**CS 2060 Programming with C - Fall 2017**

**Assignment #7**

Due Date: Oct 25, 2017 at 9:25am (MW class), Oct 26, 2017 at 9:25am (TR class)

Purpose: Understand pointers

Effort: Individual

Points: **50**

Deliverables: Upload the .c source code file to Blackboard by due date.

Please hand in a hardcopy version of your code at beginning of class.

**NEW: Please hand in a handwritten version of your memory map!**

**Assignment Description**

One way to learn about pointers is to declare and initialize pointers and then print their addresses as well as the memory addresses they store. Basically, print out a memory map. For this assignment you will declare some primitive variables, declare pointers to those primitive variables, then display a memory map.

**Specifications**

1. Create a C project called **Assignment7 (please use this exact name)**
2. Follow "CS2060 Programming Assignments Policy"
   1. Pseudocode is **NOT** needed for this assignment
3. Write a simple program to perform the following tasks in main:
   1. Declare **one int, one double, one char**, and **one array of 5 integers**
   2. Assign values to each of these variables
   3. Declare **pointers** to each of these variables
   4. Declare **one pointer** that points to the integer pointer (pointer to a pointer)
   5. Print the header as shown in the output below
   6. Print information for the integer variable as shown in table below
   7. Print information for the pointer to the integer variable as shown in table below
   8. Repeat these steps for the double and char variables
   9. Print information for the integer array variable and each element in the array
   10. Print information for the pointer to the integer array
   11. Print information for each integer array element by accessing the element through the pointer to the integer array (use pointer arithmetic)
   12. Print information for the pointer to the integer value pointer
   13. Finally, print the value stored in the integer value using the
       1. integer value ptr
       2. pointer to integer value ptr (use two levels of indirection)
4. No need to use functions – the purpose is only to create pointers and print values.
5. Using the information in your table, create an additional handwritten version of your memory map like the ones I’ve created in class.
   1. Show each variable
   2. Show address of each variable
   3. Show value stored in each variable
   4. Show where the array values are stored
   5. Draw in lines showing where each pointer is pointing

**Turn in this memory map – please staple your hardcopy.**

**Output**

Your output will look like the following. Your memory addresses will be different. The size of types may be different.

**Output - Example #1**

----------------------------------------------------------------------------

| Type | Name | Address | Value | Size | Comment |

----------------------------------------------------------------------------

int intValue 0028FF3C 1234 4

int\* intValuePtr 0028FF14 0028FF3C 4 pointer to integer value

double doubleValue 0028FF30 367.70 8

double\* doubleValuePtr 0028FF10 0028FF30 4 pointer to double value

char charValue 0028FF2F A 1

char\* charValuePtr 0028FF0C 0028FF2F 4 pointer to character value

int intArray 0028FF18 0028FF18 20 array name address of 1st element

int intArray[0] 0028FF18 8 4

int intArray[1] 0028FF1C 1 4

int intArray[2] 0028FF20 7 4

int intArray[3] 0028FF24 2 4

int intArray[4] 0028FF28 4 4

int\* intArrayPtr 0028FF04 00000000 4 pointer set to NULL

int\* intArrayPtr 0028FF04 0028FF18 4 pointer to array

int\* intArrayPtr 0028FF18 8 4 access intArray[0] through pointer

int\* intArrayPtr 0028FF1C 1 4 access intArray[1] through pointer

int\* intArrayPtr 0028FF20 7 4 access intArray[2] through pointer

int\* intArrayPtr 0028FF24 2 4 access intArray[3] through pointer

int\* intArrayPtr 0028FF28 4 4 access intArray[4] through pointer

int\*\* pointerToIntValuePtr 0028FF08 0028FF14 4 pointer to integer value pointer

Accessing value using a pointer then pointer to pointer

-------------------------------------------------------

Accessing intValue though intValuePtr = 1234

Accessing intValue through a pointer to intValuePtr - 2 levels of indirection = 1234