

**COURSE ID: 20TS3103AA**

**COURSE NAME: JAVA FULL STACK DEVELOPMENT + MICROSERVICES**

**LAB:02**

**ID:2000030185**

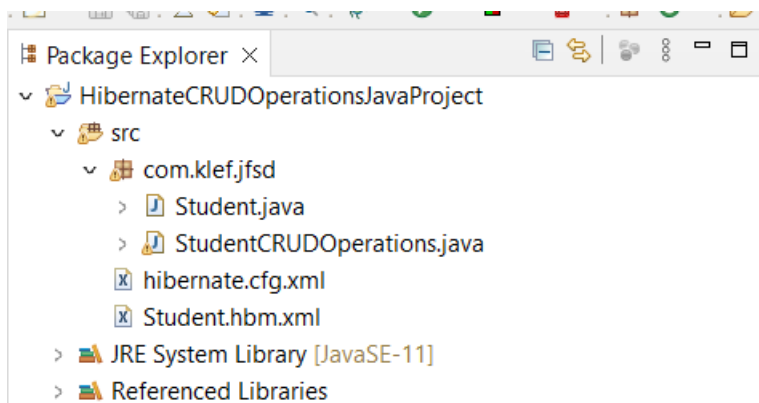
**NAME: SUPRITH CH**

QUESTION:

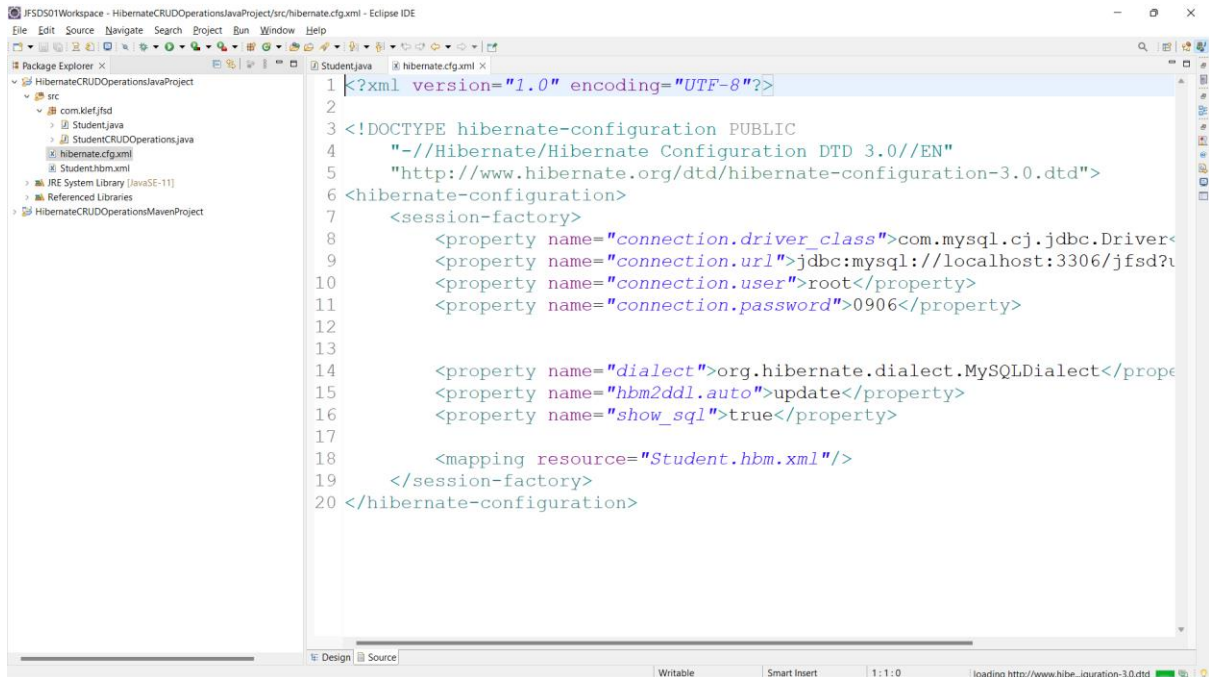
Rama, working as Assistant Professor at K L University, needs to keep up the information about his students in his database. He wants to save student details such as Id, name, gender, department, program, date of birth, contact number, graduation status, CGPA, no of backlogs.

Help him to develop a hibernate maven project to perform following operations

1. Insert the details of student graduated. The application should ask the user whether he wants to insert a detail of a student each time until he says 'no'. For every 'yes' it should gather the details of the student i.e., Id, name, gender, department, program, date of birth, contact number, graduation status, CGPA, no of backlogs.
2. Fetch a student record by asking student id as input using load () or get () method
3. Update specific student record by providing student id and details to be updated.
4. Delete specific student record based on student id.

