Database Application Development Assignment (25%)

Submission: Your submission will be a single text-based `.java` file including your Java program for the Database Application assignment. `AS_Group##.java` Your submission needs to be commented.

Milestone 1 (10%)

Objective: In this assignment, you will create a simple HR application using the Java programming language and an Oracle database. This assignment helps students learn a basic understanding of application development using Java and an Oracle database.

Instructions: In this assignment, we use the same database that you use for the labs. Note: For each query in your assignment, make sure you handle the errors and display the proper message including the error code and the error message.

```
try {
    ...
} catch (SQLException sqlExcp) {
    System.out.println(sqlExcp.getErrorCode() + ": " + sqlExcp.getMessage());
}
```

Connecting to an Oracle database from a Java Program

In your `main` method, create a connection to your database.

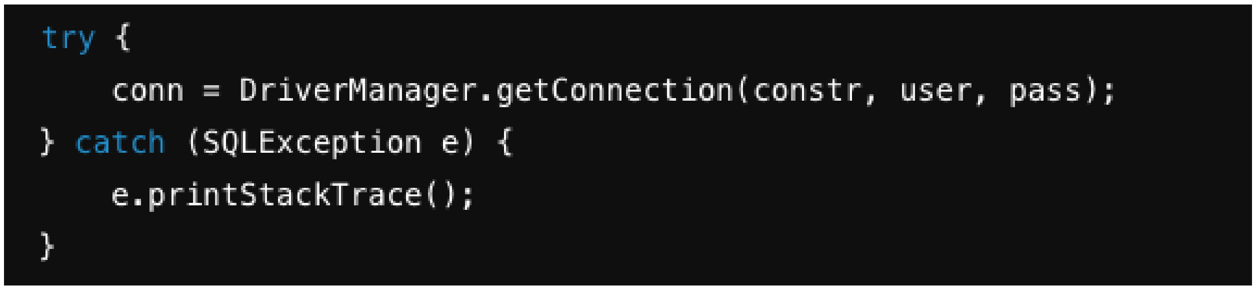
First, declare the connection variables.

```
Connection conn = null;
```

Define and initialize the variables to store the username, password, and host address.

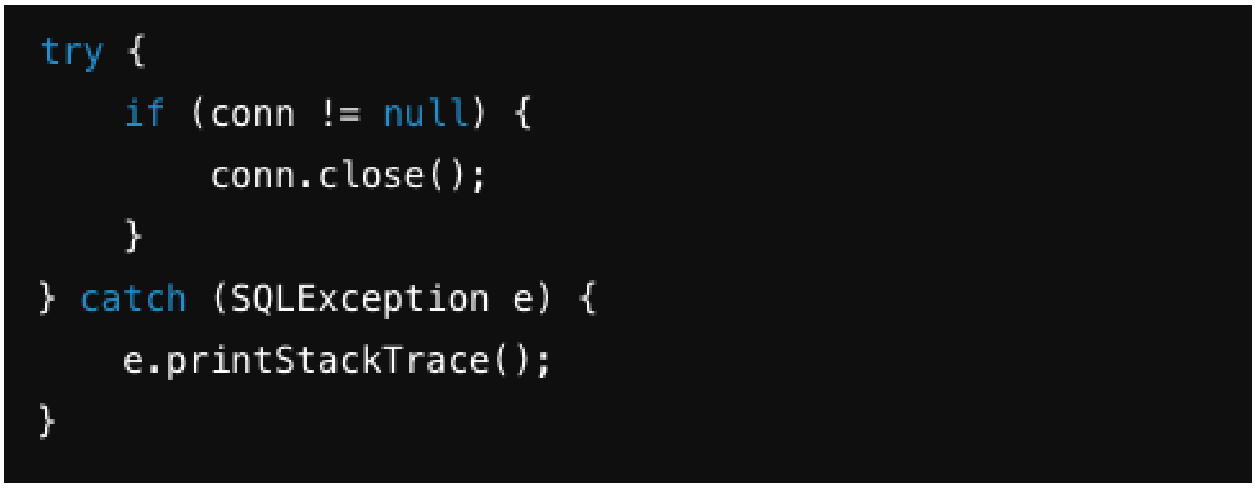
```
String user = "username";
String pass = "password";
String constr = "jdbc:oracle:thin:@myoracle12c.senecacollege.ca:1521:oracle12c";
```

Use the same Oracle username and password that you use for your labs and assignments. Create the connection and make sure you handle any errors that may be thrown as your program is executed.



