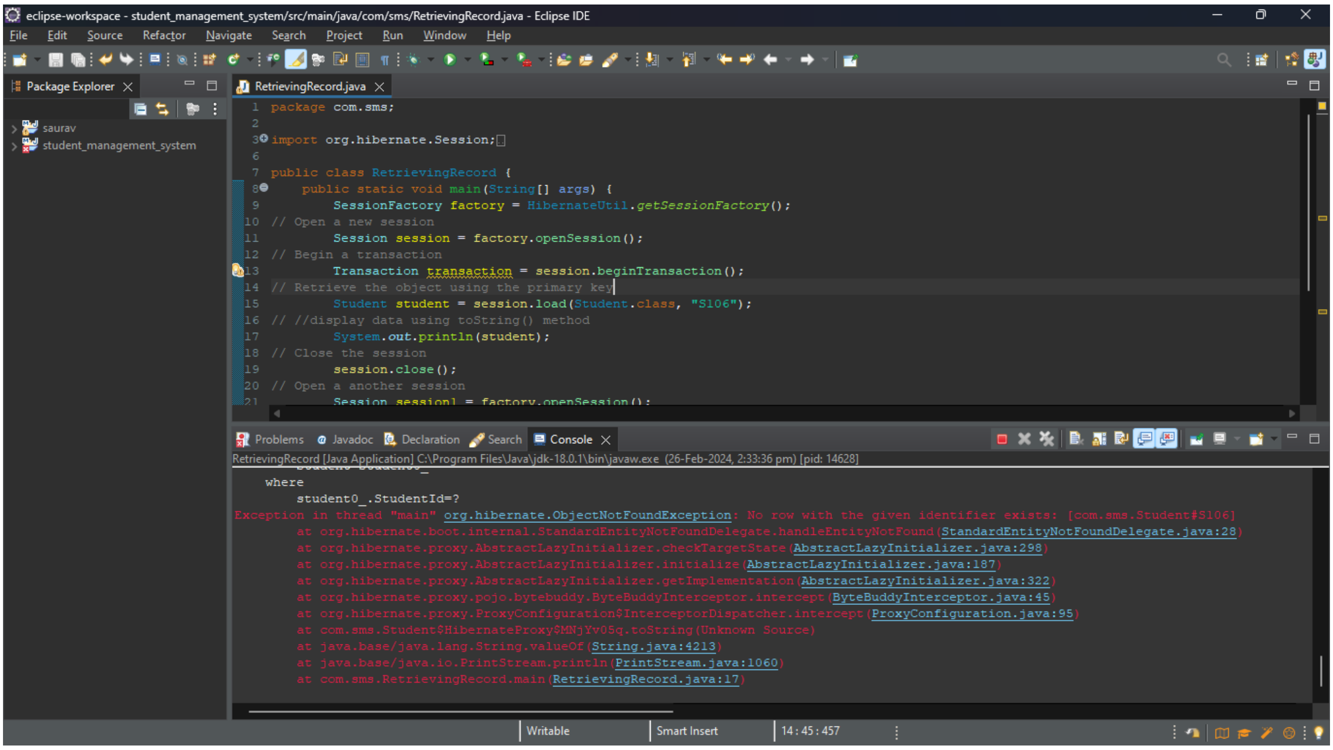
In your Hibernate project, you used the get() method to fetch a student object with an ID that doesn't exist in the database. What will be the result, and how would you handle it?

```
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();
            }
}
```


eclipse-workspace - student_management_system/src/main/java/com/sms/RetrievingRecord.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
saurav
student_management_system
RetrievingRecord.java
Problems Javadoc Declaration Search Console
Writable
Smart Insert
14 : 45 : 457

```java
package com.sms;

import org.hibernate.Session;

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 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();
```

RetrievingRecord [Java Application] C:\Program Files\Java\jdk-18.0.1\bin\javaw.exe (26-Feb-2024, 2:33:36 pm) [pid: 14628]

```
    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.ProxyConfiguration$InterceptorDispatcher.intercept(ProxyConfiguration.java:95)
	at com.sms.Student$HibernateProxy$MNjYv05q.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.