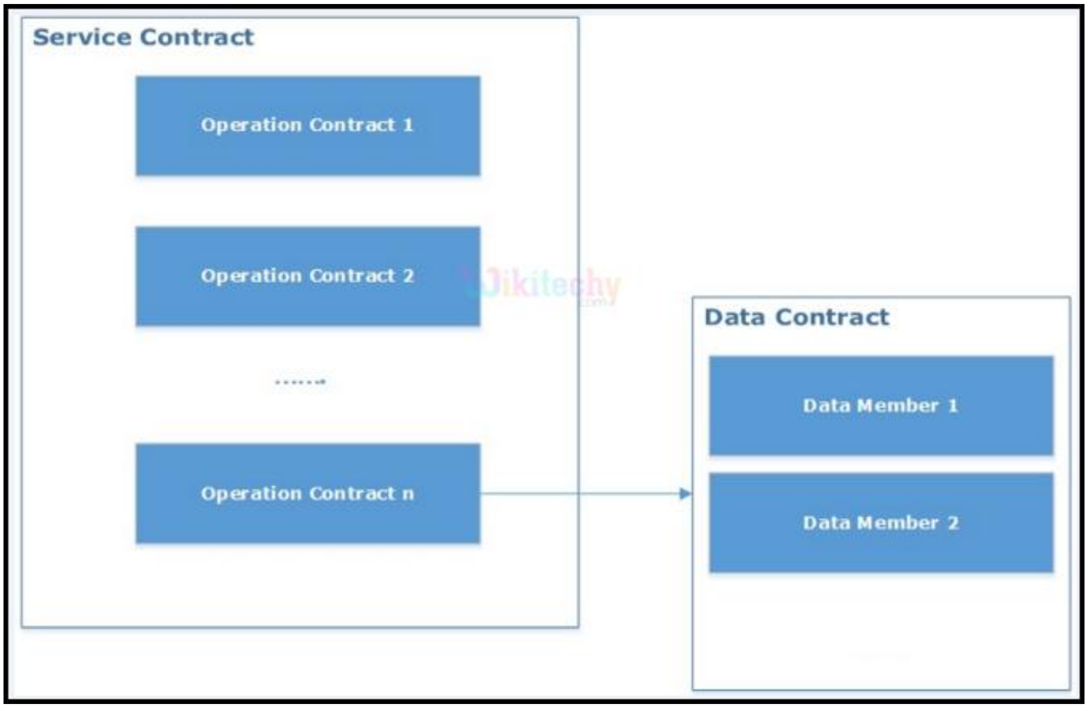
WCF-DATA-CONTRACT

- Data Contract is a formal agreement between a service and a client that abstractly describes the data to be exchanged.
- That is, to communicate, the client and the service do not have to share the same types, only the same data contracts.



Interaction of Operation Contract with Data Contract

Data Contract:

- Data contract defines the data type of variable exposed to the client.
- Message with primitive data types are automatically serialized for exchange.
- User defined types need an explicit Data Contract for serialization process and hence users using [DataContract] and [DataMember] attributes.
- So to conclude, the DataContract makes the client to be aware of variable's data type that are **coming from** or **passed** to a Service Contract.

Service Contract:

- A **service contract** defines the operations which are exposed by the service to the outside world.
- It is the interface of the WCF service and it tells the outside world what the service can do.
- It may have service-level settings such as the **name of the service** and **namespace for the service.**

Operation Contract:

