## Exercise 5: Inheritance, Abstract Classes, and Interfaces:

Write a Java program that calculates and display a customer water bill. Water costs more per gallon as use increases. Follow the following rules in the table for calculating the water bill.

At program start, the program must (FIRST) ask the user for the number of customers, such as **n** customers. Then the program creates an array(Customer[] _customers_) of size n to store Customer details.

For every customer information, you must create a customer object and add it to the **_customers[]_** array in **_TestWaterBill_** class. The Customer class has an association with Waterbill class that represents the customers' water bill (**SingleFamily** or **Duplex**). The Customer class implements the **Comparable** interface, thus it must implement the **compareTo()** method. The **SingleFamily** and **Duplex** is derived from the abstract **_WaterBill_** class. The abstract **_WaterBill_** class has data fields of gallons, value, and date created and it implements the **BaseCharge** interface. The **_WaterBill_** abstract class has an abstract **computeBill()** method that calculates the customer bill value.

UML DIAGRAM:
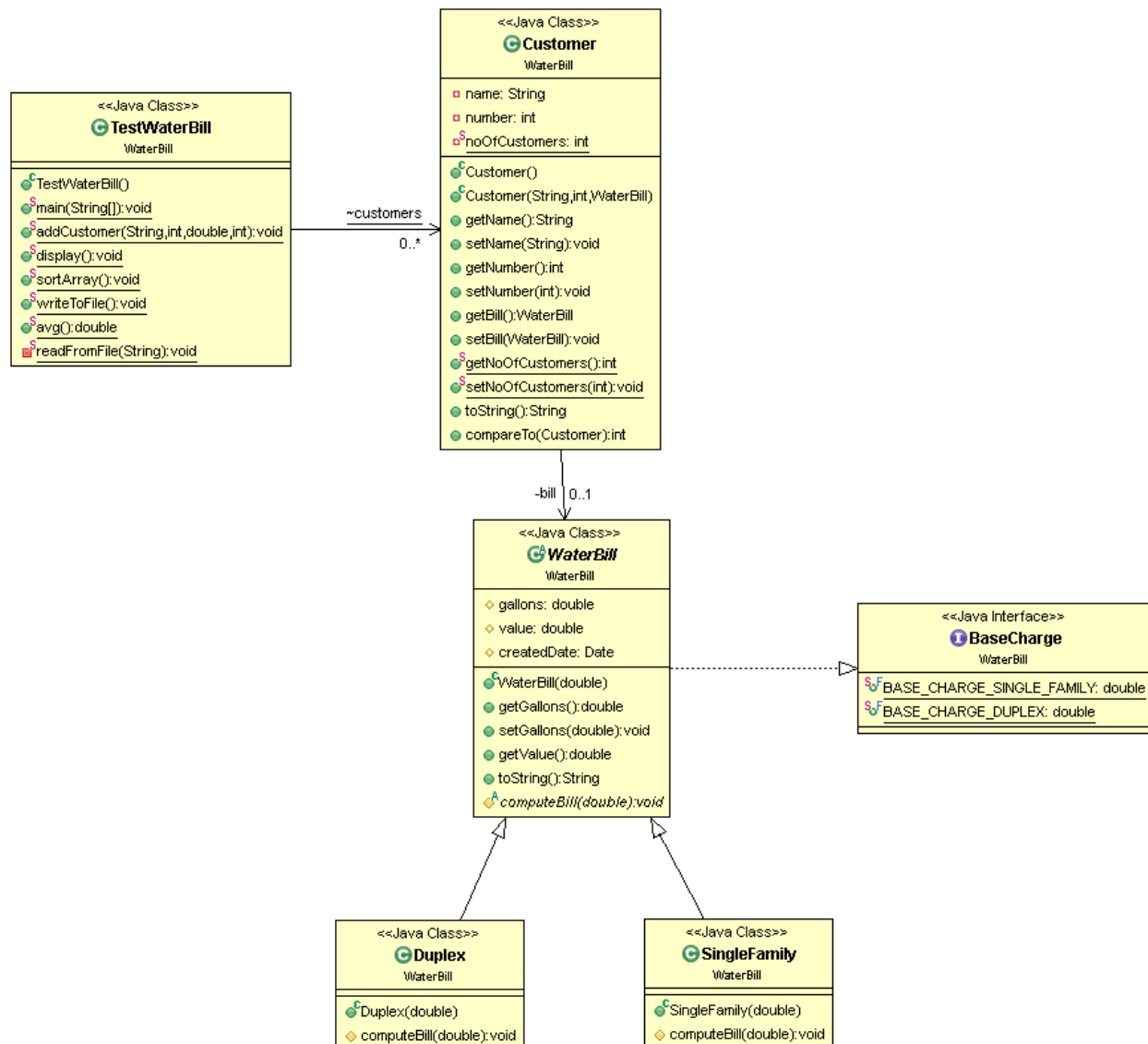
After reading all customers data and computing bills, display in one dialog box, the customer information along with bill details line by line. Finally, store the array of Customer objects into a file. Each Customer object is stored in one line.

Single-family and Duplex Water Rates

Single-family and duplex water rates have a base charge and three tiers.

Water		Single-Family	Duplex
Base Charge		$13.21	$15.51
Tier	Volume Charge per 1,000 Gallons		
1	0–7,000 Gallons	$2.04	
	0–9,000 Gallons		$1.97
2	7,001–13,000 Gallons	$2.35	
	9,001–13,000 Gallons		$2.26
3	Over 13,000 Gallons	$2.70	$2.60
Charges rounded to the nearest cent. Actual costs vary due to rounding.			

Note: Even though water rates are given as a rate per 1,000 gallons, the actual rates are still calculated on a per gallon rate. (i.e. for tier 1 the rate is $.00204 per gallon)

Use **JOptionPane** for all input and output.

Program output sample:
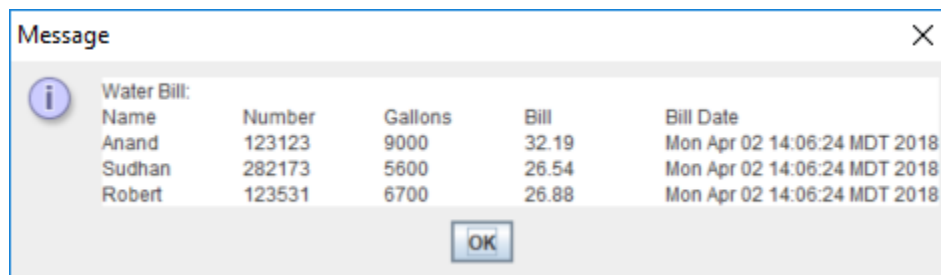
Instructions: Use the JOptionPane class for the program input and output, and if you need to format the program output, use the **String formatValue = *String. format("%.2f", value)* or** any Java utility class to format the output.

Also, the program must not crash on an invalid user input format. Hence, use try-catch statements to handle a user's invalid inputs.