

Lab Requirements

1. Create a new C++ console application called PointersLab
2. In your main() function, create two variables. One variable needs to be an intrinsic integer variable named num1 and the second needs to be a pointer to an int pNum.
3. Assign the value 3 to num1.
4. Assign the value 5 to the memory location stored in pNum. (Hint, there is a special operator used for this)
5. Create a function called PassByValue that accepts an integer argument.
6. Inside PassByValue, output a text string indicating the name of the function you are in.
7. Call PassByValue and pass in num1 as the argument.
8. Modify the value of num1 inside the PassByValue function and output the new value to the console window.
9. After the call returns to main(), output the value of num1 to the console window again.
10. Create a function called PassByRef that accepts a reference variable as an argument
11. Call PassByRef and send pNum as the argument.
12. Output a statement indicating the current function name.
13. Modify the value of pNum, setting it to 50, and output the value to the console.
14. After the call returns to main(), output the value of pNum to the console.
15. Call PassByValue and pass in pNum as an argument.
16. Document any error messages for the lab assessment that might occur during your lab.
17. Create a pointer variable to a double, and allocate memory for it using the new keyword.
18. Assign a value to the memory location.
19. Use the dereference operator, get the value and output it to the console.
20. Deallocate the memory for that variable.
21. Output the value stored in the memory location for that variable again.
22. Document any errors experienced for the lab assessment.

- !3. Create a simple class in your C++ application called Person.
- !4. Give Person member variables for name, age, height, and weight. Feel free to make these private with public accessors or make them public without accessors. It is not important for this exercise.
- !5. Create a function in your application called ModifyPerson() that accepts a Person object as an argument. NOTE: This function should be in the application area and not inside the Person class.
- !6. Instantiate a Person object in your main() method and assign values to the member variables.
- !7. Output the values stored in the member variables to the console window
- !8. Call the ModifyPerson() function, passing in your Person object you just created.
- !9. Inside the ModifyPerson() function, change the value in the name member variable.
- !0. Back in main(), output the values stored in the Person object you created. Did the name change? It should have if you passed the object correctly to the function.
- !1. Close Visual Studio and take the lab assessment.