PareDox
Education Solutions

Aamir Shabbir Pare

Problem Solving Programming

Design Patterns

PareDox
Education Solutions

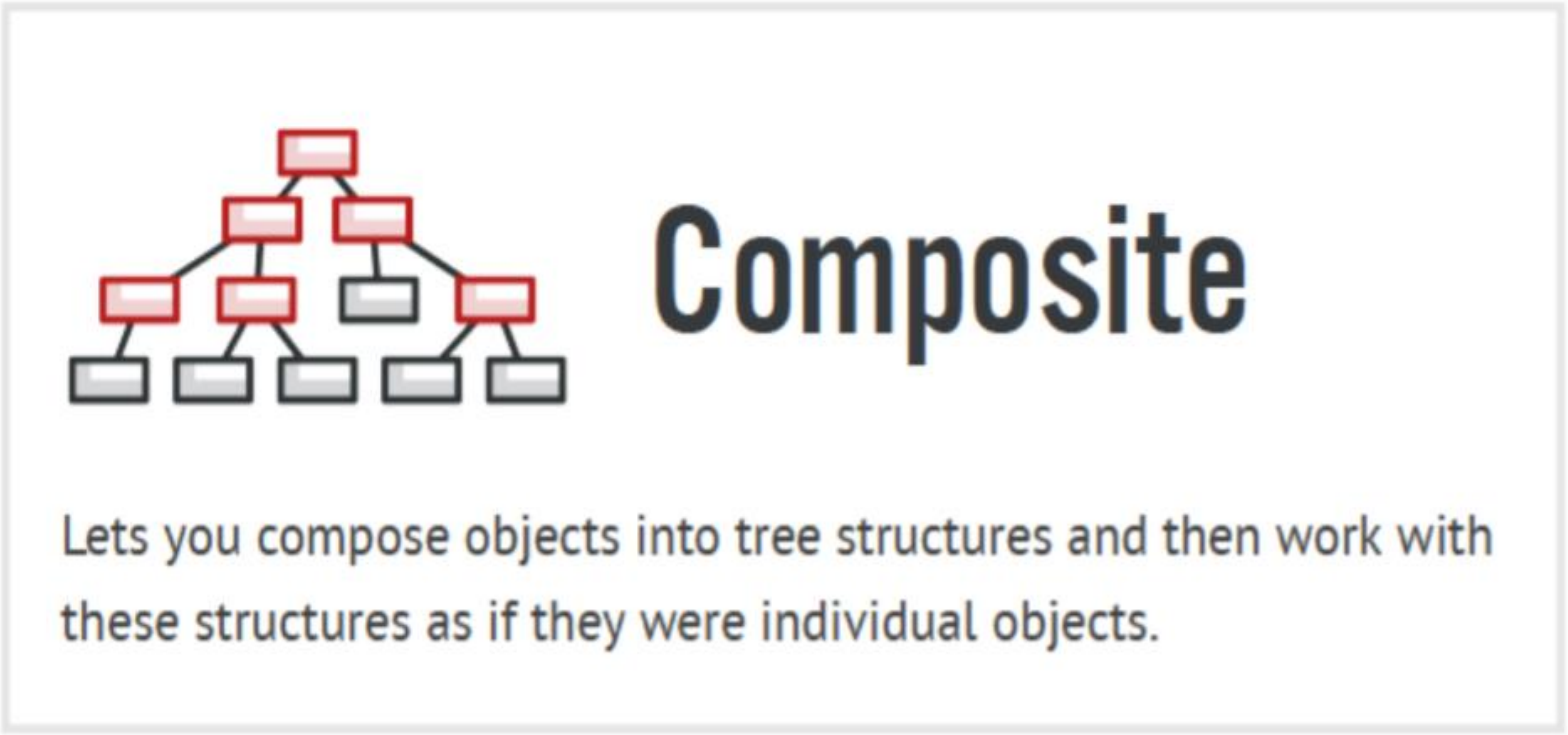
Previous Lecture

- Adapter Pattern
- Go to teams and watch video lecture

Aamir Shabbir Pare

PareDox
Education Solutions

Composite Pattern



Composite

Lets you compose objects into tree structures and then work with these structures as if they were individual objects.

