

Instructor: Michael Hackett
Department: Computer Science
Email: mhackett@ccp.edu

Thermometer Application

Create a program that the simulates a thermometer and displays its temperature using different units.

The program must use five classes including the class that contains the main method. The classes have been prepared for you to begin your work. I recommend completing the classes in the order shown in this document.

Temperature Interface (Temperature.java)

This interface should declare the following two abstract methods:

- `getTemperature`
 - Returns a String
 - Accepts no arguments
- `setTemperature`
 - Returns void
 - Accepts one double argument

Abstract Thermometer Class (Thermometer.java)

This abstract class must implement the Temperature interface and contain:

- One private field (a double) named `degrees`
- One constructor that accepts a double argument to set the `degrees` field
- A getter and setter for the `degrees` field
- An abstract method named `convert`
 - Accepts no arguments
 - Returns a String
- Since Thermometer is abstract, it does not need to implement the abstract methods from the Temperature interface. However, non-abstract subclasses (like FahrenheitThermometer and CelsiusThermometer) will need to implement them.

FahrenheitThermometer Class (FahrenheitThermometer.java)

This class must be a subclass of Thermometer and include:

- No fields
- One constructor that accepts a double argument to set the degrees field in the superclass.
- An implementation of the getTemperature method (required by the abstract superclass's interface)
 - Returns a String that is the *degrees F*
 - For example, **45.7 F** (if the superclass's degrees field was 45.7)
- An implementation of the setTemperature method (required by the abstract superclass's interface)
 - Simply sets the superclass's degrees field
- An implementation of the convert method (required by the abstract superclass)
 - Returns the temperature in Celsius
 - Degrees Celsius = (Degrees Fahrenheit - 32) * (5/9)
 - Returns a String that is the *degrees C*
 - For example, **7.6 C** (if the superclass's degrees field was 45.7)

CelsiusThermometer Class (CelsiusThermometer.java)

This class must be a subclass of Thermometer and include:

- No fields
- One constructor that accepts a double argument to set the degrees field in the superclass.
- An implementation of the getTemperature method (required by the interface)
 - Returns a String that is the *degrees C*
 - For example, **62.8 C** (if the superclass's degrees field was 62.8)
- An implementation of the setTemperature method (required by the interface)
 - Simply sets the superclass's degrees field
- An implementation of the convert method (required by the abstract superclass)
 - Returns the temperature in Fahrenheit
 - Degrees Fahrenheit = (Degrees Celsius * (9/5)) + 32
 - Returns a String that is the *degrees F*
 - For example, **145.04 F** (if the superclass's degrees field was 62.8)

Thermometer Application (ThermoApp.java)

In the main method:

- Prompt the user to choose either Fahrenheit or Celsius.
 - You may implement this as the user entering “1 or 2”, “F or C”, etc.
- Prompt the user to enter the degrees.
- Instantiate either a FahrenheitThermometer or CelsiusThermometer object (based on the user’s choice) and assign it to the Thermometer variable that has been declared for you.
 - For example: **therm = new FahrenheitTemperature(degrees);**
- Uncomment the two lines that pass the Thermometer variable to the displayTemperature and displayConversion methods.
 - The methods have been provided for you. Do not modify them except for uncommenting the code in the two methods.

UML Diagram

You will also need to create a UML diagram that includes all classes except ThermoApp. You may do this in MS Word, MS PowerPoint, or even MS Paint (or comparable programs). Acceptable file formats will be PDF, JPG, or PNG.

Sample Input/Output

```
Choose Fahrenheit (F) or Celsius (C): F  
Enter the degrees: 89.5
```

```
The temperature is 89.5 F  
The temperature converted is 31.9 C
```

- Your program’s output must exactly match the formatting in the above example.
- You don’t have to round the temperatures that are printed, but you are welcome to do so.
- Be sure to use comments to document your code. Comments show me that YOU can explain, in plain English) what your program’s code is doing.

Grading

See Assignment Rubric in Canvas.