Project 09 Temperature Class

Purpose: The purpose for this project is to reinforce the knowledge from Chapter 9 of the textbook. The students will learn how to write a user defined class.

Project Objectives:

1. Apply UML design on user defined class
2. Write overloaded constructors of a class
3. Write mutators (i.e. get methods) and accessors (i.e. set methods) of a class
4. Write overloaded methods
5. Write main method to test the user defined class

Class Diagram:

Student must implement the Temperature class according to the following class design.

Temperature
-degree: double -scale: char
+Temperature() +Temperature(degree: double) +Temperature(scale: char) +Temperature(degree: double, scale: char) +getDegreeInCelsius(): double +getDegreeInFahrenheit(): double +setDegree(degree: double): void +setDegree(scale: char): void +setDegree(degree: double, scale: char): void +equals(obj: Temperature): boolean +isLessThan(obj: Temperature): boolean +isGreaterThan(obj: Temperature): boolean

Method details

The name of a method explains it usage.

1. getDegreeInCelsius will return the temperature's degree in its equivalent Celsius degree. If the temperature's scale is 'C', then the return value is temperature's degree. If the temperature's scale is 'F', the return value is calculated by the following formula: C = (F-32)*5/9. For example, if the temperature degree is 77 and scale is 'F', then the method will return 25 since (77-32)*5/9 = 25. For another example, if the temperature degree is 77 and scale is 'C', then the method will return 77.

2. getDegreeInFahrenheit will return the temperature’s degree in its equivalent Fahrenheit degree. If the temperature’s scale is ‘F’, then the return value is temperature’s degree. If the temperature’s scale is ‘C’, the return value is calculated by the following formula: F = 1.8C+32. For example, if the temperature degree is 25 and scale is ‘F’, then the method will return 25; For another example, if the temperature degree is 25 and scale is ‘C’, then the method will return 77 since 1.8*25+32 = 77.

3. void setDegree(double degree) will reset the temperature to given degree without change the scale of the temperature.

4. void setDegree(char scale) will reset the temperature to given scale without change the degree of the temperature.

5. void setDetree(double degree, char scale) will reset the temperature to given degree of given scale.

6. equals method returns true if this temperature is equal to parameter Temperature; false otherwise. You need to compare tow temperatures under same scale. For example, you either compare whether getDegreeInCelsius return same value for both temperatures, or whether getDegreeInFahrenheit return the same value for both temperatures.

7. isLessThan method return true if this temperature is less than parameter Temperature ; false otherwise. Again, you need to compare two temperatures under same scale.

8. isGreaterThan method returns true if this temperature is greater than parameter Temperature; false otherwise. Again, you need to compare two temperatures under same scale.

Main method requirements

1. The main method must create four Temperature objects by using all four constructors.
2. The main method must test all set and get methods by setting the attributes according to user’s input then displaying the temperature by printing out the result of get methods.
3. The main function must test equals, isLessThan, and isGreaterThan methods by compare two Temperature objects and print out the result.

Sample pseudo code for main function

1. Create a Temperature object by calling default constructor
2. Print out the temperature’s degree in Celsius
3. Print out the temperature’s degree in Fahrenheit
4. Ask user to enter the scale and degree of the temperature
5. Call set method(s) to set the temperature’s degree and scale
6. Print out the temperature’s degree in Celsius
7. Print out the temperature’s degree in Fahrenheit

8. Repeat step 1 to 7 for all other three constructors to create three more temperatures
9. Test if temperature1 equals temperature2 and print out appropriate message
10. Test if temperature1 is less than temperature2 and print out appropriate message
11. Test if temperature1 is greater than temperature2 and print out appropriate message
12. Repeat step 9 to 11 for other pairs of temperatures

Note on main function and debugging

Fix one error a time.

1. If step 2 output is wrong, then either constructor is wrong or getDegreeInCelsiu function is wrong
2. If step 3 output is wrong, then either constructor is wrong or getDegreeInFahrenheit function is wrong
3. If step 6 output is wrong, then either set method(s) is wrong or getDegreeInCelsiu function is wrong
4. If step 7 output is wrong, then either set method(s) is wrong or getDegreeInFahrenheit function is wrong
5. If step 9 output is wrong, then the equals method has problem
6. If step 10 output is wrong, then the isLessThan method has problem
7. If step 11 output is wrong, then the isGreaterThan method has problem

Submission:

Submit the YourProj09.java file via blackboard link. The due date will be announced on blackboard

Sample Run (The highlight ones are input):

```
The first Temperature has been created using the default constructor which sets

the degree to a deafault value of 0.0 and the scale to a default value of C.

The first Temperature is 0.00 C.

The first Temperature is 32.00 F.

Set the degree (a number) and the scale (F or C) of the first Temperature.

First set the degree: 77

Now set the scale: C

Now the first Temperature is 77.00 C.

which is 170.60 F.
```

```
The second Temperature has been created using the constructor with a double argument.
Which set degree to this argument and the scale to a default value of C.
In my code, I passed 32 to the double argument. So
The second Temperature is 32.00 C.
The second Temperature is 89.60 F.
Set the degree (a number) and the scale (F or C) of the second Temperature.
First set the degree: 77
Now set the scale: F
The second Temperature is 25.00  C.
The second Temperature is 77.00  F.

The third Temperature has been created using the constructer which sets
the degree to a default value of 0.0 and the scale to the argument.
I passed 'F' to the argument. So
The third Temperature is 0.00 F.
which is -17.78 C.
Set the degree (a number) and the scale (F or C) of the third Temperature.
First set the degree: 25
Now set the scale: C
The third Temperature is 25.00 C.
which is 77.00 F.

The fourth Temperature has been created using the constructer which sets
the degree to double argument and and the scale to char argument.
I passed 98.6 and 'F' to the arguments. So
```

```
The fourth Temperature is 37.00 C.

The fourth Temperature is 98.60 F.

Set the degree (a number) and the scale (F or C) of the fourth Temperature.

First set the degree: 100

Now set the scale: C

The fourth Temperature is 100.00 C.

which is 212.00 F.


In order of creation the Temperatures in Celcius are: 77.00, 25.00, 25.00, 100.00

In order of creation the Temperatures in Fahrenheit are: 170.60, 77.00, 77.00, 212.00


The first Temperature is not equal to the second.

The first Temperature is not less than the second.

The first Temperature is greater than the second.


The first Temperature is not equal to the third.

The first Temperature is not less than the third.

The first Temperature is greater than the third.


The first Temperature is not equal to the fourth.

The first Temperature is less than the fourth.

The first Temperature is not greater than the fourth.


The second Temperature is equal to the third.

The second Temperature is not less than the third.
```

The second Temperature is not greater than the third.

The second Temperature is not equal to the fourth.

The second Temperature is less than the fourth.

The second Temperature is not greater than the fourth.

The third Temperature is not equal to the fourth.

The third Temperature is less than the fourth.

The third Temperature is not greater than the fourth.