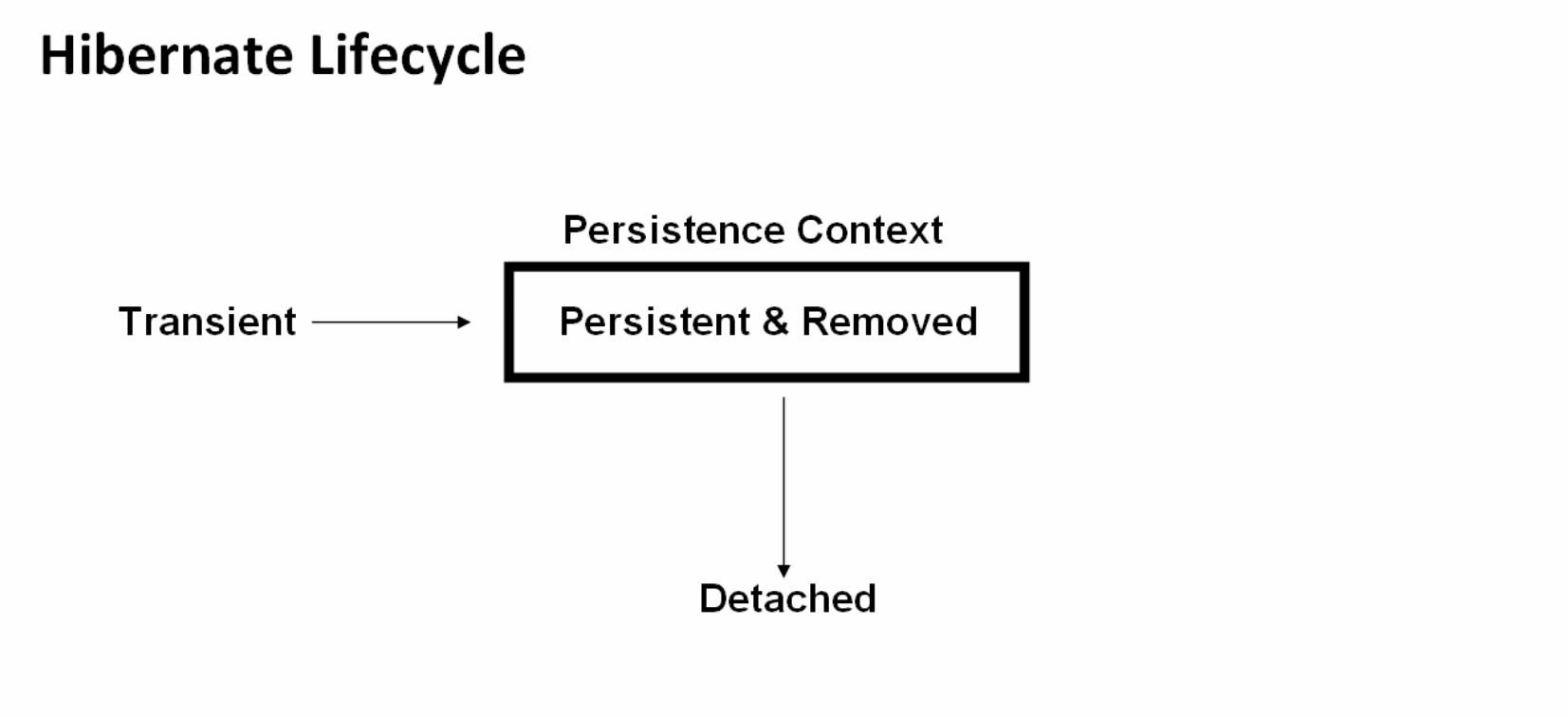
**Persistence Lifecycle**

* When working with the hibernate, we are going to invoke different method on entities. And when we do that the entities changes/transists into different states.
* Persistence Lifecycle is all about what states the entities are when we invoke different method on them.



**Entity States**

* Transient
* Persistent
* Detached
* Removed

**Transient:** Transient state are the state when a entity is being created using the **new** operator and it is not associated with any database row or in other words, When you first create a plain old Java object (via the new keyword) this object can be thought of as being in a transient state.

**Persistent:** When an object is in a persistent state, Hibernate is totally aware of it and can keep the database updated/synchronized with its values. There are two main ways to make the entity from a transient state to a persistent state:

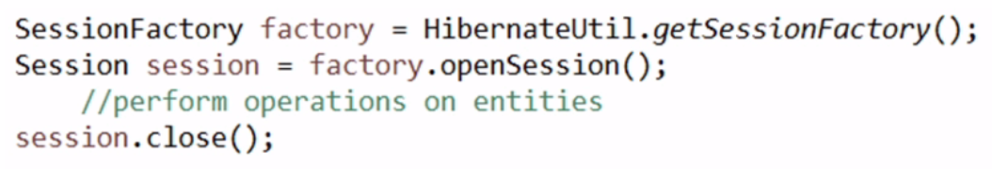
1. Loading the object from the database via Hibernate APIs (using get() or load() )
2. Saving the object to the database via Hibernate APIs ( save(), persist() or saveOrUpdate() )

**Removed:** If an entity is scheduled for deletion then this is called as Removed state.

**Detached:** This state happens when an entity object is in a persistence state and then if we close the persistence context and the entity becomes detached from the persistence context.

**Persistence Context:**

* Persistence context will cache all the persistence entities instances.
* Persistence context is usually created with a session object.



* And destroyed when the session object is closed.