JAVA Syllabus

Java

Introduction to C#

- ✓ **Overview of Java**: Features, history, and advantages.
- ✓ **Java Virtual Machine (JVM)**: Understanding JVM, JRE, and JDK.
- ✓ **Java Program Structure**: Writing and running the first Java program (Hello World).
- ✓ **Setting Up Environment**: Installing JDK and configuring IDEs like IntelliJ IDEA, Eclipse, or NetBeans.

Basics of Java

- ✓ **Syntax and Program Structure**: Java syntax, structure, and basic conventions.
- ✓ **Data Types and Variables**: Primitive data types (int, float, double, char, boolean) and reference types.
- ✓ **Operators**: Arithmetic, logical, relational, bitwise, assignment, and ternary operators.
- ✓ **Type Casting**: Implicit and explicit type conversions.
- ✓ **Input and Output**: Using Scanner and System.out for input and output operations.

Control Flow

- ✓ **Decision-Making Statements:**
  - if, else if, else, switch-case.
- ✓ **Loops:**
  - for, while, do-while, and enhanced for loop.
- ✓ **Jump Statements:**
  - break, continue, and return.

Arrays and Strings

- ✓ **Arrays:**
    - Single-dimensional and multi-dimensional arrays.
    - Array initialization, manipulation, and iteration.
- ✓ **Strings:**
    - Creating strings using String, StringBuilder, and StringBuffer.
    - String methods like length(), substring(), replace(), split(), and toLowerCase().

Object-Oriented Programming (OOP)

- ✓ **Classes and Objects**:
    - Defining classes, creating objects, and accessing members.
- ✓ **Constructors**:
    - Default, parameterized, and copy constructors.
- ✓ **Inheritance**:
    - Types of inheritance, method overriding, and super keyword.
- ✓ **Polymorphism**:
    - Method overloading, method overriding, and dynamic method dispatch.
- ✓ **Encapsulation**:
    - Private fields, public methods, and use of getters and setters.
- ✓ **Abstraction**:
    - Abstract classes and interfaces.
- ✓ **Static and Final Keywords**:
    - Static fields, methods, and blocks; final methods and variables.

Exception Handling

- ✓ **Try-Catch Blocks**: Handling runtime exceptions.
- ✓ **Finally Block:** Cleanup operations.
- ✓ **Throw and Throws:** Raising and propagating exceptions.
- ✓ **Custom Exceptions**: Creating user-defined exceptions.

Java Collections Framework

- ✓ **Collections Hierarchy:** Overview of interfaces and classes in the framework.
- ✓ **List Interface:** ArrayList, LinkedList, Vector.
- ✓ **Set Interface:** HashSet, LinkedHashSet, TreeSet.
- ✓ **Map Interface:** HashMap, TreeMap, LinkedHashMap.
- ✓ **Queue Interface:** PriorityQueue, Deque.
- ✓ **Iterators:** Iterating over collections using Iterator and ListIterator.

Input/Output (I/O) Streams

- ✓ **Streams in Java:** Byte and character streams.
- ✓ **File Handling:** Reading and writing files using FileReader, FileWriter, BufferedReader, and BufferedWriter.
- ✓ **Serialization and Deserialization:** Saving and restoring object states.

Java Multithreading

- ✓ **Threads in Java:** Creating threads using Thread class and Runnable interface.
- ✓ **Thread Lifecycle:** States of a thread.
- ✓ **Synchronization:** Thread-safe operations using synchronized blocks and methods.
- ✓ **Inter-Thread Communication:** Using wait(), notify(), and notifyAll().

Generics and Collections

- ✓ **Introduction to Generics:** Using generics to create type-safe classes and methods.
- ✓ **Generic Collections:** List<T>, Map<K,V>, and more.
- ✓ **Wildcards in Generics:** Upper-bound and lower-bound wildcards.