CIS 278 (CS1) Programming Methods: C++

Assignment 6: Pointers and Dynamic Memory

Skip to Main Content

Learning Objectives

After the successful completion of this learning unit, you will be able to:

- Implement syntactically correct pointers
- Allocate and deallocate dynamic memory using techniques that will best meet the objectives of effective programming practice.

Assignment 6.1 [25 points]

This assignment will give you a chance to perform some simple tasks with pointers. The instructions below are a sequence of tasks that are only loosely related to each other. Start the assignment by creating a .cpp file with an empty main function, then add statements to accomplish each of the tasks listed below. Some of the tasks will only require a single C++ statement, others will require more than one. Make sure that the tasks appear in your code in the same order that they are listed here.

No documentation is required for this part of the assignment.

1. Create two integer variables named x and y
2. Create an int pointer named p1
3. Store the address of x in p1
4. Use only p1 (not x) to set the value of x to 99
5. Using cout and x (not p1), display the value of x
6. Using cout and the pointer p1 (not x), display the value of x
7. Store the address of y into p1
8. Use only p1 (not y) to set the value of y to -300
9. Create two new variables: an int named temp, and an int pointer named p2. Make p2 point to x.
10. Use only temp, p1, and p2 (not x or y) to swap the values in x and y (this will take a few statements. Don't use a swap function)
11. Write a function with the following signature: void noNegatives(int *x). The function should accept the address of an int variable. If the value of this integer is negative then it should set it to zero
12. Invoke the function twice: once with the address of x as the argument, and once with the address of y. Use x or y for the argument (not p1 or p2)
13. Use p2 to display the values in x and y (this will require both assignment statements and cout statements). You can use x and y in assignment statements, but not in your cout statement. this should produce the output

    ```
    x is: 0
    y is: 99
    ```

14. Create an int array named 'a' with two elements. Make p2 point to the first element of a.
15. Use only p2 and x (not a) to initialize the first element of a with the value of x.
16. Use only p2 and y (not a) to initialize the second element of a with the value of y. Leave p2 pointing to the first element of a. Don't use pointer arithmetic. Hint: don't forget that pointers and arrays are the same thing.
17. Using cout and p2 only, display the address of the first element in a.
18. Using cout and p2 only, display the address of the second element in a. Leave p2 pointing to the first element of a. Don't use pointer arithmetic.
19. Use p1, p2, and temp to swap the values in the two elements of array 'a'. (first point p1 at a[0], then point p2 at a[1], then do not use "a" again. After this the swapping steps should look very similar to

step 10. Don't use a swap function.)
20. Display the values of the two elements. (The first element should be 99, the second 0).
21. Write a function named 'swap' that accepts two pointers to integers as arguments, and then swaps the contents of the two integers. Do not use any reference parameters.
22. Invoke your swap function with the addresses of x and y (using the address-of operator), then print their values. (x should be 99, y should be 0).
23. Invoke your swap function with the address of the two elements in array 'a', then print their values. (a[0] should be 0, a[1] should be 99)

Assignment 6.2 [20 points]

Rewrite your High Scores program so that it uses Dynamic Memory Allocation to create the names and scores arrays.

You will need to make slight modifications to your main function and no modifications anywhere else in your code if you wrote your original program correctly using the function signatures required for that assignment. **Your function signatures for this assignment should be exactly the same as the signatures required in the original High Scores assignment.**

Your modified high scores program should start out by asking the user how many scores will be entered. It should allocate appropriate arrays, and then proceed just like the original High Scores assignment. The output from your program should look approximately like this:

```
How many scores will you enter?: 4
Enter the name for score #1: Suzy
Enter the score for score #1: 9900
Enter the name for score #2: Kim
Enter the score for score #2: 1000000
Enter the name for score #3: Armando
Enter the score for score #3: 822
Enter the name for score #4: Tim
Enter the score for score #4: 514

Top Scorers:
Kim: 1000000
Suzy: 9900
Armando: 822
Tim: 514
```

Submit Your Work

Name your source code files according to the assignment numbers (a1_1.cpp, a4_2.cpp, etc.). Execute the programs and copy/paste the output that is produced by your programs into the bottom of the corresponding source code file, making it into a comment. Use the Assignment Submission link to submit the source files. When you submit your assignment there will be a text field in which you can add a note to me (called a "comment", but don't confuse it with a C++ comment). In this "comments" section of the submission page let me know whether the programs work as required.

Keep in mind that if your code does not compile you will receive a 0.