**Overloadable Operators (C# Programming Guide)**

This page is specific to

**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:

[Microsoft Visual Studio 2003/.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/8edha89s(VS.71).aspx)

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/8edha89s(VS.80).aspx)

[.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/8edha89s(VS.85).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4](http://msdn.microsoft.com/en-us/library/8edha89s(VS.100).aspx)

C# allows user-defined types to overload operators by defining static member functions using the [operator](http://msdn.microsoft.com/en-us/library/s53ehcz3.aspx) keyword. Not all operators can be overloaded, however, and others have restrictions, as listed in this table:

|  |  |
| --- | --- |
| **Operators** | **Overloadability** |
| [+](http://msdn.microsoft.com/en-us/library/k1a63xkz.aspx), [-](http://msdn.microsoft.com/en-us/library/wch5w409.aspx), [!](http://msdn.microsoft.com/en-us/library/f2kd6eb2.aspx), [~](http://msdn.microsoft.com/en-us/library/d2bd4x66.aspx), [++](http://msdn.microsoft.com/en-us/library/36x43w8w.aspx), [--](http://msdn.microsoft.com/en-us/library/wc3z3k8c.aspx), [true](http://msdn.microsoft.com/en-us/library/eahhcxk2.aspx), [false](http://msdn.microsoft.com/en-us/library/67bxt5ee.aspx) | These unary operators can be overloaded. |
| [+](http://msdn.microsoft.com/en-us/library/k1a63xkz.aspx), [-](http://msdn.microsoft.com/en-us/library/wch5w409.aspx), [\*](http://msdn.microsoft.com/en-us/library/z19tbbca.aspx), [/](http://msdn.microsoft.com/en-us/library/3b1ff23f.aspx), [%](http://msdn.microsoft.com/en-us/library/0w4e0fzs.aspx), [&](http://msdn.microsoft.com/en-us/library/sbf85k1c.aspx), [|](http://msdn.microsoft.com/en-us/library/kxszd0kx.aspx), [^](http://msdn.microsoft.com/en-us/library/zkacc7k1.aspx), [<<](http://msdn.microsoft.com/en-us/library/a1sway8w.aspx), [>>](http://msdn.microsoft.com/en-us/library/xt18et0d.aspx) | These binary operators can be overloaded. |
| [==](http://msdn.microsoft.com/en-us/library/53k8ybth.aspx), [!=](http://msdn.microsoft.com/en-us/library/3tz250sf.aspx), [<](http://msdn.microsoft.com/en-us/library/z5wecxwa.aspx), [>](http://msdn.microsoft.com/en-us/library/yxk8751b.aspx), [<=](http://msdn.microsoft.com/en-us/library/hx063734.aspx), [>=](http://msdn.microsoft.com/en-us/library/a59bsyk4.aspx) | The comparison operators can be overloaded (but see the note that follows this table). |
| [&&](http://msdn.microsoft.com/en-us/library/2a723cdk.aspx), [||](http://msdn.microsoft.com/en-us/library/6373h346.aspx) | The conditional logical operators cannot be overloaded, but they are evaluated using **&** and **|**, which can be overloaded. |
| [[]](http://msdn.microsoft.com/en-us/library/a3hd7ste.aspx) | The array indexing operator cannot be overloaded, but you can define indexers. |
| [()](http://msdn.microsoft.com/en-us/library/0z4503sa.aspx) | The cast operator cannot be overloaded, but you can define new conversion operators (see [explicit](http://msdn.microsoft.com/en-us/library/xhbhezf4.aspx) and [implicit](http://msdn.microsoft.com/en-us/library/z5z9kes2.aspx)). |
| [+=](http://msdn.microsoft.com/en-us/library/sa7629ew.aspx), [-=](http://msdn.microsoft.com/en-us/library/2y9zhhx1.aspx), [\*=](http://msdn.microsoft.com/en-us/library/s2bkaksf.aspx), [/=](http://msdn.microsoft.com/en-us/library/d31sybc9.aspx), [%=](http://msdn.microsoft.com/en-us/library/ydwa9zh0.aspx), [&=](http://msdn.microsoft.com/en-us/library/e669ax02.aspx), [|=](http://msdn.microsoft.com/en-us/library/h5f1zzaw.aspx), [^=](http://msdn.microsoft.com/en-us/library/0zbsw2z6.aspx), [<<=](http://msdn.microsoft.com/en-us/library/ayt2kcfb.aspx), [>>=](http://msdn.microsoft.com/en-us/library/23as4533.aspx) | Assignment operators cannot be overloaded, but **+=**, for example, is evaluated using **+**, which can be overloaded. |
| [=](http://msdn.microsoft.com/en-us/library/sbkb459w.aspx), [.](http://msdn.microsoft.com/en-us/library/6zhxzbds.aspx), [?:](http://msdn.microsoft.com/en-us/library/ty67wk28.aspx), [->](http://msdn.microsoft.com/en-us/library/s8bz4d5h.aspx), [new](http://msdn.microsoft.com/en-us/library/51y09td4.aspx), [is](http://msdn.microsoft.com/en-us/library/scekt9xw.aspx), [sizeof](http://msdn.microsoft.com/en-us/library/eahchzkf.aspx), [typeof](http://msdn.microsoft.com/en-us/library/58918ffs.aspx) | These operators cannot be overloaded. |
| **NoteNote:** | |
| The comparison operators, if overloaded, must be overloaded in pairs; that is, if **==** is overloaded, **!=** must also be overloaded. The reverse is also true, and similar for **<** and **>**, and for **<=** and **>=**. | |

To overload an operator on a custom class requires creating a method on the class with the correct signature. The method must be named "operator X" where X is the name or symbol of the operator being overloaded. Unary operators have one parameter, and binary operators have two parameters. In each case, one parameter must be the same type as the class or struct that declares the operator, as demonstrated in the following example:

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl51_code');" \o "Copy Code)

public static Complex operator +(Complex c1, Complex c2)

For more information, see [How to: Use Operator Overloading to Create a Complex Number Class (C# Programming Guide)](http://msdn.microsoft.com/en-us/library/6fbs5e2h.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif