C # Programming

suraj Kumar mishra

2021

.NET MVC

MODEL:DB

VIEW:DESIGNING

CONTROLER:PROGRAMMING

DESIGNING

DATABASE

-------------------

.NET :MICROSOFT

DESKTOP APPLICATION

CONSOLE APPLICATION

WEB APPLICATION

WINDOW SERVICES

WINDOWS SERVICES

MOBILE APPLICATION

30+PROGRAMMING LANGUAGES

--------------------------- Note-----------------------------------------------

.NET Framework

.NET : it was developed by microsoft team mainly by anders & hegalberg in 2000.

it is a framework by microsoft that provides a envirenment to write and run the

whole application on a single plateform.

.NET IDE -VISUAL STUDIO

During the development of .NET ,C# was developed.

c - c++ - c#

c#=> c# is a general purpose,high level,object oriented programming language.

many other programming language are just used to develope a single type of application but c# is used to develope console based application,web based application and mobile application.

Besides c# other 30+ programming languages are provided by Microsoft with .net

framework.According to the developer choice any language can be used in development.

But mostly programming language with .NET is c#. It is the extension of C++.

Object Oriented Programming:- Object Oriented Programming language mainly provide

two feature to the programming.First is Security and second is Code Reusability

.In object oriented programming language every methods are encapsulated/written within classes.

OOP'S C#:- Namespace -> Classes -> Methods -> Statement

Raw data(input)-> Processing-> Output

---------------------------------- 04/12/2021 ----------------------------------

What is NameSpace:-

---

Name space is the coollection of multiple classes.NameSpace is the logical collection normaly vcan be think as folder containing multiple .cs file.

By default the main namespace of the c# is System.

If you want to access any class declared within a namespace then firstly we have to include the namespace in file.

Syntax:-

-------

using NameSpace\_Name:-

Classes:-

------------

Classes is the collection of data members and member function.

so the in simple word classes is the collection of multiple variables and methods.

Any variable/method declared in the class is directly accessible any other methods you don't need instance of the class.

Only static member can not access the non static members of the class.

Syntax:-

---

class class\_name

{

data\_members(variable);

Member\_function;

}

Any non-static member of class is not directly accessible out of the class.

To access any member of a class,form the outside of class,you have to create instance variable/object of the class.

object are mainly a variable of a class type which is used to access all member(variable,method) of a class anywhere.

class\_name object\_name=new class\_name()

second method():

Class\_name object\_name; // normal variable of class type

object\_name=new class\_name();//instance variable of class

when a class variable which is not declared by using new keyword stores the refrence of another instance variable then this is known as Refrence variable

of class.

Suppose there is a class named Program:

Program p=new Program();

program p1;

p1=p;//now p1 is the refrence variable of class program which holds the refrence

of another instance variable that is p.

-------------------

Structer of c# first Program:

--------------------------

using system

namespace myproject1

{

class Myclass1

{

static void main(String [] arg)

}

}

Data type is the pre-fix which specifies the property of any variable, function.Data type defines total memory spaces consume by a object(variable,method) and which type of value it can store.

1.Value Type

Simple type

Signed Integral Value: short ,byte ,int ,long

Unsigned Integral Value: ushort , ubyte, uint ,ulong

Unicode Character: char

Floating Type Value: float , double , decimal

Boolean Value: bool

Structure :struct {---------}

Enum:- enum e {------------}

Nullable Type:- null

2.Reference Type

Class type

Interface

Array

static void Main(string[] args)

        {

            int num1, num2, sum;

            Console.WriteLine(" How to Calculate the sum of two numbers:");

            Console.Write(" Input number 1: ");

            num1 = Convert.ToInt32(Console.ReadLine());

            Console.Write(" Input number 2: ");

            num2 = Convert.ToInt32(Console.ReadLine());

            sum = num1 + num2;

            Console.Write(" {0} + {1} = {2}",num1,num2,sum);

            Console.ReadKey();

        }

**Program:**

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

//float f = 3.5;

float val= (float)3.5;

float flt= 3.5f;

double d = 3.5;

decimal v = 5.6m;

//small data type=big data type;

long a = 32;

}

}

}

**Program**

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

char ch;

ch = 'a';

char a = 'b';

Console.WriteLine("value of a variable is:" + a);

Console.WriteLine($"value of a variable is:{a} \n Value of second variable is:{ch}" );

}

}

}

**Program:**

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

//char ch;

//ch = 'a';

//char a = 'b';

//Console.WriteLine("value of a variable is:" + a);

//Console.WriteLine($"value of a variable is:{a} \n Value of second variable is:{ch}" );

string first\_name = "Suraj";

string last\_name = "Mishra";

//Console.WriteLine(first\_name + " " + last\_name);

Console.WriteLine($"{first\_name} {last\_name}");

}

}

}

**Program:**

string name;

Console.WriteLine();

