DUE: 2/14/2013

Project #2—PSE Invoice

Puget Sound Energy's monthly gas bill consists of several charges:

- Customer charge—a flat fee of $6.25
- Delivery charge—$0.29062 per thermal unit
- Cost of gas—$0.69156 per thermal unit for the first 50 thermal units, $1.23863 per thermal unit for 51-100 thermal units and $2.0389 per thermal unit thereafter
- Gas Conservation Program Charge—$0.00437 per thermal unit
- City Tax—6% tax. The tax applies to all charges

Write a Java program to display a simplified GUI-based gas bill for PSE—a sample of the UI is provided below. The customer's name, address, the previous and present meter readings should be read from text fields. When a Display Bill JButton is clicked, the customer's name and summary of the bill should be displayed in a JListBox or a JTextArea, as shown in the sample run below. The date shown should be the system date shown on form load, i.e., you do not type the date.

1. The project is worth 30 points and it will be graded on accuracy, completeness, presentation, comments, and the like.
2. Make sure that your name, project number, and due date appear as comments in your code (use Javadocs throughout your program code and generate them).
3. Make sure that you validate the input for present and previous meter readings and that you apply appropriate rates. Validation for other inputs is required.
4. This project requires inclusion of at least one other class, Invoice class, for calculating and displaying the bill. About form, Splash Screen, and printing are also required.
5. Submit a zipped file that contains all pertinent files and subfolders via Blackboard by midnight of the due date.

EXTRA CREDIT (Optional—4 points). Enhance the form by adding a Customer class and a combo box for the customer name. Once a name is selected, the remaining fields (address, state, and zip) should be not editable and should be filled automatically by reading the required fields from a Customers.txt file. Consider a comma-delimited text file such as this one:

Kahte Lamb,459 S. Super High Avenue, Edmonds,WA,98022-5729
Peter Rabbit,123 Bunny Road,Cabbage Town,OR,97555-6312
Captain Spiff,4444 Pluto Way,Andromeda Town, NV 83112
Flupsy Bunny,69 Carrot Court,Bunnyville,IO,55519

Requirements and Guidelines for Project 2

Drawing a detailed flowchart prior to coding this project might save you a lot of headaches later! Other than that, here are few hints (read carefully the grading rubric at the end of this file as well):

⇒ Declare all constants (customer, delivery charge, 3 rates for cost of gas, conservation charge, city tax and any others).

⇒ Declare meter readings and total units as integer and calculate total units using input validation for both type and range (e.g., previous reading should not exceed present).

⇒ Calculate delivery charge and conversation charge (based only on total units used).

⇒ Calculate cost of gas with nested if statements:

- If total units used <= 50 multiply total by 0.69156.
- Else if total units used <= 100 multiply first 50 by 0.69156 and the remaining units by 1.23863.
- Else multiply first 50 by 0.69156 and next 50 by1.23863 and the remaining units over 100 by 2.0389.

⇒ Calculate subtotal, tax on subtotal, and total.

⇒ Display all charges as currency, with right-justification for the second column.

⇒ Check your invoice against known results (see screen captures above and below).

⇒ The Invoice class is responsible for calculating the bill and returning a StringBuffer (or StringBuilder) object for displaying the bill. It should contain at least:

- Private constants and instance variables.
- Two constructors, one default.
- Set and get methods as well as a methods for calculating the bill.
- A method for padding spaces in the output for right-justification.
- A method that returns the formatted StringBuffer object containing the bill output.

⇒ Ability to print the bill (the content of the JTextArea or the JList).

Grading Rubric for Project 2

```
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*Comments by the prof:
*Great effort.  Here are suggestions for improvement:
*1. Use and generate Javadocs. Use comments to describe each method and complicated
code.
*2. An image of PSE logo would enhance the look of the form.
*3. Declare constants for customer charges, delivery charge, cost of gas (3 separate
*constants), delivery charge, city tax, and conservation charge.
*4. Correct the formulas for calculating the total bill. Make sure you apply the
*correct rates for different thermal units.
*5. Right-justify the $$ fields and the customer name—set JTextArea with Courier New
font.
*6. Use JFormattedTextFields for state and zip with proper masking.
*7. Make Display Bill default button.
*8. Use at least two classes for the project: PSEGUI and an Invoice class.
*9. Make date, total units and output all read-only fields.
*10. Show current date as the form loads.
*11. Disable maximization and resizing of all forms.
*12. Instantiate invoice object and customer object in the PSEGUI constructor.
*13. It is good style to tab code inside a method.
*14. Clean up empty methods stubs.
*15. Enable menu choices.
*16. Validate quantity inputs for type and range—quantity inputs should be
*integers > 0 and up to some maximum amount.
*17. Previous meter reading should not exceed present meter reading.
*18. Set focus to present meter reading as form loads.
*19. Program crashes on empty fields.
*20. Avoid excessive instance variables and unnecessary duplication of code.
*21. Check types and range for each input.
*22. Missing title or icon in About form.
*23. Follow Java naming convention for naming classes, methods, variables, projects.
*24. Missing print.
```

```
*25. Add icons and images to all forms.
*26. Center all forms as they start.
*27. Missing an About form.*28. Missing Splash Screen.
*29. Redesign your Invoice class: it should be able to calculate and return a
display StringBuffer object for the bill.
*30. Right-justify meter readings fields and state and zip.
*31. Missing title in main or About form.
*
*The ones that apply to your project are:
*3, 11, 29
*
*27+1=28/20
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
```