

## Re: when to use "new"

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.cpp/2004-11/1970.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-11/1970.html)

---

**From:** Sam (*manseesaw\_at\_hotmail.com*)

**Date:** 11/15/04

Date: Mon, 15 Nov 2004 08:10:18 -0000

Maybe you should understand what the stack is and what the heap is. The memory for a local variable in a function (or a method of a class) is in the stack. When the function runs, the memory for local variables are allocated in a special memory area called the stack. Once the function returns, the memory allocated for its local variables is free immediately and all the values of its local variables are lost. An instance of a class constructed in this way acts exactly like an ordinary variable such as int, float, etc.

Now we get a problem. You can write the following Java codes:

```
Polynomial createNew() {  
    return new Polynomial();  
}
```

the function creates a new instance of class Polynomial and returns the reference of that new instance.

But in C++, if you write

```
Polynomial* createNew() {  
    Polynomial pn;  
    return &pn;  
}
```

It doesn't work. It may be compiled and linked but actually it's very dangerous. The instance of Polynomial is constructed when the function is called, but also is destroyed as soon as the function returns. However, the caller gets the pointer of that nonexistent object returned by the function and takes it as a valid pointer.

We want a way to create new objects other than in the stack, so that we can hold the objects even if the functions that create them return. So we get heap. We use "new expression" to create new objects in the heap – another special memory area where objects created by "new expressions" exist. Thus we can write the following codes in C++:

```
Polynomial* createNew() {  
    Polynomial* ppn = new Polynomial;  
    return ppn;  
}
```

Memory allocated by "new expressions" can and only can be freed by "delete" in the C++ codes. So here comes another thing: delete anything you created

comp.lang.c++: Re: when to use "new"

by new (after they are useless, but before you lose the pointers of them).  
Unlike in Java, the system doesn't clear the useless objects created by new  
here.

"Rv5" <rmolney@adelphia.net> Ð`ÈëÏÛçÐÁÎ:6aSdnVDuQInG0wXcRVn-3w@adelphia.com...

- > *Rookie c++ question, but Ive spent the last 5 years doing Java, where*
- > *everytime I created an object I used new. In c++ I can create my objects*
- > *without and its confusing me just a little.*
- >
- > *I have a class called polynomial. Its a nothing little class right now,*
- > *with just int variables, a basic container class. Im using it as I go*
- > *through some tutorials, but in this particular tutorial its telling me to*
- > *do*
- > *polynomial \*first = new polynomial();*
- > *but before I found this site I was just doing*
- > *polynomial first;*
- > *Im also struggling through pointers. I understand the basics, but fail to*
- > *see the advantage using them with my objects so quickly. Is one way*
- > *better than the other?*
- >
- > *Thanks*
- >
- >