

## JAVA INTERVIEW CONCEPTS

### 1. Core Java Fundamentals:

- JVM, JDK, JRE differences.
- Java data types (primitive vs reference types).
- Java memory model (Stack, Heap, Method Area).
- Access modifiers (public, private, protected, default).
- Static vs non-static.
- Final keyword (variable, method, class).
- Constructor vs method.
- this and super keyword.
- String vs StringBuilder vs StringBuffer.

### 2. Object-Oriented Programming:

- Class and Object.
- Inheritance (single, multilevel, hierarchical).
- Abstraction (abstract class vs interface).
- Encapsulation.
- Polymorphism (compile-time vs runtime).
- Method overloading and overriding.
- Interfaces vs Abstract classes.
- default and static methods in interfaces (Java 8+).

### 3. Exception Handling:

- try, catch, finally.
- throw vs throws.
- Checked vs unchecked exceptions.
- Custom exceptions.

### 4. Collections Framework:

- List, Set, Map interfaces.
- ArrayList vs LinkedList.
- HashSet vs TreeSet vs LinkedHashSet.
- HashMap vs TreeMap vs LinkedHashMap.
- Iterator vs ListIterator.
- fail-fast vs fail-safe iterators.
- Comparable vs Comparator.

### 5. Multithreading & Concurrency:

- Thread lifecycle.
- Runnable vs Thread class.

- Synchronization (synchronized keyword).
- volatile keyword.
- Deadlock, livelock, starvation.
- wait(), notify(), notifyAll().
- ExecutorService, Callable, Future.
- ConcurrentHashMap and other concurrent utilities.

#### 6. Java 8+ Features:

- Lambda expressions.
- Functional interfaces.
- Streams API (intermediate vs terminal operations).
- Optional class.
- Method references.
- Default and static methods in interfaces.

#### 7. Memory Management & GC:

- Garbage Collection process.
- finalize() method (deprecated).
- Strong, Weak, Soft, Phantom references.

#### 8. Design Patterns Basics:

- Singleton pattern.
- Factory pattern.
- Strategy pattern.
- Observer pattern.

#### 9. Java I/O & Serialization:

- File handling basics.
- Byte stream vs character stream.
- Serialization and transient keyword.

#### 10. Miscellaneous:

- immutability concept.
- enum usage.
- Reflection API basics.
- String pool concept.
- Deep copy vs shallow copy.
- Best coding practices and common pitfalls.