**Week 10 Day 4 Lab Coding Challenges**

# 

# Question 1

# 

# Objective: In continuation of Question 1, 2 & 3 Of Day 3 Lab Exercises

To enable the Mfg Based business to automate the insertion of **Employees** information into Database

- create Java Based back end modules to perform insert Operations from RDBMS using JDBC

**Step 1:** In the Project which is already Created as per Question 1 ,

implement the functionality insertEmployeeRecord(Employee empI) in **EmployeeService class**

**Step2:**  In the Project which is already Created as per Question 1 ,

implement the functionality insertEmployee(Employee empI) in **EmployeeDAO class**

**Step3:** In the Project which is already Created as per Question 1 ,

Implement the Menu Option for **inserting the Employee record i**n the EmployeeDataManagement Class

(The EmployeeId which was auto generated through Lab Exercise 3 functionality, is to be used as the EmployeeId for the insertion , all other fields are accepted from the User)

**Stub Code: Along with the Stub** As in Lab Question 1

**EmployeeDAO class**

public class EmployeeDAO {

Connection con;

ResultSet rs;

Statement stmt;

PreparedStatement pstmt;

Employee employee;

ArrayList <Employee> employees;

MyConnection mycon;

public ArrayList <Employee> getAllEmployees()

{

}

public Employee getEmployeeById(String empId)

{

}

public boolean deleteEmployeeById(String empId)

{

}

public boolean updateEmployee(Employee empU)

{

}

public boolean insertEmployee(Employee empI)

{

**//IMPLEMENT HERE**

}

public ArrayList <String> getEmployeeIds()

{

}

}

**EmployeeService class**

public class EmployeeService {

EmployeeDAO edao;

public EmployeeService()

{

}

public ArrayList <Employee> getEmployeeRecords()

{

}

public Employee searchEmployeeById(String empId)

{

}

public boolean insertEmployeeRecord(Employee emplI)

{

**//IMPLEMENT HERE**

}

public boolean updateEmployeeRecord(Employee emplU)

{

}

public boolean deleteEmployeeRecord(String empId)

{

}

public ArrayList <String> getAllEmployeeIds()

{

}

}

//For Menu Options create EmployeeDataManagement Class

public class EmployeeDataManagement

{

public void showMenu()

{

**//IMPLEMENT HERE**

}

}

**//Client Class**

public class ClientUI {

public static void main(String[] args) {

// TODO Auto-generated method stub

EmployeeDataManagement edm = new EmployeeDataManagement();

edm.showMenu();

}

}

**Sample Output 1:**

............. Employee Details Management Menu.........

1.View All Employees

2.Search Employee By ID

3.Insert Employee Record

4.Delete Employee ID

5.Update Employee

6.Exit

**Please enter Your Option...**

3

**Inserting Employee Record...**

**The New Id Generated is** E013

**Enter the Name of The Employee You wish to Insert**

Madhusudhan

**Enter the Address of The Employee You wish to Insert**

KRPuram

**Enter the Phone of The Employee You wish to Insert**

9827375873

**Enter the Salary of The Employee You wish to Insert**

10000

**Enter the Tax Eligibility of The Employee You wish to Insert**

12.34

Record Inserted Successfully...

**Do You Wish to Continue yes/no?**

no

We are out of Menu...

# 

# Learning Objective:

Students learn to create a Menu Based application with menu options & implement the option for Insert operations on Employees table , taking the Employee record , as an input given by the user .

# 

# 

# Question 2

# Objective: In continuation of Questions 1 2 & 3 Of Day 3 Lab Exercises

To enable the Mfg Based business to automate the Deletion of **Employees** information from Database

- create Java Based back end modules to perform delete Operations from RDBMS using JDBC

**Step 1:** In the Project which is already Created as per Question 1 ,

implement the functionality deleteEmployeeRecord(String empId)in **EmployeeService class**

**Step2:**  In the Project which is already Created as per Question 1 ,

implement the functionality deleteEmployeeById(String empId) in **EmployeeDAO class**

**Step3:** In the Project which is already Created as per Question 1 ,

Implement the Menu Option for **deleting the Employee record i**n the EmployeeDataManagement Class

**Stub Code: Along with the Stub** As in Lab Question 1

**EmployeeDAO class**

public class EmployeeDAO {

Connection con;

ResultSet rs;

Statement stmt;

PreparedStatement pstmt;

Employee employee;

ArrayList <Employee> employees;

MyConnection mycon;

public ArrayList <Employee> getAllEmployees()

{

}

public Employee getEmployeeById(String empId)

{

}

public boolean deleteEmployeeById(String empId)

{

**//IMPLEMENT HERE**

}

public boolean updateEmployee(Employee empU)

{

}

public boolean insertEmployee(Employee empI)

{

}

public ArrayList <String> getEmployeeIds()

{

}

}

**EmployeeService class**

public class EmployeeService {

EmployeeDAO edao;

public EmployeeService()

{

}

public ArrayList <Employee> getEmployeeRecords()

{

}

public Employee searchEmployeeById(String empId)

{

}

public boolean insertEmployeeRecord(Employee emplI)

{

}

public boolean updateEmployeeRecord(Employee emplU)

{

}

public boolean deleteEmployeeRecord(String empId)

{

**//IMPLEMENT HERE**

}

public ArrayList <String> getAllEmployeeIds()

{

}

}

//For Menu Options create EmployeeDataManagement Class

public class EmployeeDataManagement

{

public void showMenu()

{

**//IMPLEMENT HERE for Delete option**

}

}

**//Client Class**

public class ClientUI {

public static void main(String[] args) {

// TODO Auto-generated method stub

EmployeeDataManagement edm = new EmployeeDataManagement();

edm.showMenu();

}

}

**Sample Output 1:**

............. Employee Details Management Menu.........

