📅 **Week 1: Core Java Fundamentals**

**Project**: Library Management System (OOP Based)

| **Day** | **Topics** | **Daily Task** | **Project Contribution** |
| --- | --- | --- | --- |
| 1 | Java Syntax, Variables, Operators | Write a program to calculate BMI | Design Book class |
| 2 | Control Flow (if-else, loops) | Number guessing game | Add input validation |
| 3 | Methods, Constructors, this | Student grade calculator | Implement Member class |
| 4 | OOP Basics (Classes/Objects) | Bank account simulation | Create Library class |
| 5 | Inheritance, Polymorphism | Shape area calculator | Add inheritance hierarchy |
| 6 | Abstraction (Interfaces/Abstract) | Payment system interface | Design loan interface |
| 7 | Encapsulation, Packages | Secure employee data class | Finalize access controls |

📅 **Week 2: Collections & Exception Handling**

**Project**: Inventory Management System

| **Day** | **Topics** | **Daily Task** | **Project Contribution** |
| --- | --- | --- | --- |
| 8 | ArrayList/LinkedList | Employee directory | Implement product list |
| 9 | HashSet/TreeSet | Unique word counter | Create category set |
| 10 | HashMap/TreeMap | Student marks tracker | Build inventory mapping |
| 11 | Iterator/Comparable | Sort employee records | Add sorting by price |
| 12 | Exception Handling | File reader with try-catch | Add error handling |
| 13 | Custom Exceptions | Age validator | Create LowStockException |
| 14 | Collections Algorithms | Frequency counter | Implement search feature |

📅 **Week 3: Advanced Core Java**

**Project**: File-Based Student Database

| **Day** | **Topics** | **Daily Task** | **Project Contribution** |
| --- | --- | --- | --- |
| 15 | Generics Basics | Generic storage box | Make DAO generic |
| 16 | Wildcards | Universal list printer | Add report generator |
| 17 | File I/O (Byte Streams) | File copy utility | Implement save/load |
| 18 | Character Streams | Log file analyzer | Add export to CSV |
| 19 | Serialization | Save game state | Serialize student data |
| 20 | Annotations | Runtime annotation processor | Add @StudentInfo |
| 21 | Reflection | Dynamic class loader | Build plugin system |

📅 **Week 4: Multithreading Foundation**

**Project**: Multi-Threaded Bank Simulation

| **Day** | **Topics** | **Daily Task** | **Project Contribution** |
| --- | --- | --- | --- |
| 22 | Thread Basics | Number counter threads | Implement account class |
| 23 | Runnable vs Thread | Clock ticker | Add transaction logger |
| 24 | Thread States | State monitoring tool | Simulate bank queues |
| 25 | Synchronization | ATM withdrawal simulator | Add thread-safe ops |
| 26 | wait()/notify() | Producer-consumer | Implement transfers |
| 27 | Thread Priorities | CPU-intensive task manager | Add priority banking |
| 28 | Thread Groups | Bulk transaction handler | Group loan processors |

📅 **Week 5: Advanced Multithreading**

**Project**: Concurrent Web Crawler

| **Day** | **Topics** | **Daily Task** | **Project Contribution** |
| --- | --- | --- | --- |
| 29 | ExecutorService | Parallel prime checker | Create thread pool |
| 30 | Callable/Future | Async file processor | Add URL fetcher |
| 31 | ConcurrentHashMap | Word frequency analyzer | Build visited URLs cache |
| 32 | Atomic Variables | Atomic counter | Implement stats tracker |
| 33 | Lock Interface | Reentrant lock demo | Add resource locking |
| 34 | Deadlock Prevention | Banker's algorithm | Solve crawler deadlocks |
| 35 | ThreadLocal | User session manager | Add request context |