A daemon thread is a background thread that runs in a program to perform tasks such as housekeeping, monitoring, or other auxiliary work. Unlike regular (non-daemon) threads, daemon threads are not essential to keep the program running. When all non-daemon threads have finished executing, the program can terminate, even if daemon threads are still running. In other words, daemon threads will be killed automatically when the main program or all user (non-daemon) threads finish execution.

Characteristics of a Daemon Thread:

* Background Execution: Daemon threads are typically used for background support tasks (e.g., garbage collection, monitoring, or cleanup).
* Termination with Main Thread: The JVM (in Java, for example) or the program will terminate when all non-daemon threads finish, without waiting for daemon threads to complete.
* Lower Priority: Daemon threads often have lower priority compared to regular threads, although this is not a strict rule.

Use Cases:

* Garbage Collection: In languages like Java, the garbage collector runs as a daemon thread, cleaning up memory in the background.
* Monitoring: A daemon thread can be used to monitor resources or events, such as log monitoring or watching file changes.
* Housekeeping Tasks: Periodic maintenance tasks, such as freeing resources or performing routine checks.



