In C#, a weak reference is a reference that does not prevent the garbage collector from collecting the object it refers to. This means that the object referred to by a weak reference can be garbage collected even if the weak reference is still active.

Weak references can be useful when you need to maintain a reference to an object for a certain period of time, but you don't want to prevent the garbage collector from collecting the object when it's no longer needed. This is often used in scenarios where the object is expensive to create or where you want to optimize memory usage.

To create a weak reference in C#, you can use the WeakReference class. Here's an example:

```
class MyClass {
    public int MyProperty { get; set; }
}

MyClass myObj = new MyClass();
WeakReference weakRef = new WeakReference(myObj);

// ...

if (weakRef.IsAlive) {
    MyClass myObjRef = (MyClass)weakRef.Target;
    // Do something with myObjRef
}
```

In this example, the **WeakReference** object is created with a reference to **myObj**. Later on, the **IsAlive** property is used to check whether the object is still alive. If it is, the **Target** property can be used to get a reference to the object. However, if the object has been garbage collected, Target will return **null**.