Aamir Shabbir Pare

Composite Pattern Concept

Gang of Four Definition

- Compose objects into tree structures to represent whole-part hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.
- Composite is an entity made up of several elements.
- The composite pattern is a partitioning design pattern, it describes that the group of objects are treated same way as a single instance of the same type of object.

Aamir Shabbir Pare

Implementation Guidelines

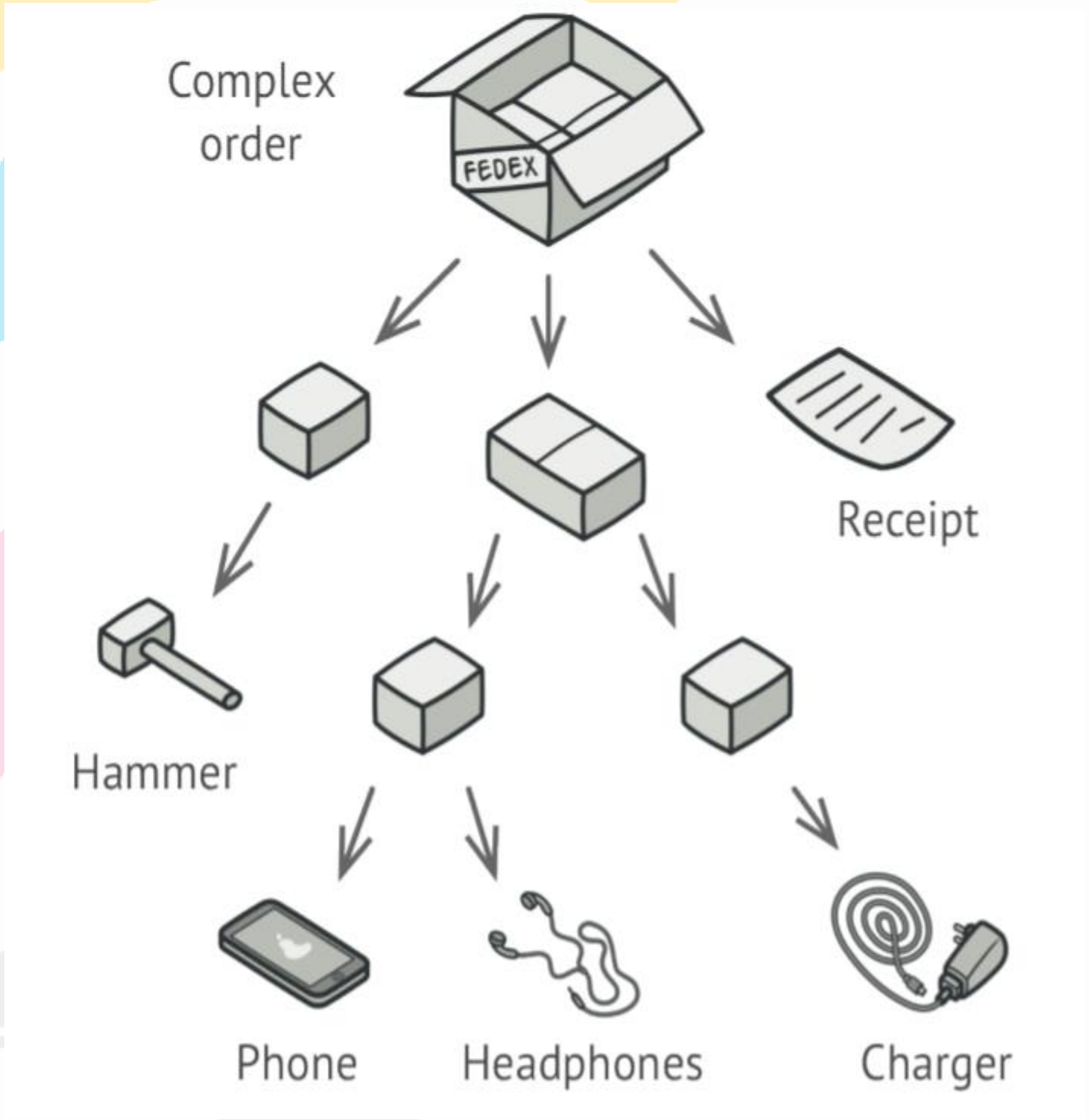
When to select this design pattern

- When to represent whole-part hierarchies of objects.
- When clients need ignore the difference between compositions and individual objects.
- The composite pattern is a partitioning design pattern, it describes that the group of objects are treated same way as a single instance of the same type of object.

