Technical Topic Essay - David Glover:

One of the most widely misunderstood coding concepts, at least in college, is pointers. Pointers are an aspect of languages like C and C++ that allow direct memory access. You declare them the same way you would a variable, but with an asterisk in front of the variable name, and you will usually set them to equal another variable. The reason pointers are so useful is the manipulation of the pointer allows you to access the value stored in the address the pointer points to. This allows for dynamic memory allocation, or for memory to be allocated and deallocated during the program's execution, which can greatly boost the efficiency of a project. Pointers also allow a coder to have ADTs like trees, linked lists, and other Node based structures, as you can have a pointer pointing to the next and the previous nodes.

There are a few common errors when it comes to pointers, the first being initialization errors. Unlike a variable, you cannot simply initialize a pointer without a value and assign the value later, in order to make a pointer point to nothing you have to initialize it as NULL. Another common error is dangling pointers. When finished with the pointer, you have to ensure to free it and to set its value to null to ensure it doesn't cause any memory issues. Finally, the most devastating error that can occur is memory corruption based on incorrect pointer arithmetic, or unintended code that ends up messing up your system.

Pointers also have more advanced applications, like the ability to point to another pointer. When declaring a pointer that points to another pointer you use 2 asterisks instead of one. Applying this principle, you can also have a pointer that points to a pointer which is pointing to a pointer. Pointers are not only variables, you can also have functions that return variables. When working with ADTs, it is common to have functions that will return pointers to either the next or previous nodes, or a function that points to the root of the ADT.

Though challenging, pointers are super useful and helpful for program efficiency and enabling certain functions and concepts, like ADTs in low language levels. While they require caution in order to not corrupt memory, the powerful applications surely outweigh the dangers and help to make programs more advanced and efficient.