Assignment 4: Weather Forecast ⛅

*Due **February 16 at 11:59pm***

Table of Contents

Table of Contents
Overview
Part 1: Temperature Converter
Part 2: The Weekly Forecast
Part 3: Putting it all together
Feedback Survey
Submitting Instructions

Overview

For this assignment, we're focusing on using template functions and lambdas effectively!

Download the starter code here.

Part 1: Temperature Converter

The Stanford Daily has just put up a listing for a new job as the campus weatherman! Since you ran out of your meal plan dollars for the quarter after eating at TAP twice, you decide to apply. However, they request that for the interview applicants present the day's weather forecast with high and low temperatures in both Fahrenheit and Celsius.

Rather than worry about handling the conversions on the fly, you decide to write a program to do it for you!

For this part of the assignment, you will write a template function called **convert_f_to_c**. The function should take in one temperature in Fahrenheit and convert to its corresponding temperature in Celsius. If you've forgotten the conversion, the equation is:

(degrees Fahrenheit - 32) * 5/9

Since the interviewer didn't tell you what format to expect the weather in, this function must be a template function that can take in any numerical type (such as ints, doubles, floats, etc..). It should, however, always return the result as a double.

Notes/tips:

- Remember the format for making a template function! Always use **template <typename [your typename here]** before the rest of the declaration.
- We only want this function to take in numerical types! Use a set of constraints or a concept to handle this. Try the function with different inputs to see whether it functions correctly. Feel free to use one that already exists in the STL!

Part 2: The Weekly Forecast

Congratulations, you got the job! No time to celebrate though; your first day is today! You've been put on the weekly forecast duty; with only a few hours till your big debut, you decide to write one more function to prompt you with each day of the week's highs and lows so you can ace your section. Who knows: if you keep this up, maybe you'll graduate from the weather and they'll put you on the next big MTL story!

You will be writing:

- **get_forecast(std::vector<std::vector> readings, void* fn):** given a vector of weekly temperatures, where each element in the vector is another vector of temperature readings from one day of the week, return a std::vector<double> of the correct statistics for each day. These statistics will be gathered by running the function **fn** over all of the readings for that day.

For example, if the readings were:

[[20, 40, 30], // Monday

```
[ 15, 30, 18], // Tuesday
[ 50, 30, 20], // Wednesday
[ 60, 45, 30], // Thursday
[ 33, 40, 45], // Friday
[ 25, 30, 44], // Saturday
[ 45, 40, 50] ] // Sunday
```

And **fn** returned the maximum temperature, then **get_forecast** should return:

[40, 30, 50, 60, 45, 44, 50]

Part 3: Putting it all together

Finally, it’s time to call your functions in the main! We’ve given you a vector of temperature readings over a week. In the main, we want you to:

- Convert the temperatures to Celsius
- From these new Celsius readings, find the maximum temperature for each day
- Find the minimum temperature for each day
- Find the average temperature for each day

To convert the temperature, we recommend checking out the STL algorithms library for a function that will let you apply **convert_f_to_c** across an entire container. For the other steps, we want to see you write three lambda expressions and use these in **get_forecast**.

Once you’ve created all of these vectors, write each of them on their own line in a new file called **output.txt**. You only need to print the numbers in the vector, separated by spaces. This will be submitted with your code for grading.

Feedback Survey

Please fill out this anonymous feedback form to help us improve these short assignments in the future!

Submitting Instructions

When you have completed this assignment, upload all of the files to Paperless under the correct assignment heading.
Your deliverables should be:

- **main.cpp**
- **output.txt**

You may resubmit as many times as you'd like before the deadline. Please let us know if you have any questions as you work on the assignment!