Aamir Shabbir Pare

Understanding the Problem

- When the core model of the application can be represented as a part-whole tree structure.
- An order might comprise various products, packaged in boxes, which are packaged in bigger boxes and so on. The whole structure looks like an upside down tree.

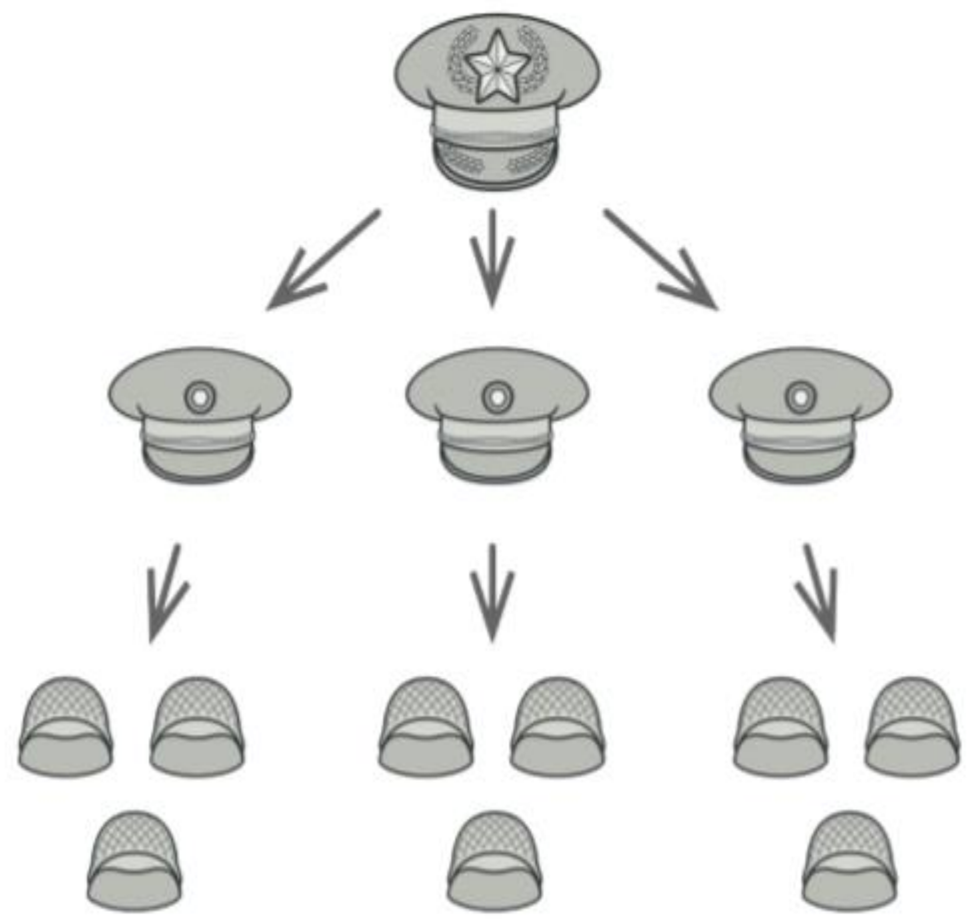


Aamir Shabbir Pare

PareDox
Education Solutions

Real World Analogy

- Armies of most countries are structured as hierarchies.
- An army consists of several divisions; a division is a set of brigades, and a brigade consists of platoons, which can be broken down into squads.
- Finally, a squad is a small group of real soldiers. Orders are given at the top of the hierarchy and passed down onto each level until every soldier knows what needs to be done.

