

HackForge

Core Java Certification Training

40 Hours

Live Online in English Andhra Pradesh | Telangana | Tamil Nadu | Karnataka

Phone :9884789292

Core Java

Module 1: Java Fundamentals

- * History & Features of Java
- * JVM vs JRE vs JDK
- * Java Program Structure
- * Compilation & Execution Flow
- * Data Types (Primitive & Non-Primitive)
- * Variables (Local, Instance, Static)
- * Type Casting (Implicit & Explicit)
- * Operators
- * Control Statements
 - * if / else
 - * switch
 - * for loop
 - * while loop
 - * do-while loop
 - * break & continue

Module 2: Object-Oriented Programming (OOPs)

Core Concepts

- * Class & Object
- * Encapsulation
- * Inheritance
- * Polymorphism
- * Abstraction

Inheritance Types

- * Single
- * Multilevel
- * Hierarchical

Important Keywords

- * this
- * super
- * final
- * static

Module 3: Constructors

- * Default Constructor
- * Parameterized Constructor
- * Constructor Overloading
- * Constructor Chaining
- * Private Constructor

Module 4: Strings & Wrapper Classes

String Handling

- * String class
- * String immutability
- * equals() vs ==
- * hashCode()
- * String methods
- * String vs StringBuilder vs StringBuffer

Wrapper Classes

- * Primitive vs Wrapper
- * Autoboxing & Unboxing
- * Use in Collections

Module 5: Exception Handling

Concepts

- * What is Exception
- * Checked vs Unchecked Exceptions
- * Throwable hierarchy

Keywords

- * try
- * catch
- * finally
- * throw
- * throws

Custom Exceptions

- * User-defined exceptions

- * Best practices

Module 6: Interfaces & Abstract Classes

- * Interface vs Abstract class
- * Multiple inheritance using Interface
- * Functional Interface
- * Marker Interface
- * Default & static methods in interfaces

Module 7: Java Collections Framework

Collection Hierarchy

- * List
- * Set
- * Queue
- * Map

List

- * ArrayList
- * LinkedList
- * Vector
- * Stack

Set

- * HashSet
- * LinkedHashSet
- * TreeSet

Map

- * HashMap
- * LinkedHashMap
- * TreeMap
- * Hashtable

Utilities

- * Comparable vs Comparator
- * Iterator
- * Collection vs Collections

Module 8: Java 8 Features

Lambda Expressions

Functional Interfaces

- * Predicate
- * Consumer
- * Supplier
- * Function

Stream API

- * Stream creation
- * filter
- * map
- * reduce
- * forEach
- * Collectors

Other Features

- * Optional class
- * Method references
- * New Date & Time API

Module 9: Multithreading & Concurrency

- * What is Thread
- * Thread Life Cycle

Creating Threads

- * Thread class
- * Runnable interface

Synchronization

- * Thread safety
- * Deadlock
- * wait()
- * notify()
- * notifyAll()

Advanced Concepts

- * Executor framework
- * volatile keyword

Module 10: File Handling & Serialization

- * File class
- * Byte streams
- * Character streams
- * Serialization
- * Deserialization
- * transient keyword

Module 11: Java Memory Management

- * Stack vs Heap
- * JVM Memory Areas
- * Garbage Collection
- * Memory leaks
- * finalize()

Module 12: equals(), hashCode(), toString()

- * Contract rules

- * Usage in HashMap
- * Usage in HashSet
- * Overriding best practices

Module 13: Important Java Keywords

- * static
- * final
- * transient
- * volatile
- * synchronized
- * native
- * strictfp

Module 14: Design Principles & Best Practices

- * SOLID principles (core understanding)
- * Immutability
- * Defensive copying
- * Clean code practices
- * Readable class design

Module 15: JDBC Connectivity

- * JDBC Architecture
- * Driver types
- * Connection setup
- * Statement vs PreparedStatement
- * ResultSet handling
- * CRUD operations

MySQL

Module 1: Introduction to Databases

- What is DBMS
- What is RDBMS
- Features of MySQL
- Installing MySQL Server & Workbench
- Creating database
- Selecting database

