

In your Hibernate project, you used the `get()` method to fetch a student object with an ID that doesn't exist in the database. What will be the result, and how would you handle it?

```
package com.sms;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

public class RetrievingRecord {

    public static void main(String[] args) {

        SessionFactory factory = HibernateUtil.getSessionFactory();

        // Open a new session

        Session session = factory.openSession();

        // Begin a transaction

        Transaction transaction = session.beginTransaction();

        // Retrieve the object that does not present using the primary key

        Student student = session.load(Student.class, "S106");

        // //display data using toString() method

        System.out.println(student);

        // Close the session

        session.close();

        // Open a another session

        Session session1 = factory.openSession();

        // Retrieve the same object using another session

        Student student1 = session1.load(Student.class, "S105");

        //display data using toString() method
```

```

        System.out.println(student);

// Close the session

        session1.close();

//close the session Factory

        factory.close();

    }

}

```

The screenshot shows the Eclipse IDE with a Java project named 'student_management_system'. The main file, 'RetrievingRecord.java', is open in the editor. The code defines a 'RetrievingRecord' class with a 'main' method that attempts to retrieve a student record with ID 'S106'. The code includes comments for each step: opening a session, beginning a transaction, loading the student, displaying data, closing the session, and opening another session.

The console output shows an exception: `org.hibernate.ObjectNotFoundException: No row with the given identifier exists: [com.sms.Student#S106]`. The stack trace indicates the error occurred in the 'main' method of 'RetrievingRecord.java' at line 17, where the student object was loaded. The stack trace also shows the Hibernate framework's internal handling of the exception.

```

1 package com.sms;
2
3 import org.hibernate.Session;
4
5
6
7 public class RetrievingRecord {
8     public static void main(String[] args) {
9         SessionFactory factory = HibernateUtil.getSessionFactory();
10        // Open a new session
11        Session session = factory.openSession();
12        // Begin a transaction
13        Transaction transaction = session.beginTransaction();
14        // Retrieve the object using the primary key
15        Student student = session.load(Student.class, "S106");
16        //display data using toString() method
17        System.out.println(student);
18        // Close the session
19        session.close();
20        // Open a another session
21        Session session1 = factory.openSession();

```

```

where
student0_.StudentId=?
Exception in thread "main" org.hibernate.ObjectNotFoundException: No row with the given identifier exists: [com.sms.Student#S106]
    at org.hibernate.boot.internal.StandardEntityNotFoundDelegate.handleEntityNotFound(StandardEntityNotFoundDelegate.java:28)
    at org.hibernate.proxy.AbstractLazyInitializer.checkTargetState(AbstractLazyInitializer.java:298)
    at org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:187)
    at org.hibernate.proxy.AbstractLazyInitializer.getImplementation(AbstractLazyInitializer.java:322)
    at org.hibernate.proxy.pojo.bytebuddy.ByteBuddyInterceptor.intercept(ByteBuddyInterceptor.java:45)
    at org.hibernate.proxy.ProxyConfigurations$InterceptorDispatcher.intercept(ProxyConfigurations.java:95)
    at com.sms.Student$HibernateProxy$9MjYv05q.toString(Unknown Source)
    at java.base/java.lang.String.valueOf(String.java:4213)
    at java.base/java.io.PrintStream.println(PrintStream.java:1060)
    at com.sms.RetrievingRecord.main(RetrievingRecord.java:17)

```

We can observe that the "Student" with ID S106 is not found in our database, which is why the "get" method is throwing an ObjectNotFoundException result.