Problem 2

Consider the following 2 Entity classes

```
class Student {
    rollNo
    name
    email
    mobile
    courseId(Foreign key)
}

class Course {
    courseId
    courseName
    duration
    fee
}
```

The above two tables has following relationship :

- **One To Many, Many to One (Bidirectional)** : One Student can belongs to One Course and One course belong to many students

Implements the following methods :

1. **public void addCourse(Course course);**

- this method should add a course with the student details.

2. **public void getStudent(int roll_no)**
    - This method should get student along with the course details
    - If the Student is not present then throw **Student Not Found Exception**

3. **public void getCourse(int course_id)**
    - This method should get course details along with the details of all the students enrolled in this course
    - If the Course is not present then throw **Course Not Found Exception**

Reference

To make use of the persistence.xml file and for hibernate and mysql related dependencies make use of the following persistence.xml and pom.xml files with the required changes.

Sample of persistence.xml:

—----------

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd" version="2.0">
<persistence-unit name="" >
<properties>
<property name="hibernate.connection.driver_class"
value="com.mysql.cj.jdbc.Driver"/>
<property name="hibernate.connection.username" value=""/> <property name="hibernate.connection.password" value=""/> <property name="hibernate.connection.url"
value="jdbc:mysql://localhost:3306/db1"/>
<property name="hibernate.show_sql" value="true"/> <property name="hibernate.hbm2ddl.auto" value="update"/>
```

```
</properties> </persistence-unit>
</persistence>
```

Note: make appropriate changes in the above file like (persistence-unit name, database name, username, password, etc)

pom.xml

—----------

```
<!-- for changing the java version -->
<properties> <maven.compiler.source>1.8</maven.compiler.source> <maven.compiler.target>1.8</maven.compiler.target>
</properties>
<dependencies>
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core --> <dependency>
<groupId>org.hibernate</groupId> <artifactId>hibernate-core</artifactId> <version>5.6.11.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java --> <dependency>
<groupId>mysql</groupId> <artifactId>mysql-connector-java</artifactId> <version>8.0.28</version>
</dependency> </dependencies>
```

Note: make sure the MySQL-connector version in the above dependency, for different versions, make the change that version only.