University of Wyoming
COSC 1030
Computer Science I

Program 05
Numerical Approximation of the Value of π

1 Purpose

This program introduces the need for design of a simple algorithm which can perhaps be best completed with a set of auxiliary functions. The design and implementation will provide the student with practice in function declaration, definition and use. The required computations also demand care in data type manipulation.

2 Procedure

1. You need to plan the program first. So create a file containing pseudocode for this program.
2. Your task is to write a C++ program which computes the approximation of π by a series summation known as the Madhava-Leibniz series. A concise overview is provided on the following web link:

   http://en.wikipedia.org/wiki/Leibniz_formula_for_pi

   The formula is given as follows:

$$\pi = 4 \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

3. Your program should prompt the user for one integer, the largest value of the index k in the truncated (no computer program really goes to infinity) summation of the formula.
   (a) NOTE: your program should demand that the value input be non-negative. More specifically, it should continue to prompt for a value until an integer equal to or greater than 0 is entered.
   (b) The result should be displayed to at least 10 digits to the right of the decimal point.
   (c) Your program will **NOT** use **any** specialized mathematical library functions such as pow() or atan().
4. Upload your solution to the WyoCourses site.
5. Your solution should consist of three files (NOTE: no .exe file!!!!).
   (a) Use the comment heading format from the first assignment. Lack of this in any of the submitted files will result in a deduction.

(b) A text file, **Prog05Pcode.txt**, which contains a pseudo-code 'plan' for your program. Really, write this BEFORE you program and try to be complete.

(c) A single C++ source code file, **Prog05.cpp**

(d) A text file, **Prog05Test.txt** which contains demonstration dialog of your programs behavior.

Insert comments in this file regarding the validity of your results (for example, demonstrate that the user inputs are handled as prescribed above, and that as you increase the maximum value of k, the computed result approaches the value of π, which we can find in many resources, to be as follows (at least to 15 digits of precision)...)

3.141592653589793

An example of the test file for this programming assignment would be as follows: (See Next Page)

```
// Prog05Test.txt
// Kim Buckner
// COSC 1030
// Program 05
// Sep 12, 2021

V:\TMP> Prog05
// KB: This term is easy to check.
Computing pi Series Summation by ML Formula
============++++++++++====================
Enter maximum value of k in truncated series (non-negative): 0
Approximation of pi is 4.000000000000000

V:\TMP> Prog05
// KB: disallows negative inputs.
Computing pi Series Summation by ML Formula
============++++++++++====================
Enter maximum value of k in truncated series (non-negative): -9
Enter maximum value of k in truncated series (non-negative): -1
Enter maximum value of k in truncated series (non-negative): 1
Approximation of pi is 2.666666666666667

V:\TMP> Prog05
Computing pi Series Summation by ML Formula
============++++++++++====================
Enter maximum value of k in truncated series (non-negative): 50
Approximation of pi is 3.161198612987051

V:\TMP> Prog05
Computing pi Series Summation by ML Formula
============++++++++++====================
Enter maximum value of k in truncated series (non-negative): 100
Approximation of pi is 3.151493401070991

V:\TMP> Prog05
// KB: slowly approaching pi!
// KB: I am sure theres a better series!
Computing pi Series Summation by ML Formula
============++++++++++====================
Enter maximum value of k in truncated series (non-negative): 10000
Approximation of pi is 3.141692643590535

V:\TMP>
```