Creating a Hibernate Framework

You Already Know

Course(s):

Java Certification Training Course

Recap

- Explain Hibernate
  - Hibernate features
  - Hibernate architecture

- Demonstrate CRUD operations
  - CRUD operations

- Use Hibernate queries and relationships
  - HQL

- Demonstrate Hibernate mapping
  - Hibernate mapping using XML file
  - Hibernate mapping using Annotations

A Day in the Life of a Full Stack Developer

As the new e-commerce product is the company's priority, Joe is again added to another sprint. This sprint focuses majorly on creating a database for the new e-commerce website.

So to enhance the performance of the website, Joe is working on database creation. He has to develop a form where the user can enter the product details, and the validated details are stored in the database.

In this lesson, we will learn how to solve this real-world scenario to help Joe complete his task effectively and quickly.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain Hibernate features and architecture
- List the technologies and databases supported by Hibernate
- Demonstrate Hibernate configuration
- Explain Hibernate logging using Log4j
- Describe mapping in Hibernate
- Explain the integration of Hibernate with Spring

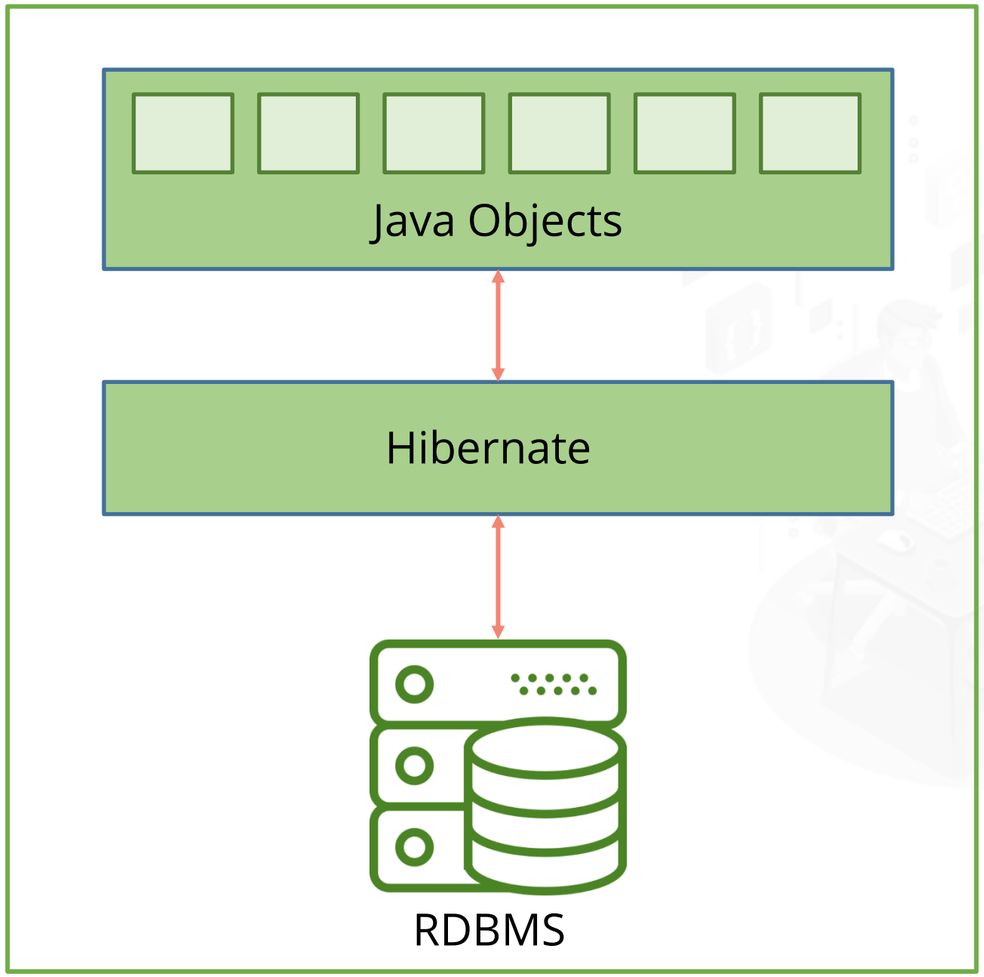
Hibernate Overview

Hibernate Overview

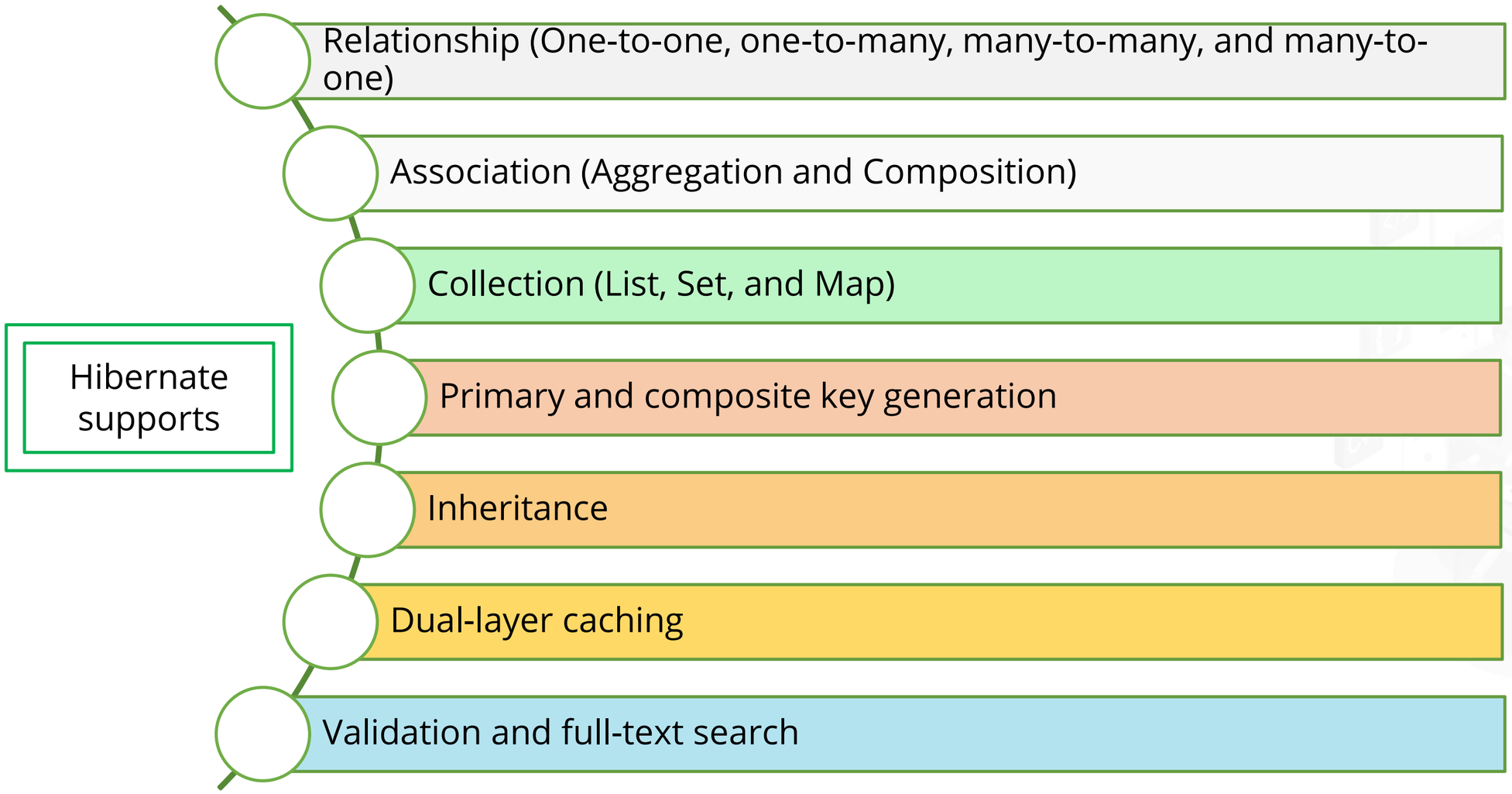
Hibernate is a lightweight open source **O**bject **R**elational **M**apping (ORM) framework.

