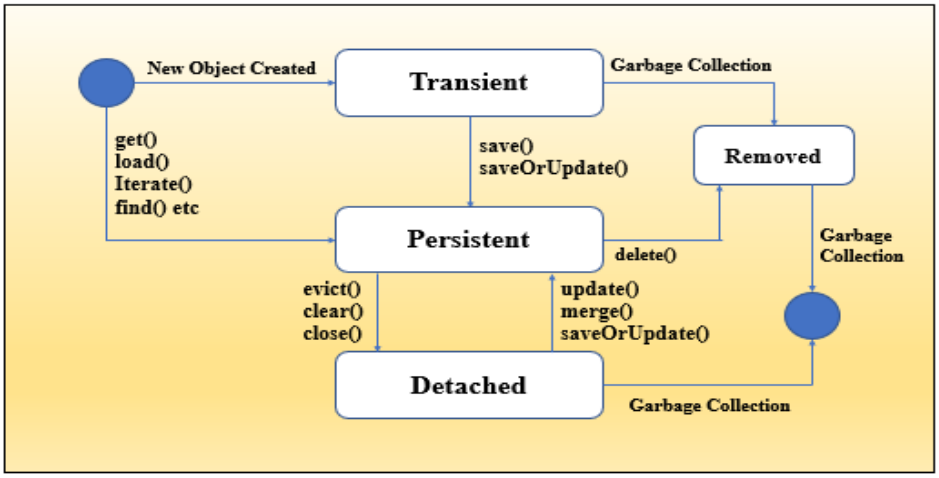
3) Hibernate works with POJO. Without any Hibernate specific annotation and mapping, Hibernate does not recognize these POJOs. Once properly annotated with required annotation, hibernate identifies them and keeps track of them to perform database operations such as create, read, update and delete. These POJOs are considered as mapped with Hibernate. An instance of a class mapped with Hibernate, can be any of the four persistence states which are known as hibernate entity lifecycle states.
Same as Figure 10.5: Hibernate Entity Lifecycle States

1. Transient
2. Persistent
3. Detached
4. Removed

4) Check section 10.4.3



Check Your Progress 3

1) **Difference between load() and get() methods**

Although load() and get() methods perform the same task, still the differences exist in their return values in case the identifier does not exist in the database.

- The get() method returns NULL if the identifier does not exist.
- The load() method throws a runtime exception if the identifier does not exist.

2) Differences between save() and persist() methods are listedin the below table.

S.NO	Key	Save()	Persist()
1	Basic	It stores object in database	It also stores object in database
2	Transaction Boundaries	It can save object within boundaries and outside boundaries	It can only save object within the transaction boundaries
3	Return Type	It returns generated id and return type is serializable	It does not return anything. Its return type is void.

4	Detached Object	It will create a new row in the table for detached object	It will throw persistence exception for detached object
5	Supported by	It is only supported by Hibernate	It is also supported by JPA

3) This saveOrUpdate() method either performs**save()** or **update()** based on identifier existence. E.g. If an identifierexists for the instance, update() method will be called, otherwise save() will be performed. The saveOrUpdate() method handles the cases where we need to save a detached instance. Unlike the save() operation on detached instances, the saveOrUpdate() method does not result in a duplicate record. Similar to update(), this method is used to reattach an instance to the session. This method can be considered as a universal tool to make an object persistent regardless of its state,whether it is transient or detached.

4) Spring Data JPA is one of the many sub-projects of Spring Data which simplifies the data access for relational data stores. Spring Data JPA is not a JPA provider; instead it wraps the JPA provider and adds its own features like a no-code implementation of the repository pattern. Spring Data JPA uses Hibernate as the default JPA provider. JPA provider is configurable, and other providers can also be used with Spring Data JPA. Spring Data JPA provides complete abstraction over the DAO layer into the project. The advantages of using Spring Data JPA are as follows:

DAO Abstraction with No-Code Repositories

Spring Data JPA defines many Repository Interfaces such as CrudRepository, PagingAndSortingRepository, JpaRepository having methods to store, retrieve, sorted retrieval, paginated result and many more.With Spring Data JPA, we don't have to write SQL statements; instead, we just need to extend the interface defined by Spring Data JPA for one of the entities.

Query Methods
Another robust and comfortable feature of Spring Data JPA is the Query Methods. Based on the name of methods declared in the repository interfaces are converted to low level SQL queries at runtime.

Seamless Integration
Spring Data JPA seamlessly integrates ***JPA*** into the Spring stack, and its repositories reduce the boilerplate code required to the JPA specification. Spring Data JPA also helps the DAO layer integration and interaction with other Spring based components in your application, like accessing property configurations or auto-wiring the repositories into the application service layer. It also works perfectly with Spring Boot auto-configuration.

10.8 REFERENCES/FURTHER READING

- Craig Walls, "Spring Boot in action" Manning Publications, 2016. (https://doc.lagout.org/programmation/Spring%20Boot%20in%20Action.pdf)

- Christian Bauer, Gavin King, and Gary Gregory, "**J**ava Persistence with Hibernate",Manning Publications, 2015.
- Ethan Marcotte, "Responsive Web Design", Jeffrey Zeldman Publication, 2011(http://nadin.miem.edu.ru/images_2015/responsive-web-design-2nd-edition.pdf)
- Tomcy John, "Hands-On Spring Security 5 for Reactive Applications",Packt Publishing,2018
- https://docs.jboss.org/hibernate/orm/3.5/reference/en/html/tutorial.html
- https://www.w3spoint.com/get-vs-load-hibernate
- https://www.baeldung.com/learn-jpa-hibernate
- https://www.javatpoint.com/hibernate-with-annotation
- https://www.tutorialspoint.com/jpa/index.htm
- https://howtodoinjava.com/hibernate/hibernate-jpa-2-persistence-annotations-tutorial/
- https://docs.jboss.org/hibernate/orm/3.6/quickstart/en-US/html/hibernate-gsg-tutorial-jpa.html
- https://www.baeldung.com/hibernate-save-persist-update-merge-saveorupdate
- https://www.journaldev.com/3472/hibernate-session-get-vs-load-difference-with-examples
- https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa
- https://dzone.com/articles/spring-data-jpa