Program #1

Due Oct 1 by 11:59pm **Points** 100

Begin Programming in C++

Introduction

Every assignment in this course is graded by demoing your work for 10 minutes with a TA. You are required to **meet with a TA within two weeks of the due date** to demo. You can schedule a demo with a TA from the TA Office Hours tab in Canvas. The available times can be viewed in the far right column of the table (labeled "Grading Hours").

- **Demo Outside 2 Weeks**: Assignments that are not demo'd within the acceptable time period will be subject to a 50 point deduction.
- **Demo Late Assignments**: Late assignments must still be demoed within the two week demo period beginning from the assignment's original due date.
- **Missing a Demo**: If you miss your demo with a TA, you will receive a 10 point (one letter grade) deduction to that assignment for each demo missed.

Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program. ☺

Each program needs to be written according to the style guidelines (https://web.engr.oregonstate.edu/~goinsj/resources/general/cpp_style_guideline.pdf) for this class. Remember that an important part of computer programming is making sure that your work is easily understandable by other programmers.

We will spend the first lab session learning how to compile a C++ program. It may be helpful to wait until after you have completed Lab 1 before attempting this assignment.

(90 pts) Problem Statement

(20 pts) Write a C++ program that first prints the maximum and minimum values for the signed and unsigned ints, longs, and shorts using the macros from the climits library, http://www.cplusplus.com/reference/climits (http://www.cplusplus.com/reference/climits) . Notice there is not a minimum unsigned! You should just enter the numerical literal for this information, and make sure you print proper messages for each min and max you are printing.

Clarification (9/26/18): The example code shown in the lecture notes used the <limits (http://www.cplusplus.com/reference/limits) > library (rather than the <climits

(http://www.cplusplus.com/reference/climits) > library). You are welcome to use either of these libraries to complete the assignment.

(62 pts) Now, write a program to calculate and print the maximum and minimum signed and maximum and minimum (which of course is a literal number, that you still need to assign!) unsigned number stored in n bytes, and store the result in a variable to print.

Clarification (9/26/18): n is a variable that will contain an integer provided by the user at the terminal. Your code should prompt the user for a value and store the input in a variable. You do not have to handle the user entering a number of bytes greater than 8. So the user could enter a number between 0 and 8.

Think about the equation from class and recitation to determine how many numbers can be represented in 8 bits with two possible choices for each bit. How are you going to express an exponent? In C++, you need to use a built-in function, pow(base, exp), from the cmath library. For example:

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
   int num = pow(2,3);
   cout << “2^3 is: ” << num << endl;
   return 0;
}
```

Hint (9/28/18): Please see the announcement that I posted regarding 8 byte max/min values.

(8 pts) Now, insert statements to add 1 to the unsigned and signed maximum number stored in the variable after you calculate the answer, and print the result of increasing the variable by one. Do the same for the unsigned and signed minimum number calculated by subtracting one from the variable and printing the value afterward.

(10 pts) Extra Credit

Now, make sure you can handle and print error messages for numbers <= 0 and > 8 bytes from the user.

(10 pts) Program Style/Comments

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the **style guidelines for this class (https://web.engr.oregonstate.edu/~goinsj/resources/general/cpp_style_guideline.pdf)** , and begin trying to follow them, i.e. don’t align everything on the left or put everything on one line!

```
/*******************************************************
** Program: numbers.cpp
** Author: Your Name
** Date: 09/24/2018
** Description:
** Input:
** Output:
******************************************************/
```

Electronically submit your C++ programs (up to two .cpp files, not your executables!!!) by the assignment due date, using **TEACH** **(https://engineering.oregonstate.edu/teach)** .

Program #1 Rubric

Criteria	Ratings	Pts
Macros printed 9 macros printed (2 pts each) + 2 free pts		20.0 pts
Read bytes (or bits) from user		10.0 pts
Program Header/Indentation Programs Header, 5 pts Indentation, 5 pts		10.0 pts
Extra Credit (10pts) Print error message with numbers <=0 and >8.		0.0 pts
(10 pts) min/max signed and max unsigned calculated for n bytes (or bits) (15 pts each) Print output (7 pts) Common errors: Calculation for signed max doesn't -1 for starting at 0, -5 Calculation for unsigned max doesn't -1 for starting at 0, -5 Don't cast pow for signed max to long or unsigned long to make whole number, -5 Don't cast pow for signed min to long or unsigned long to make whole number, -5 Don't cast pow for unsigned max to unsigned long to make whole number, -5 It is okay if the calculations for 8 bytes are hardcoded (manually set). For 0-7 byte inputs, subtract 10 points if the values are hardcoded.		52.0 pts
Additional required expressions Add one to signed and unsigned max values (4 pts) Subtract one from unsigned and signed min values (4 pts) For 8 bytes, max unsigned + 1 should rollover to 0. (-2 if broken)		8.0 pts
		Total Points: 100.0