# CSE 331 Structural Patterns worksheet

You have been designated chief software architect at WeatherCorp International, an globally renowned weather forecasting agency. Right now, the agency’s codebase has this class which allows certain software to access the temperature at a given location, in Celsius[[1]](#footnote-1).

**public class** CelsiusReporter {

 **double** temperatureInC;

 **public** CelsiusReporter() {}

 **public** **double** getTemperature() {

 **return** temperatureInC;

 }

 **public** **void** setTemperature(**double** temperatureInC) {

 **this**.temperatureInC = temperatureInC;

 }

}

However, since your agency has international reach, and not every country uses Celsius, the agency would like you to write an **adapter** class that will convert between Celsius and Fahrenheit. They have already provided you the skeleton. All you need to do is write the corresponding getter/setter, but for Fahrenheit.

In case you do not recall: °C to °F conversion is given by the formula $F= \frac{9}{5}∙C+32$.

**public class** FahrenheitReporter {

 **private** CelsiusReporter cr; // assume this has already been initialized

 **public** FahrenheitReporter() {...} // you don’t need to write this

 // your code goes below

 **public double** getTemperature() {

 **return** cr.getTemperature() \* 9 / 5 + 32

 }

 **public void** setTemperature(**double** temperatureInF) {

 cr.setTemperature((temperatureInF – 32) \* 5 / 9);

 }

}

WeatherCorp Int’l is very pleased with your code and requests another feature. They want to be able to access historical data from their logs, in order to make more accurate forecasts. You look around in the WeatherCorp codebase and find their static WeatherLogger class, which has the following operation:

append(String event); // adds an event to the internal weather log

Every time you get the current temperature, you want to add a string of the following format, to the log:

 <TIME>: GET Temperature = <DEGREES> Celsius

Every time you set the current temperature, you want to add a string of the following format, to the log:

 <TIME>: SET Temperature to <DEGREES> Celsius

<TIME> is a Unix timestamp[[2]](#footnote-2) representing when the method was called, and <DEGREES> represents the retrieved temperature or set temperature at the method call.

In the space below, complete the **decorator** class for the existing CelsiusReporter class.

**import** com.weathercorp.util.WeatherLogger

**public class** CelsiusReporterLogger **extends** CelsiusReporter {

 **private final** **static** WeatherLogger logger;

 **public** CelsiusReporterLogger() {

 super();

}

 **public** **double** getTemperature() {

 **long** currentTime = getCurrentTime();

 **double** temperatureNow = this.temperatureInC;

 **String** logMessage = currentTime + “: GET Temperature = ” + temperatureNow + “ Celsius”;

 logger.append(logMessage);

 **return** *temperatureInC*;

 }

 **public** **void** setTemperature(**double** temperatureInC) {

 **long** currentTime = getCurrentTime();

 **double** temperatureNow = temperatureInC;

 **String** logMessage = currentTime + “: SET Temperature = ” + temperatureNow + “ Celsius”;

 logger.append(logMessage);

 **this**.*temperatureInC* = temperatureInC;

 }

 **private long** getCurrentTime() {

 **return** Instant.now().getEpochSecond(); // since Java 8

 }

}

1. Adapted from http://www.avajava.com/tutorials/lessons/adapter-pattern.html?page=1 [↑](#footnote-ref-1)
2. You may use online resources to figure out how to access a Unix timestamp in Java [↑](#footnote-ref-2)