**DataType**

Definition :

Types of DataTypes :

1. Primitive DataTypes (Value Type)

* Int16 (short)
* Int32
* Int64 (long)
* Decimal
* Float
* Double (8 bytes/64 bits)
* Boolean (1 byte/8bit)

1. Non Primitive DataTypes (Reference Type).
2. Inbuilt Type

* String (immutable).
* StringBuilder (mutable).

1. User-Defined Type (Custom Type)

* Any user defined Model/DTO with Properties.

Explanations :

**Primitive Types** / Value Type : Directly stores the Value in the Variable.

Int a = 15;

Int a => define a variable by the name “a” of type “integer” of size “4 bytes / 32 bits”.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | . | . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | . | . | 0 | 0 | 0 |

A =100 1 2 3 4 5 6 7 8 9 10 11 12…………………………………………………………………….30 31

Int a = 15;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | . | . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | . | . | 0 | 0 | 1 | 1 | 1 | 1 |

A =100 1 2 3 4 5 6 7 8 9 10 11 12………………………………………..……………………………….30 31

**Non-Primitive/Reference Type :**

string b = “Hello”; (in stack memory).

string b => Create a reference by the name “b” of type string.

*The reference/variable “b” itself is basically a pointer with size as 32 bits on a 32 bit OS, 64 bits on a 64 bit OS.*

On 64 bit processor in bits.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 0 | . | . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | . | . | 0 | 0 | 0 |

b = 100 1 2 3 4 5 6 7 8 9 10 11 12…………………………………………………………………….62 63

On 64 bit processor in bytes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | A | D | 2 | 9 | 1 | 0 | 1 |

b=100 108 116 124 132 140 148 156 163

Hello

3AD29101 (In Heap Memory)

**Conversion of DataTypes.**

String to int etc.

Methods used to convert

Convert.ToInt32(“123”); => exception is not handled internally. (explicit Exception Handeling is   
 required.)

Int32.TryParse(“123”, out c); (internally handled)

Parse method (not handled internally)

**OverFlow and UnderFlow**

**Conversion of Binary to Decimal and Visa-versa.**

**Nullable DataTypes**

**ENUM**

**STRUCT**

**Byte[]**

**Operators**

**null coalescing operator**

**Ternary Operator**

**Pre and Post Increment Operator**

**Pre and Post Decrement Operator**

**String and its methods**

**Tostring**

**Trim**

**IsNullorEmpty**

**SubString**

**Random Class**

**While Loop**

**ForLoop**

**ForEach loop**

**Continue statement;**

**Break statement**

**If Else**

**If , else if , else**

**Switch Case**

**Arrays**

**Collection**

**Generics**

**Exception Handling**

**Types of Exception Handling (Global Exception Handling etc)**

**Exception Propagation**

**File Handling**

**WriteStream and ReadStream**

**Regular Expression**

**LINQ**

**OOPS**

**SOLID Principle**

**Design Pattern (Factory design Pattern)**

**API Creation Explanation.**

**Complier ROSLYN and CLR (Explain Run time environment in dot net)**

**MVC**

**ASP net WEB API**

**Worker**

**Console**

**MYSQL**

**Select**

**Update**

**Create**

**Delete**

**SubQuery**

**Join**

**Group By / Having**

**StoredProcedures.**