**Session 2: Variables & Datatypes**

**Varibale:-** It’s nothing but a name which is used to store the value of something. When variable define it allocate some memory. Variable is declared before used. And the value can be changed during execution of program. Variable contains declaring and initialisation.

* **Types of Variable: - Local Variable: -** defined within block or method. This is created when function is created and destroyed when exit. Scope is limited to that block only.
  + **EG:**- string myName;
* **Constant Variable: -** const keyword is used. Cannot change.
  + EG:- const int myName
* **Instance or Non-Static Variable: -** non-static variable and declared in class but outside of any method, constructor. This variable created when object is created and destroyed when object destoyed.
  + Eg:- int myName;
* **Static or Class Variable: -** declared with static modifier or any static block. Its also declared within class outside any method. Static variable can be access by using class name. Static variable define only one time.
  + EG:- static int myName;
* **Read-only Variable: -** readonly keyword is used. Cant modified. Its value is fixed for specific instance of class. Value can be change during runtime.
  + EG:- readonly int myName;

**Defining and Declaring Variable:-** Variable is unique, contain letter, digit, underscore ( \_ ), always start with letter, case sensitive.(num & Num).

EG:- string abc = “Imran”; string AbcXyzPqr = “ganesh”;

**Datatypes & Types: C#** is strongly typed language. datatype indicate the variable which type of variable we are using. Each datatype require memory and it’s depend on system such as 32bit or 64 bit. Int 6bit upto 5.99 6.

String Name= “pankaj”; int float double bigint bool char,string long datetime etc….. (structure) string classs = name;

**Types: Value Types:** allocate value in stack(LIFO). Integer and floating point based.

**Predefined: -** int, bool, float short, int , char, float, double etc etc...

**UserDefined:-** structure**, Enum:-** used to assign constant name to numeric integer. Values access by dot(.), enum keyword is must. Automatically avalues assign to enum if not assign. Enum week{ mondy, 0 tue 1……}

a

10

Name

name

Heap

**Stack int a=10;**

**String name= “Imran”;**

**Reference Type: -** store value in heap: handle big value. it will allocate whenever it require. dosnt contain the actual stored value data stored in variable but they are havning refrence only. If the one variable change it will reflect automatically to other.

**Predefined Types** - such as Objects, String

**User defined Types** - such as Classes, Interface.

**Pointer Datatype:** - locator or indicator that points to an address of value. Symbol is used & \*.

**Mutable and Immutable :-** mutable means value can be change.: cannot create memory used allocated memory. Data can be change after instance cration EG:- Stringbuilder.

**Immutable** value cannot be change: allocate new memory space. Value cant be changed after instace creation. Eg;-String. WHY:. Array or character and array is fixed in size.

**Escape Sequence:-** unprintable character escape sequence used to specify actions such as tab movement(\t) new line(\n) backspace(\b) single Quote(\’) double Quote(\”) etc..

@D:\imran\txt…………………..\n\n

**Verbatim Literal: -** for disabling the support of escape sequence we write @ symbol before the escape sequence. It ignores escaper character and print out as a string. It will make string in printable format.

**EXAMPLE:--**

using System;

using System.Collections.Generic;

using System.Text;

namespace AllSession

{

class S2\_VaribaleAndDatatypes

{

enum Months

{ Jan, Feb, March, April, May, June, July, August, Sepetember,Octobor, November, December};

// Instance variabel or Non static variabel

int[] numbers = { 1, 2, 3, 4, 5 };

//Static Variable

static string[] studentName = { "Abhilasha", "Amit", "Eqbal", "Faheem", "Prakash", "Akhil", "Ganesh", "Faisal", "Asad", "Sonu" };

public void VariablesAndUses()

{

//Local Variable...

int digit = 10;

Console.WriteLine("Integer Digit " + digit);

string helloMsg = "Hello Team";

Console.WriteLine("Message:-- " + helloMsg);

//Constant varibale

const string game = "Holyball";

//game = "Cricket";

Console.WriteLine("Event :-- " + game);

for (var i = 0; i < studentName.Length; i++)

{

Console.Write(studentName[i] + " ");

}

Console.WriteLine();

for (int i = 0; i < numbers.Length; i++)

{

Console.Write(numbers[i] + " ");

}

//Console.WriteLine("\nNumbers Array " +numbers);

Console.WriteLine();

foreach (int month in Enum.GetValues(typeof(Months)))

{

Console.Write($"{Enum.GetName(typeof(Months), month)}");

Console.WriteLine($" {month}");

}

//ESCAPE SEQUNCE AND VERBATRIM

string path = " "; //Escape Sequence \t \b \n \' \"" \?

path = @"D:\Resume\ImranResume"; // Verbatim

Console.WriteLine(path);

Console.ReadLine();

}

}

}