**Serialization and Deserialization**

**Introduction :**

Serialization is the process of converting an object or data structure into a format that can be easily stored or transmitted. Deserialization is the reverse process ofSerialization .

**Types of Serialization in C#**

* Binary Serialization
* XML Serialization
* JSON Serialization
* Custom serialization(using ISerializable class)

**Binary Serialization**

Converts objects into a binary format, preserving the complete state of objects. It uses the BinaryFormatter class

**Using-This keyword dispose the file after usage(CLEAN UP).**

**Provide custom Path and remove USING keyword to view the created file.**

using System;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

**[Serializable]** // For binary serilization we need to put serilizable . keyword above the class which going to serlized .

class Person

{

public string Name { get; set; }

public int Age { get; set; }

}

class Program

{

static void Main()

{

Person person = new Person { Name = "John Doe", Age = 30 };

**// Serialization**

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream fs = new FileStream("person.bin", FileMode.Create))

{

formatter.Serialize(fs, person);

}

Console.WriteLine("Object serialized successfully.");

**// Deserialization**

using (FileStream fs = new FileStream("person.bin", FileMode.Open))

{

Person deserializedPerson = (Person)formatter.Deserialize(fs);

Console.WriteLine($"Deserialized Person: {deserializedPerson.Name}, {deserializedPerson.Age}");

}

}

}

**XML Serialization:**

Serializes objects into XML format, providing human-readable output. It uses the XmlSerializer class.

using System;

using System.Xml.Serialization;

public class Book

{

public string Title { get; set; }

public decimal Price { get; set; }

}

class Program

{

static void Main()

{

Book book = new Book { Title = "C# Programming", Price = 49.99m };

// Serialization

XmlSerializer serializer = new XmlSerializer(typeof(Book));

**// XmlSerializer it is not an static class so need to create object**

using (FileStream fs = new FileStream("book.xml", FileMode.Create))

{

serializer.Serialize(fs, book);

}

Console.WriteLine("Object serialized");

// Deserialization

using (FileStream fs = new FileStream("book.xml", FileMode.Open))

{

Book deserializedBook = (Book)serializer.Deserialize(fs);

Console.WriteLine($"Deserialized Book: {deserializedBook.Title}, ${deserializedBook.Price}");

}

}

}

3. Converts objects into JSON format, which is lightweight and widely supported. It uses the JsonSerializer class from the System.Text.Json namespace.

* It has key value pair

using System;

using System.Text.Json;

public class Product

{

public int Id { get; set; }

public string Name { get; set; }

public double Price { get; set; }

}

class Program

{

static void Main()

{

Product product = new Product { Id = 1, Name = "Laptop", Price = 999.99 };

// Serialization

string jsonString = JsonSerializer.Serialize(product);

Console.WriteLine($"Serialized JSON: {jsonString}");

// Deserialization

Product deserializedProduct = JsonSerializer.Deserialize<Product>(jsonString);

Console.WriteLine($"Deserialized Product: ID={deserializedProduct.Id}, Name={deserializedProduct.Name}, Price=${deserializedProduct.Price}");

}

}