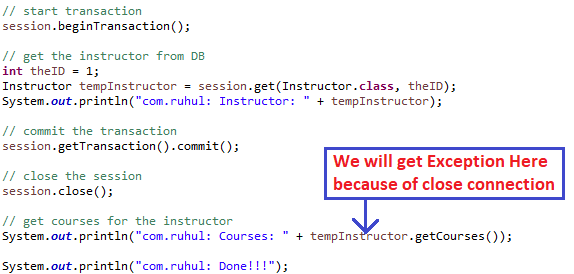
**25.4. Lazy Fetching Close the Session**

**Lazy Fetching Close the Session**:

1. When we use lazy load, the data is only retrieved on demand.
2. However, this requires an open Hibernate session.
3. We need a connection to database to retrieve data.
4. If the hibernate session is closed and we attempt to retrieve lazy data Hibernate will throw an exception

**File: LazyFatchingCloseSession.java**:

Now in our main app (**LazyFatchingCloseSession.java**) we will close our session and try to get the courses.



**RuntimeException**:

Exception in thread "main" org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role: com.ruhul.odduu.hibernate.entity.lazy.Instructor.courses, could not initialize proxy - no Session

. . . .

At com.ruhul.odduu.hibernate.demo.LazyFatchingCloseSession.main( LazyFatchingCloseSession.java:43)

**File: LazyFatchingCloseSession.java**

**package** com.ruhul.odduu.hibernate.demo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**import** com.ruhul.odduu.hibernate.entity.lazy.Course;

**import** com.ruhul.odduu.hibernate.entity.lazy.Instructor;

**import** com.ruhul.odduu.hibernate.entity.lazy.InstructorDetail;

**public** **class** LazyFatchingCloseSession {

**public** **static** **void** main(String[] args) {

// create session factory

SessionFactory factory = **new** Configuration(). configure("hibernate.cfg.xml").

addAnnotatedClass(Instructor.**class**).

addAnnotatedClass(InstructorDetail.**class**).

addAnnotatedClass(Course.**class**).

buildSessionFactory();

// create session

Session session = factory.getCurrentSession();

**try** {

// start transaction

session.beginTransaction();

// get the instructor from DB

**int** theID = 1;

Instructor tempInstructor = session.get(Instructor.**class**, theID);

System.***out***.println("com.ruhul: Instructor: " + tempInstructor);

// commit the transaction

session.getTransaction().commit();

// close the session

session.close();

// get courses for the instructor

System.***out***.println("com.ruhul: Courses: " + tempInstructor.getCourses());

System.***out***.println("com.ruhul: Done!!!");

} **finally** {

// add clean up code

session.close();

factory.close();

}

}

}

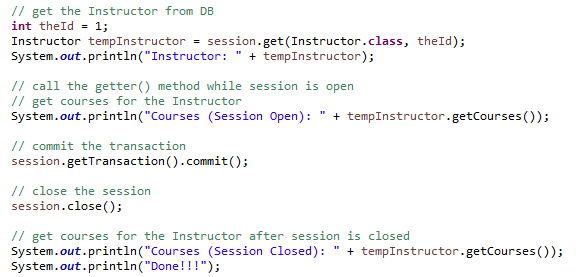
**Solution**:

We can solve this Exception in two ways.

1. **Option 1**: session.get() and call appropriate getter method(s)
2. **Option 2**: Hibernate query with HQL

**Option 1: session.get() and call appropriate getter method(s)**:

1. For resolve the lazy loading issue, we have to call the getter() method while session is open.
2. Now Courses will be loaded in memory and after close session if we call it again it don’t show any error.
3. After closing session, we get data from memory.



**File: LazyFatchingCloseSessionSolution1.class**:

**package** com.ruhul.odduu.hibernate.demo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**import** com.ruhul.odduu.hibernate.entity.lazy.Course;

**import** com.ruhul.odduu.hibernate.entity.lazy.Instructor;

**import** com.ruhul.odduu.hibernate.entity.lazy.InstructorDetail;

**public** **class** LazyFatchingCloseSession {

**public** **static** **void** main(String[] args) {

// create session factory

SessionFactory factory = **new** Configuration(). configure("hibernate.cfg.xml").

addAnnotatedClass(Instructor.**class**).

addAnnotatedClass(InstructorDetail.**class**).

addAnnotatedClass(Course.**class**).

buildSessionFactory();

// create session

Session session = factory.getCurrentSession();

**try** {

// start transaction

session.beginTransaction();

// get the instructor from DB

**int** theID = 1;

Instructor tempInstructor = session.get(Instructor.**class**, theID);

System.***out***.println("com.ruhul: Instructor: " + tempInstructor);

// call the getter() method while session is open

// get courses for the Instructor

System.***out***.println("Courses (Session Open): " +

tempInstructor.getCourses());

// commit the transaction

session.getTransaction().commit();

// close the session

session.close();

// get courses for the instructor

System.***out***.println("com.ruhul: Courses: " + tempInstructor.getCourses());

System.***out***.println("com.ruhul: Done!!!");

} **finally** {

// add clean up code

session.close();

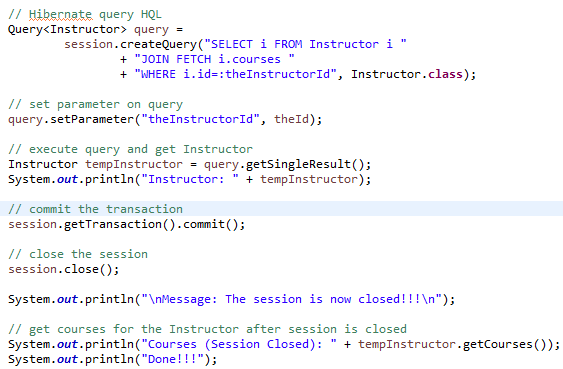
factory.close();

}

}

}

**2) Option 2: Hibernate query with HQL**:



25.4. Lazy Fetching Close the Session