CSC 355. Discrete Structures & Basic Algorithms
Project 1. Programming in Java
“Self pay kiosk”

General Guidelines.

The method signatures indicate the required methods. You may need additional methods or classes that will not be directly tested but may be necessary to complete the assignment, and you are welcome to add those into the classes.

Unless otherwise stated in this handout, you are welcome to add to/alter any provided java files as well as create new java files as needed, but make sure that your code works with the provided test cases. Your solution must be coded in Java.

Note on academic dishonesty: Please note that it is considered academic dishonesty to read anyone else’s solution code, whether it is another student’s code, code from a textbook, or something you found online. You MUST do your own work! It is also considered academic dishonesty to share your code with another student. Anyone who is found to have violated this policy will be subject to consequences according to the syllabus and university policy.

Learning Goals.

- Practice with generics in Java.
- Gain an understanding of how to implement basic Java instructions.

Project Overview.

Write a SelfPayKiosk class to support basic operations such as scan item, cancel transaction, checkout, and make payment. SelfPayKiosk.java is provided with method stubs. Follow each step to gradually complete all methods.

Step 0. Declare private fields for number of customers served (int), total sales (double), and current amount due (double). Note the provided final variable for sales tax of 7%.

Step 1 (5 pts). 1) Complete the constructor to initialize all private fields to zero. 2) Complete the accessor methods to return the number of customers served, total sales, and current amount due.

Step 2 (10 pt). Complete the scanItem() method. Increase the amount due by parameter price. Do not update amount due if parameter price is negative.

Step 3 (5 pt). Complete the checkOut() method. Multiply amount due by SALES_TAX and add to amount due.

Step 4 (10 pts). Complete the makePayment() method. If parameter payment is enough to pay the amount due, increase total sales by amount due, increment number of customers served, and reset amount due to zero in preparation for the next customer. However, if parameter payment is **not** enough, update total sales by payment and reduce amount due by payment. Do not make any changes if parameter payment is negative.

Step 5 (10 pt). 1) Complete the resetKiosk() method to reset all private fields to zero. 2) Complete the cancelTransaction() method to reset amount due to zero.

Step 6 (10 pts). Complete the simulateSales() method to perform multiple transactions with increasing prices. Use a loop to simulate parameter numSales transactions. Within the loop, call scanItem() with parameter initialPrice. Call checkOut() and makePayment() to make a payment of $1 more than the amount due. Finally, increase the item price by parameter incrPrice in preparation for the next transaction.

Step 7 (10 pt). Add a boolean private field to indicate if the customer has checked out and is ready to make a payment. Only allow payment after customer has checked out. The cancelTransaction() method should **not** reset amount due if the customer has checked out. Update the following methods by inserting assignment statements and if statements related to the boolean filed: constructor, checkOut(), makePayment(), and cancelTransaction(). Ex: Set the boolean field to false only after full payment has been made.

Lastly, from the files provided in D2L. Modify *SelfPayKiosk.java* and *Main.java* documents to solve the exercise.

Submission Procedure.

Submit your java files in D2L.

Grading

Total points = 60 points.