```java
package class19;

import java.sql.Connection;

public class OJDBCDemo {

    public static void main(String[] args) {
        try {
            // load the Oracle driver into the JDBC driver manager
            // "oracle.jdbc.OracleDriver" is the name that must be used
            // there is an older reference oracle.jdbc.driver.OracleDriver that is now deprecated
            Class.forName("oracle.jdbc.OracleDriver");
        } // end try
        catch (ClassNotFoundException ex) {
            System.out.println("The Oracle driver is not available.");
            System.out.println(ex.getMessage());
        } // end catch
        // conn must be defined here so that is "in scope" for the finally block
        // this allows us to defer .close() all the way until then
        Connection conn = null;
        // JDBC database operations require exception handling (throw SQLException)
        try {
            // create a connection to the local Oracle database using the jdbc
            // API and the Oracle driver/wrapper
            // create conn once and pass it to all following method invocations as an argument
            // this allows us to define the endpoint one time for the entire application and then
            // only have to change it in one place if necessary

            // the following is a connection string for UCA's Oracle server
            conn = DriverManager.getConnection("jdbc:oracle:thin:@161.31.3.67:1521:xe","MIS3339","MIS3339");
            // explicitly set transactional processing to be off
            conn.setAutoCommit(true);
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM BOOKS");
            // get the table's metadata
            ResultSetMetaData md = rs.getMetaData();
            // databases start counting at 1, not 0 like arrays
            for (int i = 1; i <= md.getColumnCount(); i++) {
                System.out.print(String.format("%1$"+20+ "s", md.getColumnName(i).toUpperCase()));
            } // end for
            // insert a CR/LF after all the column names are printed out
            System.out.print("\n");
            // iterate through the rowset
            while (rs.next()) {
                // iterate through the columns in each row
                for (int i = 1; i <= md.getColumnCount(); i++) {
                    System.out.print(String.format("%1$"+20+ "s", rs.getObject(i)).toUpperCase());
                } // end for
                System.out.print("\n");
            }// end while
            System.out.println(""); // enter a blank line after outputting the db contents
            stmt.close(); // close the Statement
```

```
58              rs.close(); // close the ResultSet
59              conn.close(); // close the Connection
60          } catch (SQLException e) {
61              System.out.println(e.getMessage());
62              e.printStackTrace();
63          } // end catch
64      } // end main
65  } // end OJDBCDemo
```