UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

Name	:	KAUTA MARVIN
Registration Number	:	S20B23/204
Faculty/School	:	SCIENCE AND TECHNOLOGY
Course	:	BACHOLERS OF SCIENCE IN COMPUTER SCIENCE
Lecturer	:	***MR LUBAMBO SIMON***

Write short notes about the façade design patterns

The facade design pattern is a software design pattern used in object-oriented programming that provides a simplified interface to a larger body of code, such as a class library.

The facade pattern encapsulates the complexity of the system by providing a simplified interface that clients can use to interact with the system.

the facade pattern acts as a layer of abstraction between the client code and the complex subsystems, shielding the client code from the underlying complexity and providing a simplified view of the system.

The facade pattern is often used in large systems where there are many subsystems with complex interrelationships. By using a facade, clients can interact with the system without having to know the details of how the subsystems are interconnected. This makes the code easier to understand and maintain, and reduces the coupling between the client code and the subsystems.

reasons why one might choose to use the facade design pattern

1. **Simplifies the interface:** The facade pattern provides a simplified interface to a complex system or set of objects, making it easier to use and reducing the learning curve for developers.
2. **Encapsulates complexity:** This pattern encapsulates the complexity of the system by providing a simplified interface to the client code. This reduces the amount of code needed to interact with the system and makes it easier to modify or refactor the underlying code without affecting the client code.
3. **Improves maintainability**: Because the facade pattern encapsulates the complexity of the system, it makes it easier to maintain the code over time. This is because changes made to the underlying code are isolated from the client code, reducing the risk of introducing bugs or breaking existing functionality.
4. **Reduces coupling:** The facade pattern reduces coupling between the client code and the subsystems by providing an abstraction layer between them. This makes the code more modular and easier to maintain over time.
5. **Enhances reusability:** The facade pattern can improve the reusability of the underlying code by providing a simplified interface that can be used across multiple projects. This reduces the amount of code duplication and makes it easier to maintain a consistent architecture across multiple projects.