As a first step, it checks whether the input received from the user is a valid Json. If an incorrect value is entered, the program prints a clear message on the console.

The program accepts one json at a time.

I used the Newtonsoft.Json library to serialize, deserialize and manipulate the data. I used the following two documents for this:

<https://www.newtonsoft.com/json/help/html/T_Newtonsoft_Json_Linq_JToken.htm>

<https://www.newtonsoft.com/json/help/html/T_Newtonsoft_Json_Linq_JContainer.htm>

There is no logic related operation in the **Transform()**. It checks the json. It sends Json's properties to the GetOrderedJsonChilderen method and converts the incoming value back to json and returns it.

**IsValidObject()** converts string to json. It returns false if it is not valid. if the operation is completed successfully, it returns the json object with the true value.

**ConvertFromStringToStream()** converts string type data to stream type.

**GetPriority()** returns the priority order according to the type of Json token. Lower number indicates higher priority.

**GetOrderedJsonChilderen()** takes properties in json as a list. First, it sorts the Property in order of priority (it takes priorties from **GetPriorty**) with the help of LinQ (It uses OrderBy). Actually, the process ends here. But objects and arrays in json can contain nested objects and arrays. Nested objects and arrays must also be sorted. That's why the properties go into the foreach loop.

If nested jsons are object types. The properties of the object are taken and sent to the same method(GetOrderedJsonChilderen) again. If nested jsons are array types. The properties of the array are taken and sent to the same method(GetOrderedJsonChilderen) again. **This process continues until there are no nested arrays and objects.**

So all json data is sorted by the priorities of the properties.