In this module, we will explore working with pointers in the C++ programming language.

How are pointers used with arrays, and what are the benefits to utilizing pointers?

Pointers are variables that hold memory locations. A pointer variable for an array can be used to point to the first element in that array. Pointers can be used to access array elements, similar to using indices. The example below can demonstrate:

Pointers for an array can also be incremented to point to the next value in the array. If we have an array of integers starting at memory location 1000, incrementing the pointer will make it point to location 1004 as integers are 4 bytes. If I understand correctly, this cannot be used in lieu of an array as we cannot guarantee the next space in memory is available. An array remains a contiguous place in memory.

Pointers can also be help to dynamically size arrays. Dynamically sizing an array can be done using the new key word. A programmer can create a pointer and then make it point to a new array with a different size.

The address-of operator, denoted by an ampersand &, is used to store the address of a variable or function. It is used directly with the variable or function, for example, &x. The dereference operator, denoted by \*, on the other hand is used to store the value of a variable at an address location in another variable. In other words, the dereference operator “points” to the value of a variable while dereference points to the address location. The sample code below can demonstrate the difference.

Are arrays necessary when using a pointer?

Additionally, what is the difference between the address-of operator and the dereference operator in C++?

Be sure to provide an appropriate example to illustrate your viewpoints.