CSci 1113: Introduction to C/C++, Spring 2019
Homework 0

Due Date and Time: Monday, Feb. 4 at 5:30pm. Submit your source code file using the link on the class website.

Purpose: Welcome to Homework 0. This is a short practice homework that ensures you know how to do some essential tasks, such as compiling C++ code, naming your code file carefully, and submitting it using the class website.

Instructions: Here are some important instructions. Unless otherwise stated, these will hold not only for homework, but also for all subsequent homework assignments in this class.

1. Unlike the computer lab assignments, this is **not** a collaborative assignment. You must design, implement, and test the solution to each problem on your own without substantial assistance from anyone other than the course instructor or TAs. As part of this, you may not include solutions or portions of solutions obtained from any source other than those provided in class: examples from the textbook, lectures, or code you and your partner write to solve lab problems. Using code obtained in other ways, or letting others view or use your code is considered academic misconduct. For more information, see the collaboration rules file on the class website, or ask the instructor or TAs if you have any questions on what is and is not allowed.

2. Because homework problems are submitted and tested electronically, the following are important:

   - You follow the naming conventions mentioned at the end of this file.
   - You submit the correct file or files though the correct link on the course website.
   - You submit the file or files by the due date and time.
   - You follow exactly any example input and output formats given in the problem description.
   - Regardless of where you develop your solution, your program compiles and runs on the cselabs Linux machines.

3. The problem descriptions will usually show at least one test case and the resulting correct output. However, you should test your program on other test cases (that you make up) as well. Making up good test cases is a valuable programming skill, and is part of ensuring your code solution is correct.

For this homework, Homework 0, there is one problem worth 20 points total. Solve the problem below by yourself (unlike the labs, where you work collaboratively), and submit the solution through the class website.

Problem A: Spherical Coordinates (20 points)

There are a variety of ways to specify a point in three dimensions. The most common one is by using Cartesian coordinates, that is, specifying a point by giving its *x, y, z* coordinates. However, there are other ways that are useful. For example, one way is by spherical coordinates. The spherical coordinates for a point with Cartesian coordinates (x, y, z) is a triple (r, θ, ϕ) with

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \tan^{-1}\left(\frac{y}{x}\right) \\ \phi &= \cos^{-1}\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \end{aligned}$$

Note that different fields have different conventions for spherical coordinates. For example, some fields interchange the θ and ϕ coordinates. For this assignment use the coordinate ordering and equations above.

Your task is to construct a C++ program that allows a user to input the x, y, z coordinates of a point, and then computes and outputs the spherical coordinates. The output format should give the coordinates surrounded by parentheses, separated by commas and a space (as shown below), and with θ and ϕ given in degrees, not radians.

Here is what an example run of the program should look like for a test case (1.0, 1.0, 0.5). Input values are shown by underlining. Your program should not print out underlines — the underlines are there only to distinguish user input from what your program should print.

```
Input Cartesian coordinates x, y, z:  1.0 1.0 0.5
Spherical coordinates:  (1.5, 45, 70.5288)
```

And here is a second example with different input values:

```
Input Cartesian coordinates x, y, z:  -4.2 0.0 0.0
Spherical coordinates:  (4.2, 180, 90)
```

Additional instructions:

1. Test your program using not only the example input above, but also other test cases you make up.
2. For this program you do not need to check for problems like division by 0 errors.
3. Be particularly careful not only that your output values are correct, but also that your answer follows the format above. Avoid misspellings, extra spaces, or anything else that is not in accordance with the formatting.
4. In addition to the specifications here, some expectations about your program will be explained in lecture (that you write your program using reasonably good style, and that you use the appropriate type of variables, etc.).
5. Add a comment of the following type to the very beginning of file:

   ```
   // CSci 1113 HW 0
   // Your name
   ```

   (where you replace "Your name" with your actual name).
6. Name the source code file <username>_0A.cpp, where you replace <username> with your U of M email address; for example, if your email address is smithx1234@umn.edu, your file should be named smithx1234_0A.cpp.
7. When it is complete and correct, submit your program using the HW 0 Problem A submission link on the course website.

It is important that you follow the file naming conventions very carefully. For example, note your username should be all in lowercase, you should not include "@umn.edu", the file name should contain an underscore (not a dash), the 'A' in the "0A" part is upper case, the '0' is a zero, not the letter 'O', the extension is .cpp, etc. Following rigorous naming conventions is something computer programmers often must do in "real life" programming, and so submitting your program with the correct name is part of doing this assignment correctly.