CS 11 Data Structures and Algorithms

Assignment 1: 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 1.1 [20 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
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
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 1.2 [25 points]

Write a program that records high-score data for a fictitious game. The program will ask the user to enter the number of scores, create two dynamic arrays sized accordingly, ask the user to enter the indicated number of names and scores, and then print the names and scores sorted by score in descending order.

The output from your program should look exactly like this (given the same input):

```
How many scores will you enter?: 4
Enter the name for score #1: Suzy
Enter the score for score #1: 600
Enter the name for score #2: Kim
Enter the score for score #2: 9900
Enter the name for score #3: Armando
Enter the score for score #3: 8000
Enter the name for score #4: Tim
Enter the score for score #4: 514

Top Scorers:
Kim: 9900
Armando: 8000
Suzy: 600
Tim: 514
```

Additional Requirements

The data must be stored in two dynamic arrays: an array of strings named names, and an array of ints named scores. These arrays must be declared and allocated using "new" in the main function.

The user input of the names and scores should be done in a function named initializeArrays(). It should have the following signature:

```
void initializeArrays(string names[], int scores[], int size)
```

You must also write two more functions: one to sort both arrays in descending order by score, and one to display the final list of names and scores. They should have the following signatures.

```
void sortData(string names[], int scores[], int size)
void displayData(const string names[], const int scores[], int size)
```

The main function should be very short. It should just get the number of scores, allocate the two arrays, and then invoke these three functions.

Some of you may not have studied sorting algorithms. Sorting is covered in **lesson 9**, section 9.6. You can copy as much code from that lesson as you like. Do not use C++'s sort() function.

Submit Your Work

Name your source code file(s) according to the assignment number (a1_1.cpp, a4_2.cpp, etc.). Execute each program and copy/paste the output into the bottom of the corresponding source code file, making it into a comment. Use the Assignment Submission link to submit the source file(s). 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 program(s) work as required.