Hibernate

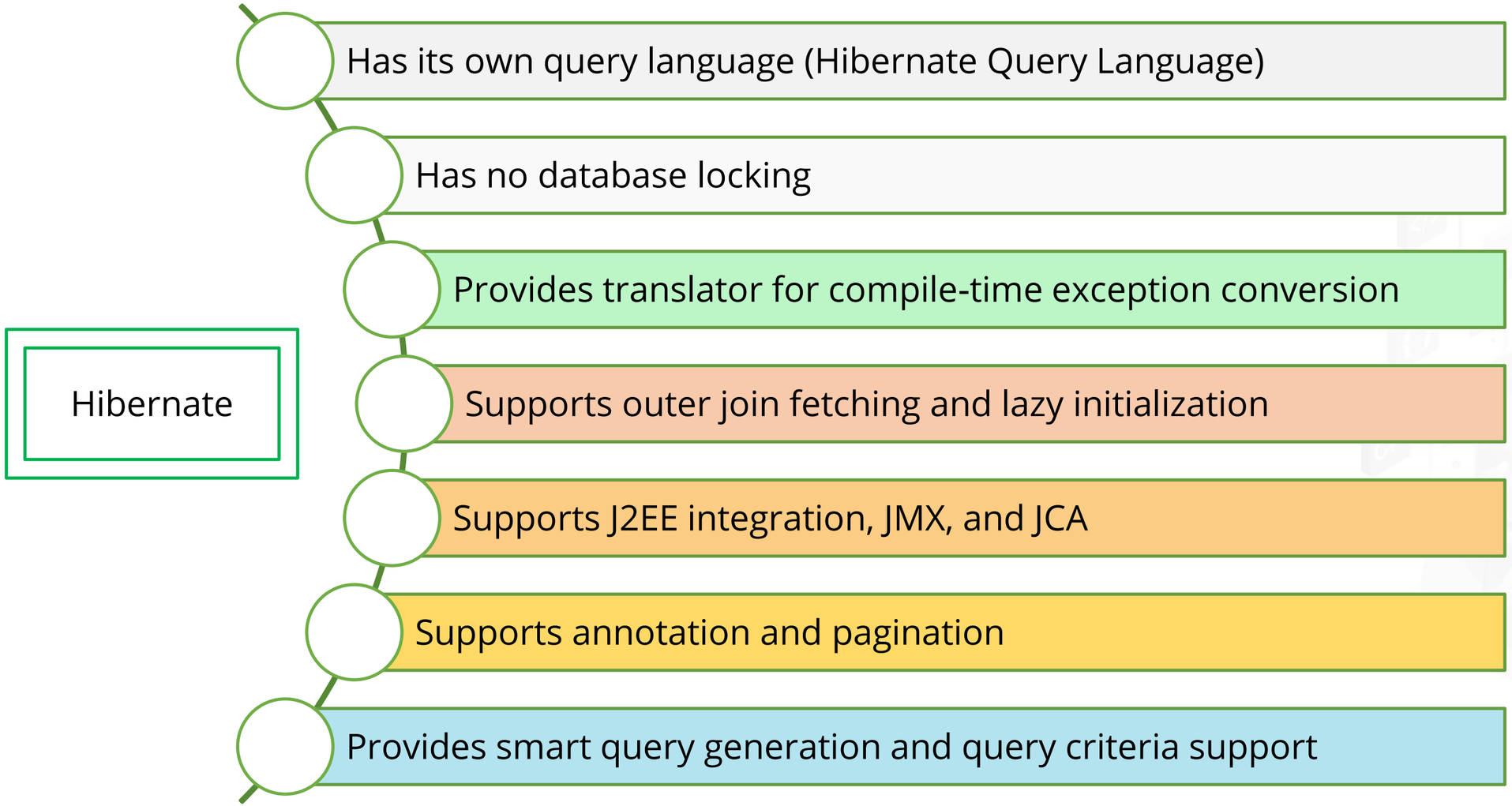
- Allows mapping of an object-oriented model to a relational database
- Implements the specifications of Java Persistence API (JPA)
- Sits between Java objects and the database server



Hibernate Features



Hibernate Features



Hibernate-Supported Databases and Technologies

Databases

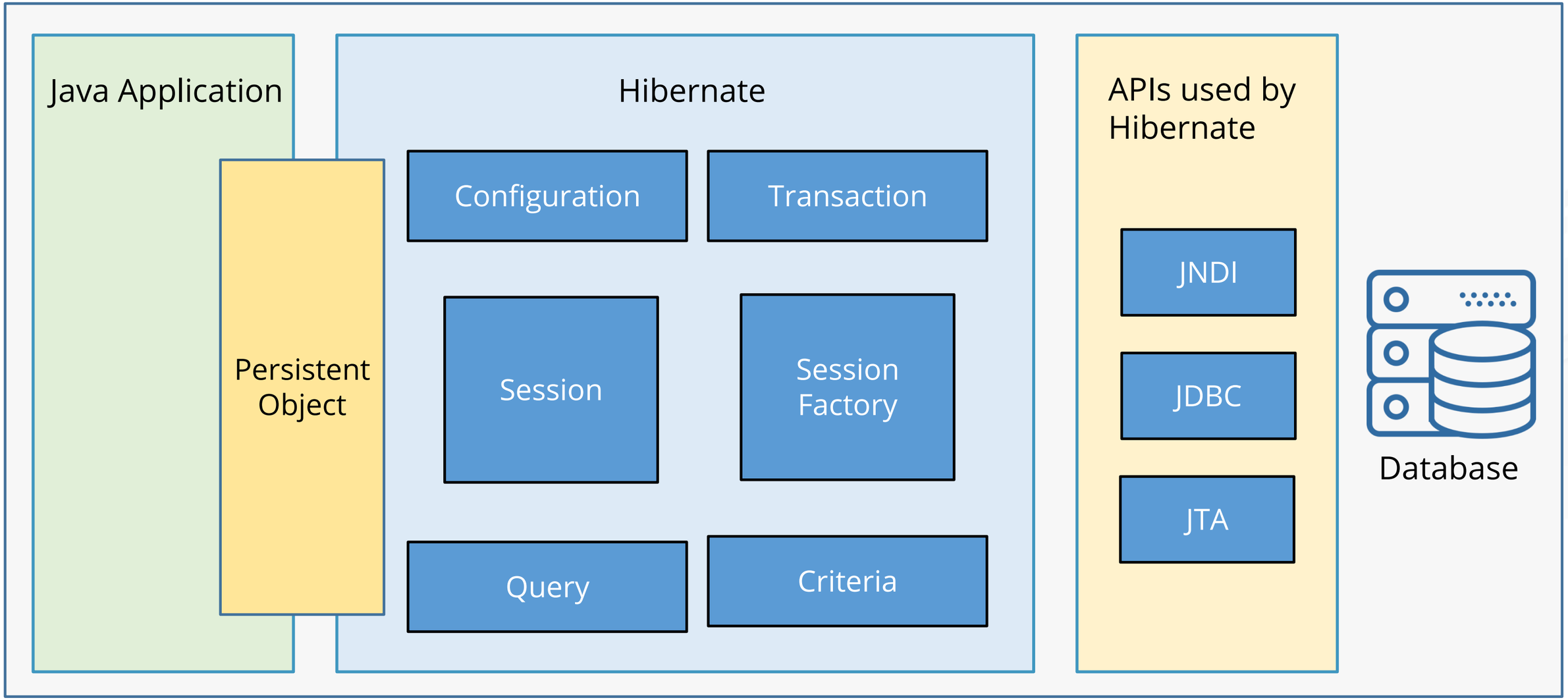
- Oracle
- MySQL
- PostgreSQL
- FrontBase
- Microsoft SQL Server Database
- Sybase SQL Server
- HSQL Database Engine
- DB2/NT
- Informix Dynamic Server