Console.ReadLine();// uee for user input

Console.WriteLine(Console.ReadLine());

string name;

name = Console.ReadLine();

Console.WriteLine(Console.ReadLine());

**Program:**

string name1, name2, name3;

name1 = Console.ReadLine();

name2 = Console.ReadLine();

name3 = Console.ReadLine();

Console.WriteLine("Full Name is:" + name1+" " + name2 +" "+ name3);

**Programm:-**

string name;

name = Console.ReadLine();

string[] breakstring = name.Split('a', 2);

System.Console.WriteLine(breakstring.Length);

**Program:-**

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

string password="suraj";

string passuser = Console.ReadLine();

int result = password.CompareTo(passuser);

Console.WriteLine(result);

Console.WriteLine(password == passuser);

}

}

}

------------------------------------ Notes ---------------------------

Data Type:-

**Sign integral values:** can hold both positive and negative whole numbers.

Sbyte : - Sbyte can hold value in range of -128 to +127 and takes 1 byte in memory space.

Short:- It takes four bytes in memory space. -32768 to +32767

Int

long : very large whole numbers ,it can store.

Int a=32;

Long val = 343434434434434;

Floating Value :can hold decimal numbers. By default all the floatig value in c# is double.

**Float:-**can store floating numbers.

Always floating value are declared with suffix f in c#.

**double :** Large real numbers.

Double d=234567.765489456

**decimal:** used for calculating exact result like money or any scientific calculation.

Values of decimal always written with suffix m.

Decimal val=3344.56545323456m;

**Unicode character:** can store a single letter.

Char val=’a’;

**String**:- string values are always enclosed within “ ”.

String v=” s ”;

String v=”techpile”;

String v=’’100”;

**Some pre-define function of string:-**

**Type Conversion:**

Type conversion is a technique to convert a value into different data type values.

So type conversion can be done in many ways like

1. Type casting: There are mainly two of casting .
2. **Implicit type casting:** When a value is being convert into another type of valuer and there is no chance of data lost then this type of casting is implicit type casting.Implicit type casting is done by compiler automatically.

**Largest\_size\_variable=small\_size\_value;**

Example:

Int a=40,b=60;

double res=a+b;

1. **Explicit Type Casting:-** Explicit type casting is forcefully done by user. There may be any possibility of data loss. When you trying to hold a larger size value in any smaller size variable then their may be data loss and this is a compile time error .To do this user have to perform type casting explicitly.

Data\_type\_var\_name=(data\_type) expression/value;

Double d=3.5;

Float v=(float)(d+10); // float v=(float) 3.5;

**2.Parsing-** Parsing is the method to convert string type value in another data type.

Parse function is used to convert string value into another type value.

Int a=int.Parse(Console.ReadLine());

Here Console.ReadLine() is a user input function and always return string type value.

Int a=Convert.ToInt32(string);

Parse function and Convert . data\_type is same in functioning .the only difference between convert and parse is Convert returns 0 for the null value and Parse return a Exception for null value.

**Program:-**

//wap to find total character in your name

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

int a, b;

float c;

//a = int.Parse(Console.ReadLine());

//b = int.Parse(Console.ReadLine());

//Console.WriteLine($"Addition of Two Number is={a + b}");

Console.WriteLine("Enter Your Fisrt Number:");

a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Your Second Number:");

b = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"Addition of two number is :\n{a+b}");

}

}

}

**Program:- Wap to swap two nubers:**

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

//wap to swap two int type value

//wap to swap two float type value

**Method 1:**

int a, b, c;

Console.WriteLine("Enter Fisrt Number");

a = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Second Number");

b = int.Parse(Console.ReadLine());

c = a;

a = b;

b = c;

Console.WriteLine($"Value of a:{a}");

Console.WriteLine($"Value of b:{b}");

**Method 2:**

float a, b, c;

Console.WriteLine("Enter Fisrt Number");

a = float.Parse(Console.ReadLine());

Console.WriteLine("Enter Second Number");

b = float.Parse(Console.ReadLine());

c = a;

a = b;

b = c;

Console.WriteLine($"Value of a:{a}");

Console.WriteLine($"Value of b:{b}");

**Method 3:**

int a, b, c;

Console.WriteLine("Enter Value of a:");

a = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Value of b:");

b = int.Parse(Console.ReadLine());

a = a + b;

b = a-b;

a = a - b;

Console.WriteLine($"Value of a:{a}");

Console.WriteLine($"Value of b:{b}");

}

}

}

**Date:-**

using System;

namespace ConsoleApp6

{

class Program

{

static void Main(string[] args)

{

int[,] arr = new int[3, 3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

arr[i, j]= int.Parse(Console.ReadLine());

}

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Console.Write(arr[i, j] + " ");

}

Console.WriteLine();

