JAVA INTERVIEW REVISION NOTES

1) OOPS Concepts (Simple)

Concept	Meaning
Encapsulation	Binding data + methods in a class; hiding details
Inheritance	One class acquires properties of another
Polymorphism	Same method behaves differently
Abstraction	Showing essential things and hiding implementation
Association	Relationship between two classes
Aggregation	Weak relationship (child can exist independently)
Composition	Strong relationship (child cannot exist independently)

2) Exception Handling

Keyword	Meaning
Exception	Abnormal event that interrupts program
throw	Used to manually throw exception
throws	Declares exception in method signature
finally	Always runs (even if exception occurs)

3) Introduction (2-Min Short)

My name is Abhay. I have completed my education in _____. I have strong knowledge of Core Java, SQL, OOPs, Collections, and API basics.
I have worked on small projects with CRUD operations using Java + MySQL.
I enjoy solving problems like arrays, strings, patterns.
I am self-motivated, quick learner, and preparing for Java roles (10 LPA+).
My skills and learning attitude make me a good fit for this job.

JAVA PROGRAMS

A) Number Programs

Program	Logic
Factorial	Multiply numbers 1 to n
Sum of digits	Add each digit
Greatest	Compare using if/else
Smallest	Same as above
Prime	Divisible by 1 & itself
Odd or Even	num % 2 == 0
Fibonacci	Sum of previous two numbers

B) Pattern Programs

Examples:

- Star

- Pyramid
- Butterfly
- Right triangle

(Ask if you need code for all.)

C) Array Programs

Requirement	Description
Zig-zag	Alternate elements from 2 arrays
Merge arrays	Combine both arrays
Reverse array	Loop from last to first
Find duplicate	Compare and count
Remove duplicate	Use Set
Count duplicates	Example: {1,2,3,2,1,3} → 1-2, 2-2, 3-2
Bubble sort	Swap adjacent elements
Swap digits	Swap using temp
Left rotation	Move first element to last
Right rotation	Move last element to first

D) String Programs

Task	Logic
Substring	"manikandan" → "kandan"
Palindrome	Reverse and compare
Reverse String	Use loop / StringBuilder

Task	Logic
Duplicate chars	Frequency approach
Count characters	Loop and count
Remove duplicates	Use Set
Separate char/num/special	Example: "1abc287" $\rightarrow num = "187", char = "abc", spl =$ ""

E) Collections Programs

Conversion	Method
Array → ArrayList	Arrays.asList(arr)
ArrayList → Array	list.toArray()
Set → List	new ArrayList<>(set)
List → Set	new HashSet<>(list)
Remove duplicates	Use Set

COLLECTION THEORY

Type	Meaning
List	Ordered, allows duplicates
Set	No duplicates
Map	Key-value pairs

MAP TYPES

Feature	HashMap	LinkedHashMap	TreeMap
Order	No order	Insertion order	Sorted order
Null Keys	1 allowed	1 allowed	Not allowed
Speed	Fastest	Moderate	Slowest

SQL BASIC QUERIES

- SELECT → `SELECT * FROM table;`
- WHERE → `SELECT * FROM emp WHERE id=1;`
- ORDER BY → `ORDER BY name DESC;`
- GROUP BY → `SELECT city, COUNT(*) FROM emp GROUP BY city;`
- JOIN → `SELECT * FROM emp e JOIN dept d ON e.did = d.id;`
- INSERT → `INSERT INTO table VALUES(...);`
- UPDATE → `UPDATE table SET name='A' WHERE id=1;`
- DELETE → `DELETE FROM table WHERE id=1;`

API BASICS

1) What is an API?

An interface that allows two software applications to communicate.

2) REST vs SOAP

Feature	REST	SOAP
Format	JSON, XML	Only XML
Speed	Fast	Slow
Flexibility	High	Strict
Protocol	HTTP/HTTPS	SOAP protocol
Complexity	Easy	Complex

3) Response Time

The time taken by server to respond to client.

