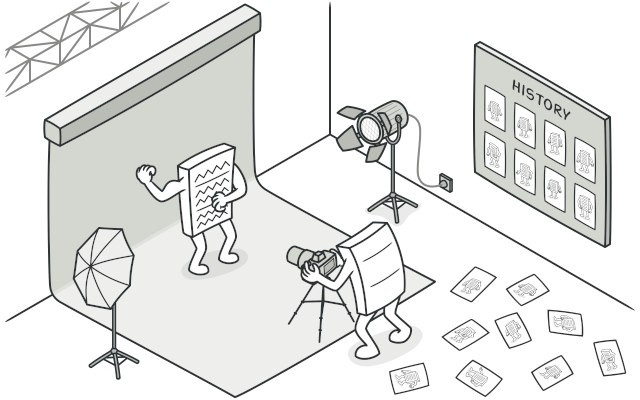
1. The following high-level GoF description defines the \_\_\_\_\_\_\_\_\_\_\_\_ design pattern

The intention of the pattern is to *avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Client initiates a request and passes it to a handler. Each handler performs its processing logic, and then potentially passes the processing request onto the next link (i.e. handler).*

* Observer
* **Chain of Responsibility**
* Visitor
* Command

1. Are you able to recognized a design pattern in the following funny cartoon?

* State
* **Memento**
* Decorator
* Observer

1. [Composite] Which entities of composite pattern need to inherit from Component?

* **Leaf and Composite**
* Leaf
* Composite
* Delegate

1. [Memento] The Memento design pattern is **NOT** implemented with the object:

* Caretaker
* **Checkpoint**
* Originator
* Memento

1. [Observer] Is it true that ...

*Is it true that observer pattern essentially establishes a one-to-many relationship between objects and has a loosely coupled design between interdependent objects?*

* **Yes**
* No, it's not worth it.
* No, better to use other design pattern
* All above are incorrect

1. What is TRUE regarding the Memento design pattern?

* **provides recovery technique**
* low maintenance costs
* it is NOT consuming a lot of RAM
* caretaker tracks the originator