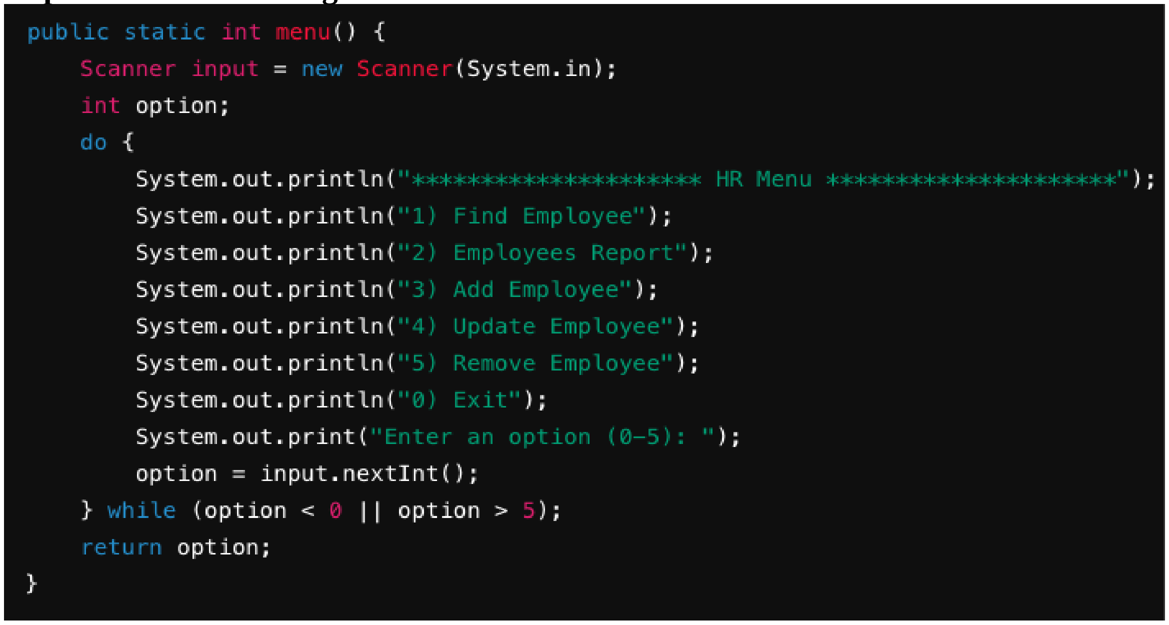
```
try {
    conn = DriverManager.getConnection(constr, user, pass);
} catch (SQLException e) {
    e.printStackTrace();
}
```

Remember to close the connection when your program terminates.