4) HTTP Request

A message sent from client to server asking for data/action.

5) HTTP Methods

Method	Use
GET	Read Data
POST	Create Data
PUT	Full Update
PATCH	Partial Update
DELETE	Delete Data

6) PUT vs PATCH

Feature	PUT	PATCH
Update	Full	Partial
Replaces	Entire object	Only changed fields
Use Case	Replace user	Update email only

7) Why POST?

Used to create new data.
More secure than GET (data not visible in URL).

8) Status Codes

Code	Meaning
200	OK
201	Created
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
500	Server Error
502	Bad Gateway

9) Request Body Formats

Format	Example
JSON	{"id":1}
XML	1
form-data	Used for images/files
x-www-form-urlencoded	key=value
Raw Text	Plain text

Spring & Spring Boot Notes

1) What is Spring?

Spring is a powerful, lightweight, open-source Java framework used to build enterprise-level applications easily and efficiently.

2) Inversion of Control (IoC)

Spring container manages object creation and their dependencies.

3) Dependency Injection (DI)

Objects are injected by the container instead of being created manually using `new`.

4) Spring Ecosystem Includes:

- **Spring Boot** – simplifies Spring application setup and deployment.
- **Spring MVC** – for building web applications.
- **Spring Data JPA** – simplifies database operations.
- **Spring Security** – for authentication and authorization.
- **Spring Cloud** – for building microservices.
- **Spring Batch**, **Spring Integration**, etc.

5) What is Spring Boot?

Spring Boot is a framework built on top of the Spring Framework that helps developers create stand-alone, production-ready Spring applications quickly and easily, with minimal configuration.

Spring Boot simplifies Spring development by eliminating the need for complex XML configurations — it auto-configures your application based on the dependencies you add.

6) Key Features of Spring Boot

Feature	Description
1. Auto Configuration	Automatically configures your Spring application based on the dependencies present in the classpath.
2. Starter Dependencies	Provides pre-defined dependency sets like `spring-boot-starter-web`, `spring-boot-starter-data-jpa`, etc.
3. Embedded Servers	Comes with built-in servers like Tomcat, Jetty, and Undertow — no need for external deployment.
4. Stand-alone Applications	Just run your app with a simple `main()` method — no need for WAR files.
5. Actuator	Provides monitoring and management endpoints for health, metrics, and logs.

Feature	Description
6. Production Ready	Includes features like health checks, metrics, and security by default.
7. Less Configuration	Uses annotations like `@SpringBootApplication` instead of long XML files.

JAVA + JDBC + SERVLET + SPRING + SPRING BOOT – FULL INTERVIEW NOTES

⭐ CORE JAVA TOPICS

1) OOPS

- Class, Object
- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

2) Array

- Stores multiple values.
- Fixed size.
- Indexed access.
- Used for sorting, searching, rotating, duplicates etc.

3) String

- Immutable object.
- Common operations: reverse, substring, split.
- Mutable versions: StringBuilder, StringBuffer.

4) Exception Handling

- Exception = runtime error.
- try–catch → handle exception
- throw → manually throw
- throws → declare
- finally → always executes

5) Collections

- List → duplicates allowed
- Set → no duplicates
- Map → key–value

6) Multithreading

- Multiple tasks run together.
- Thread creation: extends Thread / implements Runnable
- Methods: start(), run(), sleep(), join()

7) Java 8

- Lambda Expressions
- Stream API

- Functional Interfaces
- Default & Static Interface Methods
- Optional class

⭐ JDBC (Java Database Connectivity)

JDBC 5 Steps

1. Load Driver → `Class.forName("org.postgresql.Driver")`
2. Create Connection → DriverManager
3. Create Statement
4. Execute Query
5. Close Connection

Types of JDBC Driver

- Type 4 → Thin Driver (most used)

⭐ SERVLET

What is Servlet?

