San José State University
Department of Computer Engineering

CMPE 180A
Data Structures and Algorithms in C++
Spring 2018

Instructor: Ron Mak

Assignment #1

Assigned:	Thursday, January 25
Due:	Thursday, February 1 at 5:30 PM
CodeCheck:	http://codecheck.it/codecheck/files/18012520543wfclc1nwxr92426aaisknqm6
Canvas:	Assignment 1. Pi Formulas
Points:	100

Pi formulas

This assignment will give you practice using control statements, arithmetic expressions, and the built-in math functions to implement some programming logic. Your program will compute the value of pi (π) using several formulas and an infinite series.

At the above CodeCheck URL, complete the program **PiFormulas.cpp** in CodeCheck's edit box, and then press the "Submit" button. CodeCheck will compile and run your program and compare your output to a master report. You can type into CodeCheck's edit box directly, or you can first edit and test your program in the Eclipse or NetBeans IDE, and then cut and paste it into CodeCheck.

Choose descriptive variable names. Include meaningful comments in your code, but don't over-comment. Include your name in a comment at the top of your program. Follow the coding style (formatting, braces, indentation, etc.) of the Savitch textbook.

The course syllabus describes how to set up your programming environment.

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.

Ramanujan's formulas

Srinivasa Ramanujan (1887-1920) was one of India's greatest mathematical geniuses. In 1913, when he was but a poor clerk living in Madras, he wrote to the famous mathematician G.H. Hardy at Cambridge University in England. Hardy was so impressed by the theorems included in the letter that he had Ramanujan brought to the university. Until he became seriously ill and returned to India in 1919, Ramanujan made major contributions to number theory, elliptic functions, continued fractions, and infinite series.

In 1914, Ramanujan published several remarkable formulas that gave approximate values for π that were accurate to 15 to 31 decimal places:

Decimal places	Ramanujan's Formulas for π
15	$\pi = \frac{12}{\sqrt{130}} \ln\left\{\frac{\left(2+\sqrt{5}\right)\left(3+\sqrt{13}\right)}{\sqrt{2}}\right\}$
16	$\pi = \frac{24}{\sqrt{142}} \ln\left\{\sqrt{\frac{10+11\sqrt{2}}{4}} + \sqrt{\frac{10+7\sqrt{2}}{4}}\right\}$
18	$\pi = \frac{12}{\sqrt{190}} \ln\left\{\left(2\sqrt{2}+\sqrt{10}\right)\left(3+\sqrt{10}\right)\right\}$
22	$\pi = \frac{12}{\sqrt{310}} ln\left\{\frac{(3+\sqrt{5})(2+\sqrt{2})\left[\left(5+2\sqrt{10}\right)+\sqrt{61+20\sqrt{10}}\right]}{4}\right\}$
31	$\pi = \frac{4}{\sqrt{522}} \ln\left\{\left(\frac{5+\sqrt{29}}{\sqrt{2}}\right)^3 \left(5\sqrt{29}+11\sqrt{6}\right)\left(\sqrt{\frac{9+3\sqrt{6}}{4}} + \sqrt{\frac{5+3\sqrt{6}}{4}}\right)^6\right\}$

Code these formulas in C++ using the **`double`** data type, which will give you 15 digits after the decimal point. As a comparison, also calculate π using the built-in arctan function: **`4*arctan(1)`**. To use the built-in math functions in a program, you need to **`#include <cmath>`**.

Euler's convergence towards π

Famous Swiss mathematician Leonhard Euler (1707-1783) devised this infinite series that converges towards $\frac{\pi^2}{6}$

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots$$

from which you can compute π.

Compute this sum using the **`float`** data type, which will give you 7 digits after the decimal point. Continue adding terms as long as each term is > 10^{-8}. (Why is that a good time to stop?) Remember how many iterations it took. Compare the sum that you compute with the value of $\frac{\pi^2}{6}$ computed directly using the **`4*arctan(1)`** value of π.

To compute the squares of successive integers, use the following:

$$1^2 = 1$$

$$2^2 = 4 = 1 + 3$$

$$3^2 = 9 = 1 + 3 + 5$$

$$4^2 = 16 = 1 + 3 + 5 + 7$$

etc.

In other words, to compute the square of the next integer, add the next odd number to the square of the previous integer. (Later in the semester, we will prove this relationship between squares and odd numbers.)

Expected output

```
pi   = 3.141592653589793 = 4*arctan(1)

Ramanujan's pi formulas:
pi15 = 3.141592653589793
pi16 = 3.141592653589794
pi18 = 3.141592653589794
pi22 = 3.141592653589794
pi31 = 3.141592653589793

Euler's infinite sum for pi*pi/6 = 1.6449341
                     Converged to 1.6447253 after 10001 iterations

Euler's estimate for pi = 3.1413934
                  error = 0.0001992305
```

Submission into Canvas

When you're satisfied with your program in CodeCheck, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution. Submit this zip file into Canvas. You can submit as many times as you want until the deadline, and the number of submissions will not affect your score. Only your last submission will be graded.

CodeCheck expects an exact match of the output. Your program will not lose points if a value is slightly off in the last decimal place, or if your iteration count is different by a small amount.

Submit the signed zip file from CodeCheck into Canvas: **Assignment 1. Pi Formulas**.

Note: You must submit the signed zip file that you download from CodeCheck, or your submission will not be graded. Do not rename the zip file.

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Correct program output (as determined by CodeCheck) • Correct output values. • Correct output format.	**45** • 30 • 15
Correct program design • Correct use of arithmetic expressions and math functions. • Correct use of data types. • Correct use of odd numbers to compute squares. • Correct use of control statements.	**40** • 10 • 10 • 10 • 10
Good program style • Descriptive variable names. • Meaningful comments. • Follows the coding style (formatting, braces, indentation, etc.) of the Savitch textbook.	**15** • 5 • 5 • 5