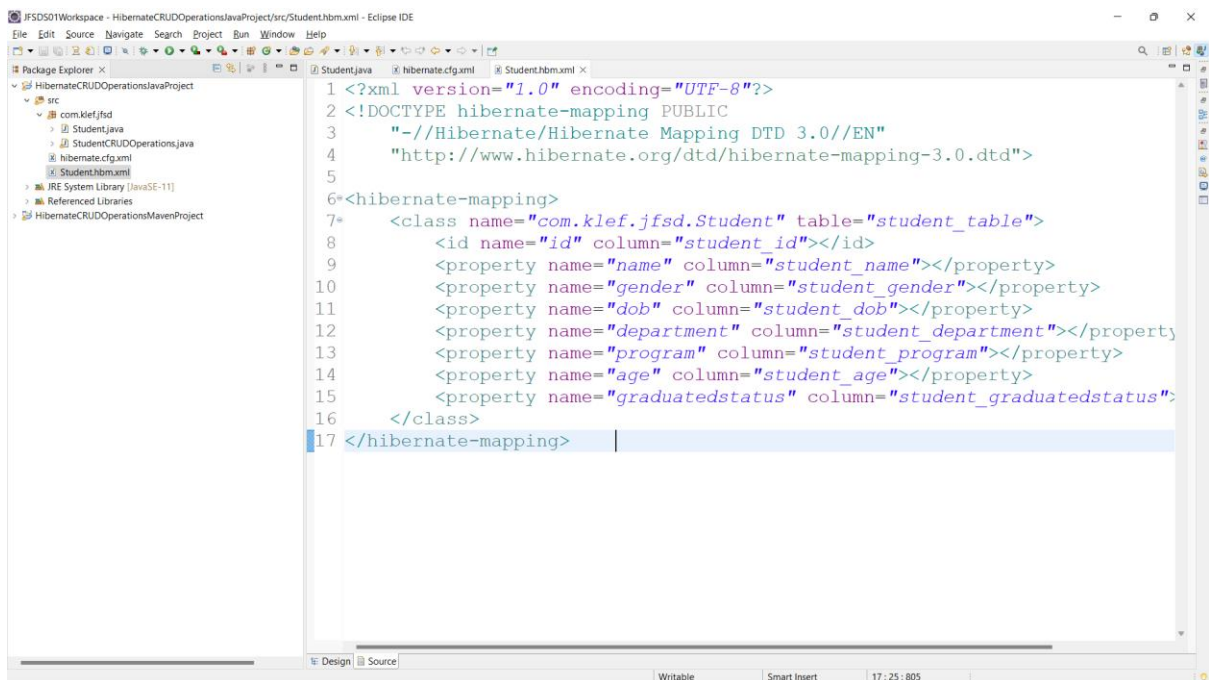


# Hibernate.cfg.xml



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!DOCTYPE hibernate-configuration PUBLIC
4     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
5     "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
6 <hibernate-configuration>
7     <session-factory>
8         <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
9         <property name="connection.url">jdbc:mysql://localhost:3306/jfsd?useUnicode=true&characterEncoding=utf8</property>
10        <property name="connection.user">root</property>
11        <property name="connection.password">0906</property>
12
13
14        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
15        <property name="hbm2ddl.auto">update</property>
16        <property name="show_sql">true</property>
17
18        <mapping resource="Student.hbm.xml"/>
19    </session-factory>
20 </hibernate-configuration>
```

# Student.hbm.xml



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3     "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4     "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
5
6 <hibernate-mapping>
7     <class name="com.klef.jfsd.Student" table="student_table">
8         <id name="id" column="student_id"></id>
9         <property name="name" column="student_name"></property>
10        <property name="gender" column="student_gender"></property>
11        <property name="dob" column="student_dob"></property>
12        <property name="department" column="student_department"></property>
13        <property name="program" column="student_program"></property>
14        <property name="age" column="student_age"></property>
15        <property name="graduatedstatus" column="student_graduatedstatus"></property>
16    </class>
17 </hibernate-mapping>
```

