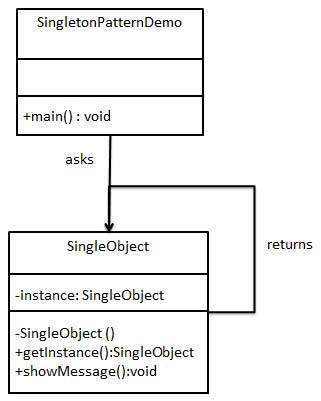
WEEK-9

**1. Creational Patterns**

These design patterns provides way to create objects while hiding the creation logic, rather than instantiating objects directly using new operator. This gives program more flexibility in deciding which objects need to be created for a given use case.

(e.g., Singleton, Factory Method, Abstract Factory, Builder, Prototype).

**Singleton Pattern**

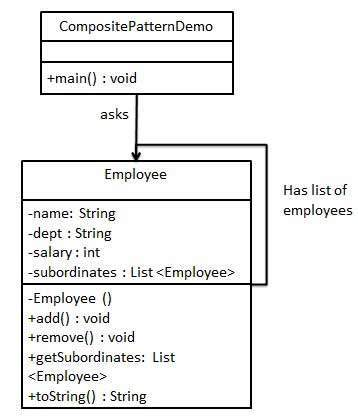
****

**2. Structural Patterns**

These design patterns concern class and object composition. Concept of inheritance is used to compose interfaces and define ways to compose objects to obtain new functionalities.

(e.g., adapter, composite, proxy, flyweight, facade, bridge, decorator).

**Composite Pattern**



**3. Behavioral Patterns**

These design patterns are specifically concerned with communication between objects.

(for example, Strategy, Observer, Command, Iterator, Mediator, Memento, State, Visitor, Template Method, Chain of Responsibility, Interpreter).

**Mediator Pattern**

