**🔒 final – Keyword**

Used to declare:

* **Final variables** (cannot be changed)
* **Final methods** (cannot be overridden)
* **Final classes** (cannot be extended)

java

CopyEdit

final int maxBids = 10; // Variable can't be reassigned

final class AuctionUtils {} // Class can't be subclassed

final void logBid() {} // Method can't be overridden

✅ Compile-time behavior  
❌ Not related to exception handling or memory

**🧹 finalize() – Method (deprecated in Java 9+)**

Used to define cleanup logic before an object is garbage collected.

java

CopyEdit

@Override

protected void finalize() throws Throwable {

System.out.println("Cleaning up...");

super.finalize();

}

⚠️ Deprecated — you should use **AutoCloseable** or **try-with-resources** instead  
🧠 Called by JVM — not guaranteed when or even if it runs

**🧯 finally – Block (exception handling)**

Always runs after try and catch, **even if an exception occurs or return is called**.

java

CopyEdit

try {

// risky code

} catch (Exception e) {

// handle exception

} finally {

// always runs

closeResources();

}

✅ Ensures cleanup  
✅ Good for closing files, DB connections, etc.

**✅ Summary Table**

| **Term** | **Type** | **Purpose** | **Notes** |
| --- | --- | --- | --- |
| final | Keyword | Prevent modification/extension | Used with variables, methods, classes |
| finally | Block | Always-execute block in exception handling | Comes after try / catch |
| finalize | Method | Cleanup before GC (deprecated) | Avoid using in new code |