**CSCE 740 - Code Smells Activity**

**Name(s):**

The following code contains several “bad smells” - signs that the code is not well-designed and needs to be refactored.

**1: Identify at least three code smells in this code. Explain why each is a problem.**

**2: Explain how you would fix each code smell.**

**-----------------------------------------------Order.java-----------------------------------------------**

package com.nq.util;  
  
import java.io.\*;  
import java.util.\*;  
import java.lang.\*;  
import java.sql.\*;  
  
public class Order {  
  
 private lineitemlist lineItemList;  
  
 public Order(lineitemlist lis) {  
 lineItemList = lis;  
 }  
  
 public boolean equals(Object aThat) {  
 if ( this == aThat ) return true;  
 if ( !(aThat instanceof Order) ) return false;  
 Order that = (Order)aThat;  
 return this.lineItemList.equals(that.lineItemList);  
 }  
  
 // writes this order object to the specified print writer  
 public void writeOrder(Order order, PrintWriter pw) {  
 // get a vector of line items  
 Vector lineItems = order.getLineItemList().getLineItems();  
  
 // ------------------------------------------------------  
 // calculate total  
 // ------------------------------------------------------  
 // create an iterator for the vector  
 Iterator iter = lineItems.iterator();  
 lineItem item;  
 // set total to zero  
 int total = 0;  
 while (iter.hasNext()) {  
 item = (lineItem)iter.next();  
  
 // calculate total for line item  
 int unitPrice = item.getUnitPrice();  
 int qty = item.getQuantity();  
 int lineitemtotal = unitPrice \* qty;  
  
 total += lineitemtotal;  
 }  
 // ------------------------------------------------------  
 // END calculate total  
 // ------------------------------------------------------  
  
 // ------------------------------------------------------  
 // write order  
 // ------------------------------------------------------  
 // create an iterator for the vector  
 iter = lineItems.iterator();  
 while (iter.hasNext()) {  
 item = (lineItem)iter.next();  
  
 // calculate total for line item  
 int unitPrice = item.getUnitPrice();  
 int qty = item.getQuantity();  
 int productID = item.getProductID();  
 int imageID = item.getImageId();  
 int lineitemtotal = unitPrice \* qty;  
  
 pw.println("Begin Line Item");  
 pw.println("Product = " + productID);  
 pw.println("Image = " + imageID);  
 pw.println("Quantity = " + qty);  
 pw.println("Total = " + lineitemtotal);  
 pw.println("End Line Item");  
 }  
 pw.println("Order total = " + total);  
 }  
  
 public int getTotal() {  
 // get a vector of line items  
 Vector lineItems = lineItemList.getLineItems();  
 // create an iterator for the vector  
 Iterator iter = lineItems.iterator();  
 lineItem item;  
 // set total to zero  
 int total = 0;  
 while (iter.hasNext()) {  
 item = (lineItem)iterator.next();  
  
 // calculate total for line item  
 int unitPrice = item.getUnitPrice();  
 int qty = item.getQuantity();  
 int lineitemtotal = unitPrice \* qty;  
  
 total += lineitemtotal;  
 }  
 return total;  
 }  
  
 /\*\* This method saves the order to the database \*/  
 public void saveOrder() throws SQLException  
 {  
 //create connection  
 Connection conn = null;  
  
 java.sql.Date date = new java.sql.Date((new java.util.Date())  
 .getTime());  
 PreparedStatement orderStatement = null;  
 PreparedStatement getStatement = null;  
 String sql = null;  
 sql = new StringBuffer().append("INSERT INTO T\_ORDER " )  
 .append("(AUTHORIZATION\_CODE, " )  
 .append("SHIPMETHOD\_ID, USER\_ID, ADDRESS\_ID) " )  
 .append("VALUES ( ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,  
 ?)" ).toString();  
 conn = setConnection();  
 orderStatement = conn.prepareStatement(sql);  
 //set all parameters  
 ...  
 //execute statement  
 orderStatement.executeUpdate();  
 }  
}

**---------------------------------------------------lineitemlist.java---------------------------------------------------**

package com.nq.util;  
  
import java.io.\*;  
import java.util.\*;  
import java.lang.\*;  
  
class lineitemlist {  
 private Vector LIList;  
  
 public void setLineItems(Vector lineItems) {  
 LIList = lineItems;  
 }  
  
 Vector getLineItems() {  
 return LIList;  
 }  
}

------------------------------------------------------**lineItem.java----------------------------------------------------**

package com.nq.util;  
  
import java.io.\*;  
import java.util.\*;  
import java.lang.\*;  
  
class lineItem {  
 protected int productId;  
 private int ImageID;  
 private int qty;  
 private int Unitprice;  
  
 public lineItem(int prodID, int ImageID, int inQty) {  
 productId = prodID;  
 this.ImageID = ImageID;  
 qty = inQty;  
 }  
  
 public void setLineItems(Vector lineItems) {  
 LineItems = lineItems;  
 }  
  
 Vector getLineItems() {  
 return LineItems;  
 }  
  
 int getProductID() {  
 return productId;  
 }  
  
 int getImageID() {  
 return imageID;  
 }  
  
 int getQuantity() {  
 return qty;  
 }  
  
 int getUnitPrice() {  
 return Unitprice;  
 }  
  
 public void setProductID(int id) {  
 productId = id;  
 }  
  
 public void setImageID(int ID) {  
 imageID = ID;  
 }  
  
 public void setQty(int qty) {  
 this.qty = qty;  
 }  
  
 public void setUnitPrice(int i) {  
 Unitprice = i;  
 }  
}