Learning Java for Data Structures and Algorithms (DSA) is a great choice, as Java is a versatile and widely used programming language. To get started, here are the essential topics you should learn:

1. **Basic Java Concepts:**
2. Variables and data types
   * Operators (arithmetic, relational, logical)
   * Control flow (if, else, switch)
   * Loops (for, while, do-while)
   * Arrays
3. **Object-Oriented Programming (OOP):**
   * Classes and objects
   * Inheritance
   * Polymorphism
   * Encapsulation
   * Abstraction
4. **Data Structures:**
   * Arrays
   * Linked Lists (singly and doubly)
   * Stacks
   * Queues
   * Trees (binary trees, binary search trees, AVL trees, etc.)
   * Heaps (min-heap, max-heap)
   * Hashing (hash tables)
5. **Algorithms:**
   * Searching algorithms (linear search, binary search)
   * Sorting algorithms (bubble sort, insertion sort, quicksort, mergesort)
   * Recursion
   * Dynamic programming
   * Greedy algorithms
   * Graph algorithms (DFS, BFS)
6. **Complexity Analysis:**
   * Big O notation
   * Time and space complexity analysis
   * Best-case, worst-case, and average-case analysis
7. **Advanced Data Structures (optional but helpful):**
   * Graphs (adjacency matrix, adjacency list)
   * Hashing (collision resolution techniques)
   * Advanced trees (red-black trees, B-trees)
8. **Java Collections Framework:**
   * ArrayList
   * LinkedList
   * HashMap
   * HashSet
9. **Exception Handling:**
   * Try-catch blocks
   * Custom exceptions
10. **File I/O:**
    * Reading and writing files
11. **Java Standard Library:**
    * Understanding commonly used classes and methods in the Java Standard Library.
12. **Practice and Problem Solving:**
    * Solve DSA problems on online platforms like LeetCode, HackerRank, or Codeforces.
13. **Version Control (optional but recommended):**
    * Learn how to use Git for code version control.
14. **IDE (Integrated Development Environment):**
    * Familiarize yourself with an IDE like Eclipse, IntelliJ IDEA, or Visual Studio Code for Java development.
15. **Testing:**
    * Learn how to write unit tests using JUnit or TestNG.
16. **Build Tools (optional):**
    * Understand build tools like Apache Maven or Gradle for project management.
17. **Java 8+ Features (optional but recommended):**
    * Lambdas and functional programming concepts.
18. **Concurrency (optional but useful):**
    * Basics of multithreading and synchronization.
19. **Design Patterns (optional but helpful):**
    * Familiarize yourself with common design patterns like Singleton, Factory, and Observer.
20. **Database Connectivity (optional):**
    * Basics of JDBC for database interaction.
21. **Web Development (optional):**
    * Servlets and JSP if you're interested in web applications.

Remember that learning programming and DSA takes time and practice, so don't rush. Start with the basics and gradually move towards more advanced topics. Practice solving problems regularly to reinforce your understanding. Additionally, there are many online tutorials, courses, and textbooks available that can help you learn Java and DSA effectively.