**Can I overload all operators in C#?**

No, you cannot overload all operators in C#. There are certain operators that you can't overload, and some that have restrictions on their overloading. Here are some points to keep in mind:

* Operators You Cannot Overload:
  + && (conditional AND)
  + || (conditional OR)
  + ?: (conditional)
  + sizeof (size of)
  + typeof (type of)
  + -> (member access)
  + . (member access)
  + checked and unchecked
* Operators with Restrictions:
  + = (assignment) can be overloaded only by defining a user-defined conversion to the type of the left-hand operand.
  + ?: (conditional) can be overloaded only if the types of the second and third operands are the same.
* Unary Operators Restrictions:
  + + (unary plus) and - (unary minus) cannot be overloaded for bool, byte, sbyte, ushort, short, uint, int, ulong, or long.
* Binary Operators Restrictions:
  + The assignment operators (+=, -=, \*=, /=, and so on) can be overloaded, but their behavior is determined by the behavior of the corresponding binary operator.
* Equality and Comparison Operators Restrictions:
  + The == and != operators must be overloaded together. The same applies to <, >, <=, and >= when overloading.
* Indexing Operator Restrictions:
  + The [] operator can only be overloaded if the containing type is an array or an indexer property.
* Conversion Operators Restrictions:
  + implicit and explicit conversion operators can be overloaded with certain restrictions. They must involve user-defined types and follow specific rules.

In summary, while you can overload many operators in C#, there are some that you cannot overload, and others have specific restrictions or requirements. Always refer to the C# language specification for the most accurate and up-to-date information regarding operator overloading.