[The **double-checked locking** design pattern is used to reduce the overhead of acquiring a lock by testing a locking criterion before obtaining the lock, commonly employed for **lazy initialization** in multi-threaded environments1](https://www.youtube.com/watch?v=jma-h9lgiRU).

Here are **five free reference links** where you can learn more about this pattern:

1. [Wikipedia: Double-checked locking](https://www.youtube.com/watch?v=jma-h9lgiRU)
2. [CodeProject: Strategized Locking Pattern](https://www.codeproject.com/Articles/33241/Strategized-Locking-Pattern)
3. [TechNet Articles: Lock Design Pattern](https://social.technet.microsoft.com/wiki/contents/articles/13240.lock-design-pattern.aspx)
4. [Java Design Patterns: Lockable Object](https://java-design-patterns.com/patterns/lockable-object/)
5. [GitHub: PatternLock](https://github.com/zhanghai/PatternLock)

Feel free to explore these resources to deepen your understanding of the double-checked locking pattern! 🚀