

Pointers

For all the assignments below, integrate Git into your workflow when you create code: commit regularly and make sure the results end up in your online CSD repository in a folder named 'pointers'.

A. Basic pointer interaction

Complete the code given in [basic_pointers.cpp](#) by putting **cout lines** below the assignments. Keep the compiler happy and make sure the program keeps running correctly after each addition.

B. Functions using pointers

Create a C++ program containing a function that performs a small calculation on a given number, e.g. take the square or add a fixed number.

Pass a variable to the function **by value**, **by a pointer** and **by reference**. In all these cases, show the contents of the variable outside and inside the function, change its value inside the function and show the differences and similarities of these approaches. Make clear in which case(s) the function creates a copy and in which case(s) the function operates on the original variable.

C. Pointers to objects

1. Create a simple class containing at least one private variable, a method to give this variable a new value (a.k.a. a 'setter') and a method to get its value (a 'getter').
2. Create several objects of this class
3. Let a pointer point to one of the objects and call the set() and get() methods through the pointer using the -> notation
4. Create an array of 50 pointers for your class type
5. Create 50 objects and let the array cells point to these objects (if you get tired after assigning 10 or more objects, move on to assignment D).
6. Give all objects a value and call their get() functions to show the values

D. Dynamic object creation

Fill the array of assignment C with pointers to **dynamically created** objects of your class and give all objects a value. Show the values of all objects using their get() functions.