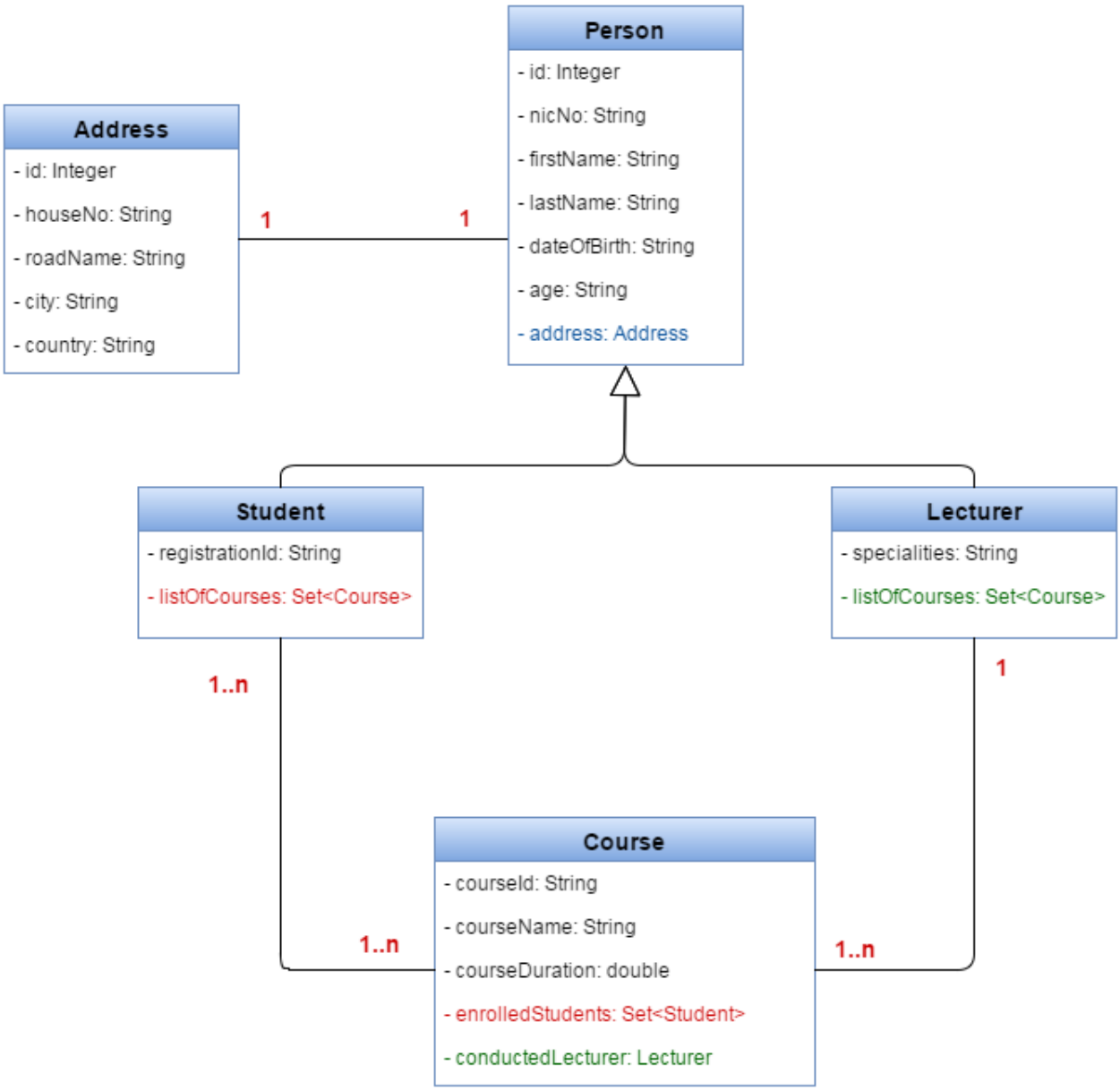
Consider the following scenario of the Hypothetical Demo Institute.

