[The **Builder Design Pattern** is a creational pattern used in software design to construct a complex object step by step, allowing the construction process to vary based on the type of product being built1](https://www.geeksforgeeks.org/builder-design-pattern/). Here are **five free references** where you can learn more about it:

1. [**GeeksforGeeks**: Provides an in-depth explanation and example of the Builder pattern1](https://www.geeksforgeeks.org/builder-design-pattern/).
2. [**Wikipedia**: Offers a concise overview of the pattern’s intent and separation of object construction from representation](https://www.geeksforgeeks.org/builder-design-pattern/)[2](https://en.wikipedia.org/wiki/Builder_pattern).
3. [**Baeldung**: Demonstrates how to use the **FreeBuilder** library to simplify builder class implementation in Java](https://www.geeksforgeeks.org/builder-design-pattern/)[3](https://www.baeldung.com/java-builder-pattern-freebuilder).
4. [**Scaler Topics**: A comprehensive resource covering various design patterns, including the Builder pattern](https://www.geeksforgeeks.org/builder-design-pattern/)[4](https://www.scaler.com/topics/design-patterns/).
5. [**DigitalOcean**: Provides a tutorial on implementing the Builder pattern in Java, along with code examples](https://www.geeksforgeeks.org/builder-design-pattern/)[5](https://www.digitalocean.com/community/tutorials/builder-design-pattern-in-java).

Feel free to explore these resources to deepen your understanding of the Builder pattern! 🛠️👷‍♂️