Commands:

```
CREATE DATABASE academy;  
USE academy;  
SHOW DATABASES;
```

Module 2: Data Types in MySQL

- Numeric types (INT, BIGINT, FLOAT, DECIMAL)
 - String types (CHAR, VARCHAR, TEXT)
 - Date & time types (DATE, DATETIME, TIMESTAMP)
 - Boolean type
 - ENUM & SET (optional)
-

Module 3: Table Creation & Structure

- CREATE TABLE

- DESCRIBE TABLE
- ALTER TABLE
- DROP TABLE
- TRUNCATE TABLE
- Rename table

Example:

```
CREATE TABLE students (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(100),  
  email VARCHAR(100),  
  age INT  
);
```

Module 4: CRUD Operations (Core SQL)

INSERT

```
INSERT INTO students(name, age)  
VALUES ('Praveen', 22);
```

SELECT

```
SELECT * FROM students;
```

UPDATE

```
UPDATE students  
SET age = 23  
WHERE id = 1;
```

DELETE

```
DELETE FROM students
```

WHERE id = 1;

Module 5: Constraints

- PRIMARY KEY
- FOREIGN KEY
- UNIQUE
- NOT NULL
- DEFAULT
- CHECK (concept)

Example:

email VARCHAR(100) UNIQUE

Module 6: Filtering Data

- WHERE clause
- AND, OR, NOT
- BETWEEN
- IN
- LIKE
- IS NULL

Example:

```
SELECT * FROM students  
WHERE age BETWEEN 18 AND 25;
```

Module 7: Sorting & Limiting Data

- ORDER BY ASC / DESC
- LIMIT
- DISTINCT

Example:

```
SELECT * FROM students
ORDER BY age DESC
LIMIT 5;
```

Module 8: Aggregate Functions

- COUNT()
- SUM()
- AVG()
- MIN()
- MAX()

Example:

```
SELECT COUNT(*) FROM students;
```

Module 9: GROUP BY & HAVING

- GROUP BY usage
- HAVING vs WHERE difference

Example:

```
SELECT age, COUNT(*)
FROM students
GROUP BY age;
```

Module 10: Joins (Very Important)

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- SELF JOIN (optional)

Example:

```
SELECT students.name, courses.title  
FROM students  
INNER JOIN courses  
ON students.course_id = courses.id;
```

Module 11: Keys & Relationships

- Primary key concept
- Foreign key relationships
- One-to-One
- One-to-Many
- Many-to-Many

Example:

```
FOREIGN KEY (course_id)  
REFERENCES courses(id)
```

Module 12: Indexes

- What is indexing
- Creating index
- Unique index
- Composite index
- Performance benefits

Example:

```
CREATE INDEX idx_email  
ON students(email);
```

Module 13: Views

- What is a view
- Creating view
- Updating view
- Dropping view

Example:

```
CREATE VIEW active_students AS  
SELECT * FROM students  
WHERE status = 'active';
```

Module 15: Functions

- Creating functions
- Returning values

Example:

```
CREATE FUNCTION total_students()
```

RETURNS INT
RETURN (SELECT COUNT(*) FROM students);

Module 16: Transactions

- START TRANSACTION
- COMMIT
- ROLLBACK
- SAVEPOINT
- ACID properties

Example:

```
START TRANSACTION;  
UPDATE accounts SET balance = balance - 500  
WHERE id = 1;  
COMMIT;
```

Module 17: Subqueries

- Single-row subquery
- Multi-row subquery
- Nested queries

Example:

```
SELECT name  
FROM students  
WHERE age > (SELECT AVG(age) FROM students);
```

Module 18: Advanced Queries

- EXISTS
- ANY
- ALL
- CASE statement

Example:

```
SELECT name,  
CASE  
  WHEN age >= 18 THEN 'Adult'  
  ELSE 'Minor'  
END  
FROM students;
```

HackForge

SpringBoot Certification Training

40 Hours

Live Online | Chennai, Bangalore, Hyderabad & Nationwide Enrolment

Phone :9884789292

Social Media Links:

<https://www.linkedin.com/company/hackforge-academy>

https://www.instagram.com/hackforge_english/

https://www.youtube.com/@hackforge_code

**Daily coding whatsapp community in java,React,JavaScript,Python
+919884789292**

hackforge.in