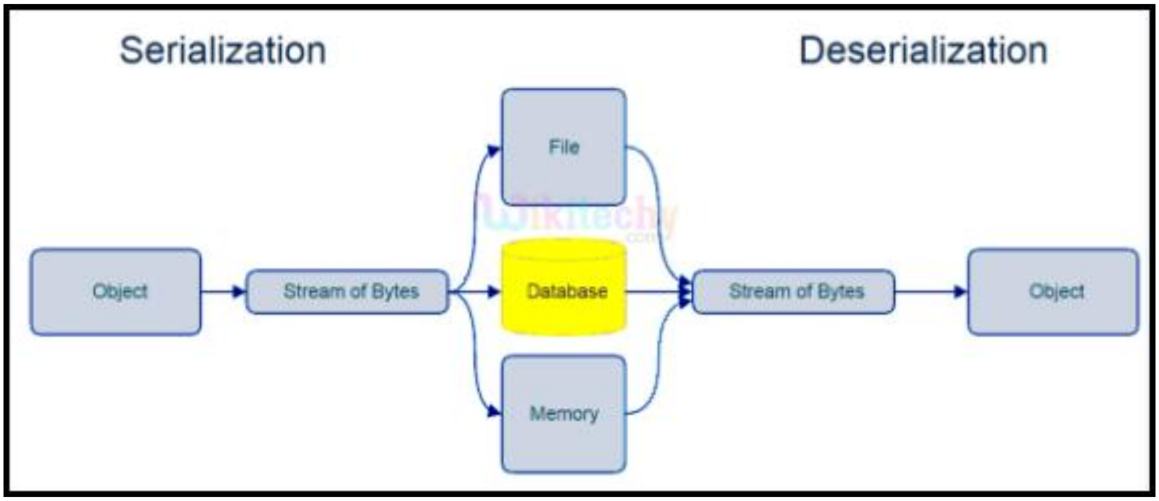
- An operation contract is defined within a service contract such that the parameters and return type of an operation has been declared.
- It can also define operation-level settings like transaction flow of the operation, the directions of the operation (one-way, two-way, or both ways) and fault contract of the operation.

Fault Contract:

- A **fault contract** defines errors raised by the service, and how the service handles and propagates errors to its clients.
- An operation contract can have zero or more fault contracts associated with it.

Serialization / Deserialization:

- **Serialization** is a process used at sender's place to transform an object into a sequence of bytes for transmission over the network.
- **Deserialization** again transforms the sequence of bytes received,back to the object form.
- A reference for System.Runtime.Serialization is required in the project code to use **DataContract and DataMember attributes.**