Technologies

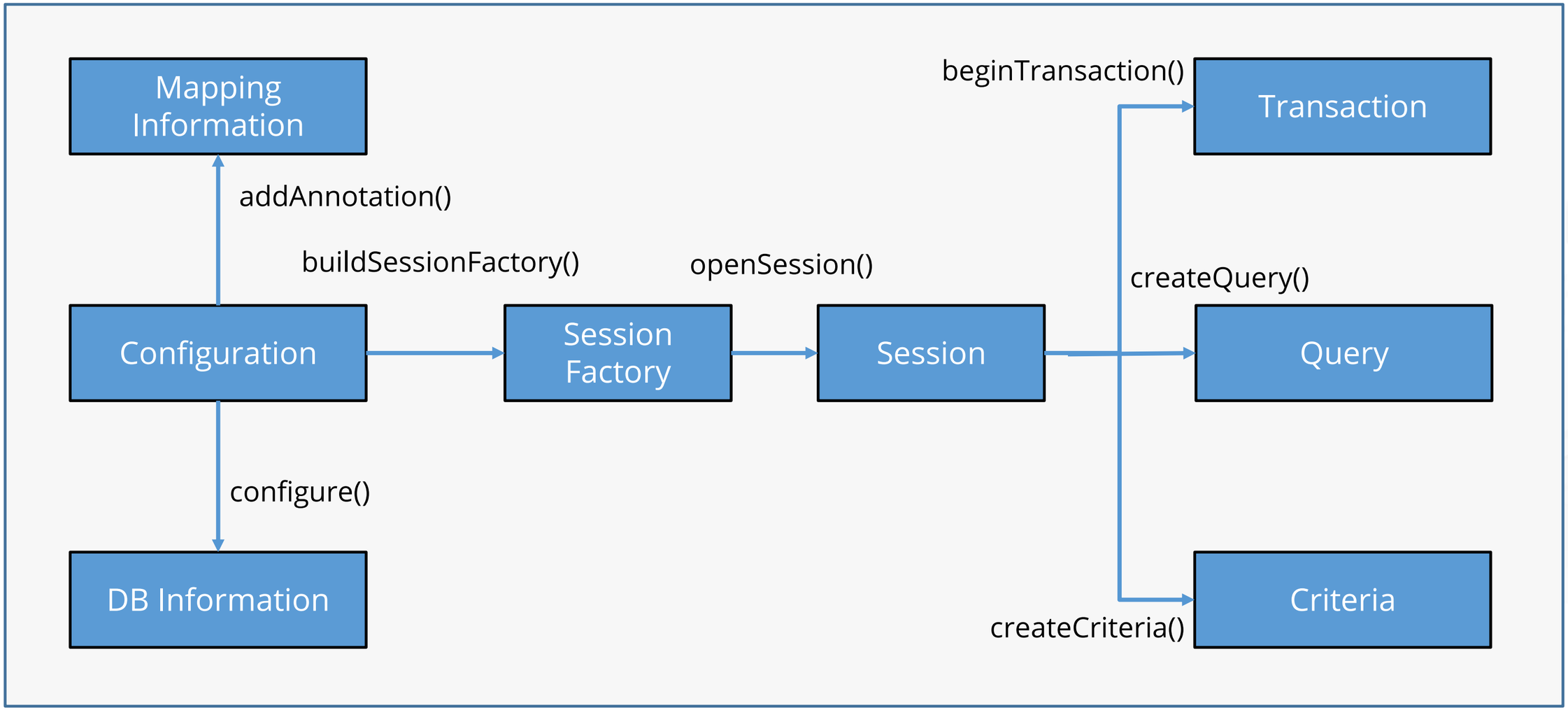
- XDoclet Spring
- J2EE
- Eclipse plug-ins
- Maven

High-Level Hibernate Architecture

Hibernate is a layered architecture that has Java Application layer, Hibernate layer, Backend API layer, and Database layer.



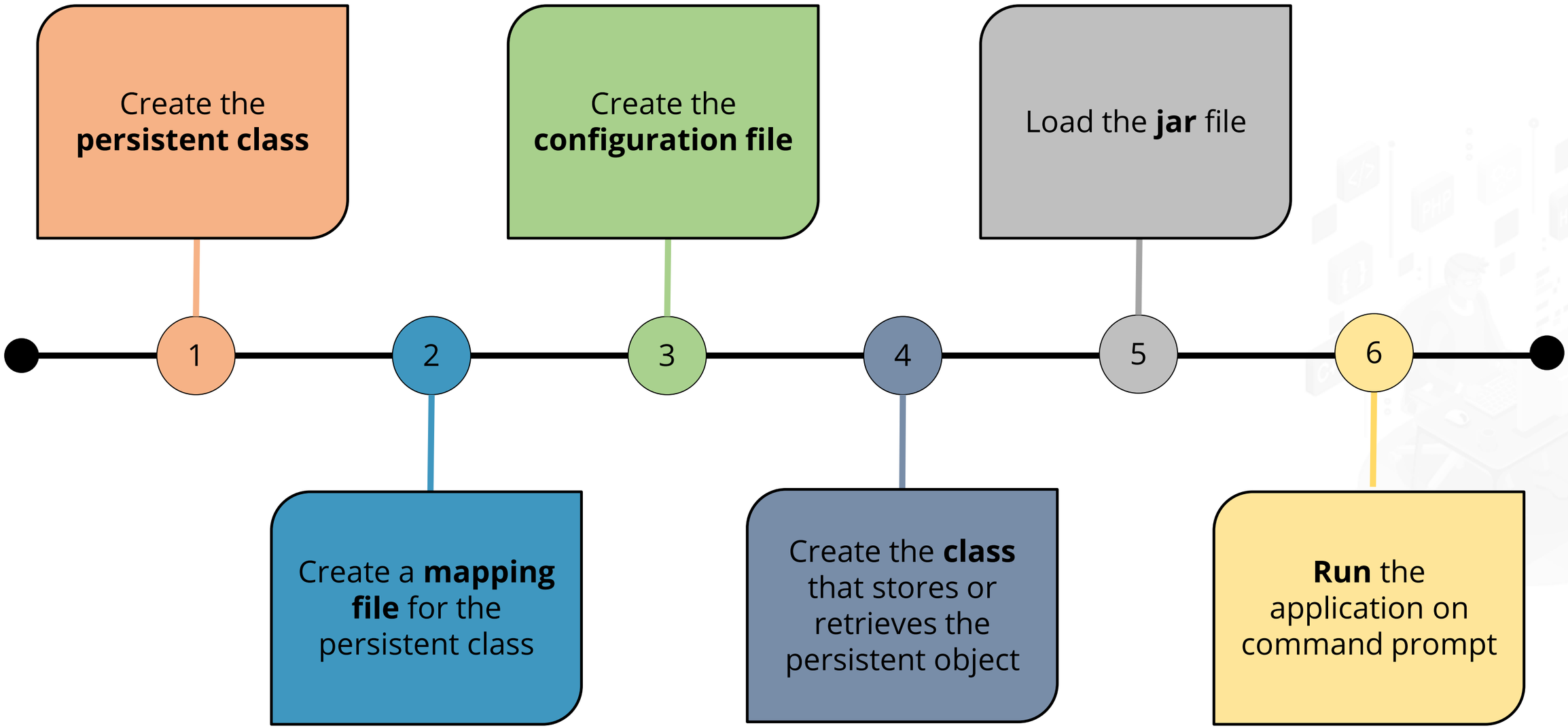
Hibernate Architecture



Hibernate Architecture: Elements

SessionFactory	Interface to hold optional (second level cache) data; factory for Session objects
Session	Interface to hold mandatory (first level cache) data; factory of Query, Criteria, and Transaction objects
Transaction	Interface with methods for transaction management
TransactionFactory	Optional factory of Transaction
ConnectionProvider	Optional JDBC connection factory that abstracts application from DriverManager or DataSource

Steps to Create a Hibernate Application on Command Prompt



Configuration of Hibernate

Duration: 20 min.

Problem Statement:

Configure Hibernate in Eclipse IDE.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to demonstrate hibernate configuration:

1. Create a dynamic web project in Eclipse IDE and add .jar files to load Hibernate and its dependencies.
2. Configure Hibernate with hibernate.cfg.xml
3. Create an HTML file that calls the servlet.
4. Run the HTML code on your browser.
5. Initialize the .git file.
6. Add and commit the program files.
7. Push the code to your GitHub repository.

Hibernate Configuration Using XML in Eclipse

Duration: 20 min.

Problem Statement:

Configure Hibernate using XML in Eclipse IDE.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to demonstrate hibernate using XML:

1. Create a dynamic web project in Eclipse IDE and add .jar files to load Hibernate and its dependencies.
2. Create a database (ecommerce) and a table (eproduct) that contains laptops for sale; fill in sample data.
3. Configure Hibernate with hibernate.cfg.xml and database table with .hbm.xml files.
4. Create Data access object class for the table.
5. Create a servlet that retrieves data and an HTML file that calls the servlet.
6. Run the HTML code on your browser.
7. Initialize the .git file.
8. Add and commit the program files.
9. Push the code to your GitHub repository.

Hibernate Configuration Using Annotations in Eclipse

Duration: 20 min.

