

In above problem, we have an operation which can be done in multiple ways (connect to friend) and user can choose desired way on runtime. So it's good candidate for strategy design pattern.

To implement the solution, let's design one participant one at a time.

- **ISocialMediaStrategy** – The interface which abstract the operation.
- **SocialMediaContext** – The context which determines the implementation.
- **Implementations** – Various implementations of ISocialMediaStrategy.  
E.g. FacebookStrategy, GooglePlusStrategy, TwitterStrategy and OrkutStrategy.
- `Java Collections.sort(list, comparator)` method where client actually passes suitable comparator based on the requirement in runtime to the method and the method is generic to accept any comparator type. Based on the comparator being passed, same collection can be sorted differently.