The **singleton design pattern** ensures that a class has only one instance and provides a global point of access to it throughout an application. [It is useful when you want to restrict the instantiation of a class to a single object](https://www.geeksforgeeks.org/singleton-design-pattern/) [1](https://www.geeksforgeeks.org/singleton-design-pattern/)[2](https://www.educative.io/courses/software-design-patterns-best-practices/singleton-pattern).

Here are **five free reference links** where you can learn more about the singleton design pattern:

1. [**GeeksforGeeks**: Provides a detailed explanation, use cases, and implementation examples in Java](https://www.geeksforgeeks.org/singleton-design-pattern/) [1](https://www.geeksforgeeks.org/singleton-design-pattern/).
2. [**Educative**: Offers a comprehensive tutorial with code snippets and best practices for implementing singletons in Java](https://www.geeksforgeeks.org/singleton-design-pattern/) [3](https://www.squash.io/tutorial-best-practices-for-java-singleton-design-pattern/).
3. [**MakeUseOf**: A concise article explaining the purpose and usage of the singleton pattern](https://www.geeksforgeeks.org/singleton-design-pattern/) [4](https://www.makeuseof.com/singleton-design-pattern/).
4. [**Pentalog**: Discusses the definition, implementation, and benefits of the singleton pattern](https://www.geeksforgeeks.org/singleton-design-pattern/) [5](https://www.pentalog.com/blog/design-patterns/singleton-design-pattern/).
5. [**TutorialsTeacher**: Covers singleton design pattern in C# with examples](https://www.geeksforgeeks.org/singleton-design-pattern/) [6](https://www.tutorialsteacher.com/csharp/singleton).

Feel free to explore these resources to deepen your understanding of the singleton pattern! 🚀