## POJO CLASS:

## Student.java

JFSD01Workspace - HibernateCRUDOperationsJavaProject/src/com/klef/jfsd/Student.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- hibernateCRUDOperationsJavaProject
  - src
    - com.klef.jfsd
      - Student.java
      - StudentCRUDOperations.java
      - hibernate.cfg.xml
      - Student.hbm.xml
  - JRE System Library [JavaSE-11]
  - Referenced Libraries
- hibernateCRUDOperationsMavenProject

```
1 package com.klef.jfsd;
2
3 public class Student
4 {
5     private int id;
6     private String name;
7     private String gender;
8     private String dob;
9     private String department;
10    private String program;
11    private double age;
12    private boolean graduatedstatus;
13    public int getId() {
14        return id;
15    }
16    public void setId(int id) {
17        this.id = id;
18    }
19    public String getName() {
20        return name;
21    }
22    public void setName(String name) {
23        this.name = name;
24    }
25    public String getGender() {
26        return gender;
```

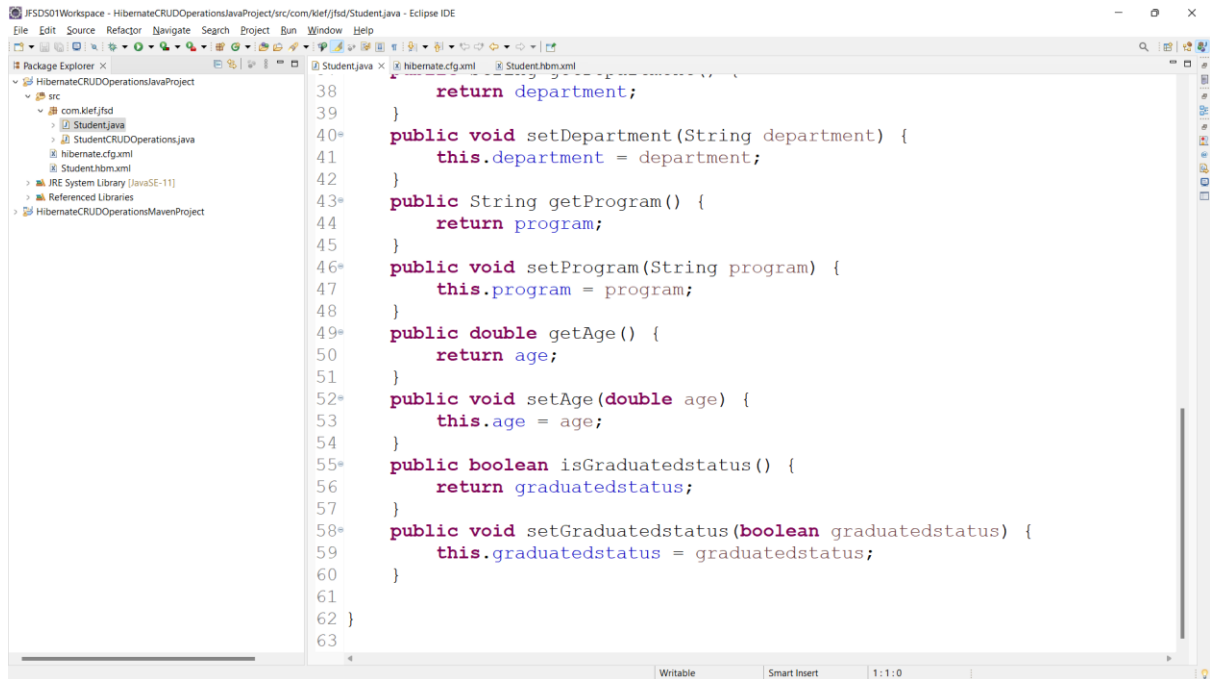
JFSD01Workspace - HibernateCRUDOperationsJavaProject/src/com/klef/jfsd/Student.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

