Pointers, Arrays, and Pointer Arithmetic

The near equivalence of pointers and array names stems from pointer arithmetic and how C++ handles arrays internally. First, let's check out the arithmetic. Adding 1 to an integer variable increases its value by 1, but adding 1 to a pointer variable increases its value by the number of bytes of the type to which it points. Adding 1 to a pointer to double adds 8 to the numerical value on systems with 8-byte double, whereas adding 1 to a pointer-to-short adds 2 to the pointer value if short is 2 bytes. Listing 4.14 demonstrates this amazing point. It also shows a second important point: C++ interprets the array name as an address.

Listing 4.14 addpntrs.cpp

```cpp
// addpntrs.cpp--pointer addition
#include <iostream>
using namespace std;
int main()
{
    double wages[3] = {10000.0, 20000.0, 30000.0};
    short stacks[3] = {3, 2, 1};

    // Here are two ways to get the address of an array
    double * pw = wages;     // name of an array = address
    short * ps = &stacks[0]; // or use address operator
    // with array element
    cout << "pw = " << pw << "\n", *pw = " << *pw << "\n";
    pw = pw + 1;
    cout << "add 1 to the pw pointer:\n"
    cout << "pw = " << pw << "\n", *pw = " << *pw << "\n";

    cout << "ps = " << ps << "\n", *ps = " << *ps << "\n";
    ps = ps + 1;
    cout << "add 1 to the ps pointer:\n"
    cout << "ps = " << ps << "\n", *ps = " << *ps << "\n";

    cout << "access two elements with array notation\n"
    cout << "wages[0] = " << wages[0] << "\n";
}```