Problem Statement:

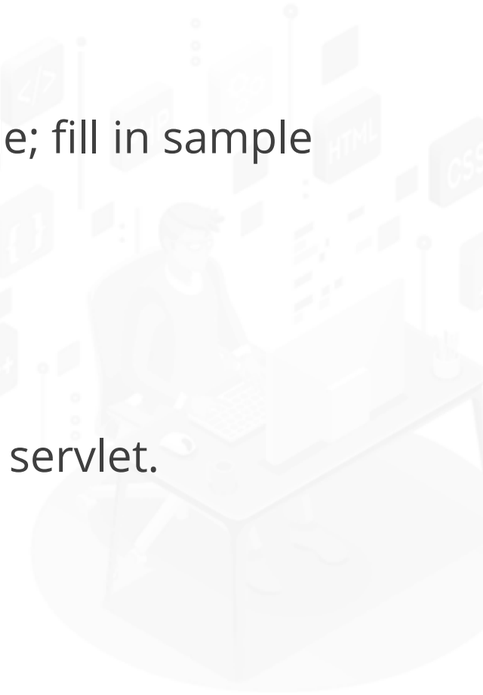
Configure Hibernate using Annotations in Eclipse IDE.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to demonstrate hibernate using annotations:

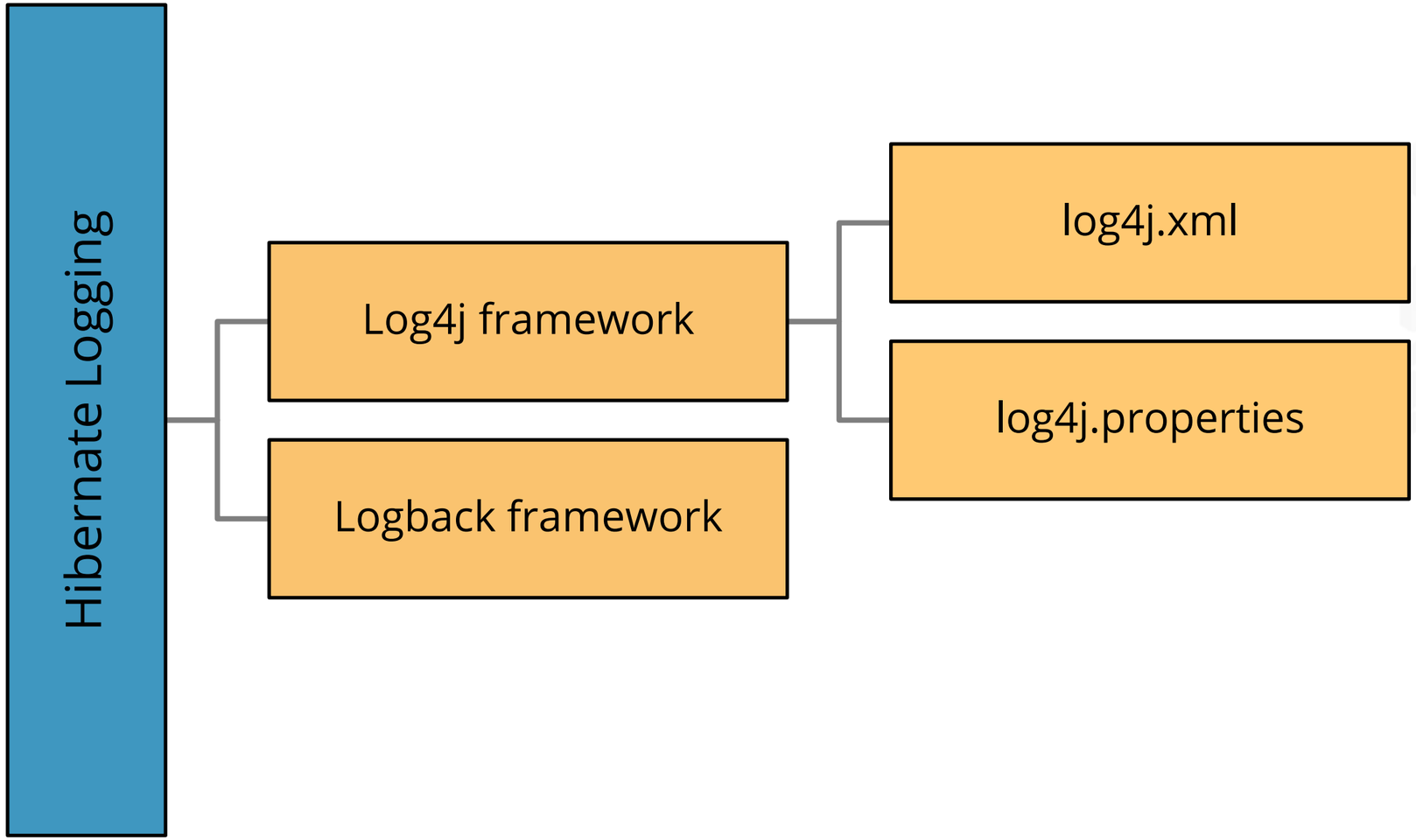
1. Create a dynamic web project in Eclipse IDE and add .jar files to load Hibernate and its dependencies.
2. Create a database (e-commerce) and a table (e-product) that contains laptops for sale; fill in sample data.
3. Configure Hibernate with hibernate.cfg.xml and database table with .hbm.xml files.
4. Create Data access object class for the table using Annotations.
5. Create a servlet that retrieves data using Annotations and an HTML file that calls the servlet.
6. Run the HTML code on your browser.
7. Initialize the .git file.
8. Add and commit the program files.
9. Push the code to your GitHub repository.