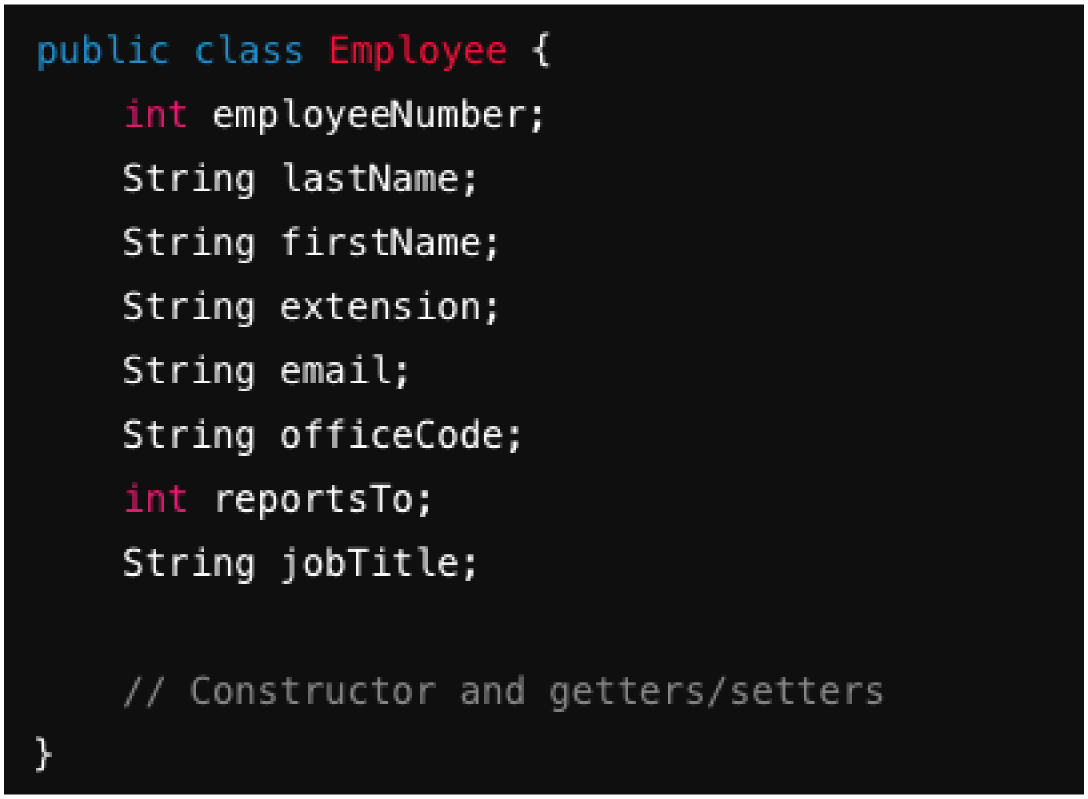
```
try {
    if (conn != null) {
        conn.close();
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

Implement the following functions:



```
public static int menu() {
    Scanner input = new Scanner(System.in);
    int option;
    do {
        System.out.println("********************* HR Menu **********************");
        System.out.println("1) Find Employee");
        System.out.println("2) Employees Report");
        System.out.println("3) Add Employee");
        System.out.println("4) Update Employee");
        System.out.println("5) Remove Employee");
        System.out.println("0) Exit");
        System.out.print("Enter an option (0-5): ");
        option = input.nextInt();
    } while (option < 0 || option > 5);
    return option;
}
```

Employee Class:



```
public class Employee {
    int employeeNumber;
    String lastName;
    String firstName;
    String extension;
    String email;
    String officeCode;
    int reportsTo;
    String jobTitle;

