DATA TYPES IN C#

* A data type is a classification that specifies which type of value a variable can hold, what operations can be performed on that variable, and how the data is stored in memory.
* There are two types of data types in C

1.Value type

2.Reference type

* 1. Value type : These data types are primitive data types. These type of data type contains a object with fixed value.

Ex: 1. Byte

System.Byte - system type

8-bits - size ( 8 bits = 1 byte)

Range - 0 to 255

2. char

System.Char

16 bits

U+0000 to U+ffff

3.short

System.Int16

16 bits

-32,768 to 32,767

4.int

System.Int32

32 bits

-2,147,483,648 to 2,147,483,647

5.long

System.Int64

64 bits

-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

6.Sbyte

System.Sbyte

8 bits

-128 to 127

7.ushort

System.Int16

16 bits

0 to 65,535

8.uint

System.Int32

32 bits

0 to 4,294,967,295

9.ulong

System.Int64

64 bits

0 to 18,446,744,073,709,551,615

10. bool

System.Boolean

8 bits

Values : true , false.

11. decimal

System.Decimal

128 bits (16 bytes)

+- 1.0\* 10^-28 to +- 7.9228\*10^28

12. float

System.Single

32 bits (4 bytes)

+-1.5\*10^-45 to +-3.4 \*10^38

13. double

System.Single

8 bytes

+-1.5\*10^-45 to +-3.4\*10^38

* 2.Reference type : These types of data type directly contains a reference to an object. These are user defined data types.
* Ex: Strings ,enumerations, classes, interfaces, arrays.

1. enum : it is used to make values as constant. Default value starts with 0.

System.Enum

4 bytes

1. Void : it is used to specify the return type of data

Var : it is used to declare an implicit type variable, which specifies the type of variable based on initialised value.

Syntax : var variable\_name = value;

**BOXING AND UNBOXING IN C#**

Boxing and unboxing are operations in C# related to the conversion between value types and reference types.

Boxing is the process of converting a value type to the type **object** or to any interface type that it implements.

Ex: int intvalue = 42

Object boxedvalue = int value;

Unboxing is the process of converting an object (previously boxed) back to its original value type.

Ex: object boxed value = 42;

Int intvalue =(int)boxed value;

C# provides the **System.Convert** class and the **as** and **is** operators to perform conversions between certain types without boxing and unboxing, when applicable.

Ex: int intValue = 42;

object boxedValue = System.Convert.ChangeType(intValue, typeof(object));

Object

It is a type which is base of all the types and one of the oldest features of the language. It means values of any types can be stored in object type variable. But type conversion (un-boxing) is required to get original type when the value of variable is retrieved in order to use it.

Ex : Object o;

o = 10;

So it is an extra overhead of un-boxing. Thus object type should only be preferred if we have no more information about the data type. Compiler has title information of object type.

Dynamic

This keyword was introduced in .net 4.0 framework. This also can be used to store any type of value like object and var but unlike object un-boxing is not required at time of using the variable.

ex: Dynamic d1 = 10;

Dynamic d2;

D2=new Product();

Dynamic d3="string value"

Compiler has no information of dynamic type but value is evaluated at run time.

**Differences between C# classes and C# Structures**

1. Classes are references types of data type, structures are value type of data type.

2. Classes support default constructor i.e. we can set default values that will be assigned while creating an object. Structures do not support the concept of the default constructor, we cannot set values like classes that can be used as default values while creating a structure object/variable.

3. Classes support the inheritance; structures do not support the inheritance.

**SEARCHING IN C#**

Searching is the technique to find particular item, here we are discussing some of the popular searching techniques to find an item from the array.

* There are two types of searching is used in computer technology.

1.Internal Searching

Searching an item into main memory or primary memory.

2.External Searching

Searching an item into external or secondary memory.

**THIS & BASE :**

THIS : It is used to refer to the current instance of the class.

To distinguish between instance variables and parameters with the same name. To invoke instance methods, properties, or indexers when there is a local variable with the same name.

BASE : It is used to access members (fields, properties, methods) of the base class in the context of inheritance. To call a method or access a member in the base class that is overridden or hidden in the derived class.

WHAT IS .NET?

.NET is a cross platform, opensource framework developed by Microsoft that provides a runtime environment and a set of libraries for building and running various types of applications. The .NET framework supports multiple programming languages, allowing developers to use their language of choice to build applications for different platforms, such as Windows, Linux, and macOS.