Lab 1

Review C++

Background

This lab is designed to give some more experience with basic constructs, pointers, heap memory and its manipulation using pointers, and memory allocation/deallocation operators.

We are going to be working with int pointers exclusively for this lab, but the memory allocation issues apply to all pointers.

Problem #1:

Write a C function int hamming_weight(int number) which takes as input an integer number > 0 and outputs its hamming weight.

The Hamming weight is the number of bits set in the binary representation of the number. For example:

Binary representation	Hamming weight
000011101	4
111010001	5
000000000	0

The function must:

- Return the number hamming weight if the number entered is greater than or equal to 0.
- Return -1 if number entered is negative.

Your code must go in the Lab_1_-_Problem_1.cpp file, which can be found in your lab1/ directory.

Problem #2:

Complete the three part for analyze pointer problem in the Lab_1_-_Problem_2.cpp file.

Part 1:

Write a function void analyze_pointer(int *ptr) that does two things:

- Write the memory location pointed by the pointer to the console.
- Write the value of the integer (which the pointer points to) to the console.

Part 2:

Use the function to complete two tasks:

- Allocate an int on the stack (e.g., “int iValue;"), assign a value to it, and get its memory location (with the reference operator—&) to pass this value to analyze_pointer.
- Allocate an int on the heap (with the new operator). Assign a value to it, and pass it to analyze_pointer.

Question #1. What happens?

Part 3:

Now, we’re going to add a couple more functions to the mix; call them int_pointer1 and int_pointer2. Both will return int pointers.

- int_pointer1 will allocate an int on the heap (via new int), assign a value to it, and return that value.
- int_pointer2 will allocate an int on the stack (via “int iValue;" or something similar), assign a value to it, and return its memory location (as both of these functions should be of the int pointer type).

Call analyze_pointer on the return of both of these functions from your main function:

- “analyze_pointer(int_pointer1());"
- “analyze_pointer(int_pointer2());"

Question #2. What happens, and why?

Two more tests on memory:

- Remember that int we allocated on the heap for function “int_pointer1()"? Delete it, and call analyze_pointer on it after deleting it.
- Call: analyze_pointer(new int);

Question #3. After completion of the problem, how many memory leaks are there?

Problem #3:

Define a struct Area that has two private variable members; units of type string and area_value of type float. Modify the Lab_1_-_Problem_3.cpp program to create a dynamic variable of type Area.

- ❖ Input from the keyboard the area_value and its units. Compute one-half and one-quarter of the area and display the results
- ❖ Destroy the dynamic variable at the end

Hand in

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS. You should hand in a single compressed/archived file named Lab_1_<your reg. No. XXX without angle brackets>.zip that contains the following.

1. All completed C++ source files representing the work accomplished for this lab: Lab_1_-_Problem_1.cpp, Lab_1_-_Problem_2.cpp, and Lab_1_-_Problem_3.cpp. The files should contain author in the comments at the top.
2. An plain text file named OUTPUT.txt that includes a) author information at the beginning, b) a brief explanation of the lab, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.
2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.
3. In-class lab time is not intended as free time for working on your program assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the program assignment.