1.View All Employees

2.Search Employee By ID

3.Insert Employee Record

4.Delete Employee ID

5.Update Employee

6.Exit

Please enter Your Option...

4

Deleting Employee Record by Id

You Are about to Delete a Record...

The Employee Ids Are...

E001

E002

Please enter the ID of the Employee whose Record you wish to Delete...

E001

Deleted The Record of E001 Successfully...

Do You Wish to Continue yes/no?

no

# 

# Learning Objective:

Students learn to create a Menu Based application with menu options & implement the option for Delete operations on Employees table ,based on the Employee Id , as an input given by the user .

# 

# Question 3

# Objective: In continuation of Questions 1 2 & 3 Of Day 3 Lab Exercises

To enable the Mfg Based business to automate the Updation of **Employees** information in the Database

- create Java Based back end modules to perform update Operations from RDBMS using JDBC

**Step 1:** In the Project which is already Created as per Question 1 ,

implement the functionality updateEmployeeRecord(Employee emplU) in **EmployeeService class**

**Step2:**  In the Project which is already Created as per Question 1 ,

implement the functionality updateEmployee(Employee empU) in **EmployeeDAO class**

**Step3:** In the Project which is already Created as per Question 1 ,

Implement the Menu Option for **updating the Employee record i**n the EmployeeDataManagement Class

[In a web application, when we wish to update a record, all details are shown in a form and we are allowed to update any field except primary key, but in here, we show all the existing records and allow the user to enter the Id of the record , the user wishes to update and accept all field values except primary key for that record and update the same in the back end.

]

**Stub Code: Along with the Stub** As in Lab Question 1

**EmployeeDAO class**

public class EmployeeDAO {

Connection con;

ResultSet rs;

Statement stmt;

PreparedStatement pstmt;

Employee employee;

ArrayList <Employee> employees;

MyConnection mycon;

public ArrayList <Employee> getAllEmployees()

{

}

public Employee getEmployeeById(String empId)

{

}

public boolean deleteEmployeeById(String empId)

{

}

public boolean updateEmployee(Employee empU)

{

**//IMPLEMENT HERE**

}

public boolean insertEmployee(Employee empI)

{

}

public ArrayList <String> getEmployeeIds()

{

}

}

**EmployeeService class**

public class EmployeeService {

EmployeeDAO edao;

public EmployeeService()

{

}

public ArrayList <Employee> getEmployeeRecords()

{

}

public Employee searchEmployeeById(String empId)

{

}

public boolean insertEmployeeRecord(Employee emplI)

{

}

public boolean updateEmployeeRecord(Employee emplU)

{

**//IMPLEMENT HERE**

}

public boolean deleteEmployeeRecord(String empId)

{

}

public ArrayList <String> getAllEmployeeIds()

{

}

}

//For Menu Options create EmployeeDataManagement Class

public class EmployeeDataManagement

{

public void showMenu()

{

**//IMPLEMENT HERE option for Update**

}

}

**//Client Class**

public class ClientUI {

public static void main(String[] args) {

// TODO Auto-generated method stub

EmployeeDataManagement edm = new EmployeeDataManagement();

edm.showMenu();

}

}

**Sample Output 1:**

............. Employee Details Management Menu.........

1.View All Employees

2.Search Employee By ID

3.Insert Employee Record

4.Delete Employee ID

5.Update Employee

6.Exit

**Please enter Your Option...**

5

Updating Employee Record...

**Records Existing are....**

Employee [employeeId=E001, employeeName=SreeHarsha, employeeAddress=RTNagar, employeePhone=9838388383, employeeSalary=10000, employeeTax=10.3]

Employee [employeeId=E002, employeeName=SumanthKumar, employeeAddress=Indiranagar, employeePhone=9786388383, employeeSalary=12000, employeeTax=12.3]

Employee [employeeId=E003, employeeName=KumareshGupta, employeeAddress=Malleswaram, employeePhone=9876543210, employeeSalary=30000, employeeTax=12.34]

Employee [employeeId=E004, employeeName=Muralidhar, employeeAddress=RTNagar, employeePhone=9812345687, employeeSalary=20000, employeeTax=13.34]

Employee [employeeId=E006, employeeName=RakeshKumar, employeeAddress=Koramangala, employeePhone=9829929292, employeeSalary=12000, employeeTax=12.34]

**Enter the Id of the Employee Whose Record you wish to Update....**

E001

**Enter the New Name of The Employee You wish to Update**

SreeHarsha

**Enter the New Address of The Employee You wish to Update**

RSNagar

**Enter the New Phone of The Employee You wish to Update**

9828282828

**Enter the New Salary of The Employee You wish to Update**

14000

**Enter the New Tax Eligibility of The Employee You wish to Update**

14.3

**Updated Record for Employee with Id E001**

**Do You Wish to Continue yes/no?**

yes

............. Employee Details Management Menu.........

1.View All Employees

2.Search Employee By ID

3.Insert Employee Record

4.Delete Employee ID

5.Update Employee

6.Exit

**Please enter Your Option...**

2

**Searching Employee By Id...**

The Employee Ids Are...

E001

E002

E003

E004

E006

**Please Enter the Employee Id whose record you wish to see...**

E001

The Employee is Employee [employeeId=E001, employeeName=**SreeHarsha,** employeeAddress=**RSNagar,** employeePhone=**9828282828,** employeeSalary=**14000**, **employeeTax=14.3**]

Do You Wish to Continue yes/no?

no

We are out of Menu...

# 

# Learning Objective:

Students learn to create a Menu Based application with menu options & implement the option for Update operations on Employees table , taking the modified Employee record , as an input given by the user .