# Easy Problem:-

//Easy Level (Spring DI with Java-based Configuration).

// Course.java

public class Course {

private String courseName;

private int duration;

public Course(String courseName, int duration) {

this.courseName = courseName;

this.duration = duration;

}

public String getDetails() {

return courseName + " (" + duration + " months)";

}

}

// Student.java

public class Student {

private String name;

private Course course;

public Student(String name, Course course) {

this.name = name;

this.course = course;

}

public void printStudentDetails() {

System.out.println("Name: " + name);

System.out.println("Course: " + course.getDetails());

}

}

// AppConfig.java

import org.springframework.context.annotation.\*;

@Configuration

public class AppConfig {

@Bean

public Course course() {

return new Course("Java", 3);

}

@Bean

public Student student() {

return new Student("Alice", course());

}

}

// Main.java

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class Main {

public static void main(String[] args) {

AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

Student student = context.getBean(Student.class);

student.printStudentDetails();

context.close();

}

}

# Medium:-

// Medium Level (Hibernate CRUD with MySQL)

// Student.java (Entity)

import javax.persistence.\*;

@Entity

@Table(name = "students")

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private int age;

// Getters and Setters

}

<!-- hibernate.cfg.xml -->

<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/testdb</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">root</property>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="show\_sql">true</property>

<mapping class="Student"/>

</session-factory>

</hibernate-configuration>

// Main.java (CRUD Operations)

import org.hibernate.\*;

import org.hibernate.cfg.Configuration;

public class Main {

public static void main(String[] args) {

SessionFactory factory = new Configuration().configure().buildSessionFactory();

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

Student s1 = new Student();

s1.setName("Bob");

s1.setAge(22);

session.save(s1);

tx.commit();

session.close();

factory.close();

}

}

# Hard :-

// Hard Level (Spring + Hibernate + Transaction Management)

// Account.java

import javax.persistence.\*;

@Entity

public class Account {

@Id

private int id;

private String name;

private double balance;

// Getters and Setters

}

// TransactionJava.java

import javax.persistence.\*;

@Entity

public class TransactionJava {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int transactionId;

private int fromAccount;

private int toAccount;

private double amount;

// Getters and Setters

}

<!-- applicationContext.xml -->

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd">

<bean id="sessionFactory" class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">

<property name="configLocation" value="classpath:hibernate.cfg.xml"/>

</bean>

<bean id="transactionManager" class="org.springframework.orm.hibernate5.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory"/>

</bean>

<tx:annotation-driven transaction-manager="transactionManager"/>

<context:component-scan base-package="com.example"/>

</beans>

// BankService.java

import org.springframework.transaction.annotation.Transactional;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class BankService {

@Autowired

private SessionFactory sessionFactory;

@Transactional

public void transfer(int fromId, int toId, double amount) {

Session session = sessionFactory.getCurrentSession();

Account from = session.get(Account.class, fromId);

Account to = session.get(Account.class, toId);

from.setBalance(from.getBalance() - amount);

to.setBalance(to.getBalance() + amount);

TransactionJava tx = new TransactionJava();

tx.setFromAccount(fromId);

tx.setToAccount(toId);

tx.setAmount(amount);

session.save(tx);

}

}

// Main.java

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BankService bankService = context.getBean(BankService.class);

try {

bankService.transfer(1, 2, 500);

System.out.println("Transaction successful.");

} catch (Exception e) {

System.out.println("Transaction failed: " + e.getMessage());

}

}

}