Lab 02: Calculating PI

CSci 515 Spring 2015

2015-01-23

Dates:

Due: In Lab, Wednesday January 28, by 4 pm (lab end time)

Objectives

- Practice writing index controlled loops
- Become more comfortable with using digital computers for calculating mathematical expressions in C.
- More practice with output formatting.
- Practice using real valued variables for mathematical calculations.
- Gain experience in translating formula into algorithmic procedures.
- Practice using uniary, binary and special assignment operators.

Description

Calculate an approximate value of π from the series:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} ...$$

Print a table that shows the approximate value of π after each of the first N terms of this series. Your program should prompt the user for the value of N to determine how many values of the table of approximate values of π will be displayed. A session of using your program from the terminal should have exactly the following output:

```
What size of table should I compute: 5

N     pi
----- -------
1     4
2     2.66667
3     3.46667
4     2.89524
5     3.33968
```

Extra Credit

If you calculate the approximation of pi correctly using the series, you will see that using standard cout output it appears that only 6 digits of precision are being calculated (e.g. create a table of size 1000 or bigger). However, the standard float data type is capable of representing 16 digits of precision. The problem is that by default sending floats to cout only display a few decimal digits. Find out how to display all of the calculated digits, using the iostream library.

Lab Submission

An eCollege dropbox has been created for this lab. You should upload your version of the lab by the end of lab time to the eCollege dropbox named `Lab 02 Calculating Pi`. Work submitted by the end of lab will be considered, but after the lab ends you may no longer submit work, so make sure you submit your best effort by the lab end time in order to receive credit.

Requirements and Grading Ruberics

Program Execution, Output and Functional Requirements

- Your program must compile, run and produce some sort of output to be graded. 0 if not satisfied.

- Your program must correctly respond to the user input when displaying the number of items in the series. 20 or more points.

- Your program must correctly calculate the values in the series. 40 or more points.

- Your program must correctly format the output as shown in the lab description. 20 points

Program Style

Your programs must conform to the style and formatting guidelines given for this course. The following is a list of the guidelines that are required for the lab to be submitted this week.

- The file header and function header for your main function must be present, and filled out correctly.
- You must indent your code correctly and have no embedded tabs in your source code. (Don't forget about the Visual Studio Format Selection command).
- You must not have any statements that are hacks in order to keep your terminal from closing when your program exits.
- You must have a single space before and after each binary operator.
- You must have a single blank line after the end of your declaration of variables at the top of a function, before the first code statement.

Failure to conform to any of these formatting and programming practice guidelines for this lab will result in a grade of 0 for the lab, and your program being returned with an indication of which of these items your program violates. Failure to follow other class/textbook programming guidelines may result in a loss of points, especially for those good programming practices given in chapters 1-5 of our textbook which you should have read by now.