

That's awesome that you're diving into Java programming! Here's a structured routine and project ideas to help you progress from basic to advanced levels in 90 days:

1. **Basic Foundations (Days 1-30)**

Week 1-2: Getting Started

- **Day 1-2***: Introduction to Java, Setting up IDE (Eclipse/IntelliJ/VS Code), Hello World program.
- **Day 3-4***: Variables, Data Types, and Operators.
- **Day 5-6***: Control Flow (if-else, switch-case, loops).
- **Day 7***: Practice with simple programs (calculator, basic algorithms).

Week 3-4: Object-Oriented Programming (OOP)

- **Day 8-9***: Classes and Objects.
- **Day 10-11***: Constructors and Methods.
- **Day 12-13***: Inheritance and Polymorphism.
- **Day 14***: Practice with OOP concepts (building simple class hierarchies).

Week 5-6: Collections and Error Handling

- **Day 15-16***: Arrays and ArrayList.
- **Day 17-18***: HashMap and HashSet.
- **Day 19-20***: Exception Handling (try-catch-finally).
- **Day 21***: Practice with collections and error handling.

Week 7-8: File Handling and Advanced Concepts

- **Day 22-23***: Reading and Writing Files.
- **Day 24-25***: Interfaces and Abstract Classes.
- **Day 26-27***: Enums and Generics.
- **Day 28***: Practice with file operations and advanced concepts.

Week 9-10: Multithreading and Concurrency

- **Day 29-30***: Basics of Threads.
- **Day 31-32***: Synchronization and Thread Safety.
- **Day 33-34***: Executors and Thread Pools.
- **Day 35***: Practice with multithreading concepts.

Week 11-12: Advanced Topics and Project Development

- **Day 36-37***: JDBC and Database Connectivity.
- **Day 38-39***: Networking (Sockets, HTTP).
- **Day 40-41***: JavaFX or Swing for GUI (optional but recommended).
- **Day 42***: Work on a larger project integrating concepts learned.

2. **Project Ideas (Throughout 90 Days)**

- **Week 1-2***: Simple Calculator App
 - Implement basic arithmetic operations with a simple GUI.
- **Week 3-4***: Address Book

- Manage contacts using OOP principles (classes for contacts, file handling for persistence).
- **Week 5-6**: Todo List Application
 - CRUD operations using ArrayList or HashMap, with file storage for persistence.
- **Week 7-8**: File Encryption/Decryption Tool
 - Implement file handling along with algorithms like AES for encryption/decryption.
- **Week 9-10**: Chat Application
 - Implement multithreading for handling multiple clients and networking for communication.
- **Week 11-12**: Library Management System
 - JDBC for database connectivity, GUI for managing books, users, and transactions.

Tips for Improvement:

- **Code Review and Refactoring**: Regularly review your code, refactor to improve readability and efficiency.
- **Read Documentation and Articles**: Dive into official Java documentation and blogs/articles to deepen your understanding.
- **Practice Algorithms and Data Structures**: Use platforms like LeetCode, HackerRank, or Project Euler for algorithmic challenges.
- **Build Personal Projects**: Work on projects that interest you outside the suggested ones to apply Java in areas you're passionate about.

By following this routine and working on diverse projects, you'll build a solid foundation in Java programming and gradually move towards advanced topics. Good luck with your learning journey!