CIS210 – Program 2 Fall 2015 Due: _____

If's and Loops Triangles (60 Points) Root Finder (40 Points) 100 Points Total

Part A – Newton's Method for Finding Roots of an Equation 40 Points

Newton's Method of finding roots of equations is a standard approximation way of doing so, although it needs a derivative of the original equation. Assuming your equation is "well behaved", this method quickly converges to the correct root. The term "well behaved" means the slopes of the curve are not too steep, nor are there any dis-continuous parts in the equation. The equation $y = \sqrt[3]{x}$ is not "well behaved" (see notes from class). Newton's Method equation is:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad \text{until } |f(x_{n+1})| < \text{Tolerance}$$

where Tolerance is how close the function is to zero at $x_{n+1}$ in order to say we are done, $x_n$ is the current approximation to the desired root, and $x_{n+1}$ is the next approximation and hopefully closer to the desired root. Using your equation given below based on the last digit of your jag number, write a complete program called <lastname_new.cpp> to find the desired root using a starting value of 0.5 and a Tolerance of $10^{-6}$ or 0.000001 or 100 times through the loop (whichever condition occurs first) and then include your results in your comments including the number of times it took to find the desired root using the loop. Note: This program does need a loop.

**Equations to use:**

| Last Digit | Equation |
|---|---|
| 0, 2, 4 | $f(x) = \arccos(x) - \cos(x)$ |
| 1, 3, 5, 7 | $f(x) = \arccos(x) - 1/\cos(x)$ |
| 6, 8, 9 | $f(x) = \arcsin(x) - 1/\sin(x)$ |

Extra Credit Part for Newton's Method – 10 Points

Change the Tolerance to $10^{-7}$ and the loop counter to 200, find the number of times **MORE** from the previous tolerance and include that difference in your comments for extra credit.

Expected Comments for each program

1. Author of Program
2. Date Started
3. Date Completed
4. Purpose of program – in your own words
5. Pseudo-code for program
6. Libraries used including functions/constants used and briefly why
7. Variables and constants created by you with type and brief purpose for each one

Item to turn-in for grading

Submit your source code to me via Sakai's assignment dropbox

145



Part B – Triangles Again 60 Points

Write a complete program called <lastname_tri.cpp> to calculate and output the perimeter, the area, the angles of a triangle in degrees, the sum of the three angles in degrees, the radius of the inscribed circle, the radius of the circumscribed circle, and the positive difference in radii using a suitable format for each calculated value by inputting lower_a, upper_a, b, and c and having a loop for side a starting at lower_a and running to upper_a inclusive and an if statement to catch non-triangles (I suggest you look at the area calculation) and output a statement about having a non-triangle for these three sides (and not doing any more calculations) or output all of the results for valid ones.

Equations:

$$\text{perimeter} = a + b + c$$

$$s = (a + b + c)/2$$

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{AngleA} = \arccos((b^2 + c^2 - a^2)/(2bc))$$

$$\text{AngleB} = \arccos((a^2 + c^2 - b^2)/(2ac))$$

$$\text{AngleC} = \arccos((a^2 + b^2 - c^2)/(2ab))$$

$$\text{Inner Radius} = \text{area}/s$$

$$\text{Outer Radius} = abc/(4 \times \text{area})$$

Part B – Triangles Again – EC Input : 10 Points Output : 10 Points

In place of having an user input the values for the program stated above, use an input file instead. In place of having the output to the console window, output the results to a text file. Using an input file is EC 10 points. Using an output file is EC 10 points. If you do this program, it takes the place of the regular Part B and state that in your comments. 2 backslash

Expected Comments for each program

1. Author of Program
2. Date Started
3. Date Completed
4. Purpose of program – in your own words
5. Pseudo-code for program
6. Libraries used including functions/constants used and briefly why
7. Variables and constants created by you with type and brief purpose for each one

Item to turn-in for grading

Submit your source code to me via Sakai's assignment dropbox