

Re: Variable declaration/definition

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2008-06/msg03084.html>

- *From:* Jon Skeet [C# MVP] <skeet@xxxxxxxx>
 - *Date:* Thu, 26 Jun 2008 20:19:18 +0100
-

Anonymous Coward <me@xxxxxxxx> wrote:

I am relatively new in C#, but have several years C++/Java. I am writing a little C# console app to "test the waters" and learn the language.

If I have a class defined as:

```
class MyClass
{
MyClass(string s1, int i2, bool b3)
{
}
}
```

Why can't I declare a variable on the stack somewhere later on in my code like this:

```
// .... code above
MyClass mc("Hello",42, true);
```

That's not how you invoke a constructor in C#. You can just declare a variable, of course:

```
MyClass mc;
```

but that won't create an instance.

Do I always have to alloc from the heap when instantiating objects whose ctor have parameters?

You always have to allocate from the heap whenever you create **any** objects, regardless of whether there are parameterless constructors. In that respect C# is identical to Java. Basically C# is closer to Java than it is to C++ in many ways.

Re: Variable declaration/definition

Note, however, that "new" doesn't always mean "allocate from the heap". For instance, consider:

```
DateTime dt = new DateTime(2008, 06, 26);
```

That calls the `DateTime` constructor, but doesn't allocate anything on the heap (assuming it's a normal local variable), because `DateTime` is a value type. In this respect C# is different to Java, as Java doesn't have custom value types, just the primitives.

Jon Skeet - <skeet@xxxxxxxx>

Web site: <http://www.pobox.com/~skeet>

Blog: http://www.msmvps.com/jon_skeet

C# in Depth: <http://csharpindepth.com>

.