    // Constructor and getters/setters
}
```

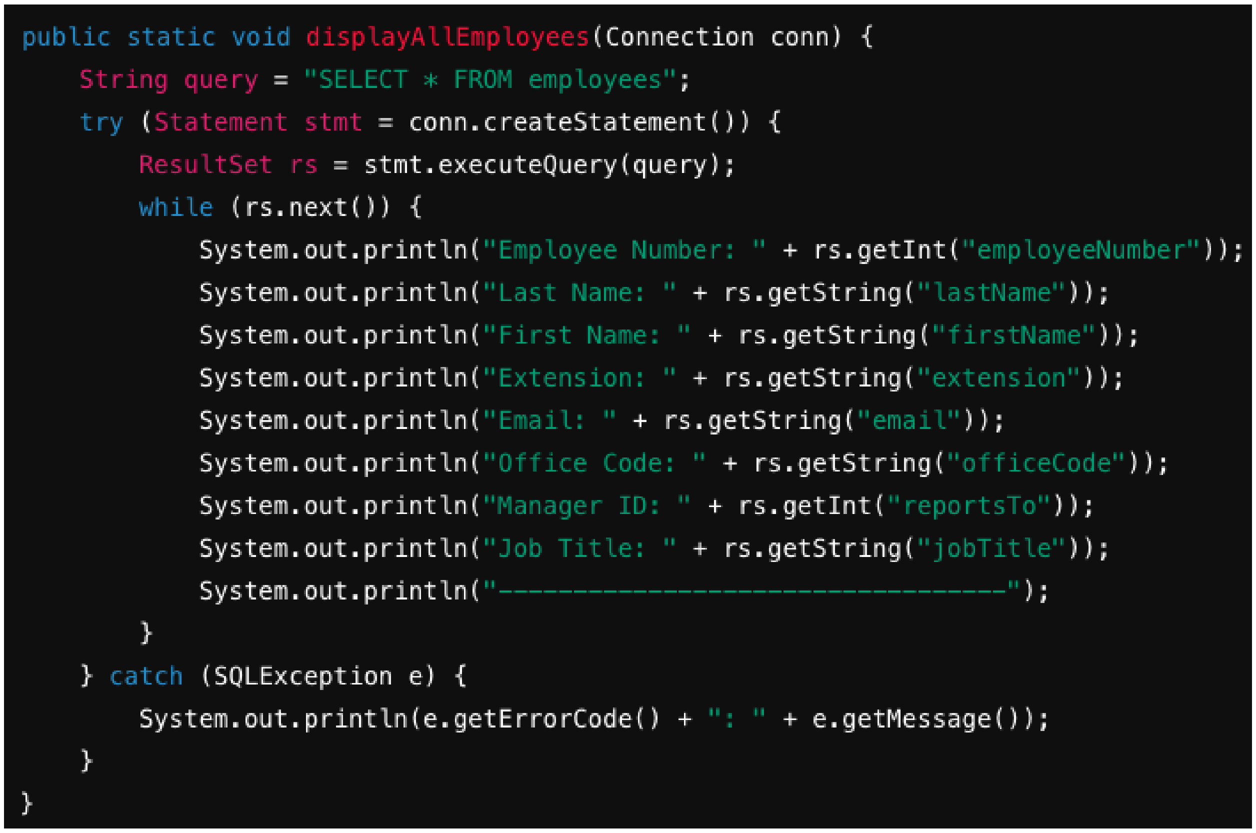
Find Employee Function:

```
public static int findEmployee(Connection conn, int employeeNumber, Employee emp) {
    String query = "SELECT * FROM employees WHERE employeeNumber = ?";
    try (PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setInt(1, employeeNumber);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            emp.employeeNumber = rs.getInt("employeeNumber");
            emp.lastName = rs.getString("lastName");
            emp.firstName = rs.getString("firstName");
            emp.extension = rs.getString("extension");
            emp.email = rs.getString("email");
            emp.officeCode = rs.getString("officeCode");
            emp.reportsTo = rs.getInt("reportsTo");
            emp.jobTitle = rs.getString("jobTitle");
            return 1;
        } else {
            return 0;
        }
    } catch (SQLException e) {
        System.out.println(e.getErrorCode() + ": " + e.getMessage());
        return 0;
    }
}
```

Display Employee Function:

```
public static void displayEmployee(Connection conn, Employee emp) {
    System.out.println("--------------- Employee Information --------------");
    System.out.println("Employee Number: " + emp.employeeNumber);
    System.out.println("Last Name: " + emp.lastName);
    System.out.println("First Name: " + emp.firstName);
    System.out.println("Extension: " + emp.extension);
    System.out.println("Email: " + emp.email);
    System.out.println("Office Code: " + emp.officeCode);
    System.out.println("Manager ID: " + emp.reportsTo);
    System.out.println("Job Title: " + emp.jobTitle);
}
```

Display All Employees Function:



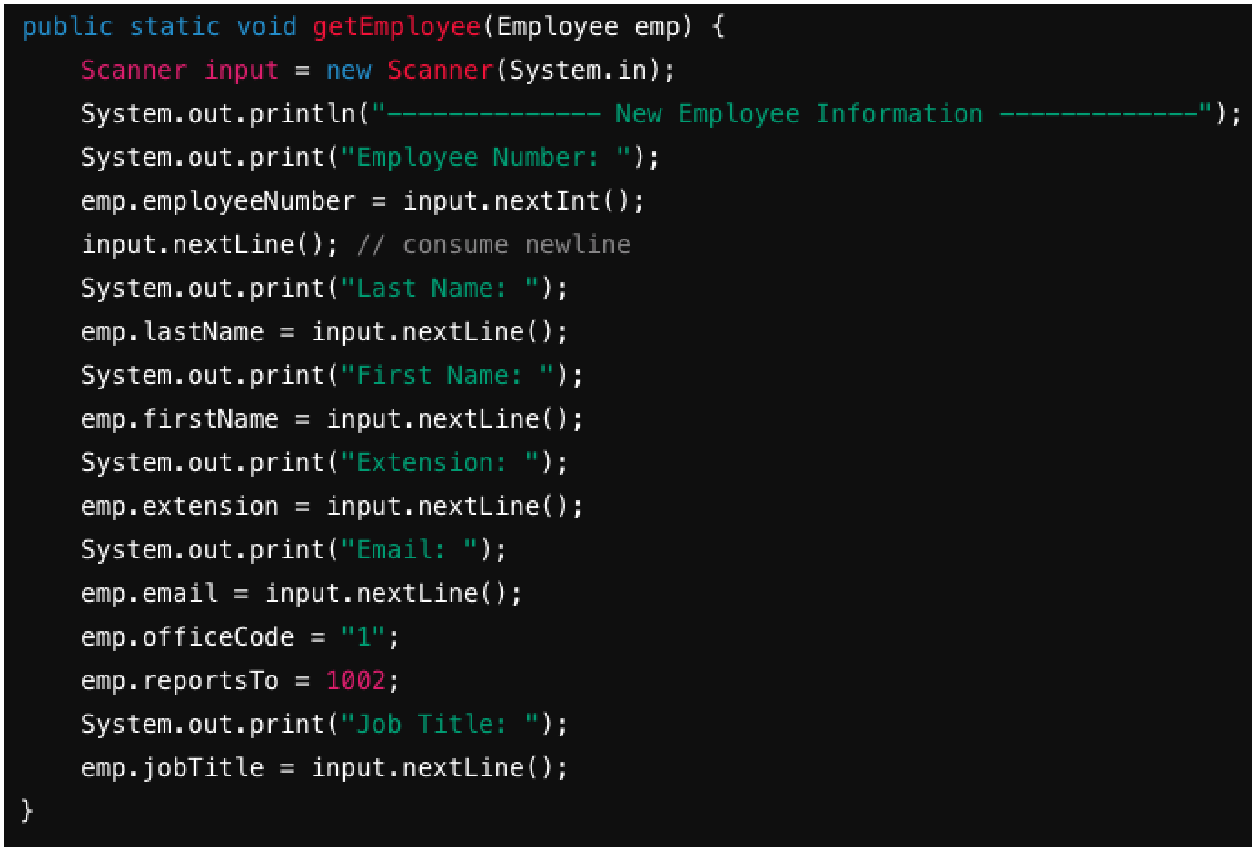
```java
public static void displayAllEmployees(Connection conn) {
    String query = "SELECT * FROM employees";
    try (Statement stmt = conn.createStatement()) {
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            System.out.println("Employee Number: " + rs.getInt("employeeNumber"));
