{  
    int * pt = new int;         // allocate space for an int  
    *pt = 1001;                 // store a value there  

    cout << "int ";  
    cout << "value = " << *pt << ": location = " << pt << "\n";

    double * pd = new double;   // allocate space for a double  
    *pd = 10000001.0;           // store a double there

    cout << "double ";  
    cout << "value = " << *pd << ": location = " << pd << "\n";
    cout << "size of pt = " << sizeof pt;
    cout << ": size of *pt = " << sizeof *pt << "\n";
    cout << "size of pd = " << sizeof pd;
    cout << ": size of *pd = " << sizeof *pd << "\n";
    return 0;
}

Here is the output:

int value = 1001: location = 0x004301a8
double value = 1e+07: location = 0x004301d8
size of pt = 4: size of *pt = 4
size of pd = 4: size of *pd = 8

Of course, the exact values for the memory locations differ from system to system.

**Program Notes**

The program uses `new` to allocate memory for the type `int` and type `double` data objects. This occurs while the program is running. The pointers `pt` and `pd` point to these two data objects. Without them, you cannot access those memory locations. With them, you can use `*pt` and `*pd` just as you would use variables. You assign values to `*pt` and `*pd` to assign values to the new data objects. Similarly, you print `*pt` and `*pd` to display those values.

The program also demonstrates one of the reasons you have to declare the type a pointer