**BASIC JAVA**

**Beginner Level:**

1. **Hello World:** Create a Java program that prints "Hello, World!" to the console.
2. **Calculator:** Build a simple calculator that can perform basic arithmetic operations (addition, subtraction, multiplication, division).
3. **Guess the Number:** Write a program that generates a random number and asks the user to guess it. Provide feedback on whether the guess is too high or too low.
4. **Factorial Calculator:** Create a program that calculates the factorial of a given number using both iterative and recursive approaches.
5. **Palindrome Checker:** Write a program that checks if a given string is a palindrome (reads the same forwards and backward).

**Intermediate Level:**

1. **Prime Number Generator:** Develop a program that generates a list of prime numbers within a specified range.
2. **String Manipulation:** Create a program that performs common string operations like reversing a string, counting characters, and checking for substrings.
3. **Data Structures:** Implement basic data structures like a linked list, stack, or queue from scratch.
4. **File Handling:** Build a program that reads data from a text file, processes it, and writes the results back to a file.
5. **Sorting Algorithms:** Implement sorting algorithms like Bubble Sort, Insertion Sort, or Selection Sort and compare their efficiency.

**Advanced Level:**

1. **Search Algorithms:** Implement search algorithms like Binary Search and Linear Search.
2. **Database Connectivity:** Develop a Java application that connects to a database, retrieves data, and performs CRUD (Create, Read, Update, Delete) operations.
3. **Multithreading:** Create a multithreaded application to demonstrate concurrency and synchronization.
4. **Network Programming:** Build a networked chat application or a client-server program using Java's socket programming.
5. **MVC Pattern:** Design a simple Java application using the Model-View-Controller (MVC) architectural pattern.

**Expert Level:**

1. **Java EE Application:** Develop a Java Enterprise Edition (Java EE) application that includes features like authentication, authorization, and database integration.
2. **Spring Framework:** Learn and practice the Spring Framework to create a web application with Spring MVC, Spring Boot, or Spring Data.
3. **RESTful API:** Design and implement a RESTful API using Java, possibly with the Spring framework.
4. **Concurrency Challenges:** Tackle advanced concurrency problems, such as the Dining Philosophers problem or the Producer-Consumer problem.
5. **Big Data:** Explore Java libraries and tools for handling big data and distributed computing, such as Apache Hadoop and Apache Spark.