

SAMPLE PROGRAM for CS READINESS TEST

General Instructions

You are to demonstrate good use of the concepts of classes, interfaces, graphical user interfaces, file reads, and methods. You must have properly commented code. Your program will be run in Java 8. You may not create Packages and you may not use JavaFX. Your program will be tested with several different files using the same format.

Specific Instructions

0. Your full name must appear as a comment in every class.
1. Write classes to implement the Yard interface shown below. Your implementing classes should be named CustomerSmallYard, CustomerMediumYard, and CustomerLargeYard. The compareTo method should compare two Yard objects using last names. In addition, you will need a class called DriverSampleYard. DriverSampleYard should contain main. You may need additional classes named PanelSampleYard and DisplaySampleYard. The Customer classes must manipulate customer information. The Panel and Display classes must store customer information and contain the GUI. You may write additional classes if you need them.
2. Add program code to read data from a file. You may read from file "greenGrow.txt". However, your program may be checked with a different file than the one used in Sample – it will have the same format. As part of your program, you must ask for and accept the file name including the dot txt. This may be done either in the console or with a JOptionPane. Each time the "Next" button is clicked in the GUI, the next set of customer info is read from the file and displayed in the GUI.
3. Keep a running total of the costs that is updated in the GUI each time the "Next" button is clicked.
4. Store all data into an array after it has been read. Alphabetize the data in the array by last name. You must use your own code to do this. When the user clicks the "Quit" button, print the alphabetized array to the console.

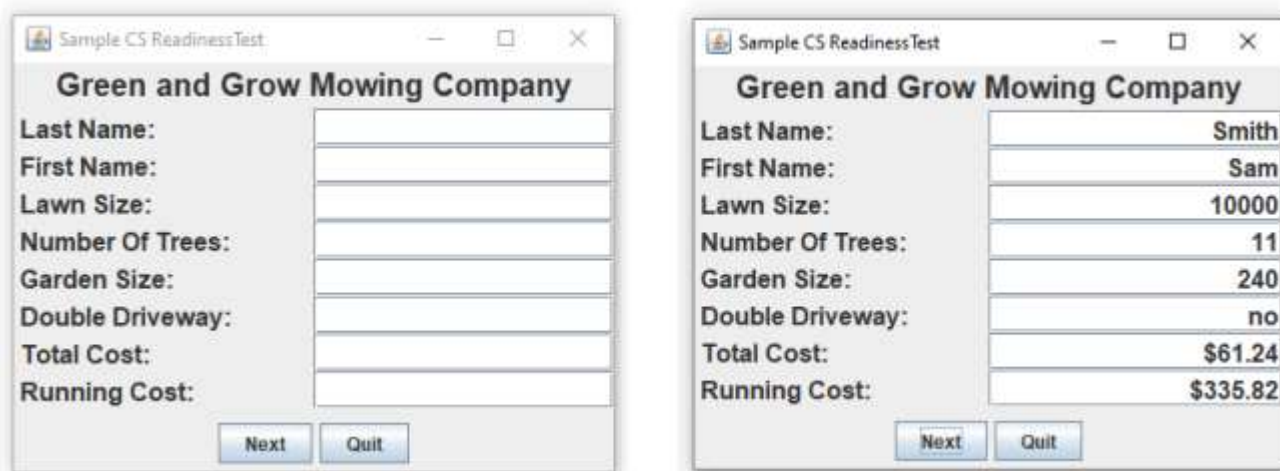
```
-----  
public interface Yard {  
    public abstract double getCost();  
    public abstract String getFirstName();  
    public abstract String getLastName();  
    public abstract int compareTo(Yard y);  
    public abstract int getNumberTrees();  
    public abstract int getGardenSize();  
    public abstract String getDoubleDriveway();  
    public abstract String toString();  
    public abstract int getSize(); }  

```

The Green and Grow Lawn Mowing Company charges its customers based on their total yard to be mowed. Those with a yard of 10,000 square feet or smaller are charged \$0.005 per square foot, those with yards larger than 10,000 square feet but not more than 20,000 square feet are charged \$0.004 per square foot, and those with yards larger than 20,000 square feet are charged \$0.003 per square foot. This constitutes the base fee. Over time, the Green and Grow Lawn Mowing Company has discovered that another factor affects the speed they are able to mow a lawn – trees. And they discovered that an even number of trees had no effect, but an odd number of trees slowed mowing and required an average of 5% more time. Customers with an odd number of trees are now charged a 5% surcharge – or 5% more than the base fee. They have also discovered that some customers have double car driveways. These take less time and receive a 3% discount from the base fee. Further, some customers have a garden. Those with a garden have the size of their garden, given in square feet, deducted from their yard size. A garden size of zero indicates no garden. However, an additional garden fee of \$10 is charged since workers must work around the garden.

Declare classes that implement Yard to include a customer's name and yard size.

Your before and after GUIs look like the following:



The Running Cost or Running Total for the five customers in the file below will be \$335.82. Sam Smith’s cost is \$61.24.

To assist you, the file “greenGrow.txt” has the information written one item per line. However, we have written the information on a single line to save space on this page. The first line of the txt file will always contain the number of entries.

Condensed “greenGrow.txt”:

```
5
Jeffers Tom 5000 10 0 yes
Nevar Tina 15000 12 360 no
Kim Lisa 20000 13 400 yes
Black Kim 30000 17 0 yes
Smith Sam 10000 11 240 no
```