Exercise 3.1: Memory Management Fundamentals

How Garbage Collection Works in Java

Garbage Collection (GC) in Java is an automatic memory management process that frees up memory occupied by objects that are no longer reachable.

Garbage Collection Process:

1. **Allocation**: Objects are allocated in the heap.
2. **Reachability Analysis**: The JVM determines which objects are still in use.
3. **Garbage Collection**: Unreachable objects are deallocated to free up memory.

GC Phases:

- **Young Generation (Eden + Survivor Spaces)**: New objects are created here.
- **Old Generation (Tenured Space)**: Long-lived objects are moved here.
- **Permanent Generation (Metaspace in Java 8+)**: Stores class metadata.

Common Pitfalls to Avoid:

1. **Memory Leaks**: Holding references to objects that are no longer needed (e.g., unclosed resources).
2. **OutOfMemoryError**: Excessive object creation without freeing memory.
3. **GC Pauses**: Long pauses in applications due to inefficient GC tuning.