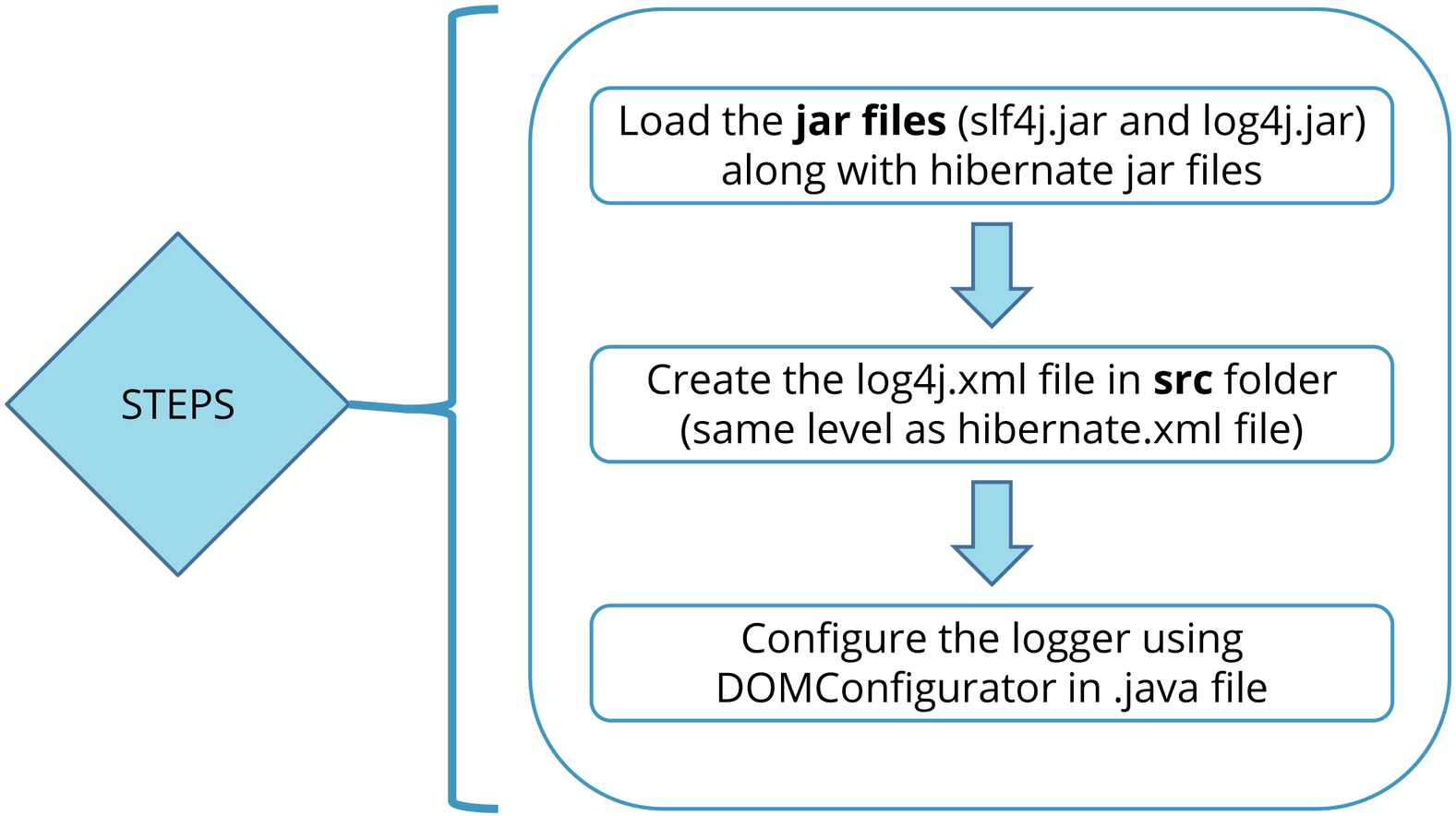
Hibernate Logging by Log4j

Hibernate Logging

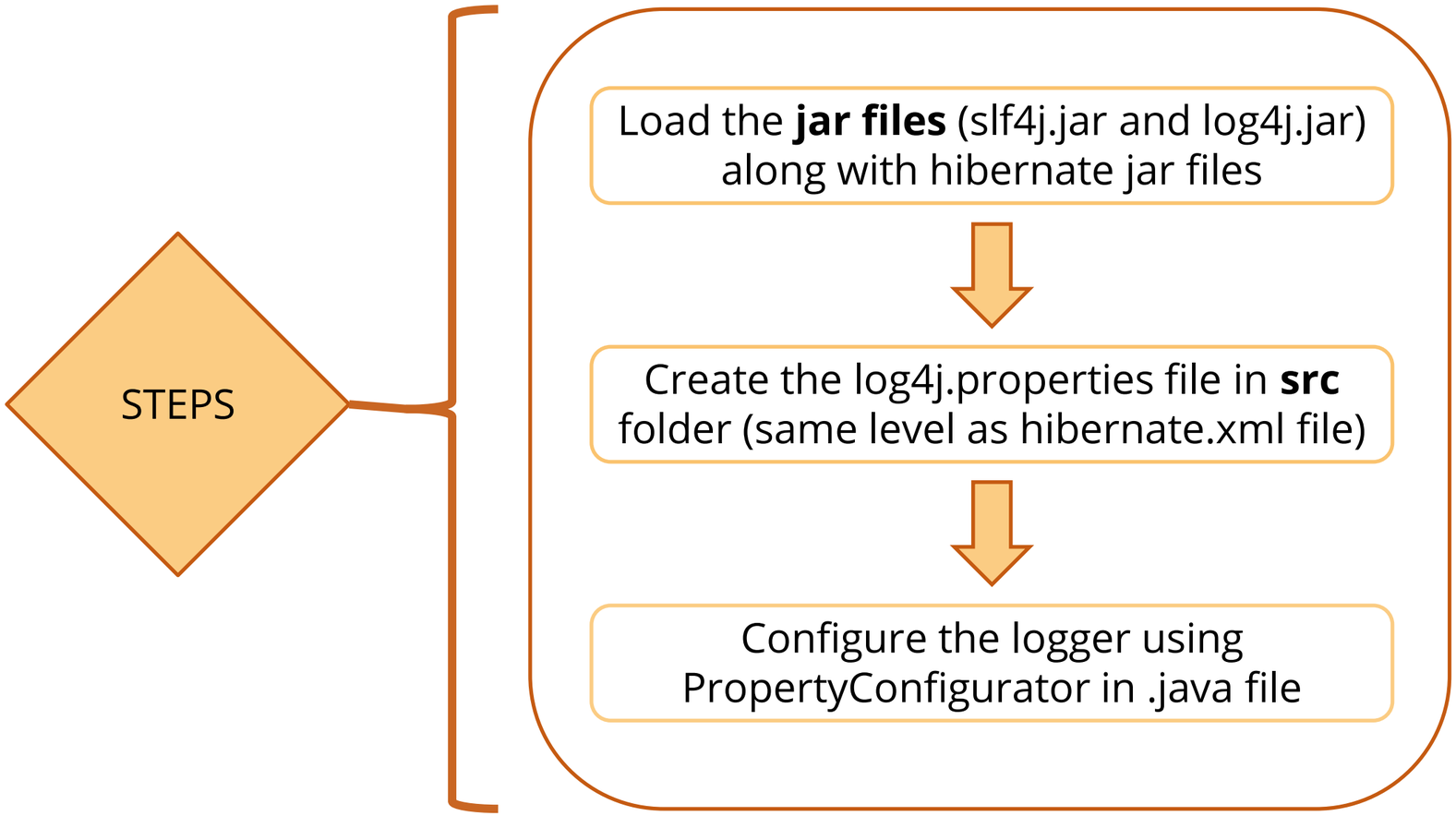
Hibernate framework uses Log4j and Logback frameworks for logging.



Hibernate Logging by Log4j Using log4j.xml



Hibernate Logging by Log4j Using log4j.properties



Hibernate Logging by Log4j

Duration: 20 min.

Problem Statement:

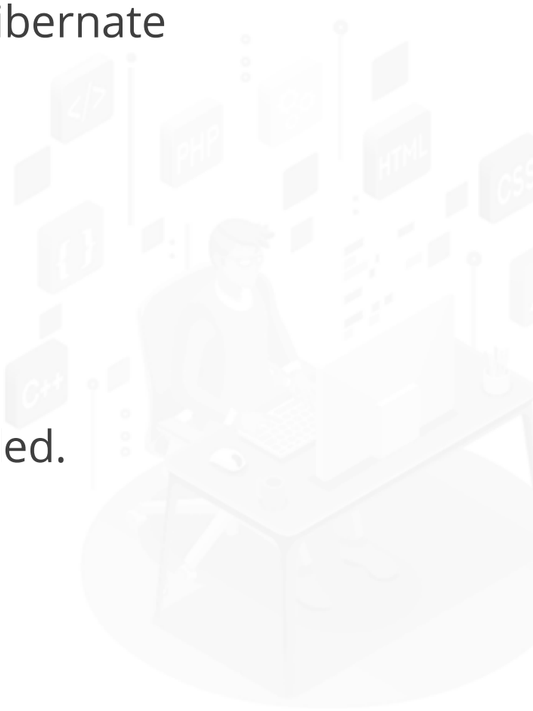
Demonstrate Hibernate logging by Log4j.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to demonstrate hibernate logging by Log4j:

1. Create a dynamic web project in Eclipse IDE and the required jar files to load Hibernate and its dependencies.
2. Add the required jar files to load log4j.
3. Configure Hibernate with hibernate,cfg.xml.
4. Create a servlet that opens a Hibernate session and closes it with logging enabled.
5. Create an HTML file that calls the servlet. Run the HTML file on the browser.
6. Initialize the .git file.
7. Add and commit the program files.
8. Push the code to your GitHub repository



Hibernate Mapping

Collection Mapping in Hibernate

Collection mapping as value type is supported by Hibernate

The type of collection in the persistent class must be one of the following types

- java.util.List
- java.util.Set
- java.util.SortedSet
- java.util.Map
- java.util.SortedMap
- java.util.Collection

Hibernate

also supports **implementation** of

org.hibernate.usertype.UserCollectionType

Collection Mapping Elements in Hibernate

Mapping elements used in Hibernate depend on the type of the interface.

Hibernate Mapping Elements

- <set>
- <list>
- <map>
- <bag>
- <array>
- <primitive-array>

Persistent objects can be mapped either in the **mapping file** or using **annotations**.

Collection Mapping Types in Hibernate

Mapping List in Collection Using XML File

In the mapping file, the **list** object can be mapped using the **<list>** element. A list is an index-based collection. **<index>** stores the key and **<element>** stores the value.

Example

```
<list name="sampleVal" table="sampleCol">
  <key column="name"></key>
  <index column="id"></index>
  <element column="samplname" type="string"></element>
</list>
```

Mapping Bag in Collection Using XML File

In the mapping file, the **bag** object can be mapped using the **<bag>** element.

Example

```
<bag name="sampleVal" table="sampleCol">
   <key column="name"></key>
   <element column="Samplename" type="string"></element>
 </bag>
```

Mapping Set in Collection Using XML File

In the mapping file, the **set** object can be mapped using the **<set>** element.

Example

```
<set name="sampleVal" table="sampleCol">
        <key column="name"></key>
        <element column="Samplename" type="string"></element>
</set>
```

Mapping Map in Collection Using XML File

In the mapping file, the **map** object can be mapped using the **<map>** element. A map is an index-based collection. **<index>** stores the key and **<element>** stores the value.

Example

```
<map name="sampleVal" table=" sampleCol " cascade="all">
   <key column="name"></key>
   <index column="id" type="string"></index>
   <element column="Samplename" type="string"></element>
</map>
```

Collection Mapping in Hibernate

Duration: 60 min.

