Delaying array size in class definition in C++?

Is there some way to delay defining the size of an array until a class method or constructor?

What I'm thinking of might look something like this, which (of course) doesn't work:

```
class Test
{
    private:
    int _array[][];

    public:
    Test::Test(int width, int height);
};

Test::Test(int width, int height)
{
    _array[width][height];
}
```

c++ arrays size delay

edited Apr 17 '09 at 2:37
RedBlueThing
23.9k 12 67 100

asked Mar 5 '09 at 6:10
Slythfox
25 1 5

4 Answers

What Daniel is talking about is that you will need to allocate memory for your array dynamically when your Test (width, height) method is called.

You would declare your two dimensional like this (assuming array of integers):

```
int ** _array;
```

And then in your Test method you would need to first allocate the array of pointers, and then for each pointer allocate an array of integers:

```
_array = new  *int [height];
for (int i = 0; i < height; i++)
{
    _array [i] = new int[width];
}
```

And then when the object is released you will need to explicit delete the memory you allocated.

```
for (int i = 0; i < height; i++)
{
    delete [] _array[i];
    _array [i] = NULL;
}
delete [] _array;
_array = NULL;
```

edited Mar 5 '09 at 6:28

answered Mar 5 '09 at 6:21
RedBlueThing
23.9k 12 67 100

Could add the pointer array allocation: _array = new int[height]; Upvoted for providing source though! – Daniel Sloof Mar 5 '09 at 6:25

Oops. Thanks Daniel. I forgot to add that :). Cheers. – RedBlueThing Mar 5 '09 at 6:28

The problem with manuall arrays managment is the fact you need to create your own copy constructors/operator = or make class explictly non-copyable to prevent future problems – Artyom Mar 5 '09 at 6:29

I will definitely read up on new/delete. Thanks. – Slythfox Mar 5 '09 at 6:33

1 I realize it is years later, but I would just like to point out (as it took me about half an hour of googling to figure it out :P) That "_array = new *int [height]" Should really be " _array = new int*[height]' " Otherwise you will run into a compiler error (using -W -Wextra -Wall -pedantic). :) – OscuroAA Oct 9 '14 at 16:58

vector is your best friend

```
class Test
{
    private:
    vector<vector<int> > _array;

    public:
    Test(int width, int height) :
        _array(width,vector<int>(height,0))
    {
    }
};
```

answered Mar 5 '09 at 6:27
Artyom
19.2k 12 81 164

1 Artyom is absolutely right. Save yourself a whole pile of pain :). However, I guess it is good to understand the underlying principles. – RedBlueThing Mar 5 '09 at 6:30

In terms of data structures, it looks like a vector is still structurally an array but wrapped into a class. Yeah? – Slythfox Mar 5 '09 at 6:32

Yes, most vector implementations use arrays 'behind the scenes' I think. You can also use the array operators to manipulate a vector. – gnud Apr 13 '09 at 9:17

2 In fact a vector *must* use an array to store the data. The standard dictates that the elements in a vector be contiguous. – John Dibling Apr 17 '09 at 2:41

I think it is time for you to look up the new/delete operators.

Seeing as this is a multidimensional array, you're going to have to loop through calling 'new' as you go (and again not to forget: delete).

Although I am sure many will suggest to use a one-dimensional array with width*height elements.

edited Mar 5 '09 at 6:17

answered Mar 5 '09 at 6:12
Daniel Sloof
6,835 11 49 84

(Months later) one can use templates, like this:

```
// array2.c
// http://www.boost.org/doc/libs/1_39_0/libs/multi_array/doc/user.html
// is professional, this just shows the principle

#include <assert.h>

template<int M, int N>
class Array2 {
public:
    int a[M][N];  // vla, var-len array, on the stack -- works in gcc, C99, but not all

    int* operator[] ( int j )
    {
        assert( 0 <= j && j < M );
        return a[j];
    }

};

int main( int argc, char* argv[] )
{
    Array2<10, 20> a;
    for( int j = 0; j < 10; j ++ )
    for( int k = 0; k < 20; k ++ )
        a[j][k] = 0;
    int* failassert = a[10];

}
```

answered Aug 7 '09 at 14:29

denis
8,590 3 30 48