Homework

16

Writing to_string

Using the `to_string()` examples from the lesson, **write a template function** and place it into the header file.

- Make sure you surround it with the `#if` preprocessor directive, so it will **only be included when compiling with early versions of C++**.
- **Fully qualify** each library name.
- **Do not** put `using namespace` in your header file.
- Add your id to the `.cpp` file. **This will be the only code** inside the implementation file.
- Do `make test` to check your code. This will compile your code using C++98 and C++17, and then test the functions using C++11. Unfortunately, all does not seem perfect.

```
Testing using C++98
------------------------------------------
Checking: to_string function -------------------
   + to_string(42)->"42"
   X to_string(3.F): expected [3.000000] but found [3]
   + to_string(-1U)->"4294967295"
   + to_string(4L)->"4"
   X to_string(1.7L): expected [1.700000] but found [1.7]
   X to_string(2.7e13): expected [27000000000000.000000] but found [2.7e+13]
   + to_string(2.17e-4)->"0.000217"
   + to_string(1.0/0.0)->"inf"
   + to_string(-1.0/0.0)->"-inf"
   + to_string(0.0/0.0)->"-nan"
----------------------------------------------------------------------
  Tests passing 7/10 (70%).
```

You want **your code** to act exactly like the new, standard library version of this function. The expected values above are what the **standard library** returns when run with each of these tests. (We're using the C++17 library as a **test oracle** here.)

As you can see, the floating-point numbers don't produce the correct output. It looks like in the C++11 version of **`to_string()`**, floating-point numbers are converted using **`fixed`** notation, even if they were originally otherwise. Change one line to solve this problem.

```
out << std::fixed << value;
```

Now you can do **`make test`**, and, if you get 100%, then **`make submit`**. Visit me in my office or on the discussion board if you need help.