Problem Statement:

Demonstrate mapping List, Set, Bag, and Map in collection using XML file.

ASSISTED PRACTICE

Assisted Practice: Guidelines

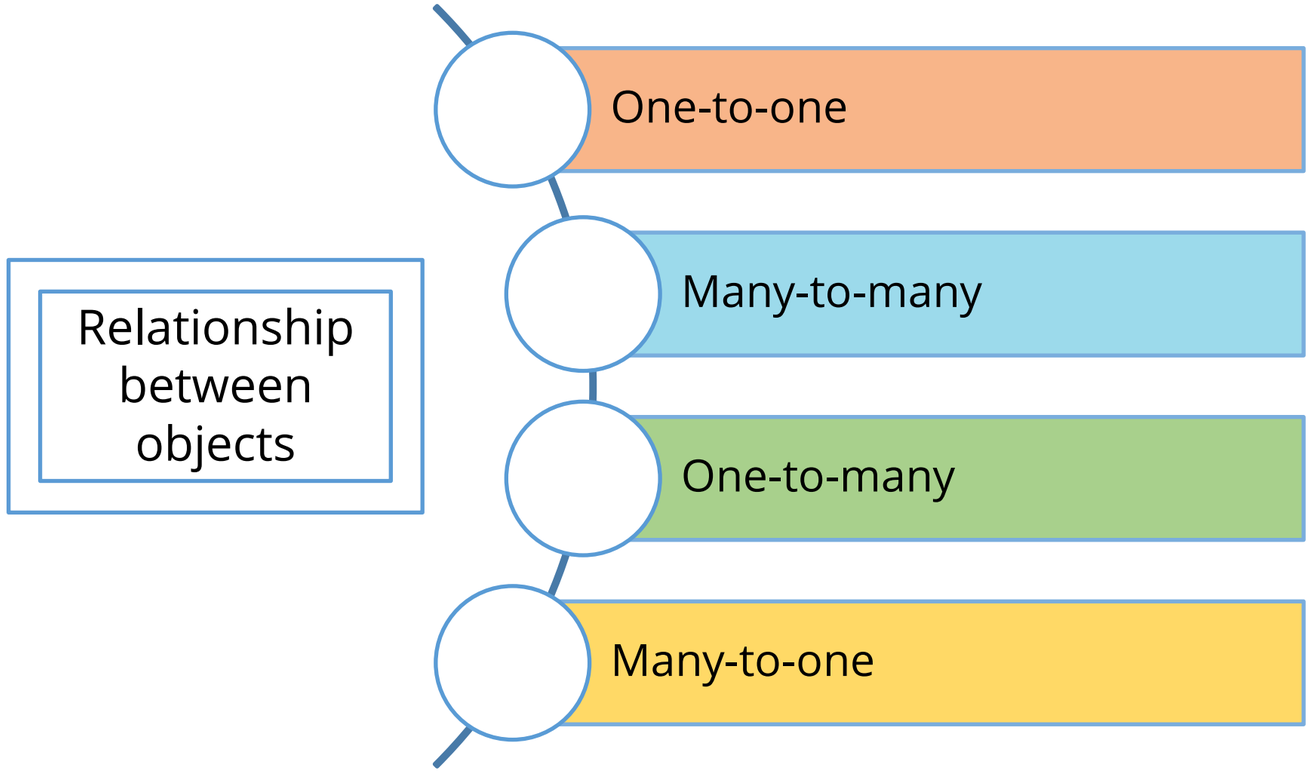
Steps to demonstrate collection mapping:

1. Create a dynamic web project in Eclipse IDE and add .jar files to load Hibernate and its dependencies.
2. Create a database (e-commerce) and a table (e-product) that contains laptops for sale; fill in sample and secondary data (operating system, colour, screen size, and finance option).
3. Configure Hibernate with hibernate.cfg.xml and database table with .hbm.xml files.
4. Create Data access object class for the table.
5. Create a servlet that retrieves data using Hibernate Collections and an HTML file that calls the servlet.
6. Run the HTML code on your browser.
7. Initialize the .git file. Add and commit the program files.
8. Push the code to your GitHub repository

Hibernate Mapping Using XML File

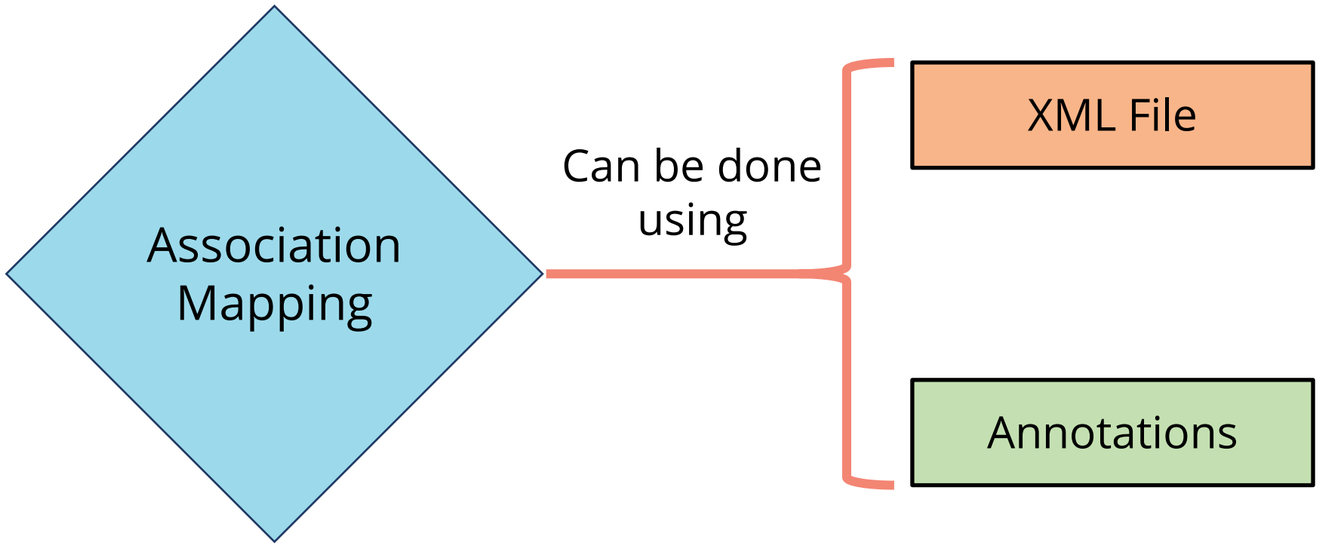
Association Mapping

Mapping of associations between entity classes and tables form the crux of Hibernate.
An association mapping can be unidirectional or bidirectional.



Mapping of java objects to database will be done in
Hibernate XML Mapping file (hbm.xml).

Association Mapping



Mapping of java objects to database will be done in **Hibernate XML Mapping file (<className>.hbm.xml)**.

Hibernate Mapping Using XML

One-to-one	One object can be associated with only one object
Many-to-many	Multiple objects can be associated with multiple objects. This mapping can be achieved using the Set collection
One-to-many	One object can be associated with multiple objects. This mapping can be achieved using the Set collection
Many-to-one	One object can be associated with multiple objects. This is the most common type of association

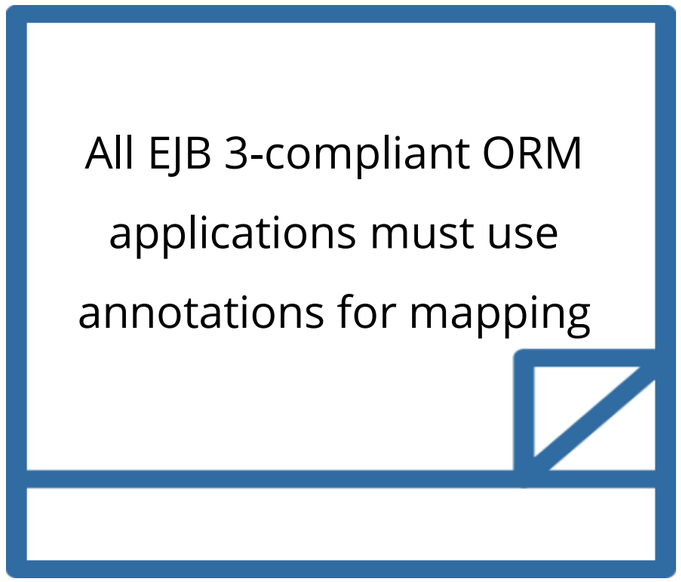
Hibernate Mapping Using XML

One-to-one	Declared using the <one-to-one> element
One-to-many	Declared using the <one-to-many> element
Many-to-many	Declared using the <many-to-many> element
Many-to-one	Declared using the <many-to-one> element

Hibernate Mapping Using Annotations

Hibernate Mapping Using Annotations

Annotations are used to define mapping. They are used in place of the XML file.



