

COSC 1P02 Assignment 7

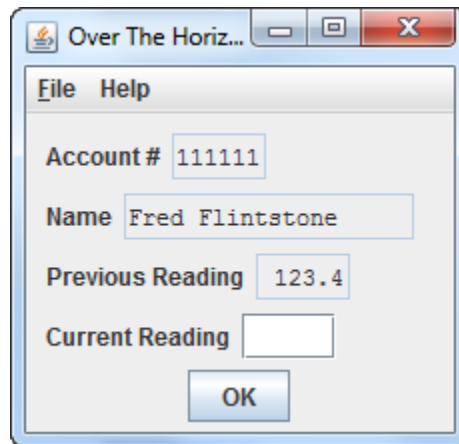
“New Horizons”

Due: Dec. 7, 2015 @ 10:00 am (late date Dec. 10 @ 10:00 am)

In preparation for this assignment, create a folder called `Assign_7` for the DrJava project for the assignment. The objective of this assignment is to apply files, reports, GUI and class design in a data processing application.

Problem

Over the Horizon Utilities is an electric utility that provides electricity to homeowners. At the end of each month, it does billing for each customer. The amount charged the customer is the consumption (in kilowatt hours (kwh)) times the current rate (\$1.525/kwh). A file (`ASCIIDataFile`) of customer data is maintained and updated (as an `ASCIIOutputFile`) when the billing is done. It contains, for each customer, the account number (`String`), customer name (`String`) and meter reading (from the last billing period, `double`). The program processes all the customer records and requests from the clerk (user of the program) the current meter reading for the customer via a `BasicForm` such as:



The screenshot shows a Java BasicForm window titled "Over The Horiz...". The window has a standard Mac OS X-style title bar with a red close button, a yellow maximize button, and a green minimize button. Below the title bar is a menu bar with "File" and "Help" menus. The main content area contains four labeled text input fields: "Account #" with the value "111111", "Name" with the value "Fred Flintstone", "Previous Reading" with the value "123.4", and "Current Reading" which is currently empty. At the bottom center of the window is an "OK" button.

It then computes the billing amount for the customer as the difference between the current and previous meter readings (consumption) times the rate and updates the meter reading in the customer record (i.e. replaces the previous reading with the current one).

The program produces a report (`ReportPrinter`) listing the relevant customer billing data and the total billed as summary such as:

Over The Horizon Utilities Billing Report				
Account #	Name	Previous	Current	Charge
111111	Fred Flintstone	123.4	234.5	\$169.43
222222	Barney Rubble	234.5	333.3	\$150.67
333333	Homer Simpson	345.6	999.9	\$997.81
444444	Wile E Coyote	0.0	1.0	\$1.52
	Total Billed			\$1,319.43

Write a class called `Account` that represents a single customer's account and a main class called `Billing` that performs the billing, reading and updating the customer data and producing the report.

Consider developing the program in phases:

0. Decide what information should be represented by an `Account` object (knowing: instance variables) and what the `Account` object should do (doing: methods).
This will be done as a class exercise in the Tutorial on Nov. 26.
1. Based on this model, write a class `Account` that captures the responsibilities of an `Account` object including those determined in 1 and the ability to read account data (constructor) and write updated account data (method `write`).
2. Write a main class `Billing` that uses the `Account` constructor to read account records and the `Account` `write` method to write to the updated account file producing a copy of the original account file.
3. Modify the `Billing` class to input (from an `ASCIIPrompter`) the current meter reading, compute the billing amount writing it and the account number to an `ASCIIDisplayer`, update the meter reading in the `Account` and write the updated account file.
4. Replace the `ASCIIPrompter` with a `BasicForm` as shown above and load it with information from the `Account` object before presenting it on the screen to get the current reading.
5. Replace the `ASCIIDisplayer` with a `ReportPrinter` as shown above, produce the appropriate detail lines and compute the total billed presenting it as the report summary.

Submission:

Details regarding preparation and submission of assignments in COSC 1P02 are found on the COSC 1P02 Sakai Site as `Assignment Guidelines` under `Course Documents`. This document includes a discussion of assignment preparation, programming standards, evaluation criteria and academic conduct (including styles for citation) in addition to the detailed assignment submission process copied below.

To prepare and submit the assignment electronically from the lab, follow the procedure below:

1. Ensure your folder (say `Assign_6`) for the assignment is stored on your Z : drive.
2. Using DrJava, print (to CutePDF Writer) the .java files (`Account` and `Billing`) of your assignment using the name `ClassName.pdf` where `ClassName` is the class name (i.e. same name as the .java file) and save the .pdf file at the **top level** of the project folder (i.e. directly within `Assign_7`).
3. Run the program using the data file `accounts.txt` entering the data
234.5
333.3
999.9
1.0
in the forms as the `Current Reading`, producing an updated account file (as `newAccounts.txt`) and a report (as `billReport.pdf`) at the **top level** of the project folder (i.e. directly within `Assign_7`). Open `newAccounts.txt` in Notepad (or similar) and print it selecting CutePDF as the printer with the name `newAccounts.pdf` at the **top level** of the project folder (i.e. directly within `Assign_7`).
4. Create a .zip file of your submission by right-clicking on the top level folder (i.e. `Assign_7`) and selecting `Send to/Compressed (zipped) folder`. A zipped version of the folder will be created. Use the default name (`Assign_7.zip`).
5. Log on to Sakai and select the COSC 1P02 site.
6. On the `Assignments` page select `Assignment 7`. Attach your .zip file (e.g. `Assign_7.zip`) to the assignment submission (use the `Add Attachments` button and select `Browse`). Navigate to where you stored your assignment and select the .zip file (e.g. `Assign_7.zip`). The file will be added to your submission. Be sure to check the `Honor Pledge` checkbox. Press `Submit` to submit the assignment. You should receive a confirmation email.
7. Assignments incorrectly submitted will not be marked.

DrJava

The .zip folder you submit should contain the project folder including all files relevant to the project—the .drjava, .java and .class files for the assignment and .pdf files for program listings and output.

Other Platforms

If you are using an IDE other than DrJava to prepare your assignment, you must include the .java source files and the .pdf files described above as well as a file (likely .class or .jar) that will execute on the lab machines.