



# 30-Day Java Programming Syllabus

## Course Information Course Title:

**Course Title:** Java Programming Fundamentals with Project-Based Learning

**Duration:** 30 Days

**Format:** Daily Hands-On Sessions

## Course Description:

This intensive 30-day training program is designed to teach students the fundamentals of Java programming through hands-on coding and real-world projects. The course focuses on building a strong foundation in Java syntax, object-oriented programming (OOP), collections, file handling, and basic multithreading. By the end of the course, students will develop a portfolio-ready **Library Management System** using Core Java concepts.

## Learning Objectives:

By the end of this course, students will be able to:

- Understand Java syntax and programming structure
- Apply object-oriented programming concepts (OOP) effectively
- Work with arrays, collections (List, Map, Set), and exception handling
- Implement file handling and basic multithreading
- Write modular and maintainable code using best practices
- Build and document a Java-based Library Management System
- Use Java 8 features like Lambdas and Stream API

## Required Materials:

- Laptop with reliable internet access

- 
- Java Development Kit (JDK 17 or latest)
  - IDE (e.g., IntelliJ IDEA, Eclipse, or VS Code with Java extensions)
  - Text editor (VS Code recommended)
  - GitHub account (free)
  - PDF or Word editor (for documentation and reporting)

## Course Schedule:

### Week 1: Java Basics & OOP Concepts

**Goal:** Build a strong foundation in Java syntax and object-oriented programming.

#### Day 1–2: Java Setup & Basics

- Install JDK and IDE (e.g., IntelliJ or Eclipse)
- Hello World, Data types, Variables, Type casting, Operators

#### Day 3: Control Flow

- if-else, switch
- Loops: for, while, do-while

#### Day 4: Methods & Arrays

- Method declaration and overloading
- 1D and 2D Arrays

#### Day 5: OOP - Part 1

- Classes, Objects, Constructors
- this keyword

#### Day 6: OOP - Part 2

- Inheritance
- Method Overriding
- super keyword

#### Day 7: Practice

- Task: Build a simple calculator using OOP

---

## Week 2: Advanced OOP, Collections, Exception Handling

**Goal:** Learn real-world coding: collections, error handling, abstraction.

**Day 8:**

- Polymorphism & Abstraction (Abstract Classes, Interfaces)

**Day 9:**

- Encapsulation, Getters/Setters, Access Modifiers

**Day 10:**

- Exception Handling (try, catch, finally, throw, throws)

**Day 11–12:**

- Java Collections (List, Set, Map)

**Day 13:**

- Wrapper Classes & Autoboxing

**Day 14: Mini Project**

- Student Management System using OOP + Collections

## Week 3: File Handling, Java 8 Features, Multithreading

**Goal:** Explore Java APIs and backend logic.

**Day 15:**

- File Handling (FileReader, FileWriter)

**Day 16:**

- Multithreading Basics (Thread class, Runnable)

**Day 17:**

- Java 8 Features (Lambda, Functional Interfaces)

**Day 18:**

- Streams + Date API (LocalDate)

**Day 19:**

- Enums, Static, Final

**Day 20–21:**

- Practice + Mock Test
- Plan Project

## Week 4: Final Project – Library Management System

**Project Overview:** Build a Library Management System using Core Java, OOP, Collections, and File Handling.

**Day 22: Project Planning**

- Define features
- Create packages and classes (Book, User, LibraryService, MainApp)

**Day 23: Core Features (Add/View/Search Books)**

- Use OOP and ArrayList for book storage

**Day 24: Issue/Return Books**

- Track issued books with Map

**Day 25: Collections & Optional File Handling**

- Use ArrayList/HashMap
- File writing/reading

**Day 26: Java 8 Features**

- Use Stream API and Optional

**Day 27: Polish Code and Test**

- Clean code
- Handle exceptions

---

### Day 28: Final Touches + README

- Menu-driven MainApp
- Project documentation

## Week 5: Project Presentation & Career Preparation

### Day 29: Final Project Polishing & Presentation Rehearsal

- Final code review and bug fixing
- Implement feedback and clean up code
- Add finishing touches: validation, documentation, etc.
- Create and polish project README for GitHub
- Rehearse your project presentation
- Peer review and feedback on presentation clarity

### Day 30: Final Project Presentations

- Students present their **Library Management System**
- Structured peer and instructor feedback
- Course recap and key concept reinforcement
- Talk: “**What’s next in Java?**” (Spring Boot, APIs, Full-Stack Dev)
- Discussion on **career paths** in backend development

### Assessment Methods

- **Daily Assignments:** 40%
- **Class Participation:** 10%
- **Final Project:** 50%

### Weekend Assignments

To reinforce weekly concepts, **weekend assignments** will include hands-on coding, mini-projects, and review challenges.