**BOXING AND UN-BOXING IN C#**

 As we all know, C# is a strongly-typed language. That means we have to declare the type of a variable that indicates the kind of values it is going to store, such as integer, double, float, boolean decimal, text, etc.

 In C#, data types are categorized into two types: Value types and Reference types. Value types include simple types (such as int, double, float, bool, and char), enum types, struct types, and Nullable value types. Reference types include class types, interface types, delegate types, and array types.

 In C#, Boxing and unboxing allows developers to convert .NET data types from value type to reference type and vice versa.

 In C#, Converting a value type to a reference type is called called boxing in C# and converting a reference type to a value type is called unboxing in C#.

 Mainly C# has two kinds of data types, value types and reference types. Value type stores the value itself, on the other hand reference type stores the address of the value where it is stored.

 In C#, Some predefined data types such as int, float, double, decimal, bool, char, etc. are value types and object, string, and array are reference types.

**What Is Boxing In C# ?**

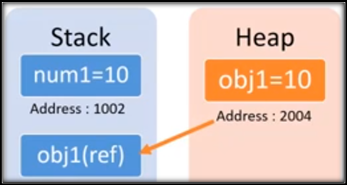
* Implicit (automatic) conversion of a value type to a reference type is called "Boxing".

**Example:**

int num1 = 10; // int is value type

object obj1 = num1; // Implicit conversion of value type into reference type.

* In Boxing process, a value type is being allocated on the heap rather than the stack.



**What Is Un-Boxing In C# ?**

* Explicit (manual) conversion of the same reference type (which is being created by boxing), back to a value type.

**Example:**

int num1 = 10; // int is value type

object obj1 = num1; // Implicit conversion of value type into reference type (Boxing)

int num2 = (int) obj1; // Explicit conversion of reference type into value type (Unboxing)

* In unboxing process, an unboxed value is being allocated to a variable on the stack rather than the heap.

