

# Java Immersion Program Reinforcement Exercise

## Background

This exercise is designed to reinforce the concepts and techniques learned in the Core Session of the Java Immersion Program. This exercise will require the student to:

- Design, code and test a Java Class that will hold data from the database, an Entity class
- Design, code and test JDBC access to the database for query, insert, update and delete functionality, DAO Interface/Implementation classes
- Use good design techniques, including creating an Interface first, writing JUnit tests to validate functionality, checking security and use coding “Best Practices”
- Read and understand existing Java code in order to add functionality

## Target Audience

This exercise is targeted to the following primary audience:

- Students who have completed the Core Session of the Java Immersion Program
- Students who will have Java experience, and will be attending only the Web Technologies Session of the Java Immersion Program

The secondary audience includes:

- The Java Mentors supporting the students outlined above
- The Java Architecture team

## Exercise Project Overview

The project incorporated within the Java Immersion Program is designed as follows:

**Layer I** → User Interface, this will be the subject of the Web Technologies Session  
This exercise will not incorporate any Layer I coding.

**Layer II** → Business Logic, this contains the core entity classes  
This exercise will require new core entity class(es)

**Layer III** → Services layer, services include a User Authentication service, a Geography service, a service to access the database via JDBC.  
This exercise will require modifying the existing JDBC service or creating a new JDBC service to add functionality.

A number of Layer II and Layer III components have been designed and built as part of the Core Session of the program. This exercise will add functionality to the existing code base. The Instructor will make available his Eclipse workspace with the existing code base included. A detailed description of this “starter” workspace is included below.

### **Pre-Work for the Exercise**

The Student web site for the Java Immersion Program is <http://JRE-Training.com/JIT>. Included here are a number of documents that provided the required background for the exercise. In particular everyone should examine:

- The SettingUpYourEnvironment.pdf – this will explain the folder structure behind the Instructor workspace, if the Student attended the Core Session of the Java Immersion Program, they will have completed the environment setup. The Instructor workspace download contains these folders, so this document is background knowledge only.
- The DatabaseInstallation.pdf – this will explain how the project database is installed and how to open the SQL Scrapbook to execute SQL statements.
- The ProjectOverview.pdf – this is the Java Immersion Program project, please study this document well, as this information is required to understand and complete this exercise.

## The Assignment

The assignment for this exercise is to code the Layer II and Layer III Java code for the following Use Case:

### **Add/Update/Delete Supplier Note – Use Case**

Pre-Conditions: The Employee is signed into the system and viewing a Employee screen, with a menu option to Add/Update/Delete a Supplier Notes record

Post-Conditions: A Supplier Notes record has been added/updated/deleted

#### Steps:

1. The Employee selects “**Suppliers**” from the Employee menu
2. The System display a list of Suppliers, there will be an indication if the Supplier has existing Notes
3. The Employee selects an existing Supplier, the System displays any current Supplier Note(s), and provides and option to “**Add/Update/Delete a Supplier Note**” for the Employee to select
4. If the Employee selects to add a new Supplier Note, the System display a blanks screen to enter the data
5. If the Employee selects to update a Supplier Note, the System displays the current Supplier Note for the Employee to edit.
6. If the Employee selected to delete a Supplier Note option, the System asks for the Employee to confirm the deletion
7. If the Employee is adding or updating a Supplier Note, the System updates or adds the Supplier Note data

#### Business Rules:

- A Supplier Note must be associated with a valid Supplier.
- If a Supplier is deleted, the associated Supplier Notes must be deleted
- A Supplier may have zero or more notes
- Any Employee (Clerk/Manager) can add/updated/delete a Supplier Note

## **Deliverables**

Each Student is requested to design, code and test the following items:

- An Interface describing the SupplierNotes database (DAO) methods required by the Use Case
- The SupplierNotes entity class with constructor(s), get/set methods, any business methods, etc.
- The JDBC database access (DAO) to support the Use Case above
- JUnit/DBUnit tests for the entity class, and the JDBC implementation

The Java code should show “Best Practices” and the Student is encouraged to make use of logging techniques and security checks. The examples within the Instructor's workspace will show logging, security checks, JavaDoc comments and Ant built files.

## **Suggested Approach**

It is recommended the Student start by outlining the functionality required for the three layers (Layer I, Talk to user, Layer II Business Logic, and Layer II Access the Database), these lists will be the start of the Interface(s) requested as deliverables in this exercise.

Layer I functionality will not developed at this time, so this list will set aside for the Web Techniques session of the Immersion program.

Layer II will define the basic business rules, such as is a field allowed to be null, or what is the max length allowed in the field, these will be coded in the set methods on the entity class and tested via the JUnit tests of the entity classes. Please remember to check for illegal characters in all the data fields to detect security violations.

Layer III will define the database actions, such as add a new records, update a record, etc. Included in the Layer III functionality will be custom queries to find entities by fields other than primary key, or queries to find all. Layer III functionality will be tested via the DBUnit tests of the DAO Implementation class.

Please note any changes to existing code, and ensure tests are written for those changes.

After the interfaces are designed, the Student should write the JUnit tests and the entity classes. Once the entity classes pass the tests, the Student should write the DBUnit tests and the DAO interface/implementation class(es).

## **Sanity Check**

The target of the exercise is between 8 to 10 hours of effort, if a student gets “stuck”, ask for help. A Java Developer recognizes they do not have to know everything, they just have to ask the right questions and understand the answers.

During this exercise the Student should seek help from:

1. Their Java Mentor
2. The Java Architecture Team (JAAG group):
  - Contact Arafa Khatun <Arafa.Khatun@irs.gov>
3. The Instructor:
  - Contact Jonathan Earl <Jonathan@Earl-Family.Net>

## Database Commands

The following commands have been used to add the Supplier\_Notes table to the Gallery Database, add a Foreign Key between the Suppliers and Supplier\_Notes tables, and add a few records to the Supplier\_Notes table.

```
CREATE TABLE SUPPLIER_NOTES (  
    NOTES_ID INTEGER GENERATED BY DEFAULT  
        AS IDENTITY(START WITH 1) NOT NULL PRIMARY KEY,  
    USER VARCHAR(10) NOT NULL,  
    SUPPLIER_ID INTEGER NOT NULL,  
    NOTE_DATE DATE NOT NULL,  
    NOTE VARCHAR(255) NOT NULL  
);
```

```
ALTER TABLE SUPPLIER_NOTES ADD CONSTRAINT  
    "SYS_FK_SUPPLIER" FOREIGN KEY ("SUPPLIER_ID")  
    REFERENCES SUPPLIERS("SUPPLIER_ID");
```

```
INSERT INTO SUPPLIER_NOTES  
    VALUES(1,'PaulaE',1,'2012-02-17','BSG Arts offers next day shipping on request for no cost');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(2,'PaulaE',23,'2012-02-17','Ask for Mary, when talking to Customer Service');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(3,'MaryL',31,'2012-02-13','Slow service');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(4,'FrankN',4,'2012-02-10','Charges 10 percent for framing');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(5,'PaulaE',1,'2012-02-17','Bulk orders at 15 percent discount');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(6,'MaryL',2,'2012-02-17','Open early, and on Saturday mornings');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(7,'MaryL',3,'2012-02-13','Expedited service on request');  
INSERT INTO SUPPLIER_NOTES  
    VALUES(8,'FrankN',14,'2012-02-10',  
        'Call the LA office after Eastern closing hours for overnight service');
```