifiers and provides access through getters and setters.

Inheritance

Inheritance allows a class to inherit properties and methods from another class.

Polymorphism

Polymorphism allows methods to take different forms: Method Overloading (same name, different parameters) and Method Overriding (child class redefines parent method).

Abstraction

Java Interview Preparation Roadmap

Abstraction hides implementation details using abstract classes or interfaces.

Access Modifiers

Java has access modifiers: private (accessible within the class), protected (accessible in subclass), public (accessible anywhere).

Exception Handling

Try-Catch Blocks

Used to handle exceptions and prevent program crashes.

Finally Block

Executes code after try-catch, regardless of exception occurrence.

Throws and Throw

Throws is used to declare exceptions, while throw is used to manually raise an exception.

Custom Exceptions

Allows creation of user-defined exceptions for specific error handling.

File Handling

Reading and Writing Files

Java provides FileReader, FileWriter, BufferedReader, and BufferedWriter for file operations.

BufferedReader and BufferedWriter

Used for efficient file reading and writing.

Serialization and Deserialization

Converts objects into a byte stream (serialization) and restores them (deserialization).

Java Collections Framework

List

Ordered collection of elements (ArrayList, LinkedList).

Set

Unordered collection with unique elements (HashSet, TreeSet).

Map

Key-value pairs for efficient lookups (HashMap, TreeMap).

Queue and Deque

Queue follows FIFO, while Deque allows insertion at both ends.

Java Interview Preparation Roadmap

Iterators

Used to traverse collections (Iterator, ListIterator).

Multithreading

Threads and Runnable Interface

Threads allow concurrent execution, created using Thread class or Runnable interface.

Synchronized Methods

Used to prevent thread conflicts by locking methods.

Locks

More flexible than synchronized, allows explicit control of thread synchronization.

Executors

Manages a pool of threads for better performance.

Java 8+ Features

Lambda Expressions

Concise way to write anonymous functions.

Streams

Used for processing collections efficiently.

Functional Interfaces

Interfaces with a single abstract method (Predicate, Consumer, Supplier).

Default and Static Methods in Interfaces

Allows method implementations in interfaces.

Optional Class

Used to avoid null pointer exceptions.

Method References

Shorthand for calling methods using :: operator.

Frameworks and Libraries

Spring Framework

Popular framework for Java backend development.

Hibernate

Object-Relational Mapping (ORM) framework.

Java Interview Preparation Roadmap

Apache Maven and Gradle

Dependency management tools.

JavaFX

Used for building desktop applications.

Debugging

System.out.println()

Simple way to debug by printing values.

Debugging in IDEs

Using breakpoints and stepping through code in IDEs.

Loggers

Logging frameworks like Log4j and SLF4J for better debugging.

Others

JVM, JRE, and JDK

JVM runs Java programs, JRE contains JVM, and JDK includes JRE + development tools.

Garbage Collection

Automatic memory management in Java.

Inner Classes

Classes within classes, useful for encapsulation.

Anonymous Classes

Inner classes without a name, useful for short implementations.

Recursion

A method calling itself to solve a problem.

Java Memory Model

Defines how Java handles memory and threads.

Annotations

Metadata to provide additional information about code.

Advanced Topics

Generics

Allows type-safe operations on objects.

Java Interview Preparation Roadmap

Reflection API

Used to inspect and modify classes at runtime.

Networking

Java supports sockets and HTTP/HTTPS communication.

Regular Expressions

Pattern matching in Java.

Dependency Injection

A design pattern used to manage dependencies in applications.