```
/**
*   Class Name          : AITPReader.class
*   Version             : Release 1 Build 2
*   Date of Change      : 2000 MAR 27
*   Author              : Don Baldwin
*   Copyright           : 2000 ASR Strategic Resources
*   Description         : A database handler that reads the Hello World string
*                         from the database, just to show how it works.
*
*   Version             Date            Author
*   Release 01          2000 MAR 27     Don Baldwin
*                                               Tested and accepted
* Build 02              2000 MAR 27     Don Baldwin
*                                               Modified the class for the AITP
*                                               presentation
*   Build 01            2000 MAR 26     Ulf Hedlund
*                                               Wrote the readGreeting() method.
*/
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class AITPReader
    {

    /**
    * A reference to the AITPReader itself. Using this to make sure only
    * one instance is ever created.
    */
    private static AITPReader objThis = null;

    /**
    * The constructor needs to be hidden so we can make sure no one can
    * create an instance without going through getAITPReader().
    */
    private AITPReader()
        {
        }

    public static void main(String[] arg)
        {
        AITPReader obj = AITPReader.getAITPReader();
        System.out.println(obj.readGreeting("swedish"));
        System.out.println(obj.readGreeting("english"));
        }

    /**
    * getAITPReader is the only way to create an instance of this class.
    * If no objThis has been created yet, we create one and return it.
    * If one already exists, we simply return the existing object.
    */
    public static final AITPReader getAITPReader()
        {

        if(objThis == null)
            {
            objThis = new AITPReader();
            }

        return objThis;
        }

    public void log(String aStrMessage)
```

```
    {
    String        strFinalMsg;

    strFinalMsg= "AITPReader, " + aStrMessage;

    System.out.println(strFinalMsg);
    }

/**
* This method setups up the connection to the database.
* Instead of dealing with possible exceptions here, we
* allow the caller of the method to do it instead for
* greater flexibility.
*/
private Connection initConnection()
    throws SQLException, ClassNotFoundException
    {

    Connection sqlConnection = null;


    // Creating a JDBC connection to the standard JdbcOdbcDriver.
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");


    // Setting up the connection to the database
    sqlConnection = DriverManager.getConnection(
                    "jdbc:odbc:helloworld");

    return sqlConnection;
    }

/**
* The readGreeting method reads the proper hello world
* string from the database.
*/
protected String readGreeting(String aStrLanguage)
    {
    Connection     sqlConnection;
    Statement   sqlStatement;
    ResultSet   sqlResult;
    String            strSQLQuery;
    String            strGreeting      = null;

    try
        {
        // Get the connection to our hello world database.
        sqlConnection = initConnection();

        // Create the SQLstatement pointing to the connection.
        sqlStatement   = sqlConnection.createStatement();

        strSQLQuery       = "SELECT * FROM HELLOWORLD WHERE LANGUAGE LIKE '" +
                        aStrLanguage + "'";

        // Access the database and acquire information based on the strSQLquery.
        sqlResult        = sqlStatement.executeQuery(strSQLQuery);

        // Jump to first row in the result set.
        sqlResult.next();
        strGreeting      = sqlResult.getString("PHRASE");

        // Close the resultset, statement, and connection objects after we are done.
        // This is especially important with a MS Access database since it tends to
        // crash frequently if we don't do this.
```

```
        sqlResult.close();
        sqlStatement.close();
        sqlConnection.close();
        }
    catch(ClassNotFoundException excCNF)
        {
        log("readGreeting(): Problems finding the database driver - " + excCNF);
        }
    catch(SQLException excSQL)
        {
        log("readGreeting(): Exception doing the SQL query - " + excSQL);
        }

    return strGreeting;
    }

} // END Class AITPReader.java
```