Lab2: Problem Solving Using Visual C++

Objective

The objective of this lab is to teach you to solve engineering problems using C++. We will focus on these issues:

- Understand the requirements of a problem and the algorithm/pseudo code solution
- Implement the solution using C++
- Debug and remove errors
- Test and verify the correctness of the results

Description

For this lab, you will solve two (2) problems:

1. Compute the area and circumference of a circle (25%)
2. Compute the sine and cosine of angles using the Taylor series (75%)

For each solution you will develop an algorithm in pseudo code, then develop and execute a C++ program using the VC++ environment.

Evaluation

There are two (2) parts to this lab:

1. Circle Program (25%)
2. Taylor Program (75%)

Visual C++ Environment

A **project** in the VC++ environment is a program. A **workspace** is a folder in which all project-related information is stored. When you create a project you may create it in a new workspace or add it to an existing workspace. Refer to the handout “Working with VC++ programs” for more details. Do the following:

1. Create a workspace called Lab2.
2. Create a project for each program and add it to the Lab2 workspace. Create project names as follows:
    a. Circle
    b. Taylor

Template for Program Header

Place the following code at the beginning of every source code file that you submit for this class.

```
/************************************************************
* NAME: your name                                           *
* PERSON NUMBER: your person number                         *
* PROGRAM:  Lab name                                        *
* PURPOSE: 1-2 line summary of the purpose of the lab       *
* DATE: Date of last update                                 *
* PLATFORM: Microsoft Visual C++ 6.0 Pro                    *
* Course & Section:                                         *
************************************************************/
```

On-line submission of your code

All source code (.cpp) will need to be submitted using the on-line command which will be given to you during the lecture and lab. The procedure for submission is as follows:

From your unix command-prompt, do the following:

- change directories to where your files are located
- run the following command for each file
  - `submit_eas230 <file>`
    - where `<file>` is the name of the file you need to submit.

Program 1: Circle 25%

This program will calculate the area and circumference of a circle given the radius. The radius will be input from the keyboard and validated (i.e., ensure that the radius is >= 0), and the area and circumference will be calculated with 2-decimal places accuracy and output.

Evaluation

This program is worth 25% of the grade. The evaluation is as follows:

1.	Define PI	5%
2.	Input Statements	5%
3.	Calculation Statements	5%
4.	Output Statements	5%
5.	Comments	5%

Algorithm

Use a constant for PI = 3.14

1. *Input radius of the circle.*
2. *Compute and print out area of the circle.*
3. *Compute and print out circumference of the circle.*
4. *Exit (return).*

Code

Create a new project called **Circle** and add it to the workspace **Lab2**. Create a C++ source file named **Circle.cpp**.

Submission

Submit the code for Circle.cpp.

Program 2: Taylor Series Program 75%

This program is slightly more complex than the Circle program as it calculates the sine and cosine of an angle using both a Taylor series approximation and the C Math library functions sin() and cos(). It calculates the percent error between the Taylor approximated and CMath calculated values and displays it to the user. This part will be conducted in three steps:

Step 1: Taylor Series Approximation
We can compute the sine and cosine from first principles using the Taylor series.
The Taylor series for computing sine and cosine are given below:

$$\sin x = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + (x^9/9!) \ldots..$$

$$\cos x = 1 - (x^2/2!) + (x^4/4!) - (x^6/6!) + (x^8/8!) \ldots..$$

Since we have not studied loop control structures, we will use an approximation of the above series limited by the number of terms as shown below to compute sine and cosine.

$$\sin x = x - (x^3/3!) + (x^5/5!)$$

$$\cos x = 1 - (x^2/2!) + (x^4/4!)$$

Design and implement a C++ program to determine sine and cosine of x using the formulae given above.

You may use <cmath> library for power (pow) function. Symbol ! represents factorial and n! is given by the expression: n (n-1)*(n-2)….*

Do not use <cmath> library for sin and cos.

Step 2: C Math Library Routines
An alternative way to compute sine and cosine is by using built-in math library available with many programming languages. Now you may use <cmath> library. Add statements to the program developed in Step 1 to determine sine and cosine of x. Use the same test values for x as given in Step 1.

Compile and execute the program. Compare the results from Step1 and Step2 for the same values of X. What do you observe? Record your observation as comments after the code for Step2.

Step 3: **Both Combined**
Let's quantify and formalize our observation in Step2. Obviously, sine and cosine computed using the C++ math library is more accurate than our limited approximate Taylor series. Determine the percent error by using value computed in Step 2 as actual and value computed in Step 1 as estimate for each input value of x given in Step1. Print out x values, sine and cosine values x, and the respective percent error, formatted appropriately and accompanied by suitable messages.

For all three steps, run the program with the following data sets:

1. angle = 7 degrees
2. angle = 27 degrees
3. angle = 87 degrees

Evaluation

This program is worth 75% of the grade. The evaluation is as follows:

1.	Data Entry	5 %
2.	Data Conversion	5 %
3.	Taylor Approximation	40 %
4.	CMath Calculation	10 %
5.	% Error calculation	10 %
6.	Comments	5 %

Algorithm

Use a constant for PI = 3.14

1. *Input angle x in degrees*
2. *Convert angle from degrees to radians*
3. *Compute sin (x) using approximate Taylor series formula.*
4. *Print it out.*
5. *Compute cos (x) using approximate Taylor series formula.*
6. *Print it out.*
7. *Compute sin (x) and cos (x) using cmath library functions and print the values.*
8. *Compute the percent error incurred by the approximation in Taylor series.*
9. *Print it out.*
10. *Exit (return).*

Code

Create a new project called **Taylor** and add it to the workspace **Lab2**. Create a C++ source file named **Taylor.cpp**.

Submission

Submit the code for Taylor.cpp, step 3.