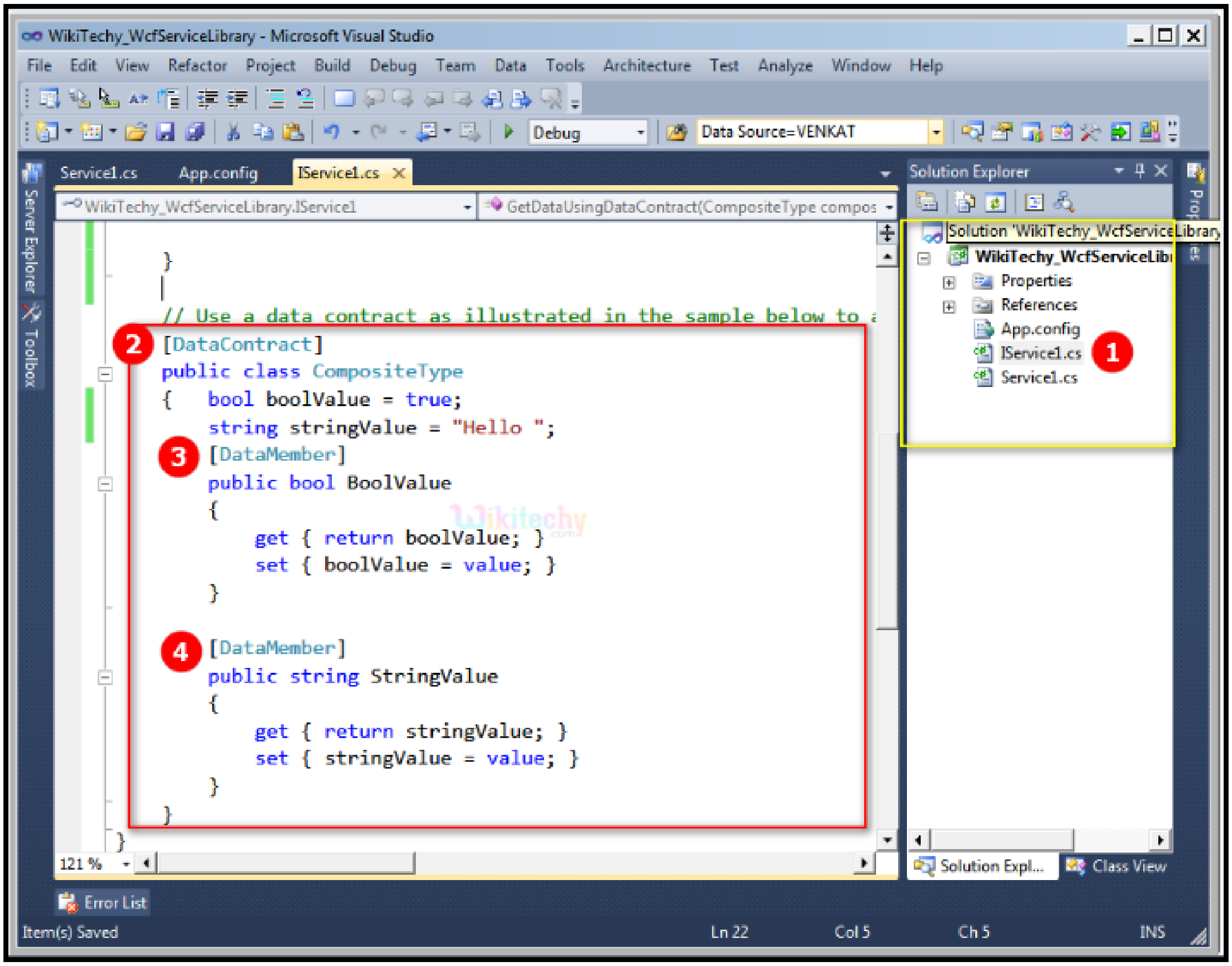


Serialization / Deserialization

Sample Code:

```
[DataContract]
    public class CompositeType
    {
        bool boolValue = true;
        string stringValue = "Hello ";
        [DataMember]
        public bool BoolValue
        {
            get { return boolValue; }
            set { boolValue = value; }
        }
        [DataMember]
        public string StringValue
        {
            get { return stringValue; }
            set { stringValue = value;}}
```

Code Explanation:


WikiTechy_WcfServiceLibrary - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Data Source=VENKAT
Service1.cs
App.config
IService1.cs
WikiTechy_WcfServiceLibrary.IService1
GetDataUsingDataContract(CompositeType compos
Server Explorer
Toolbox
Solution Explorer
Solution 'WikiTechy_WcfServiceLibrary
WikiTechy_WcfServiceLib
Properties
References
App.config
IService1.cs
Service1.cs
1
2
3
4
121 %
Solution Expl...
Class View
Error List
Item(s) Saved
Ln 22
Col 5
Ch 5
INS

