

NAME: _____

PERIOD: _____

May 2020

SAMPLE_READINESS

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. You are to mail in a flash drive containing this problem with your application. 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. Solve problem. Write classes to implement the Yard interface shown below. Your implementing classes should be named CustomerSmallYard, CustomerMediumYard, and CustomerLargeYard. In addition, you will need a class called DriverSampleYard. 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.
3. Store all data into an array after it has been read.
4. Alphabetize the data in the array by last name. You must use your own code to do this.
5. Add program code, in the form of a method, to accomplish data calculations.
6. Add program code to keep track of a running total of individual calculations.
7. Add program code to output to your data.
8. Ensure your program will process multiple data sets. Each data set starts with a last name.

```
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 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.

Declare classes that implement Yard to include a customer's name and yard size. Your before and after GUIs look like the following:

A screenshot of a GUI window titled "Sample CS Placement Test" with the subtitle "Green and Grow Mowing Company". The window contains several input fields: "Last Name:", "First Name:", "Lawn Size:", "Number Of Trees:", "Double Driveway:", "Total Cost:", and "Running Cost:". At the bottom, there are two buttons: "Next" and "Quit".

A screenshot of the same GUI window after data entry. The fields are populated with the following values: "Last Name:" is "Smith", "First Name:" is "Sam", "Lawn Size:" is "10000", "Number Of Trees:" is "11", "Double Driveway:" is "no", "Total Cost:" is "\$52.50", and "Running Cost:" is "\$310.15". The "Next" and "Quit" buttons are still present at the bottom.

The Running Total for the five customers in the file below will be \$310.15.

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.

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