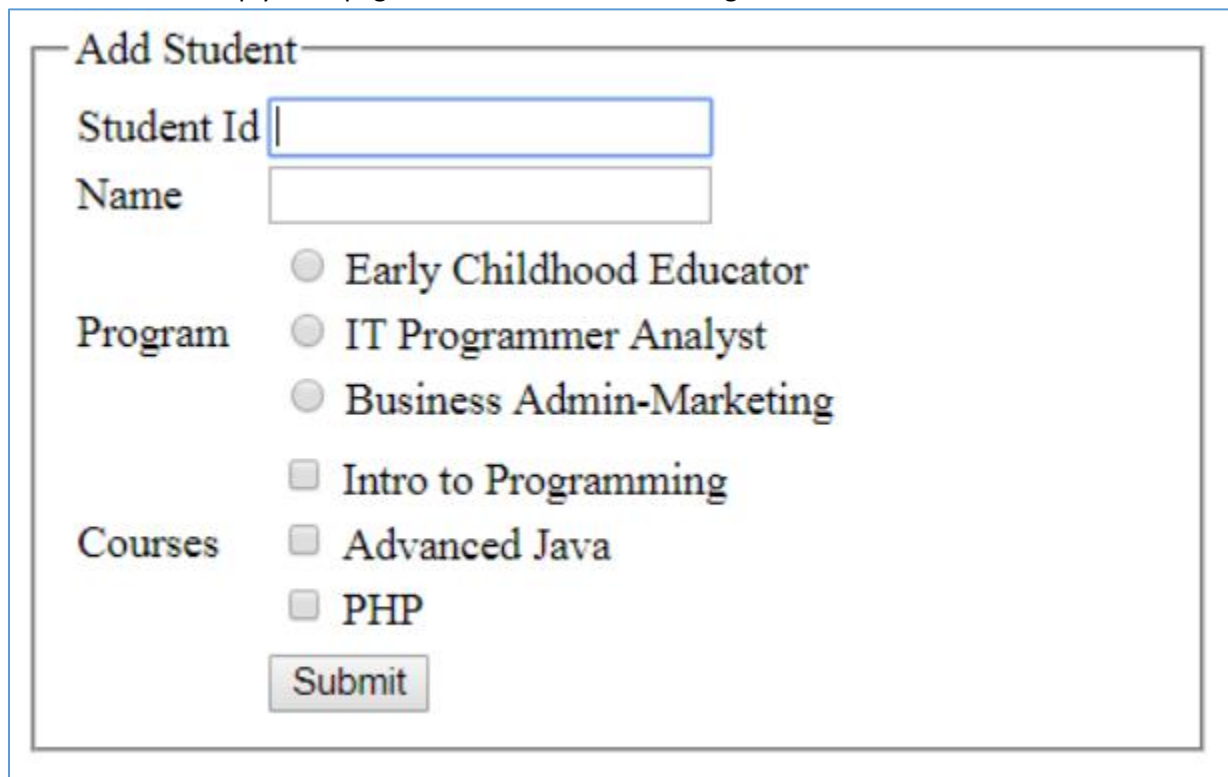Please read Chapter 10 of the book "Beginning Java EE 7" By Antonio Goncalves ©2013 on SkillsNB:

https://gnb.skillport.com/skillportfe/main.action#summary/BOOKS/RW$384622:_ss_book:55989

For this assignment, you will use the "NBCC" MySql database. I have included the dump file on BrightSpace for you. It only contains 1 small table "students". Take a look at it, it's purpose should be apparent. (NOTE: The courses that the Student is enrolled in are kept in a pipe-delimited string. I realize this isn't the best database design, but the purpose of this assignment is to learn JSFs)

1. Create a new Java Web project in NetBeans for this assignment. Make sure to add the MySQL JDBC driver to your project!

2. You **MUST** do this assignment with Java Server Faces technology. Using JSPs and/or servlets will result in a reduction of marks.

3. You must create the following items:

   Index.xhtml – this is simply a JSF page that looks like the following:

DisplayStudents.xhtml – this is a JSF page that will display AFTER the index page is submitted. It must dynamically display all the students from the database in a table like this:

In addition to the 2 xhtml files listed above, you must have 3 beans. You can keep them in 1 package for simplicity:

Student – This is a managed bean (aka: backing bean) that will have the four attributes of a "Student" along with getters, setters and constructors.

StudentBackingBean – This managed bean will connect the JSF to the Data Layer. It will contain two attributes:

private Student student;
private ArrayList<String> courses;
as well as 2 methods:
public ArrayList<Student> getAllStudents()
public String AddStudent()

StudentBean – This EJB is the database layer. It will look very similar to your database layer from the last assignment. It must contain methods:

- //constructor - will initialize the DB connection
  public StudentBean()
- //creates a database connection
  public static Connection GetConnection()
- //Inserts a new record into student table
  public static boolean InsertStudent(Student s)
- public static ArrayList<Student> FetchAllStudents()

Test it all to make you can enter some student data on index.xhtml and it will get stored in the DB and then all students will display on DisplayStudent.xhtml.