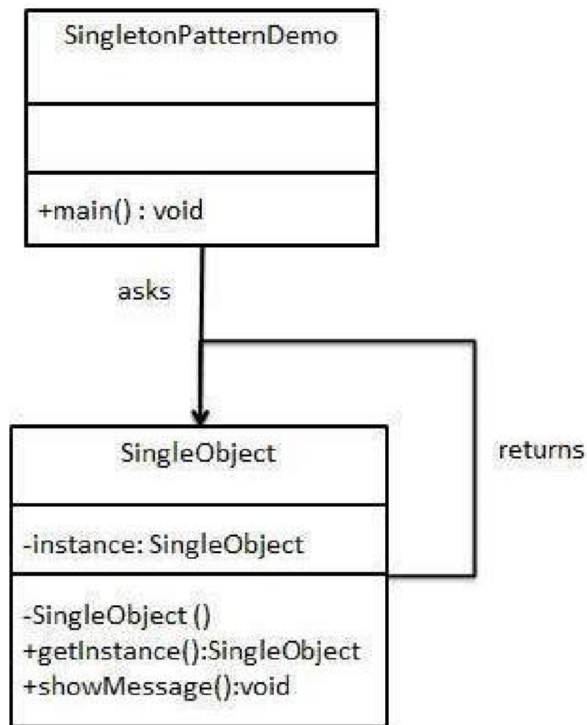


Lab 12-Gang of Four Patterns

Task 01

Create a SingleObject class. SingleObject class have its constructor as private and have a static instance of itself. SingleObject class provides a static method to get its static instance to outside world. SingletonPatternDemo, our demo class will use SingleObject class to get a SingleObject object.

Implement the given Pattern and also write the code.



Task 02

Consider a Garment Store which produces various types of garments like shirt, trousers. The consumers can request for the required types of garments through the store. However from consumer's perspective they are completely unaware of who is creating this object. They just know that the store is providing them the required garments.

Problem Statement:

It is a standard practice that objects are created by calling the “new” keyword. Imagine a scenario that there are multiple cases in the client class and we call multiple new keywords for creating new objects.

If we now have to add jacket or sweater we have to keep on modifying the client code and further add the new keyword. This creates a dependency on the client code and in turn makes it difficult to maintain.

Another problem is that the client application has to know how many types of concrete classes are available upfront. Later if we have to add another concrete class e.g. sweater or jacket then client code has to be changed and recompiled.

The objective is that:

- Client should be unaware of the object instantiation
- Client should access the objects through a common interface.

Propose a solution to the above problem by using an appropriate design pattern.