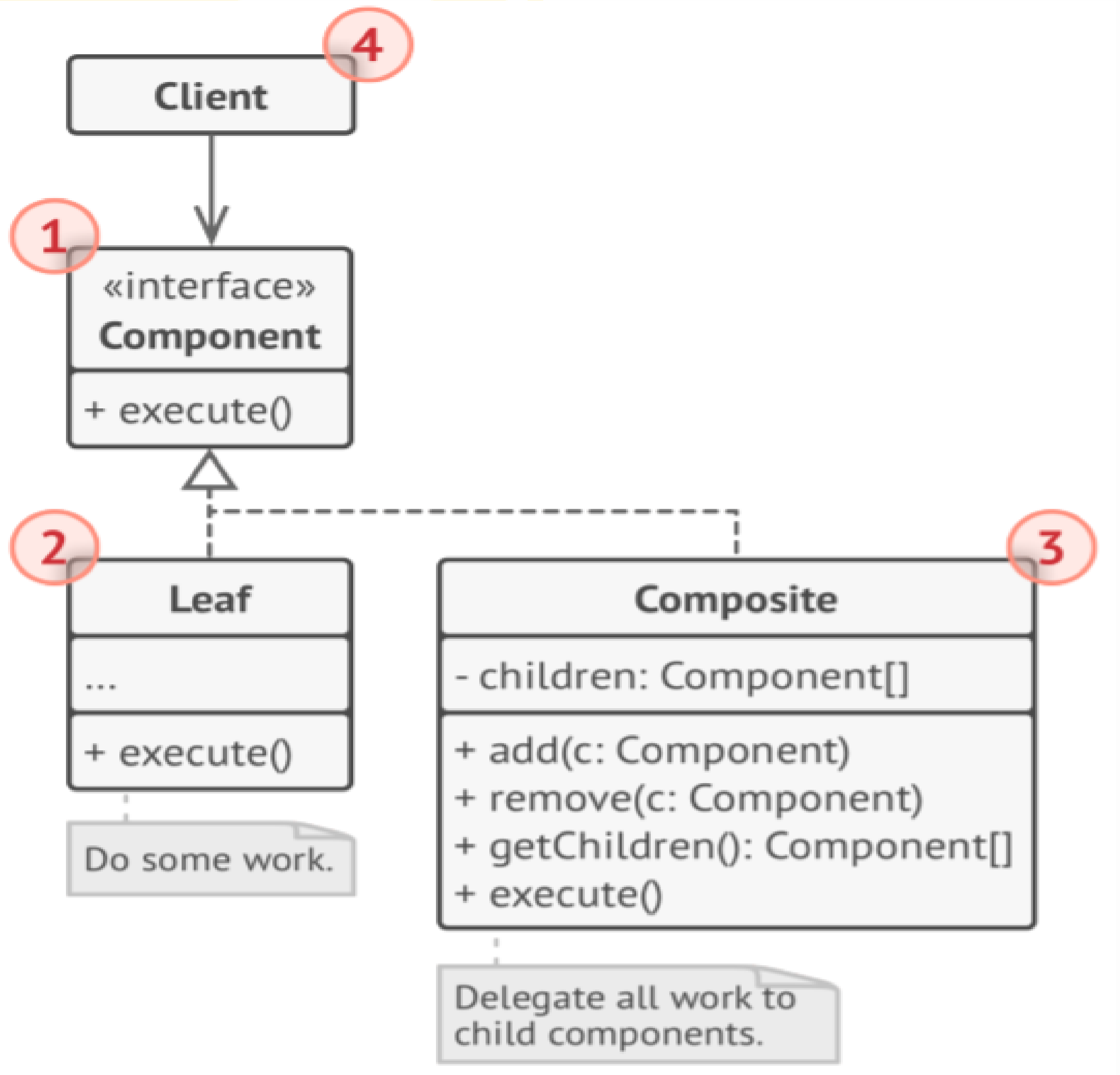


An example of a military structure.

Aamir Shabbir Pare

Structure

- Client
- Component
- Leaf
- Composite



Aamir Shabbir Pare

PareDox
Education Solutions

Structure Details

- The **Component** interface describes operations that are common to both simple and complex elements of the tree.
- The **Leaf** is a basic element of a tree that doesn't have sub-elements.
- The **Container** (aka composite) is an element that has sub-elements: leaves or other containers. A container doesn't know the concrete classes of its children. It works with all sub-elements only via the component interface.
- The **Client** works with all elements through the component interface. As a result, the client can work in the same way with both simple or complex elements of the tree.

