This week we discussed about:

1. W1D5: Persistent API
    a. Entity Life Cycle: consists of 4 states: Transient, Managed, Detached, Removed. The state of entities can be changed by methods of the EntityManager (EM).
        i. Transient entity is no database entity
        ii. Managed entity is managed by EM and has a db entity.
        iii. Detached entity has a db entity, but is not managed by the current EM / PersistenceContext.
        iv. Removed entity is removed from db and PersistenceContext, but it's still in somewhere around the application / memory, may not yet be garbage collected in Java.
    b. EntityManager or PersistenceContext works like a cache, manages entities, tracks changes and pushs them to the database.
    c. EntityManager API: methods of EntityManager class/object such .persist(), .merge(), .remove(), ... what they are and how they work.
        i. What are differences between .find() and .getReference() and best practices
    d. Updating entity objects
        i. When changes are pushed automatically to db, are on tx.commit(), em.flush() or before a query.
        ii. Why there is no .update() in Hibernate, what is its problem
        iii. How to use .merge() correctly and to solve the .update() problem.
    e. EntityManager cache
2. W2D1: Associations & Collections
    a. Associations:
        i. Concepts: Join Table, Join Column, Uni and Bi-directional, Owning side, mappedBy, Shared PK, Embedded Classes, ...
        ii. 7 types of relationships
        iii. Cascades
    b. Collections: 4 types
        i. Bag: duplicates, no order, can @OrderBy
        ii. Set: no duplicates, no order, can @OrderBy
        iii. Map: Set of Keys to a Bag of values.
        iv. List: duplicates, built-in order, no @OrderBy
3. W2D2: Complex & Queries
    a. Inheritance: Single Table, Joined Tables, Table per Concrete Class
    b. Complex Mapping: Secondary Tables, Embedded Classes, Composite Keys
    c. Queries: JPQL (select, from, where, order by, ...)
4. W2D3: Queries & Optimization
    a. Queries: JPQL continued: Joins, join fetch, join collections
    b. Optimization: lazy and eager, N + 1 problem, entity graph, join fetch, batch size, sub select, caching
5. W2D4: Web Apps & Concurrency
    a. Web containers: Servlets, Filters
    b. JSP: JSTL, EL expressions

    c. Combination of Hibernate and web applications: Single Entity Manager, customized Filters.
    d. Concurrency: 4 isolation levels, update problem, optimistic concurrency, pessimistic locking, application transactions.
6. SCI:
    a. How connections are made is as important as the parts themselves. The whole is greater than the sum of the parts. How we implement associations is very important. Because they reflect exactly the real world and make the whole system work efficiently.
    b. Purification leads to progress. Understanding what the entity manager does with the DB and Cache, we can use the methods to maximize their potential. Similarly to other parts.