Data Types

C# is a strongly typed language. It means, that you cannot use variable without data types. Data types tell the compiler that which type of data is used for processing. Such as if you want to work with string value then you will have to assign string type variable to work with. C# provides two types of data types: **Value types** and **Reference types**.

A **Value type** data type stores copy of the value whereas the **Reference type** data types stores the address of the value.

C sharp provides great range of predefined data types but it also gives the way to create **user defined data types.**

VALUE TYPES

- Value type variables can be assigned a value directly.
- They are derived from the class System.ValueType.
- The value types directly contain data

Data Types	Size	Values
sbyte	8 bit	–128 to 127
byte	8 bit	0 to 255
short	16 bit	–32,768 to 32,767
ushort	16 bit	0 to 65,535
int	32 bit	–2,147,483,648 to 2,147,483,647
uint	32 bit	0 to 4,294,967,295
long	64 bit	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
ulong	64 bit	0 to 18,446,744,073,709,551,615
char	16 bit	0 to 65535
float	32 bit	–1.5 x 1045 to 3.4 x 1038

double	64 bit	−5 x 10324 to 1.7 x 10308
decimal	128 bit	−1028 to 7.9 x 1028
bool	–––	True or false

REFERENCE TYPES

- The reference types do not contain the actual data stored in a variable, but they contain a reference to the variables.
- In other words, they refer to a memory location. Using multiple variables, the reference types can refer to a memory location. If the data in the memory location is changed by one of the variables, the other variable automatically reflects this change in value.
- Example of built–in reference type: **string**.

String
- allows you to assign any string values to a variable
- the value for a string type can be assigned using string literals in two forms: quoted and @quoted.

For example,

```
String str = "Tutorials Point";
```

A @quoted string literal looks as follows –

```
@"Tutorials Point";
```