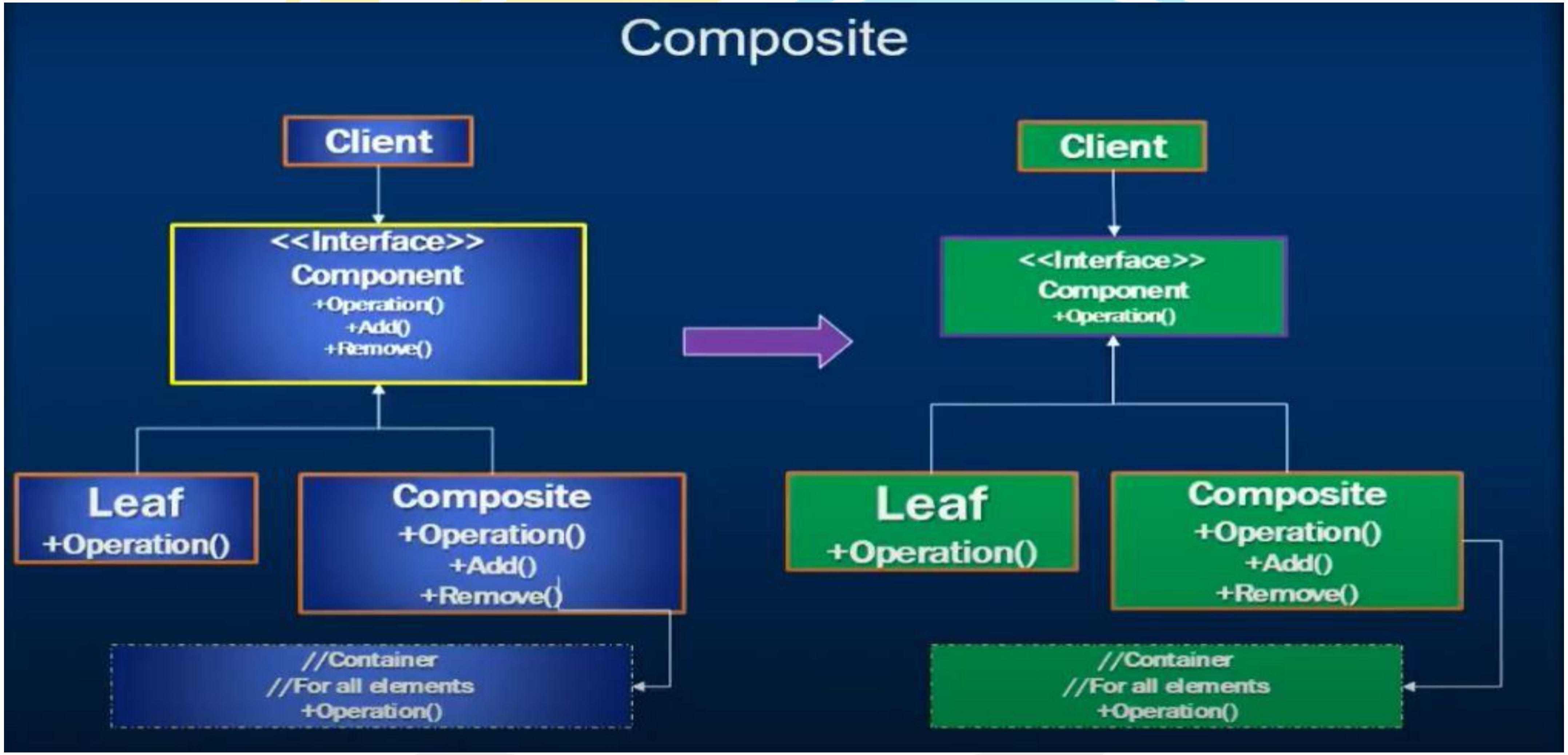
Aamir Shabbir Pare

Composite Pattern with LSP

- The Original GoF definition problem.
- **Leaf** and **Composite** both implements the same interface.
- Operations required by the Composite are forced to Leaf.
- Applying LSP to Composite Pattern
- Requires change in original definition.
- Move the operations from interface down to the composite.

Aamir Shabbir Pare

Change in Interface



Aamir Shabbir Pare