“Demo Institute” is an institute which conduct courses under two main streams such as Homicide Investigations and Martial Arts. They started their operation few years back and they relied on a manual process for maintaining their business for the past few years.

However, due the growth of their student base and the wide popularity they earned, it became hard for them to continue the manual process so that they decided to invest on a computer based system application.

The Software Architect working for the particular project has decided to implement the solution using Java technologies and decided to use Hibernate in Data Access Object layer.



The Technical lead has come up with the following class diagram and decided to deliver the requirements mentioned below in this document as for the first iteration.

Classes and related Information

We identified that both Student and Lecturer has common properties so that they were specified in Person class and it has been extended by both Student and Lecturer classes.
(**Hint:** Use Joined Inheritance strategy by using **@Inheritance(strategy = InheritanceType.JOINED)** annotation on top of Person class)

The primary key for both Student and Person will be the id field which will be auto generated when persisting a new Student or a Lecture object.
(**Hint:** use **@Id** annotation for the primary key field and use **@GeneratedValue** annotation to define the auto generation strategy)

The nicNo field should be unique.
(**Hint:** use **unique** attribute of **@Column** annotation)

The age field should not be persisted in the database.
(**Hint:** use **@Transient** annotation)

It is decided to collect the address data when inserting a student / lecturer but decided to save data in a separate table. In this scenario, you have one to one relationship with Person and Address. (The person class will have a reference field called address which is shown in blue color text in above class diagram)
(**Hint:** use **@OneToOne** annotation)

The primary key for the Address will be the id field which will be auto generated.

Students can enrolled for multiple courses and a course can be followed by multiple students. The required referenced fields for this relationship is shown in red color in above class diagram.
(**Hint:** use **@ManyToMany** annotation on both sides)

The middle relationship table getting created for above relationship should have the name STUDENT_ENROLED_COURSES_TAB and its column names should be STUDENT_ID and COURSE_ID)
(**Hint:** use **@JoinTable** annotation)

A Lecturer can conduct multiple courses and a particular course can be conducted by one lecturer only. The required referenced fields for this relationship is shown in green color in above class diagram.
(**Hint:** use **@OneToMany** and **@ManyToOne** annotations as appropriate)

The primary key for the Course will be the courseId field which you have to specify manually when persisting a course.

Please note that all the tables getting created for above classes should contain their column names changed as follows.
firstName = FIRST_NAME
courseName = COURSE_NAME
…etc
(**Hint:** use **name** attribute of the **@Column** annotation for appropriate fields)

Please note that the tables getting created for the classes should append "_TAB" as for the table name.
Ex:
Person = PERSON_TAB
Course = COURSE_TAB
…etc.
(**Hint: use @Table(name = "TABLE_NAME")** annotation)

Requirements

1. Create a Java Console application named "DemoInstitute" and add Hibernate libraries to it.

2. Configure the hibernate.cfg.xml file to do following
    a. Tables should be auto generated when loading the application.
    b. SQL statements executed by Hibernate should be logged to the standard output console.
    c. Incorporate the following property too.
    hibernate.current_session_context_class=thread

3. Create the Entity classes inside com.demoinstitute.entity package.

4. In every entity class, override the toString() method as you did in tutorial session and return the value of the fields.

5. Create the SessionFactoryUtil.java class inside com.demoinstitute.util package. This is the class which is responsible for creating the session factory by reading the Hibernate configuration file. Please refer the class you created for the tutorial session.

6. Write three service classes under the com.demoinstitute.service package each containing four methods for CRUD operations. Refer the methods you wrote in tutorial session.
    a. CourseService.java

b. LecturerService.java
c. StudentService.java

7. Write an additional business method in StudentService.java to list the students who are enrolled for a provided course as follows.

   public void listStudentsFollowingCourse(String courseId)

8. Create the Main.java class inside com.demoinstitute.console package. Declare the main method and write logic to insert Students, Lecturers, and Courses, enrolling of Students to courses and assigning lecturers to courses as mentioned below. You may call the methods you wrote in the service classes you created in step 6 to persist and update records.

