Weak References

The garbage collector cannot collect an object in use by an application while the application's code can reach that object. The application is said to have a strong reference to the object.

A weak reference permits the garbage collector to collect the object while still allowing the application to access the object.

A weak reference is valid only during the indeterminate amount of time until the object is collected when no strong references exist.

When you use a weak reference, the application can still obtain a strong reference to the object, which prevents it from being collected. However, there is always the risk that the garbage collector will get to the object first before a strong reference is reestablished.

Weak references are useful for objects that use a lot of memory but can be recreated easily if they are reclaimed by garbage collection.

Short and Long Weak References

Short	**Long**
The target of a short weak reference becomes null when the object is reclaimed by garbage collection. The weak reference is itself a managed object and is subject to garbage collection just like any other managed object.	long weak reference is retained after the object's Finalize method has been called. This allows the object to be recreated, but the state of the object remains unpredictable.