CS180 Project 1

1 Creating a simple invoice

Project assigned on: Friday, Jan 27th 2012
Project due date: Thursday, Feb 2nd, 11:59 p.m
Last revised on: -
List of revisions: -

1.1 Description

In this project you will create an invoice based on user inputs like customer name, quantity, price of items, applicable tax etc. You will read in details of three items(item code, qty and price) and applicable discount to arrive at the total amount. A quick look at figure 1 and figure 2 will give a good overall requirement of the project.

2 Learning Objectives

1. Learn how to use Scanner to read user input
2. Learn how to work with numerical data types
3. Learn how to use standard java classes like Random, String, Math, Date etc
4. Learn how to format numerical data to required precision for display.

3 Project Setup

Create a folder project1 in your cs180 folder. Save all your Java source files in the project1 folder. You will be turning in this folder when the project is completed.

Note: Use the following commands if needed.
% cd cs180
% mkdir project1
% cd project1
% pwd
/u/u9x/yourlogin/cs180/project1
% drjava
4 Project Details

Decide on the name of your java class and make sure to save it in a file with the same name. Remember that all names are case-sensitive. Once you start coding, remember to import any required classes and test your code frequently. Don’t try to write the entire program before compiling and testing. **There is no need to handle invalid user input in this project.**

4.1 Code Organization

Try to organize your code by adding one method for each section below. For example you can have a readUserInput() method and processData() method. Your main method should then instantiate an object and call these methods in correct order.

4.2 Gathering user information

Your program should read the name of a customer and other details as shown below in order to create an invoice.

![Figure 1: Reading User Input](image)

1. Read a customer’s first name and last name.

2. Read details of **3 items** that the customer purchased. This number is fixed.

3. Each item has an alphanumeric code. This will be a single word with a reasonable number of letters and digits.

4. Quantity (Qty) is the number of items purchased by the customer. This will ALWAYS be a whole number (no decimals).

5. Price of an item can be a decimal value.

6. Your program should also allow the user to enter a discount percentage. This also will be an integer value. If user enters 10, this means 10% discount should be applied.
4.3 Data Processing and Output

Use the collected data to print a receipt as shown in Figure 2.

![Figure 2: Output](image.png)

1. Display the customer name (first name followed by last name) as shown.

2. Display the date in mm/dd/yy format

3. Generate an invoice number. The invoice number should be a random 4 digit number. i.e. any number between 1000 and 9999 (including both)

4. Display details of all three items entered by user - in order, in a column wise format.

5. Calculate total before tax as:
   \[
   \text{item1.Qty} \times \text{item1.Price} + \text{item2.Qty} \times \text{item2.Price} + \text{item3.Qty} \times \text{item3.Price}
   \]

6. Calculate \( \text{Discount} = \text{Total.Before.Tax} \times \text{User.Entered.Discount} \times 0.01 \) Note that the output also shows at what percent the discount was calculated.

7. Calculate tax rate. The tax rate should be a random integer value between 2 and 7.

8. Calculate \( \text{Tax} = \text{Total.After.Discount} \times \text{Tax_Rate} \times 0.01 \)

9. Calculate \( \text{Total} = \text{Total.After.Discount} + \text{Tax} \)

4.4 Displaying Output

Keep the following in mind when displaying the output

1. All decimal values should be formatted to exactly 2 digits of precision
2. Display should be well-aligned. That is if item1 code is only two characters, item1 Qty and Price should NOT shift left. You can use format() method of String or System.out object for this. The tab character also may be used to align. Note that there is no need to handle very large user inputs.

3. Try to match your output as closely as possible to figure 2.

5 Coding Standards

Make sure you follow the coding standards specified in the course website here.

6 Grading

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Data Types</td>
<td>5</td>
</tr>
<tr>
<td>Data Input</td>
<td>25</td>
</tr>
<tr>
<td>Random Number Generation &amp; Calculations</td>
<td>15</td>
</tr>
<tr>
<td>Correct Output</td>
<td>25</td>
</tr>
<tr>
<td>Output Formatting</td>
<td>20</td>
</tr>
<tr>
<td>Coding Standard</td>
<td>10</td>
</tr>
</tbody>
</table>

7 Turnin

Make sure you have class level comments which clearly specifies your name, cs login, recitation section number, and date.

/*
 * name:
 * cs login:
 * recitation section
 * date
 */

To turnin first remove all the .class files from your project1 folder

% pwd
/u/u9x/yourlogin/cs180/project1
% rm -i *.class

Very Imp: change your current folder to your cs180 folder and only then run turnin command.

% cd ..
% pwd
/u/u9x/yourlogin/cs180
% turnin -v -c cs180=XXX -p project1 project1

Find in the following table the recitation section you are enrolled in, and substitute XXX with the corresponding value of the column XXX.
<table>
<thead>
<tr>
<th>Section Day</th>
<th>Time</th>
<th>Rec. TA</th>
<th>XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01 F</td>
<td>11:30</td>
<td>Gnanasekaran, Bala Murugan</td>
<td>REC1</td>
</tr>
<tr>
<td>R02 F</td>
<td>2:30</td>
<td>John Zage</td>
<td>REC2</td>
</tr>
<tr>
<td>R03 F</td>
<td>9:30</td>
<td>Stephen, Julian</td>
<td>REC3</td>
</tr>
<tr>
<td>R04 F</td>
<td>8:30</td>
<td>Sriram, Dinesh</td>
<td>REC4</td>
</tr>
<tr>
<td>RM1 F</td>
<td>3:30</td>
<td>John Zage</td>
<td>REC2</td>
</tr>
<tr>
<td>RM2 F</td>
<td>10:30</td>
<td>Stephen, Julian</td>
<td>REC3</td>
</tr>
<tr>
<td>RM3 F</td>
<td>4:30</td>
<td>Sriram, Dinesh</td>
<td>REC4</td>
</tr>
</tbody>
</table>