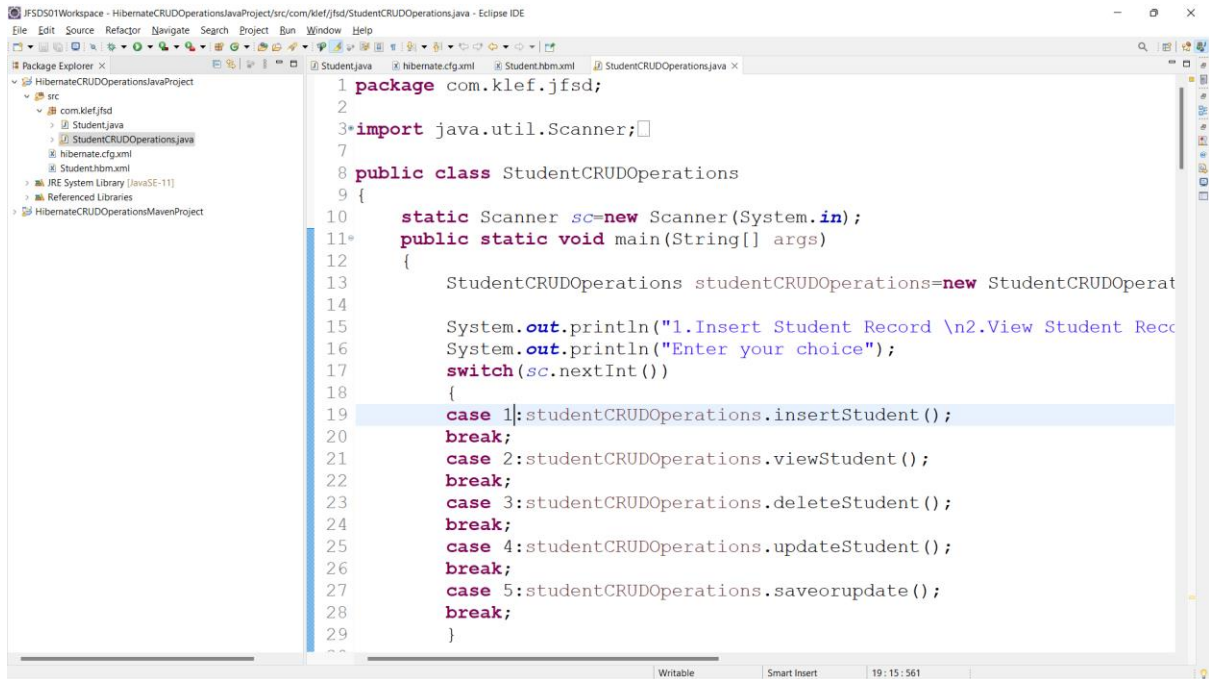
Package Explorer

- hibernateCRUDOperationsJavaProject
  - src
    - com.klef.jfsd
      - Student.java
      - StudentCRUDOperations.java
      - hibernate.cfg.xml
      - Student.hbm.xml
  - JRE System Library [JavaSE-11]
  - Referenced Libraries
- hibernateCRUDOperationsMavenProject

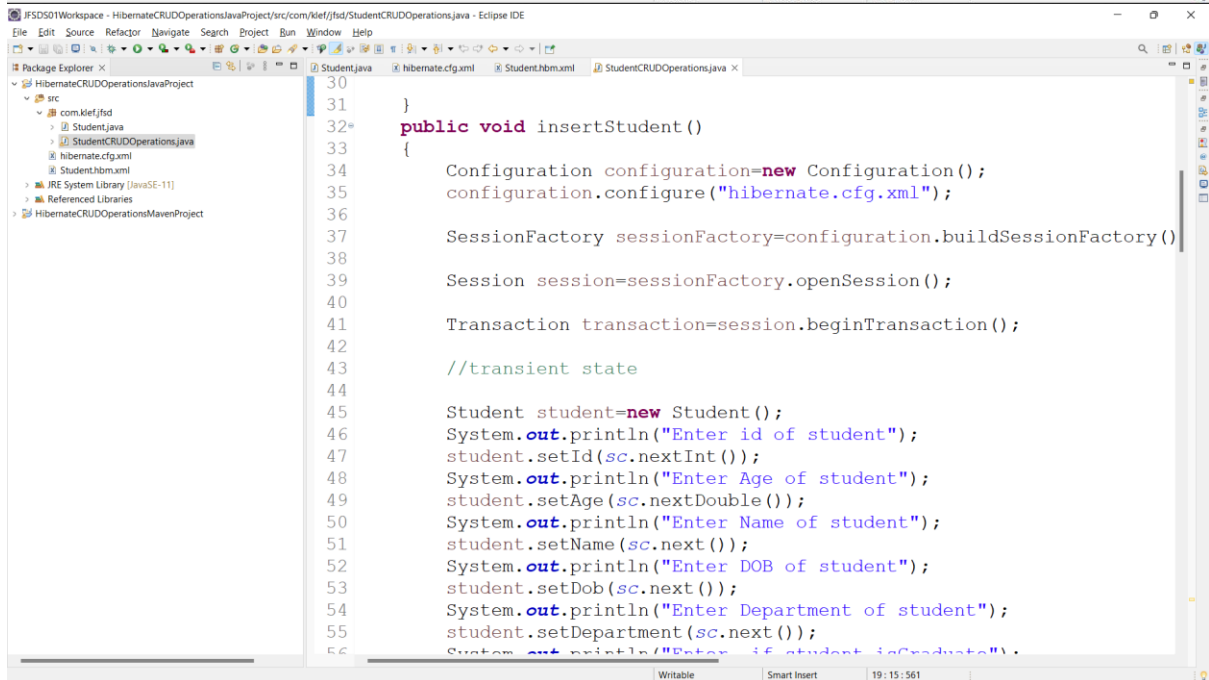
```
27
28    public void setGender(String gender) {
29        this.gender = gender;
30    }
31    public String getDob() {
32        return dob;
33    }
34    public void setDob(String dob) {
35        this.dob = dob;
36    }
37    public String getDepartment() {
38        return department;
39    }
40    public void setDepartment(String department) {
41        this.department = department;
42    }
43    public String getProgram() {
44        return program;
45    }
46    public void setProgram(String program) {
47        this.program = program;
48    }
49    public double getAge() {
50        return age;
51    }
52    public void setAge(double age) {
53        this.age = age;
```



