The MemberwiseClone method creates a shallow copy by creating a new object, and then copying the nonstatic fields of the current object to the new object. If a field is a value type, a bit-by-bit copy of the field is performed. If a field is a reference type, the reference is copied but the referred object is not; therefore, the original object and its clone refer to the same object.

For example, consider an object called X that references objects A and B. Object B, in turn, references object C. A shallow copy of X creates new object X2 that also references objects A and B. In contrast, a deep copy of X creates a new object X2 that references the new objects A2 and B2, which are copies of A and B. B2, in turn, references the new object C2, which is a copy of C. The example illustrates the difference between a shallow and a deep copy operation.

There are numerous ways to implement a deep copy operation if the shallow copy operation performed by the MemberwiseClone method does not meet your needs. These include the following:

* Call a class constructor of the object to be copied to create a second object with property values taken from the first object. This assumes that the values of an object are entirely defined by its class constructor.
* Call the MemberwiseClone method to create a shallow copy of an object, and then assign new objects whose values are the same as the original object to any properties or fields whose values are reference types. The DeepCopy method in the example illustrates this approach.
* Serialize the object to be deep copied, and then restore the serialized data to a different object variable.
* Use reflection with recursion to perform the deep copy operation.