# CS 2275 Exercise 5 Pointers, Static, Global

Module 5 Objectives:

1. Void pointers, shifts, pointer arithmetic, static and global variables.

Readings for Module 5: Lippman - 2.3.23.5.3, 12.2

Using Dev-C++, create a .cpp source file titled ex5-<your last name>.cpp that contains solutions to the following problems. Use a .doc or .txt file ex5<your last name>.doc to provide the answers to the text questions. As with all assignments in this course, be sure to include a block comment just about the function indicating its purpose and use a really good function name. Include code under main to test your function by allowing the user to enter the needed parameters. Make sure you test your function sufficiently to insure correctness

1. What will the following C++ code display in the console window?

#include <iostream>

using namespace std;

int main() {

int z = 11;

int \*pZ = &z;

cout << pZ << endl;

cout << \*pZ << endl;

int y[4];

y[0]=z/8; y[1]=y[0]+5; y[2]=y[0]+7;

cout << \*y << endl;

int \*pY = y;

cout << \*pY << endl;

pY++;

pY++;

cout << \*pY << endl;

return 0;

}

1. What value of x will generate the output 21, 21, 26 in the following code fragment?

int main() {

int x = ??? ;

int y[3];

y[0] = x / 2; y[1] = y[0] + 3; y[2] = y[1] + 2;

cout << \*y << endl;

int \*pY = y;

cout << \*pY << endl;

pY++;

pY++;

cout << \*pY << endl;

return 0;

}

1. Write a function using the following header: int \* count(const string &s) that returns an integer array of length 10 containing the frequency count of each numeric digit(e.g., 0, 1, 2, …, 9) in that string. Thus:

int counts[] = count(“123456306456”); count[0]=0; count[1]=1; count[2]=1; count[3]=2; count[4]=2; count[5]=2; count[6]=3;

1. Write a function count1() that counts how many times it is called. Use a static variable. Then write a second function count2() that uses a global variable to count how many times it is called.
2. Write a function that accepts a string and returns an unsigned long long array containing the asci values of all the characters in that string.

Grading Rubric

20 points each.

Style up to -10: poor variable or poor function names or no block comment with a goal statement for a function. Solutions that have significantly more lines that needed will be docked points for lack of elegance.

**REMINDERS**

* code that does not compile will not be graded
* the grader should not need to modify/uncomment your code to test it. Provide a test mechanism allowing the grader to enter various tests for teach function.