Hibernate Mapping Using Annotations

One-to-one	Syntax	`@OneToOne(targetEntity=<ClassName.class>>, cascade=CascadeTypeALL`
Many-to-many	Syntax	`@ManyToMany(fetch = FetchType.LAZY, cascade = CascadeType.ALL)`
One-to-many	Syntax	`@OneToMany(fetch = FetchType.LAZY, mappedBy = "<<className>>")`
Many-to-one	Syntax	`@ManyToOne(cascade = CascadeType.ALL)`

Bidirectional Mapping

Bidirectional Mapping

In bidirectional mapping, each entity has a property or relationship field that refers to the other entity. The entity class accesses the related object through the relationship field.

Example

Object A and Object B have a bidirectional mapping that means that Object A can be accessed from Object B and Object B from Object A.

Bidirectional association allows us to fetch details of dependent objects from both sides.

Bidirectional Mapping: Rules

In many-to-many relationship, either side can be the owning side

In one-to-one relationship, the side that contains the corresponding foreign key is the owning side

In many-to-one relationship, the many side is the owning side. Many side must not define ‘mappedBy’ element

In @OneToOne, @OneToMany, or @ManyToMany annotation, the owning side is referred by the inverse using ‘mappedBy’ element

Hibernate Lazy Collection

Lazy Collection

Lazy collection enables child objects to be loaded on demand. It is used to improve performance.

```
<list name="sampleList" lazy="true">

   <key column="sampleName"></key>
   <index column="type"></index>
   <one-to-many class="sampleClass"/>

 </list>
```

In Hibernate 3.0 and above, lazy collection is enabled by **default.**

Hibernate Lazy Collection

Duration: 15 min.

Problem Statement:

Demonstrate lazy collection in Hibernate.

ASSISTED PRACTICE

Assisted Practice: Guidelines

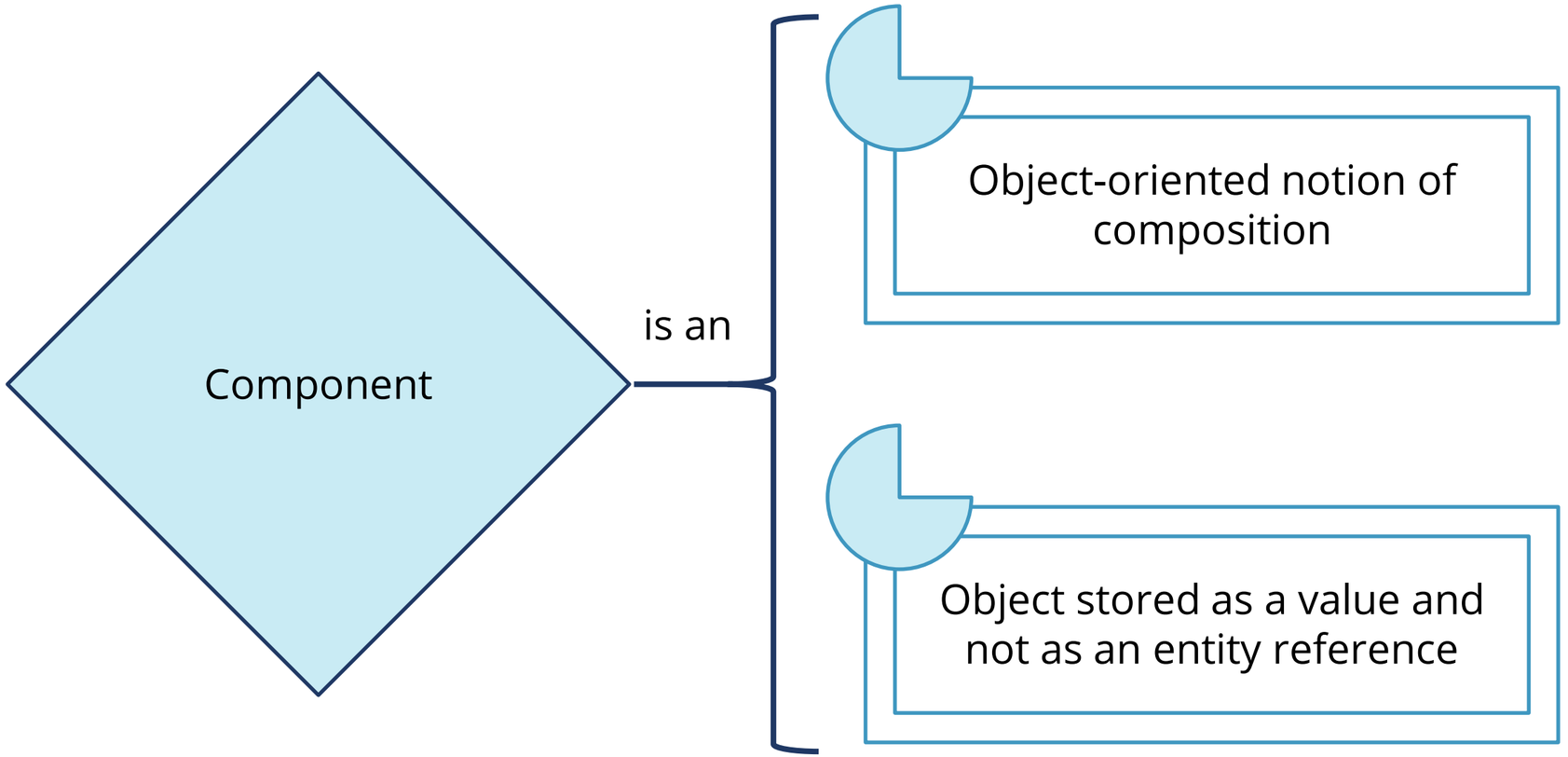
Steps to demonstrate hibernate lazy collection:

1. Create a dynamic web project in Eclipse IDE and add .jar files to load Hibernate and its dependencies.
2. Create a database (e-commerce) and a table (e-product) that contains laptops for sale; fill in sample and secondary data (operating system, colour, screen size and finance option).
3. Configure Hibernate with hibernate.cfg.xml and database table with .hbm.xml files.
4. Create Data access object class for each table.
5. Create a servlet that retrieves data using Hibernate Lazy Collections and an HTML file that calls the servlet. Run the HTML code on your browser.
6. Initialize the .git file. Add and commit the program files.
7. Push the code to your GitHub repository.

