The salesReport project allows an employee to enter a simple daily sales report and prints out the results. The employee must enter his username and password. After verifying the username and password, the program allows the employee to enter his sales details for the day. Sales details include the product name, product ID#, price, and quantity sold. An object should be created for each product and then added to the daily collection of product objects. When the salesperson is finished entering data, the program calculates the total cash value of the salesperson's sales for the day and prints out the report. Students were given the following incomplete code for the Item class and the SalesReportDriver class.

public class Item

{

private String name; // the name of the item

private String itemID; // the id # of the item

private double price; // the price

private int quantity; // the number in stock

/\*\*

\* Construct an item.

\*/

/\*

Write a constructor for an item.

It should have a parameter for each field.

It should initialize each field.

\*/

/\*

Write a separate accessor (get) method for each field (i.e. write 4 methods.)

\*/

/\*\*

\* Convert the item to a String.

\*/

/\*

Write a toString method for Item. It should

return a String composed of all 4 field values.

Make it look pretty.

\*/

}

public class SalesReportDriver

{

public static final String[] EMPLOYEE = { "Abe", "Bob", "Cud"}; // names of all employees

public static final String[] EMPLOYEE\_PSWD = { "aaa", "bbb", "ccc"}; // passwords for the employees

public static final String TITLE = "Daily Sales Report"; // title String for dialog boxes

public static final int QSTN = JOptionPane.QUESTION\_MESSAGE;

private static final int INFO = JOptionPane.INFORMATION\_MESSAGE

public static void main(String[] args)

{

String userName; // the name of the person using this program

String password; // user's password

String input; // temporary String for holding returned values from input dialog boxes

boolean done = false; // false until the employee is done entering products sold

String itemName; // the name of the item sold

String itemID; // product ID # for the item sold

double price; // price for the item

int quantity; // quantity sold

ArrayList<Item> items = new ArrayList<Item>();// holds all items sold by this employee today

/\* Ask the user for his name and store it in the appropriate variable.

Ask the user for his password and store it in the appropriate variable.

Verify the user and password using the "verifyUser" method you will write below.

Get all required information for the first item from the user, create the first item,

and store it in the item ArrayList.

Ask the user if he is done entering products. If not repeat necessary steps until he is.

Print out the sales report using the method you will write below.

Exit the program.

\*/

}

/\* Write a method called "printReport" that prints out the sales report according to the sample output. You will need to calculate the total price. \*/

/\* Write a method called "verifyUser" that checks the user's name and password against the EMPLOYEE array and the EMPLOYEE\_PSWD array. If the userName is not in the EMPLOYEE array, or if the userName and password do not correspond, show the "Invalid User" message as shown in the sample output and then exit the program. Otherwise do nothing. \*/

/\* Write a method called "testForCancel" that checks a String to see if it is null. If the String is null exit the program, otherwise do nothing. \*/

}