SOAP serialization and XML serialization

* During SOAP and binary serializatons, private fields, type information including assembly, namespace, public key token information are also serialized.
* In XML ---Serialized form of the object does not contain any type, assembly information of the class. Only the data is stored.
* Deserialization gives the same object as it was before serialization.
* Serialization can be controlled to exclude a field using [NonSerialized] attribute.
* By using XML serialization only public properties and fields can be serialized. If private members are to be serialized, other serialization methods should be used.
* [Serializable] attribute is used to mark a class as serializable.
* It does not have to be put when XML serialization is used.

.

* To use SOAP and binary serialization, classes do not need to have a parameterless constructor. Because of performance issues, the constructor of the class is not called during deserialization.
* In XML serialization It requires a parameterless constructor. This constructor is called during deserialization.
* IDictionary objects which cannot be serialized using XML serialization can be serialized using SOAP/Binary serialization.
  + Binary Serialization:
    - Compact serialization
    - Performs better
    - Creates byte stream
  + SOAP Serialization
    - Creates SOAP messages.
    - Use if the serialization and deserialization platforms are not .NET.
    - Use if the message is sent through a firewall.
    - Easy debugging.

Both SOAP and binary formatters inherit from IFormatter interface.