# Assignment 2-1-2: C++ Data types, Arrays, Pointers

## Assignment instructions:

* You must use the virtual environment that you set up in Exercise 1-1-3 for this assignment.
* Each program must be in a separate .cpp file. If you aren't sure how to create a new .cpp file, refer to the Create a New File section in the [setup instructions document](https://docs.google.com/document/d/1G4X4kad71xhylR8grJaiZsp8ptxYNh44/edit?usp=sharing&ouid=108230341302163246486&rtpof=true&sd=true). Make sure each file is named as instructed in the question.
* Each program must include appropriate input and output messages.
* Be sure to compile and test each program to be certain it works as expected. If you aren't sure how to compile and run a C++ program, refer to the Build and Execute Program section of the [setup instructions document](https://docs.google.com/document/d/1G4X4kad71xhylR8grJaiZsp8ptxYNh44/edit?usp=sharing&ouid=108230341302163246486&rtpof=true&sd=true).

## Important notes:

* At the top of your .cpp file, please include a comment with your **full name**. If your section uses **Lightweight Teams**, add the names of the teammates whom you worked with to the same comment.
* Add your own **individual comment** for each function / major portion of code that you add, briefly explaining what that part does.
* If you are asked to submit screenshots, your assignment will not be graded unless you submit screenshots. Moreover, if your submitted screenshots do not match with your program's actual behavior, we will consider that to be a **violation of academic integrity** and pursue it accordingly.
* Make sure to **organize and format** your code in a consistent way.
* If you refer to any online resource to understand a concept, see examples of the use of a particular syntax, etc., add a comment **citing** that resource (i.e., specify website name and link).
* You must only submit **.cpp** files. If you have multiple .cpp files, upload them individually and **not** as a zip / compressed file.
* No screenshot(s) will mean no grade for this assignment.

## Assignment objectives:

* The purpose of this activity is to get familiar with or refresh concepts related to C++ data types, arrays and pointers.

## Assignment Resources:

Here is a [C++ visualizer](http://pythontutor.com/cpp.html#mode=edit) that you may find useful as you write programs in C++. It allows you to quickly step through your program and visualize the output at each step.

## Assignment tasks:

1. **(3 points)** Create a new file called **pointerBasics.cpp**. Write a C++ program in this new file. In this new file, write code that will do the following:
   1. (1 point) *Declare and assign values to variables*
      1. Declare an integer variable called *myInt* and assign the value 15 to it.
      2. Declare a pointer variable called *myPointer* using the dereference operator (\*) and assign it to the memory location of *myInt* using the address-of operator (&)
   2. (1 point) *Print to console*
      1. Print the **memory address** (this should be in hexadecimal, i.e., 0x1234 format) of *myInt* (remember the address-of operator!). Then print the **value** contained in the variable *myPointer*.
      2. Print the **value** (this should be an integer) of *myInt* . Then print the **value pointed to** by *myPointer* (Hint: remember the dereference operator!)
   3. (1 point) *Change values and print to console*
      1. ***Reassign the value*** of *myInt* from 15 to 10 using an assignment operator.
      2. Print the **memory address** (this should be in hexadecimal, i.e., 0x1234 format) of *myInt* (remember the address-of operator!). Then print the **value** contained in the variable *myPointer*.
      3. Print the **value** (this should be an integer) of *myInt* . Then print the **value pointed to** by *myPointer* (Hint: remember the dereference operator!)

Take a screenshot of a sample output and upload the picture as part of your assignment submission.

1. Create a new file named **array\_ptrs.cpp** and in this file, write a program that does the following:
   1. Defines an array of integers called *my\_ints* with a pre-specified size of 4.
   2. Populates *my\_ints* with input data provided by the user.
   3. Defines an array of pointers called *my\_ptrs* of the same size as *my\_ints* and initializes the pointers in *my\_ptrs* to point to corresponding elements of the array *my\_ints* (refer to the “Initial state of *my\_ptrs*” in the example shown below)
   4. Sort the contents of the *my\_ptrs* array in ascending order. You are allowed to use any existing sorting algorithm (*two simple options are* [*Bubble Sort*](https://en.wikipedia.org/wiki/Bubble_sort) *and* [*Insertion Sort*](https://en.wikipedia.org/wiki/Insertion_sort)*; you may use either one or choose another one if you prefer*). However, you are **NOT** allowed to use a library implementation of the algorithm. You must implement the algorithm from scratch.
   5. What you have done above is to sort the *values* contained in the *my\_ptrs* array, i.e., you have sorted memory addresses. Now, update your code to sort the *my\_ptrs* array based on the integers that are *being pointed to*, i.e., the first element of *my\_ptrs* should point to the smallest element in *my\_ints*, the second element of *my\_ptrs* should point to the next larger element in *my\_ints*, etc. (refer to the “State of *my\_ptrs* after Step 2e” in the example shown below). The my\_ints array should NOT be changed in any way; the values should stay in the same position. You have already implemented this part of sorting from the steps above, but that was not using pointers, so now you need to sort it using pointers.
   6. Traverse the array *my\_ptrs* and print the values being pointed to by each pointer in this array.

Take a screenshot of a sample output and upload the picture as part of your assignment submission.

| **SANITY CHECK**: As shown in the example below, you are **NOT** supposed to modify or sort the original *my\_ints* array in any way once it is initialized with input data provided by the user. As a sanity check, print out the contents of the *my\_ints* array directly and make sure that it is **unchanged** even after the *my\_ptrs* array has been sorted. |
| --- |

Example:

| Initial state of *my\_ptrs*: | State of *my\_ptrs* after Step 2e: |
| --- | --- |
| Expected output:  *1 7 10 25* | |

**Caution: Before you submit, make sure that you have followed all the instructions under** [**Assignment instructions**](#_urewpavuo7ho) **and** [**Important notes**](#_mo4a08do3nvl) **and that you have taken screenshots as indicated in the assignment.**

## Assignment Submission Items:

The files that need to be submitted for this assignment are the following:

* pointerBasics.cpp
* array\_ptrs.cpp
* The necessary output screenshots for both cpp files.

Note: No screenshot(s) will mean no grade for this assignment.