Code | Blame | 84 lines (75 loc) · 2.27 KB | Code 55% faster with GitHub Copilot

```cpp
#include <iostream>

using namespace std;

/**
 * @brief main function will receive input from user, store values, create pointers, and display outputs
 *
 * Program prompts user to input three integer values
 * These values are stored within three separate variables
 * Pointers are created in order to dynamically allocate memory for each variable
 * The values in the variables and pointers are then displayed
 * Lastly, the allocated memory is cleaned up once the program is 'finished'
 *
 * Pseudocode:
 * int main() {
 *      - Declare variables
 *      int num1, num2, num3
 *
 *      - Prompt user for inputs
 *      OUTPUT: 'Enter first variable'
 *      INPUT: num1
 *      OUTPUT: 'Enter second variable'
 *      INPUT: num2
 *      OUTPUT: 'Enter third variable'
 *      INPUT: num3
 *
 *      - Create pointers
 *      int* ptr1 ... num1
 *      int* ptr2 ... num2
 *      int* ptr3 ... num3
 *
 *      - Output values in variables and pointers
 *      OUTPUT: num1 & ptr1
 *      OUTPUT: num2 & ptr2
 *      OUTPUT: num3 & ptr3
 *
 *      - Free dynamically allocated memory
 *      delete ptr1
 *      delete ptr2
 *      delete ptr3
 *
 *      - For safety... set pointers to null post-deletion
 *      ptr1 = null
 *      ptr2 = null
 *      prt3 = null
 *
 *      return 0; - Successful execution
 * }
 */

int main() {
    // Declare variables - storing user inputs
    int num1, num2, num3;

    // Prompt the user for inputs
    cout << "Enter your first integer: ";
    cin >> num1;
    cout << "Enter your second integer: ";
    cin >> num2;
    cout << "Enter your third integer: ";
    cin >> num3;

    // Create pointers to dynamic memory - user 'new' operator
    int* ptr1 = new int(num1);
    int* ptr2 = new int(num2);
    int* ptr3 = new int(num3);

    // Display the contents of the variables AND pointers
    cout << "Variable 1: " << num1 << " Pointer 1: " << *ptr1 << endl;
    cout << "Variable 2: " << num2 << " Pointer 2: " << *ptr2 << endl;
    cout << "Variable 3: " << num3 << " Pointer 3: " << *ptr3 << endl;

    // Use delete operator to free dynamically allocated memory
    delete ptr1;
    delete ptr2;
    delete ptr3;

    // Set the pointers to nullptr, a safety precaution when working with pointers
    ptr1 = nullptr;
    ptr2 = nullptr;
    ptr3 = nullptr;

    return 0;
}
```