A Java program that handles HTTP requests and returns responses.

How to Create Servlet

- Extend HttpServlet
- Override doGet() / doPost()
- Configure using @WebServlet or web.xml

Servlet Lifecycle

1. init()
2. service()
3. destroy()

⭐ JSP

- HTML + Java
- Easy UI creation
- Works with Servlet

⭐ HIBERNATE / JPA

ORM

Object Relational Mapping → maps Java objects to database tables.

Entity

A Java class mapped to a table using @Entity.

Caching

- Level 1 → Session
- Level 2 → Optional

⭐ SPRING CORE

1) What is Spring?

Spring is a lightweight Java framework for enterprise apps.
It provides IoC, DI, AOP, MVC, Security, JPA etc.

2) Difference Between Spring and Spring Boot

Spring	Spring Boot
Manual configuration	Auto configuration
Needs external server	Embedded server
More code	Less code
XML-based	Annotation-based
Complex setup	Quick setup

3) What is IoC?

Inversion of Control → Spring controls object creation.

4) What is DI?

Dependency Injection → Spring provides required objects, no need of `new`.

5) What is Spring Container?

Manages object creation, DI, lifecycle.

6) Types of Spring Container

- BeanFactory
- ApplicationContext (most used)

7) What is MVC?

Model–View–Controller
Used to build web apps.

Spring MVC

Spring's MVC implementation.

⭐ SPRING BOOT

8) What is Spring Boot?

It makes Spring development faster using:

- auto configuration
- starter dependencies
- embedded servers

Why use Spring Boot?

- No XML
- Fast development
- Less code
- Production-ready features (Actuator)

9) What is REST?

Representational State Transfer

- Uses HTTP
- Stateless
- Used to build JSON-based APIs

10) What is Web Service?

A service that allows communication between two applications over a network.

11) Difference Between API and REST API

API	REST API
Interface to communicate	API that follows REST rules
Any format	Mostly JSON
Not always stateless	Stateless

12) HTTP Status Codes

Category Codes

- **1xx** → informational
- **2xx** → success
- **3xx** → redirection
- **4xx** → client error
- **5xx** → server error

Common Codes

- 200 → OK
- 201 → Created
- 204 → No Content
- 400 → Bad Request
- 401 → Unauthorized
- 403 → Forbidden
- 404 → Not Found
- 500 → Internal Server Error

13) ResponseEntity (class)

Used to send:

- status code
- body
- message

14) JpaRepository (interface)

Used in Spring Data JPA for:

- CRUD operations
- Pagination
- Custom queries

⭐ IMPORTANT SPRING BOOT ANNOTATIONS

Annotation	Meaning
@Component	Generic bean
@ComponentScan	Scans packages
@Autowired	Injects dependency
@Value	Inject values
@Configuration	Bean configuration class
@SpringBootApplication	Main Spring Boot annotation
@Repository	DAO layer
@ControllerAdvice	Global exception handling
@ExceptionHandler	Handle specific exception
@RestController	Returns JSON
@Controller	Returns HTML/JSP
@Scope	Bean scope
@Primary	Highest priority bean
@Qualifier	Select specific bean
@Bean	Create bean manually
@Transactional	DB transaction
@RequestParam	Read query parameter
@PathVariable	Read path variable
@RequestMapping	Map URL pattern

⭐ COMPANY INTERVIEW QUESTIONS (Giant Leap + Others)

1) Why we use Spring Boot over Hibernate?

Hibernate = ORM only.
Spring Boot = web, JPA, server, auto config → full application framework.
Spring Data JPA makes CRUD very easy using JpaRepository.

2) Difference between Spring and Spring Boot

(Already given above.)

3) What is JPA?

Java Persistence API.
Hibernate is its implementation.

4) SQL → 3rd Max Salary

```sql
SELECT DISTINCT salary
FROM employee
ORDER BY salary DESC
LIMIT 2,1;

# END
```