are concerned with providing alternative behavior for already existing classes or objects (without touching them

These patterns are classified as structural design patterns. [GoF, p10] – Adapter provides an alternative interface for an (already existing) class or object. – Bridge\* lets an abstraction and its implementation vary independently. – Composite composes (already existing) objects into a tree structure. – Decorator provides additional functionality for an (already existing) object. – Facade provides an unified interface for (already existing) objects in a subsystem. – Flyweight\* supports large numbers of fine-grained objects efficiently. – Proxy provides additional functionality when accessing an (already existing) object. • Composite - Flyweight – Composite and Flyweight often work together. Leaf objects can be implemented as shared flyweight objects. • Flyweight - Singleton – The flyweight factory is usually implemented as Singleton.