CS 103 Lab 2 - C/C++ Expressions and Library Functions

1 Introduction
In this lab you will gain experience writing C/C++ code that uses arithmetic expressions and makes use the built-in C++ library functions.

2 What you will learn
After completing this lab you should be able:
- Use cin/cout to perform input from the user and output to the user
- Write arithmetic expressions in C/C++
- Use of the <cmath> library functions
- Simple for loops
- Use of the modulus operator

3 Background Information and Notes
None

4 Procedure
Download the following code samples by running the command:

```bash
$ cd ~
$ cd examples
$ wget http://ee.usc.edu/~redekopp/cs103/lab2.tar
```

Un-tar the files and list the new directory contents. You should see some .cpp files.

```bash
$ tar xvf lab2.tar
$ ls
```

This should bring down the following partially completed files into your examples directory:

- color_conv.cpp
- compute_sin.cpp

4.1 [2 pts.] Learning Item 1
Learn expression syntax, order of operations/precedence, variable declarations, and basic I/O.

Color representation for electronics, print, and other media can vary. Two of the most popular systems used are RGB and CMYK. RGB represents a color with 3 components (Red, Green, and Blue) that each range from 0-255. CMYK represents a color with 4 components (Cyan, Magenta, Yellow, and Black) that are each real-values between 0.0 and 1.0.
Write a program that converts from RGB format to CMYK using the following conversion method:

\[
\text{Define } \text{white} = \max \left( \frac{\text{red}}{255}, \frac{\text{green}}{255}, \frac{\text{blue}}{255} \right)
\]

\[
cyan = \frac{(\text{white} - \text{red})}{\text{white}}
\]

\[
magenta = \frac{(\text{white} - \text{green})}{\text{white}}
\]

\[
yellow = \frac{(\text{white} - \text{blue})}{\text{white}}
\]

\[
black = 1 - \text{white}
\]

Your program should take 3 integers, (red, green, and blue) as input from the keyboard and output the corresponding values for cyan, magenta, yellow, and black to the monitor. If you #include<algorithm> at the top of your program, C++ will provide you a function to compute the maximum of two numbers. You can use it as shown in the example below. Note: It can only compare (find the max of) two numbers at a time:

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

int main()
{
    double x = 4.5, y = 6.3, z;
    z = max(x, y);  // z will be assigned 6.3
    // since it is the maximum of x and y
    cout << z << endl;
    return 0;
}
```

Use gedit to edit your code in the provided colorconv.cpp. When you believe you have it working, compile your program by typing:

```
$ make color_conv
```

Then run the program.

```
$ ./color_conv
```

**Test Cases:** The following test cases should help you validate if your program is working.
For input \((r, g, b)\)
\[
20 \ 80 \ 60
\]
the correct output should be
\[
\begin{align*}
\text{cyan: } & 0.75 \\
\text{magenta: } & 0 \\
\text{yellow: } & 0.25 \\
\text{black: } & 0.686275
\end{align*}
\]

For input \((r, g, b)\)
\[
103 \ 0 \ 0
\]
the correct output should be
\[
\begin{align*}
\text{cyan: } & 0 \\
\text{magenta: } & 1 \\
\text{yellow: } & 1 \\
\text{black: } & 0.596078
\end{align*}
\]

For input \((r, g, b)\)
\[
255 \ 255 \ 255
\]
the correct output should be
\[
\begin{align*}
\text{cyan: } & 0 \\
\text{magenta: } & 0 \\
\text{yellow: } & 0 \\
\text{black: } & 0
\end{align*}
\]

For input \((r, g, b)\)
\[
12 \ 120 \ 201
\]
the correct output should be
\[
\begin{align*}
\text{cyan: } & 0.940299 \\
\text{magenta: } & 0.402985 \\
\text{yellow: } & 0 \\
\text{black: } & 0.211765
\end{align*}
\]

Adding the text to the output format is optional, it is okay if you just print out 4 separate numbers. It is also okay if you give any prompts to the user.

**Demo your program and show your code to your TA/sherpa.**

### 4.2 [3 pts.] Learning Item 2
Learn expression syntax, order of operations/precedence, variable declarations, and basic I/O.

Open the file `compute_sin.cpp` Modify the code to create a program to prompt the user for an input number in radians (say, \(x\)) and output an approximation of \(\sin(x)\). \(\sin(x)\) can be approximated use a series expansion of the form:

\[
\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \ldots
\]

Your program should take \(x\) and compute the result from the first 4 terms of the \(\sin(x)\) series expansion. To compute the factorial, we have written a factorial function for you. Simply “call” it by typing `fact(n)` where \(n\) is replaced with the
desired value. Use parentheses where necessary. Also to compute powers do not simply multiply x by itself. Instead use the pow() function that is part of <cmath> (a.k.a. <math.h>, which was included at the top. [Look up the pow() function for what inputs it expects and what it returns on the web or type man pow at the Linux command prompt for more info]. Display the output to the user. You do NOT have to use a loop.

Record the value you computed for x = 1.57? ___________________

Demo your program and show your code to your TA/sherpa.

4.3 [3 pts.] Learning Item 3
Think, code, test.

Write a program to perform the following operations. Prompt the user for an integer. We will assume the integer that the user enters is between [0 and 999]. (You don’t need to write any code to check the input.) Write code to take the single decimal value and find the 1’s digit, 10’s digit, and 100’s digit (in that order) and store them in three separate variables. If they enter a 1 or 2-digit decimal value you should produce 0 for the leading digits. Then output the result on three separate lines to the user.

Sample program execution:
Enter an integer between 0 and 999: 247
1’s digit is 7
10’s digit is 4
100’s digit is: 2

[Hint: Think how you can use integer division and the modulus operator to help you.]

Demo your program and show your code and recorded value to your TA/sherpa.

5 Review & Reflection
[2 pts.] Take a moment and reflect on the previous learning item (Item 3). Suppose we wanted to achieve the same task of finding the separate digits of any number (i.e. 3-, 4-, 5-, etc. digits) in ascending order (from 1's to 10's to ...) using a while loop? What would you do in each body/iteration? What would the condition be for repeating (i.e. the while condition)? [Re-write your program if you like using this approach].

Show your answers or re-written code to your Coach/TA