Assignment #1 – Begin Programming in C++
Due: Sunday, 10/12/14, 11:59pm

Grading: For each programming assignment, you are graded by explaining and demoing your code to a TA. You must demo your program within 7 days of turning it in, and if you fail to do so, you will automatically lose 50 points! **Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program**☺ Your grade is based on Polya’s 4 steps for solving problems:
  • Understanding the problem. (*Recognizing what is asked.*)
  • Devising a plan. (*Responding to what is asked.*)
  • Carrying out the plan. (*Developing the result of the response.*)
  • Looking back. (*Checking. What does the result tell me? Did I do it right?*)

(15 pts) Polya’s steps 1, 2, and 4 do not directly deal with you C++ code itself, but rather, the steps you took to write a correct program that solves the given problem statement. With this said, make yourself familiar with the three sections below, and make sure you can provide documentation and talk intelligently about what you did to arrive at the solution you did. **You will have to explain how you went through these steps to your TA, as part of your assignment grade!!!**

(2.5 pts) Understanding the Problem
In your own words, explain what YOU think the problem is asking you to do. In this section, document your uncertainties about the problem and anything else that you feel was unclear or vague. This is to ensure that YOUR understanding matches MY understanding of the problem☺

(2.5 pts) Devising a Plan/Design
At a minimum, provide an algorithm/pseudo code you designed to help solve the problem. In addition, include pictures/flow charts you used to help you devise your plan, as well as any other design decisions you made such as how to manage your time, how to decompose the problem, where to start first, etc. You can scan any handwritten work and attach it to the document as needed.

(10 pts) Looking Back/Self-Reflection
Report any checking/self-reflection you did while solving the problem. For instance, how did you make sense of the output from the implementation? This includes things such as using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using external sources of information such as the internet to make sense of the results, etc. Also, include a statement about what you learned from the assignment. **Provide us the test plan you used!**
(65 pts) **Problem Statement:** Write a C++ program that first prints the maximum and minimum values for the signed and unsigned numbers using the macros from the climits library, [http://www.cplusplus.com/reference/climits/](http://www.cplusplus.com/reference/climits/)

Now, create variables for the largest and smallest `short`, `int`, and `long`, both `signed` and `unsigned`. Compute the largest and smallest number using `pow()`, which is in the `cmath` library, and `sizeof()` for the specific type and assign it to its corresponding variable. (You do not need to use `pow` to compute the minimum unsigned number, as that is just zero!). The `sizeof()` function returns the number of bytes used by the type supplied to the function. You can use this function by passing the type as an argument, i.e. `sizeof(float)`, `sizeof(int)`, etc. You only need to print the decimal values for these numbers. **Make sure these numbers match the numbers printed from the macros!!!**

In this same program, **prompt the user for 5 test scores.** Calculate the test average and display it to the user. Be sure to think about the types of variables you need for this. The number displayed is in fixed-point notation with one decimal point of precision. Lastly, display the average as a **rounded whole number in decimal, binary, hexadecimal, and octal** number.

(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, and begin trying to follow them, i.e. don’t align everything on the left or put everything on one line! [http://classes.engr.oregonstate.edu/eecs/fall2014/cs161-001/161_style_guideline.pdf](http://classes.engr.oregonstate.edu/eecs/fall2014/cs161-001/161_style_guideline.pdf)

/**********************************************************************************
** Program: numbers.cpp
** Author: Your Name
** Date: 10/10/2014
** Description:
** Input:
** Output:
**********************************************************************************/

(10 pts) **Design for Assignment #2**
**Design a solution** for the following problem statement in Assignment #2. In addition, you will also **provide a test plan** for how you will make sure your program is working correctly. There is design in both the solution and testing your solution.

**Problem Statement:** Expand the problem from Assignment #1 to continue to ask the user for `n` test scores, rather than a set number of 5 test scores. **What are the**
different ways you can think of doing this? Which design are you picking and why?

In addition, these test scores should range from 0 to 100, and your program needs to check that the scores supplied are valid numbers before moving forward. This may include making sure the user doesn’t enter a letter or string of letters. **How are you going to make sure the user entered a number in the right range?** What are the various ways you can think of making sure the user enters a number, rather than letters?

Electronically submit your C++ program (.cpp file, not your executable!!) and design document, **as a pdf**, by the assignment due date, using TEACH.

**NOTE:** The easiest way to upload your program from ENGR to TEACH is to map a network drive to your home directory on ENGR. Mac or Windows, See: http://engineering.oregonstate.edu/computing/fileaccess/