Week 1: Java Basics and Control Statements

Day 1: Introduction to Java

- Java history and features
- Understanding the Java Development Kit (JDK), Java Runtime Environment (JRE), and Java Virtual Machine (JVM)
- Installing and setting up Java
- First Java program: “Hello World”
- Structure of a Java program (classes, methods, and main function)

Day 2: Java Data Types and Variables

- Primitive data types (int, char, double, etc.)
- Non-primitive data types (String, arrays, etc.)
- Declaring variables and constants
- Type casting and type conversion

Day 3: Operators in Java

- Arithmetic, relational, logical, and bitwise operators
- Assignment, increment, and decrement operators
- Ternary operator and operator precedence

Day 4: Control Statements

- If, if-else, and nested if
- Switch case
- Loops (for, while, and do-while)
- Break, continue, and return statements

Day 5: Arrays and Strings

- One-dimensional and multi-dimensional arrays

- String class and its methods
- StringBuffer and StringBuilder

Day 6-7: Weekend Practice

- Write small programs using control statements, arrays, and strings (ex: palindrome check, matrix addition, etc.)

Week 2: Object-Oriented Programming Concepts

Day 8: Introduction to OOP

- Object-oriented programming principles (Encapsulation, Inheritance, Polymorphism, Abstraction)
- Class and object basics
- Fields, methods, and constructors

Day 9: Inheritance

- Types of inheritance (single, multilevel, hierarchical)
- Super keyword
- Constructor chaining
- Method overriding

Day 10: Polymorphism

- Compile-time (method overloading) and runtime polymorphism
- Method overloading
- Method overriding and upcasting

Day 11: Abstraction

- Abstract classes and methods
- Interfaces in Java
- Difference between abstract class and interface

Day 12: Encapsulation

- Access specifiers (private, protected, public)
- Getter and setter methods
- Packages and the concept of encapsulation

Day 13-14: Weekend Practice

- Programs on inheritance, polymorphism, abstract classes, and interfaces.

Week 3: Exception Handling and Collections

Day 15: Exception Handling

- Introduction to exceptions
- Try-catch block
- Throw, throws, and finally
- Custom exceptions

Day 16: Input/Output Operations

- Using `Scanner` and `BufferedReader` classes for user input
- File I/O (FileReader, FileWriter, BufferedReader, BufferedWriter)

Day 17: Introduction to Collections Framework

- Overview of collections
- List, Set, and Map interfaces
- ArrayList, LinkedList, HashSet, TreeSet, HashMap, and TreeMap

Day 18: Collections (continued)

- Iterating through collections (Iterator, ListIterator)

- Comparing elements in collections
- Sorting collections using Comparator and Comparable

Day 19: Generics in Java

- Understanding generics
- Generic classes and methods
- Bounded types

Day 20-21: Weekend Practice

- Programs on exception handling, I/O operations, and collections.

Week 4: Multithreading, Advanced Concepts, and Project

Day 22: Multithreading Basics

- Introduction to multithreading
- Thread lifecycle
- Creating threads using `Thread` class and `Runnable` interface
- Thread synchronization and inter-thread communication

Day 23: Lambda Expressions and Functional Programming

- Introduction to lambda expressions
- Functional interfaces (`Runnable`, `Comparator`, etc.)
- Stream API overview (filtering, mapping, reducing)

Day 24: Java Memory Management and Garbage Collection

- Java memory model (Heap, Stack)
- JVM internals

- Garbage collection in Java

Day 25: Java Annotations and Reflection

- Introduction to annotations
- Built-in annotations (`@Override`, `@SuppressWarnings`, etc.)
- Reflection in Java (overview)

Day 26: Java 8+ Features (Optional)

- New features introduced in Java 8 (Streams, Optional, new Date/Time API)
- Lambda expressions in-depth

Day 27-28: Mini Project/Capstone Project

- Build a small project incorporating Java concepts such as OOP, collections, and exception handling. For example, a banking system, library management system, or student record system.