A second, uglier approach is to realize that if \( \text{ps} \) is a pointer to a structure, then \( *\text{ps} \) represents the pointed-to value—the structure itself. Then, because \( *\text{ps} \) is a structure, \((*\text{ps}).\text{price}\) is the \text{price} member of the structure. C++'s operator precedence rules require that you use parentheses in this construction.

Listing 4.16 uses \texttt{new} to create an unnamed structure and demonstrates both pointer notations for accessing structure members.

**Listing 4.16 newstrct.cpp**

```cpp
// newstrct.cpp _ using new with a structure
#include <iostream>
using namespace std;
struct inflatable   // structure template
{
    char name[20];
    float volume;
    double price;
};
int main()
{
    inflatable * ps = new inflatable; // allot structure space
    cout << "Enter name of inflatable item: ";
    cin.get(ps->name, 20);       // method 1 for member access
    cout << "Enter volume in cubic feet: ";
    cin >> (*ps).volume;          // method 2 for member access
    cout << "Enter price: $";
    cin >> ps->price;
    cout << "Name: " << (*ps).name << "\n"; // method 2
    cout << "Volume: " << ps->volume << " cubic feet\n";
    cout << "Price: $" << ps->price << "\n"; // method 1
    return 0;
}
```

Here is a sample run: