Composite in C++

Back to **Composite** description

Composite design pattern

1. Identify the scalar/primitive classes and vector/container classes
2. Create an "interface" (lowest common denominator) that can make all concrete classes "interchangeable"
3. All concrete classes declare an "is a" relationship to the interface
4. All "container" classes couple themselves to the interface (recursive composition, Composite "has a" set of children up the "is a" hierarchy)
5. "Container" classes use polymorphism as they delegate to their children

```
#include <iostream>
#include <vector>
using namespace std;

// 2. Create an "interface" (lowest common denominator)
class Component
{
  public:
    virtual void traverse() = 0;
};

class Leaf: public Component
{
    // 1. Scalar class   3. "isa" relationship
    int value;
  public:
    Leaf(int val)
    {
        value = val;
    }
    void traverse()
    {
        cout << value << ' ';
    }
};

class Composite: public Component
{
    // 1. Vector class   3. "isa" relationship
    vector < Component * > children; // 4. "container" coupled to the interface
  public:
    // 4. "container" class coupled to the interface
    void add(Component *ele)
    {
        children.push_back(ele);
    }
    void traverse()
    {
        for (int i = 0; i < children.size(); i++)
        // 5. Use polymorphism to delegate to children
          children[i]->traverse();
    }
};

int main()
{
  Composite containers[4];
```

```
  for (int i = 0; i < 4; i++)
    for (int j = 0; j < 3; j++)
      containers[i].add(new Leaf(i *3+j));

  for (i = 1; i < 4; i++)
    containers[0].add(&(containers[i]));

  for (i = 0; i < 4; i++)
  {
    containers[i].traverse();
    cout << endl;
  }
}
```

Output

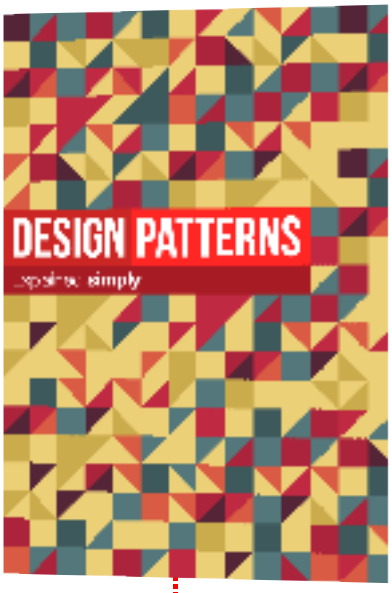
```
0 1 2 3 4 5 6 7 8 9 10 11
3 4 5
6 7 8
9 10 11
```

Read next

This article is taken from our book **Design Patterns Explained Simply**.

All of the design patterns are compiled there. The book is written in clear, simple language that makes it easy to read and understand (just like this article).

We distribute it in PDF & EPUB formats so you can get it onto your iPad, Kindle, or other portable device immediately after your purchase.



Learn more

Code examples

Java	**Composite in Java: Before and after**	**Composite in Java**	**Composite in Java**	**Composite in Java: User-configurable 'views' of a Composite**
C++	**Composite in C++**	**Composite in C++**		
PHP	**Composite in PHP**			
Delphi	**Composite in Delphi**			
Python	**Composite in**			