            System.out.println("Last Name: " + rs.getString("lastName"));
            System.out.println("First Name: " + rs.getString("firstName"));
            System.out.println("Extension: " + rs.getString("extension"));
            System.out.println("Email: " + rs.getString("email"));
            System.out.println("Office Code: " + rs.getString("officeCode"));
            System.out.println("Manager ID: " + rs.getInt("reportsTo"));
            System.out.println("Job Title: " + rs.getString("jobTitle"));
            System.out.println("------------------------------------");
        }
    } catch (SQLException e) {
        System.out.println(e.getErrorCode() + ": " + e.getMessage());
    }
}
```

Milestone 2 (15%)

Objective: In this assignment, you will complete the HR application from the first part to insert, update, and delete employee information.

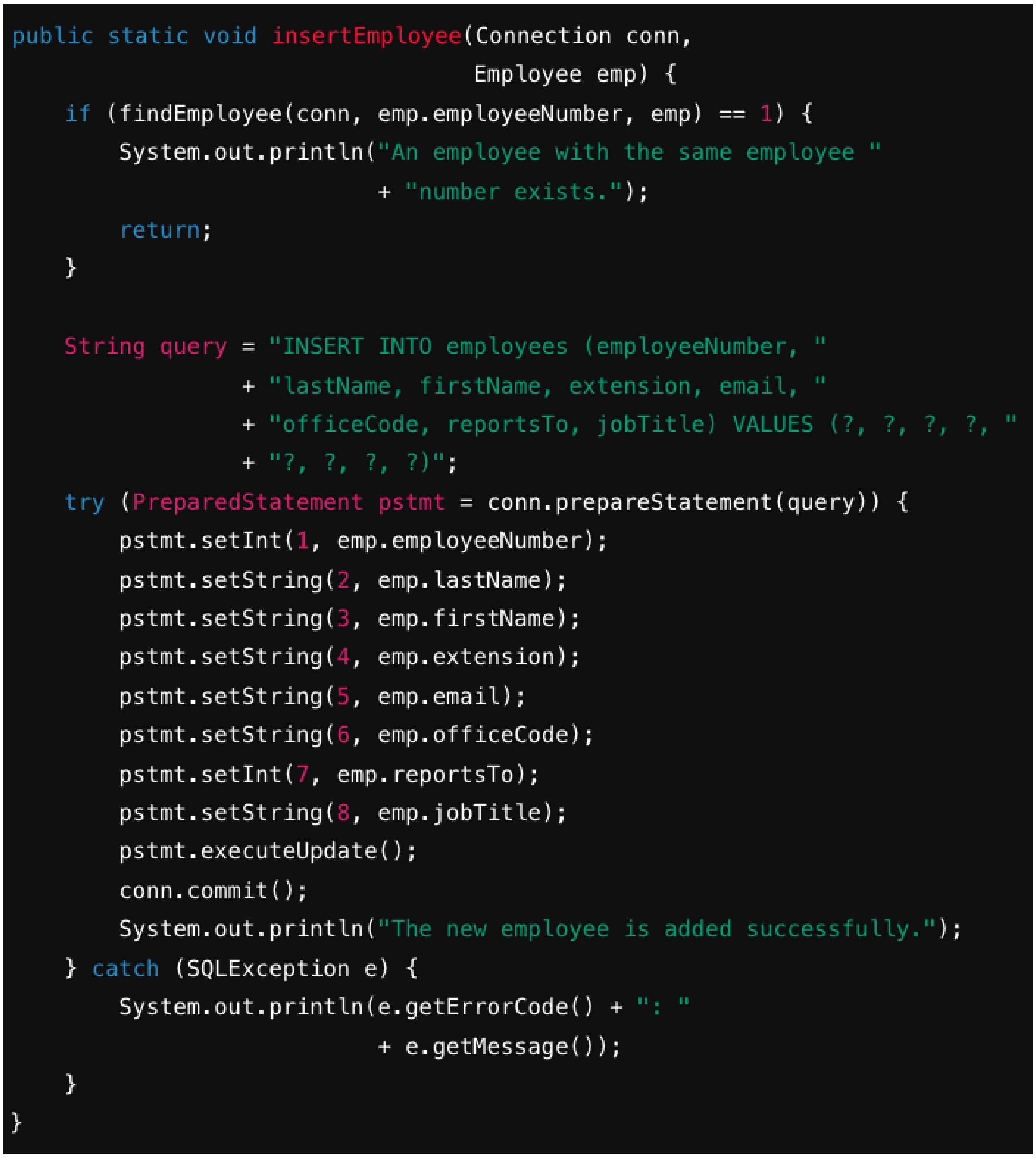
Implement the following functions:

Get Employee Function:



```
public static void getEmployee(Employee emp) {
    Scanner input = new Scanner(System.in);
    System.out.println("-------------- New Employee Information -------------");
    System.out.print("Employee Number: ");
    emp.employeeNumber = input.nextInt();
    input.nextLine(); // consume newline
    System.out.print("Last Name: ");
    emp.lastName = input.nextLine();
    System.out.print("First Name: ");
    emp.firstName = input.nextLine();
    System.out.print("Extension: ");
    emp.extension = input.nextLine();
    System.out.print("Email: ");
    emp.email = input.nextLine();
    emp.officeCode = "1";
    emp.reportsTo = 1002;
    System.out.print("Job Title: ");
    emp.jobTitle = input.nextLine();
}
```

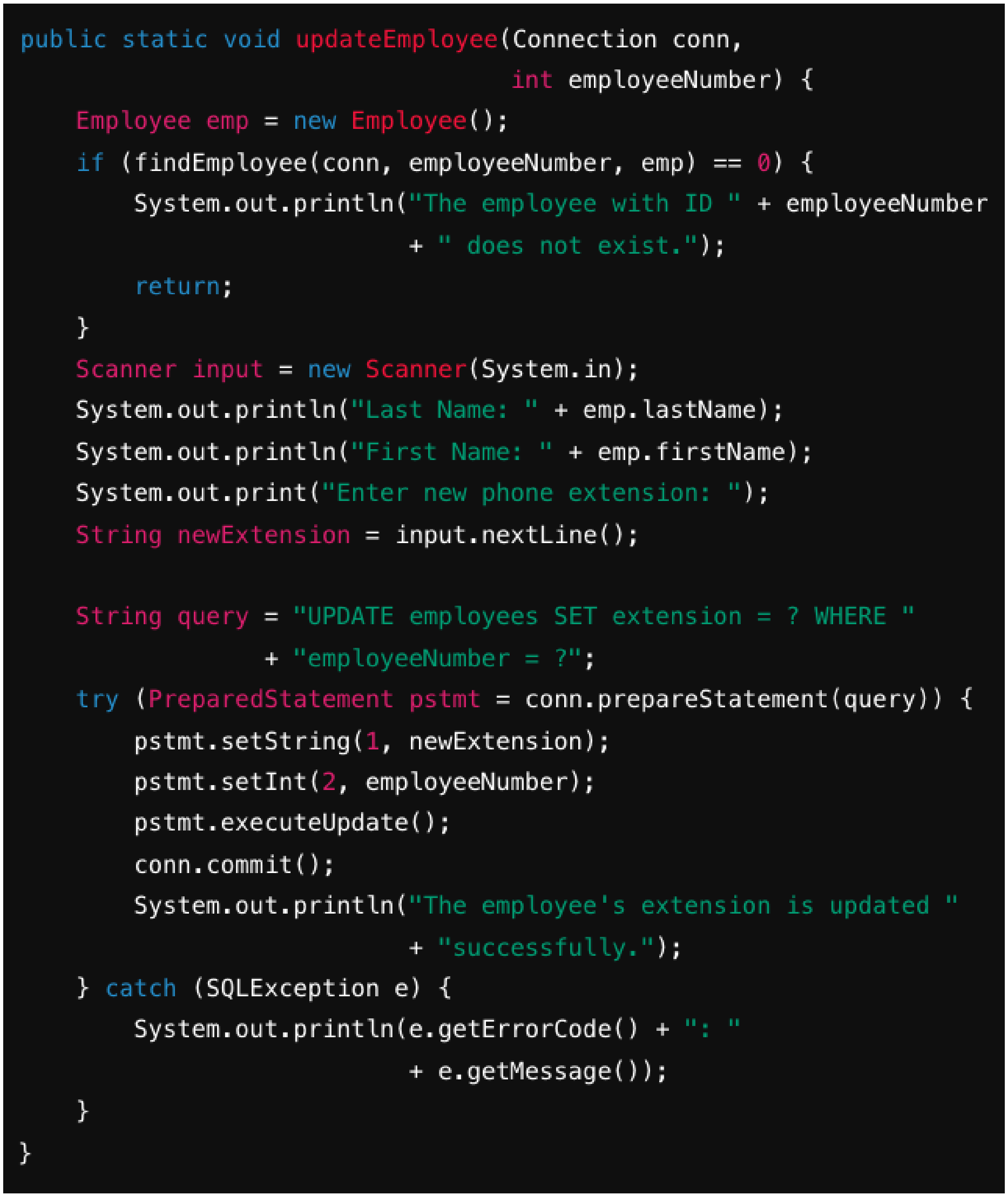
Insert Employee Function:



```java
public static void insertEmployee(Connection conn,
                                  Employee emp) {
    if (findEmployee(conn, emp.employeeNumber, emp) == 1) {
        System.out.println("An employee with the same employee "
                           + "number exists.");
        return;
    }

    String query = "INSERT INTO employees (employeeNumber, "
                 + "lastName, firstName, extension, email, "
                 + "officeCode, reportsTo, jobTitle) VALUES (?, ?, ?, ?, "
                 + "?, ?, ?, ?)";
    try (PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setInt(1, emp.employeeNumber);
        pstmt.setString(2, emp.lastName);
        pstmt.setString(3, emp.firstName);
        pstmt.setString(4, emp.extension);
        pstmt.setString(5, emp.email);
        pstmt.setString(6, emp.officeCode);
        pstmt.setInt(7, emp.reportsTo);
        pstmt.setString(8, emp.jobTitle);
        pstmt.executeUpdate();
        conn.commit();
        System.out.println("The new employee is added successfully.");
    } catch (SQLException e) {
        System.out.println(e.getErrorCode() + ": "
                           + e.getMessage());
    }
}
```

Update Employee Function:



```java
public static void updateEmployee(Connection conn,
                                  int employeeNumber) {
    Employee emp = new Employee();
    if (findEmployee(conn, employeeNumber, emp) == 0) {
        System.out.println("The employee with ID " + employeeNumber
                           + " does not exist.");
        return;
    }
    Scanner input = new Scanner(System.in);
    System.out.println("Last Name: " + emp.lastName);
    System.out.println("First Name: " + emp.firstName);
    System.out.print("Enter new phone extension: ");
    String newExtension = input.nextLine();

    String query = "UPDATE employees SET extension = ? WHERE "
            + "employeeNumber = ?";
    try (PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setString(1, newExtension);
        pstmt.setInt(2, employeeNumber);
        pstmt.executeUpdate();
        conn.commit();
        System.out.println("The employee's extension is updated "
                           + "successfully.");
    } catch (SQLException e) {
        System.out.println(e.getErrorCode() + ": "
                           + e.getMessage());
    }
}
```

Delete Employee Function:

```java
public static void deleteEmployee(Connection conn,
                                  int employeeNumber) {
    Employee emp = new Employee();
    if (findEmployee(conn, employeeNumber, emp) == 0) {
        System.out.println("The employee with ID " + employeeNumber
                           + " does not exist.");
        return;
    }

    String query = "DELETE FROM employees WHERE employeeNumber = ?";
    try (PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setInt(1, employeeNumber);
        pstmt.executeUpdate();
        conn.commit();
        System.out.println("The employee with ID " + employeeNumber
                           + " is deleted successfully.");
    } catch (SQLException e) {
        System.out.println(e.getErrorCode() + ": "
                           + e.getMessage());
    }
}
```