## StudentCRUDOperations.java



```
1 package com.klef.jfsd;
2
3 import java.util.Scanner;
4
5
6
7 public class StudentCRUDOperations
8 {
9
10     static Scanner sc=new Scanner(System.in);
11     public static void main(String[] args)
12     {
13         StudentCRUDOperations studentCRUDOperations=new StudentCRUDOperations();
14
15         System.out.println("1.Insert Student Record \n2.View Student Record\n3.Delete Student Record\n4.Update Student Record\n5.Exit\nEnter your choice");
16         switch(sc.nextInt())
17         {
18             case 1:studentCRUDOperations.insertStudent();
19             break;
20             case 2:studentCRUDOperations.viewStudent();
21             break;
22             case 3:studentCRUDOperations.deleteStudent();
23             break;
24             case 4:studentCRUDOperations.updateStudent();
25             break;
26             case 5:studentCRUDOperations.saveOrUpdate();
27             break;
28         }
29     }
30 }
```



```
30 }
31
32 public void insertStudent()
33 {
34     Configuration configuration=new Configuration();
35     configuration.configure("hibernate.cfg.xml");
36
37     SessionFactory sessionFactory=configuration.buildSessionFactory();
38
39     Session session=sessionFactory.openSession();
40
41     Transaction transaction=session.beginTransaction();
42
43     //transient state
44
45     Student student=new Student();
46     System.out.println("Enter id of student");
47     student.setId(sc.nextInt());
48     System.out.println("Enter Age of student");
49     student.setAge(sc.nextDouble());
50     System.out.println("Enter Name of student");
51     student.setName(sc.next());
52     System.out.println("Enter DOB of student");
53     student.setDob(sc.next());
54     System.out.println("Enter Department of student");
55     student.setDepartment(sc.next());
56     System.out.println("Enter if student is Graduate");
57 }
```

```
54      System.out.println("Enter Department of student");
55      student.setDepartment(sc.next());
56      System.out.println("Enter if student isGraduate");
57      student.setGraduatedstatus(sc.nextBoolean());
58      System.out.println("Enter Program of student");
59      student.setProgram(sc.next());
60      System.out.println("Enter Gender of student");
61      student.setGender(sc.next());
62
63      session.save(student); //persistent
64
65      transaction.commit(); //results will be updated in the database
66
67      System.out.println("Student Object Inserted Succsfully");
68
69      session.close(); //detach state
70      sessionFactory.close();
71  }
72  public void viewStudent()
73  {
74      Configuration configuration=new Configuration();
75      configuration.configure("hibernate.cfg.xml");
76
77      SessionFactory sessionFactory=configuration.buildSessionFactory()
78
79      Session session=sessionFactory.openSession();
80  }
```

```
72  public void viewStudent()
73  {
74      Configuration configuration=new Configuration();
75      configuration.configure("hibernate.cfg.xml");
76
77      SessionFactory sessionFactory=configuration.buildSessionFactory()
78
79      Session session=sessionFactory.openSession();
80
81      //Object obj=session.load(Student.class,101);
82      //Object obj=session.load(Student.class,105);
83      //Object obj=session.get(Student.class,101);
84      System.out.println("Enter id of student to view Details");
85      Object obj=session.get(Student.class,sc.nextInt());
86      Student student=(Student)obj;
87      if(student!=null)
88      {
89          System.out.println("Student Id: "+student.getId());
90          System.out.println("Student Name: "+student.getName());
91          System.out.println("Student Gender: " +student.getGender());
92          System.out.println("Student Dob: " +student.getDob());
93          System.out.println("Student Department: " +student.getDepartm
94          System.out.println( "Student Program: " +student.getProgram())
95          System.out.println("Student Age: " +student.getAge());
96          String b="No";
97          if(student.isGraduatedstatus())
```