9. Persist four students as follows with their addresses.

Student 1 Details	**Student 1 Address Details**
nicNo = 927656351V	houseNo = 65/33
registrationId = DIT-13-M1-0056	roadName = Kandy Road
firstName = Jackie	city = Malabe
lastName = Chan	country = Sri Lanka
dateOfBirth = 05/27/1992	

Student 2 Details	**Student 2 Address Details**
nicNo = 936724590V	houseNo = 21/5
registrationId = DIT-14-M1-0023	roadName = Galle Road
firstName = Jet	city = Mount Lavinia
lastName = Li	country = Sri Lanka
dateOfBirth = 03/12/1993	

Student 3 Details	**Student 3 Address Details**
nicNo = 854524340V	houseNo = 29/35
registrationId = DIT-06-M1-0010	roadName = Kandy Road
firstName = Adrian	city = Peradeniya
lastName = Monk	country = Sri Lanka
dateOfBirth = 10/10/1985	

Student 4 Details	Student 4 Address Details
nicNo = 903476126V	houseNo = 11/32
registrationId = DIT-11-M1-0009	roadName = Galle Road
firstName = Patrick	city = Panadura
lastName = Jane	country = Sri Lanka
dateOfBirth = 09/09/1990	

10. Persist two lecturers as follows with their addresses.

Lecturer 1 Details	Lecturer 1 Address Details
nicNo = 401782987V	houseNo = 99/99
specialities = Martial Arts	roadName = Chinatown
firstName = Bruce	city = San Francisco
lastName = Lee	country = USA
dateOfBirth = 11/27/1940	

Lecturer 1 Details	Lecturer 1 Address Details
nicNo = 541084368V	houseNo = 221B
specialities = Homicide Detective	roadName = Baker Street
firstName = Sherlock	city = London
lastName = Holmes	country = UK
dateOfBirth = 2/22/1950	

11. Persist three courses as follows.

Course Details
courseId = MA-SHKU
courseName = Shaolin Kung Fu
courseDuration = 100

Course Details
courseId = MA-WUSH
courseName = Wushu
courseDuration = 80

Course Details
courseId = HO-INVE
courseName = Homicide Investigation
courseDuration = 60

12. Assign Lecturers for the courses as follows.

    Assign Lecturer 1 for the Course 1
    Assign Lecturer 1 for the Course 2
    Assign Lecturer 2 for the Course 3

    Hint: You may write a similar code as follows by calling the update method you wrote in CourseService.java file
    course1.setConductedLecturer(lecturer1);
    main.courseService.updateCourse(course1);

13. Enroll students for the courses as follows.

    Assign Student 1 for the Course 1
    Assign Student 2 for the Course 1
    Assign Student 2 for the Course 2
    Assign Student 3 for the Course 3
    Assign Student 4 for the Course 3
    Assign Student 4 for the Course 1

    Hint: You may write a similar code as follows by calling the update method you wrote in StudentService.java class.
    student2.getListOfCourses().add(course1);
    student2.getListOfCourses().add(course1);
    main.studentService.updateStudent(student2);

14. List the students who follow course id MA-SHKU (Shaolin Kung Fu). You may call the business method you wrote in StudentService.java class. You should get a result like follows in your console output.
    *** Students who are following course MA-SHKU***
    *** Start ***
    NIC No = 927656351V | Registration No = DIT-13-M1-0056 | Full Name = Jackie Chan
    NIC No = 936724590V | Registration No = DIT-14-M1-0023 | Full Name = Jet Li
    NIC No = 903476126V | Registration No = DIT-11-M1-0009 | Full Name = Patrick Jane
    *** End ***