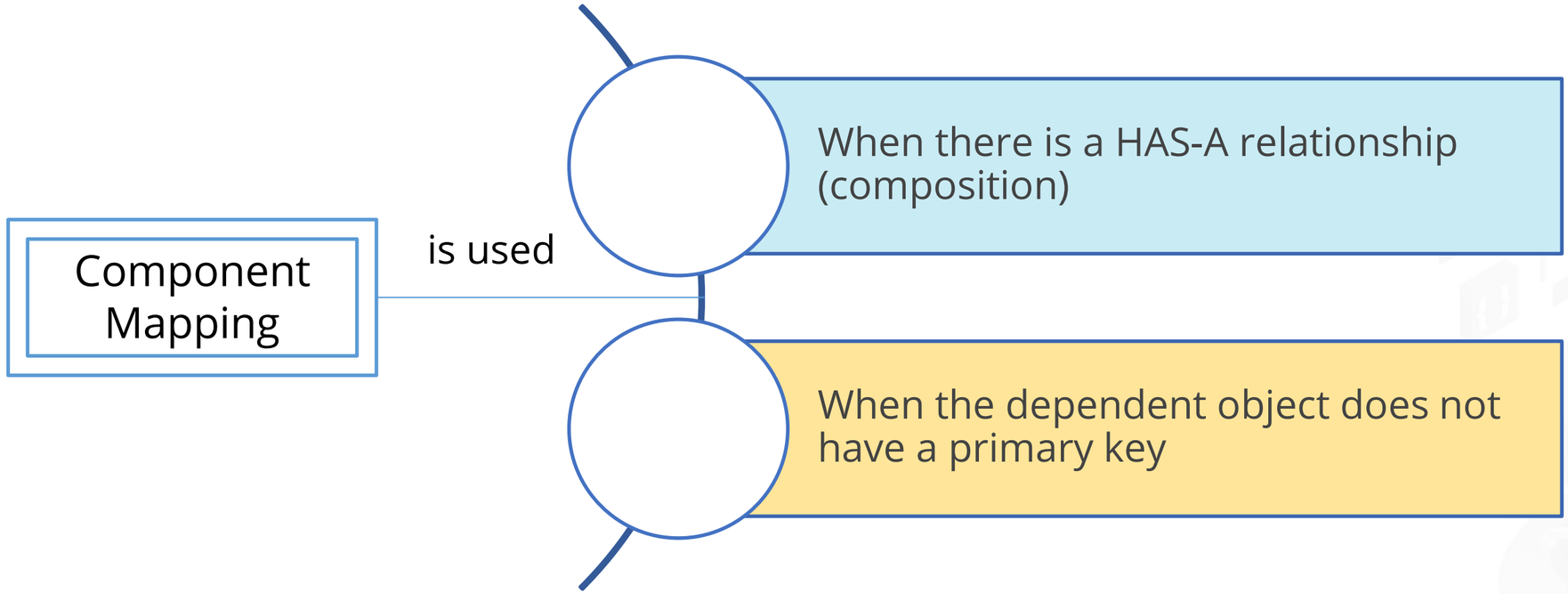
Component Mapping

Component Mapping

Component mapping allows a dependent object to be mapped as a component.



Component Mapping



Component Mapping

Duration: 15 min.

Problem Statement:

Demonstrate component mapping in Hibernate.

ASSISTED PRACTICE

Assisted Practice: Guidelines

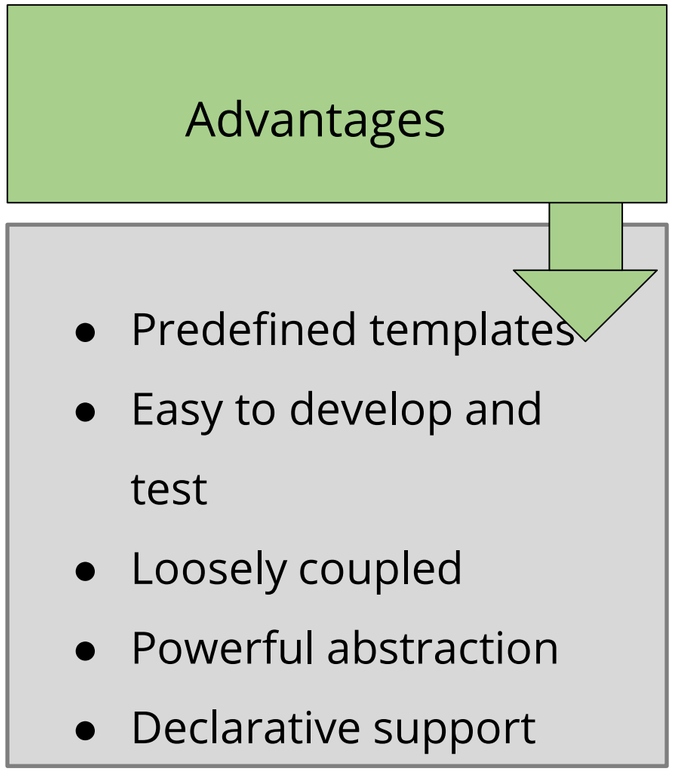
Steps to demonstrate component mapping:

1. Create a dynamic web project in Eclipse IDE and add .jar files to load Hibernate and its dependencies.
2. Create a database (e-commerce) and a table (e-product) that contains laptops for sale. Group the columns of parts_* in the table into a separate Hibernate class and link it to the main e-product class.
3. Configure Hibernate with hibernate.cfg.xml and database table with .hbm.xml files.
4. Create Data access object class for each table.
5. Create a servlet that retrieves data using main and component Hibernate classes and an HTML file that calls the servlet. Run the HTML code on your browser.
6. Initialize the .git file. Add and commit the program files.
7. Push the code to your GitHub repository.

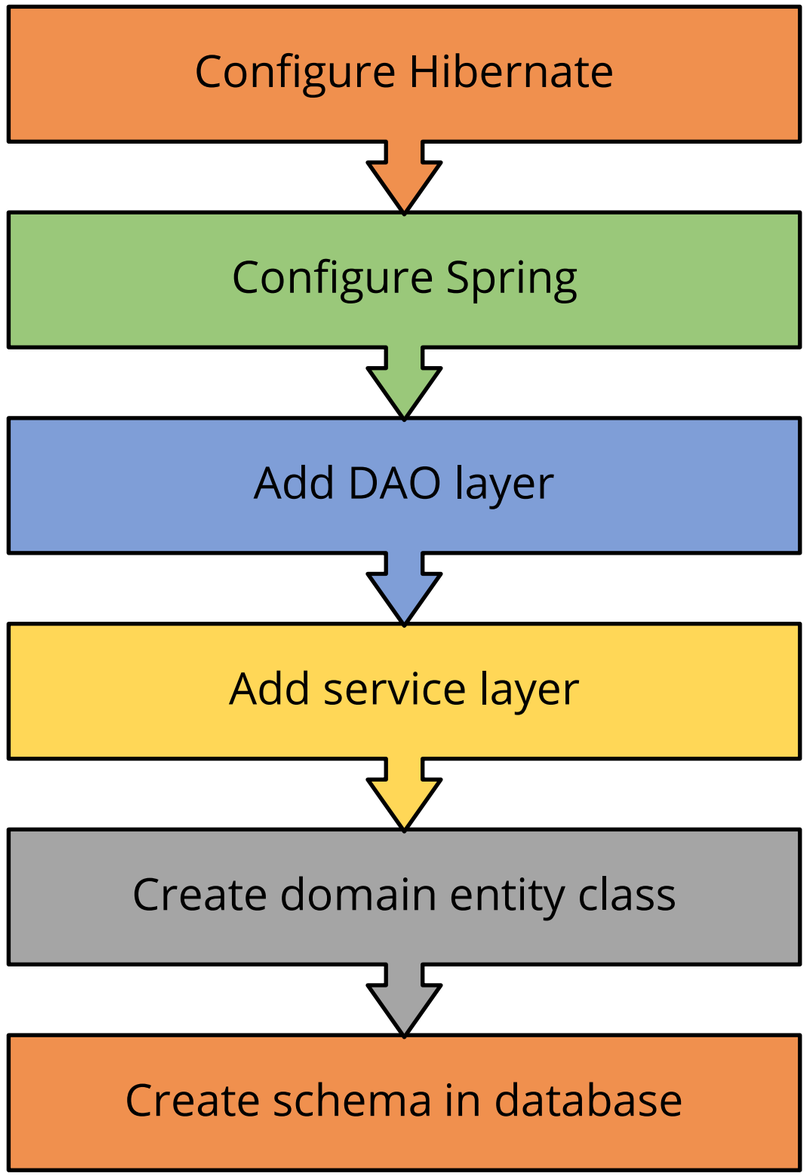
Integration of Hibernate with Spring

Spring

Spring framework is a lightweight and comprehensive configuration and programming model designed to develop Java-based applications.



Steps to Integrate Hibernate with Spring



Integration of Hibernate with Spring

Duration: 15 min.

Problem Statement:

Demonstrate integration of Hibernate with spring.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to Demonstrate Hibernate and Spring Integration:

1. Create a Maven web project in Eclipse IDE ; configure pom.xml to include the required dependencies.
2. Create a database (e-commerce) and a table (e-product) that contains laptops for sale.
3. Configure Hibernate with hibernate.cfg.xml.
4. Create entity, DAO, and controller classes for the table.
5. Setup the dispatcher servlet using XML.
6. Create a jsp page to view the e-product table data. Create an HTML file that calls the controller.
7. Run the HTML file on the browser.
8. Initialize the .git file. Add and commit the program files.
9. Push the code to your GitHub repository.

Key Takeaways

- Hibernate is open-source, performs fast, has automatic table creation, and provides query statistics.
- Hibernate supports Oracle database, MySQL database, PostgreSQL database, Front Base database, and Microsoft SQL Server database.
- Hibernate framework uses Log4j and Logback frameworks for logging.
- Spring framework is a lightweight and comprehensive configuration and programming model designed to develop Java-based applications.

Adding a New Product in the Database

Duration: 45 min.

Problem Statement:
Create a Java program to add a new product into the database.

LESSON-END PROJECT

Before the Next Class

Course: Java Certification Training Course

You should be able to:

- Explain JSP life cycle
- Create JSP elements
- Demonstrate JSP standard actions
- Use JSTL and custom tag libraries