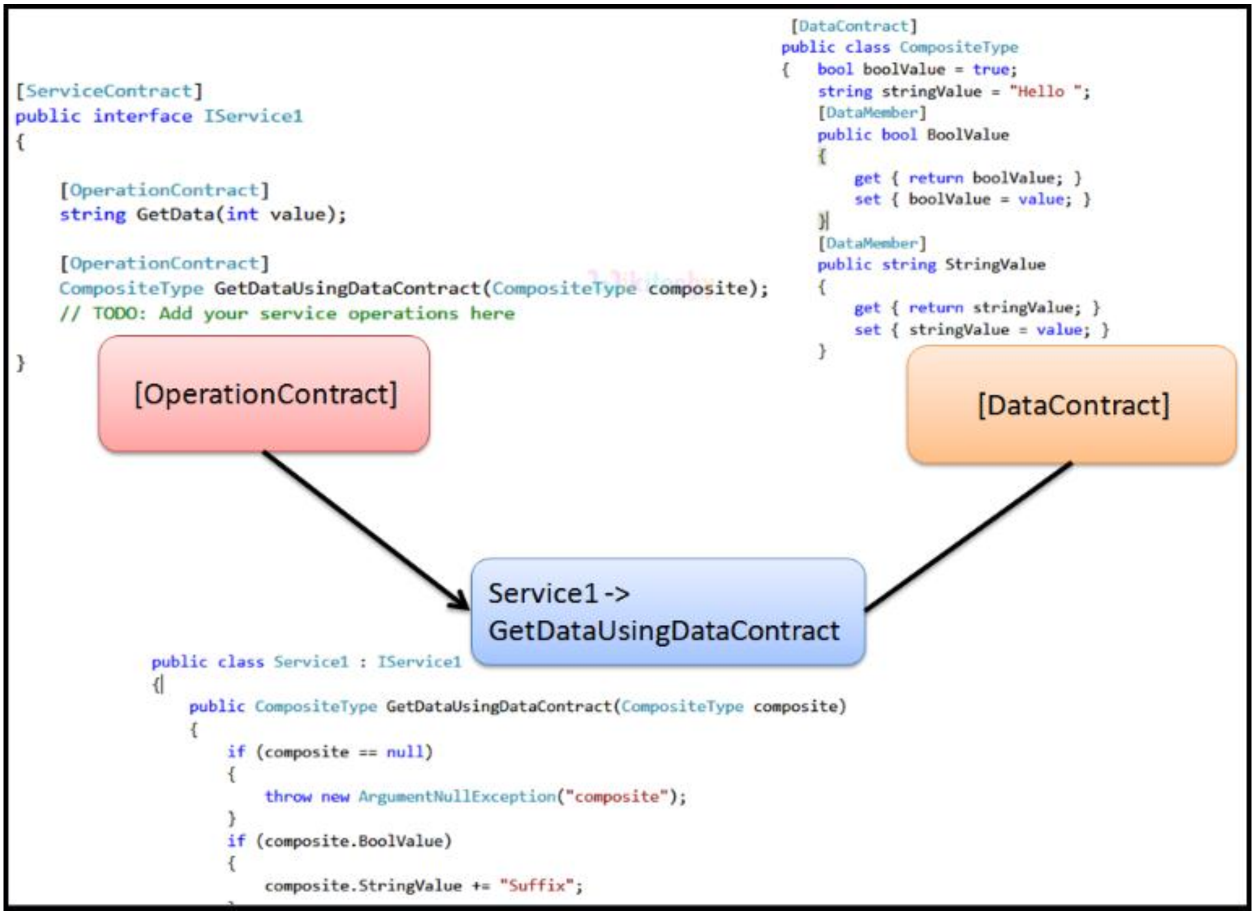
```
    }
    |
    // Use a data contract as illustrated in the sample below to
    [DataContract]
    public class CompositeType
    {   bool boolValue = true;
        string stringValue = "Hello ";
        [DataMember]
        public bool BoolValue
        {
            get { return boolValue; }
            set { boolValue = value; }
        }

        [DataMember]
        public string StringValue
        {
            get { return stringValue; }
            set { stringValue = value; }
        }
    }
}
```

1 The WCF application automatically creates a Service Interface IService1.cs file on opting a new project.

2 The file IService1.cs defines a DataContract and the class name as CompositeType.

3 The CompositeType data contract has a [DataMember] named BoolValue, which is a property used to get set values.

4 The CompositeType data contract has another [DataMember] named StringValue, which is a property with get set values.



```
[ServiceContract]
public interface IService1
{

    [OperationContract]
    string GetData(int value);

    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);
    // TODO: Add your service operations here

}
```

```
 [DataContract]
public class CompositeType
{   bool boolValue = true;
    string stringValue = "Hello ";
    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }
    [DataMember]
    public string StringValue
    {
        get { return stringValue; }
        set { stringValue = value; }
    }
```

```
public class Service1 : IService1
{
    public CompositeType GetDataUsingDataContract(CompositeType composite)
    {
        if (composite == null)
        {
            throw new ArgumentNullException("composite");
        }
        if (composite.BoolValue)
        {
            composite.StringValue += "Suffix";
```

Operation-Service-Data Contracts

- As per the above diagram, the operation/service contract, interacts with the service class to access data from data contract.