}

}

}

}

task:-

\*\*\*\*\*\*\*\*\*\*\*\*\*

using System;

namespace task2

{

class Program

{

static void Main(string[] args)

{

int count = 0;

string[] city = new string[10];

for (int i = 0; i <= (city.Length) - 1; i++)

{

city[i] = Console.ReadLine();

}

for (int j = 0; j <= (city.Length) - 1; j++)

{

if (city[j].Length >= 4)

{

count++;

}

}

Console.WriteLine($"total city whose length greater then or equal to 4 is :{count}");

string[] cities = new string[3];

for (int i = 0; i <= (cities.Length) - 1; i++)

{

cities[i] = Console.ReadLine();

}

Console.WriteLine("String in Upper case");

for (int j = 0; j <= (cities.Length) - 1; j++)

{

Console.WriteLine(cities[j].ToUpper());

}

string[] name = new string[5];

int count = 0;

for (int i = 0; i <= (name.Length) - 1; i++)

{

name[i] = Console.ReadLine();

}

for (int j = 0; j <= (name.Length) - 1; j++)

{

count = 0;

for (int i = 0; i <= 4; i++)

{

if (name[j] == name[i])

{

count++;

}

}

Console.WriteLine($"{name[j]} present {count} times in array");

}

wap to input 5 element of array and count how many dividors of element is present in array

int[] arr = new int[5];

for (int i = 0; i <= (arr.Length) - 1; i++)

{

arr[i] = int.Parse(Console.ReadLine());

}

for (int i = 0; i <= (arr.Length) - 1; i++)

{

int count = 0;

for (int j = 0; j <= 4; j++)

{

if (arr[i] == arr[j])

{

count++;

}

}

if (count == 0)

{

Console.WriteLine(arr[i]);

}

}

}

}

.NET MVC

MODEL:DB

VIEW:DESIGNING

CONTROLER:PROGRAMMING

DESIGNING

DATABASE

-------------------

.NET :MICROSOFT

DESKTOP APPLICATION

CONSOLE APPLICATION

WEB APPLICATION

WINDOW SERVICES

WINDOWS SERVICES

MOBILE APPLICATION

30+PROGRAMMING LANGUAGES

--------------------------- Note-----------------------------------------------

.NET Framework

.NET : it was developed by microsoft team mainly by anders & hegalberg in 2000.

it is a framework by microsoft that provides a envirenment to write and run the

whole application on a single plateform.

.NET IDE -VISUAL STUDIO

During the development of .NET ,C# was developed.

c - c++ - c#

c#=> c# is a general purpose,high level,object oriented programming language.

many other programming language are just used to develope a single type of application but c# is used to develope console based application,web based application and mobile application.

Besides c# other 30+ programming languages are provided by Microsoft with .net

framework.According to the developer choice any language can be used in development.

But mostly programming language with .NET is c#. It is the extension of C++.

Object Oriented Programming:- Object Oriented Programming language mainly provide

two feature to the programming.First is Security and second is Code Reusability

.In object oriented programming language every methods are encapsulated/written within classes.

OOP'S C#:- Namespace -> Classes -> Methods -> Statement

Raw data(input)-> Processing-> Output

---------------------------------- 04/12/2021 ----------------------------------

What is NameSpace:-

---

Name space is the coollection of multiple classes.NameSpace is the logical collection normaly vcan be think as folder containing multiple .cs file.

By default the main namespace of the c# is System.

If you want to access any class declared within a namespace then firstly we have to include the namespace in file.

Syntax:-

-------

using NameSpace\_Name:-

Classes:-

------------

Classes is the collection of data members and member function.

so the in simple word classes is the collection of multiple variables and methods.

Any variable/method declared in the class is directly accessible any other methods you don't need instance of the class.

Only static member can not access the non static members of the class.

Syntax:-

---

class class\_name

{

data\_members(variable);

Member\_function;

}

Any non-static member of class is not directly accessible out of the class.

To access any member of a class,form the outside of class,you have to create instance variable/object of the class.

object are mainly a variable of a class type which is used to access all member(variable,method) of a class anywhere.

class\_name object\_name=new class\_name()

second method():

Class\_name object\_name; // normal variable of class type

object\_name=new class\_name();//instance variable of class

when a class variable which is not declared by using new keyword stores the refrence of another instance variable then this is known as Refrence variable

of class.

Suppose there is a class named Program:

Program p=new Program();

program p1;

p1=p;//now p1 is the refrence variable of class program which holds the refrence

of another instance variable that is p.

Note:- we can create multiple object of a single class.

for each object separate memory spaces initialize by compiler.

Memory spaces are initialized for only those object created by using new keyword. So New are use to initialize instance variable of class

-------------------

Structer of c# first Program:

--------------------------

using system

namespace myproject1

{

class Myclass1

{

static void main(String [] arg)

}

}

Data type is the pre-fix which specifies the property of any variable, function.Data type defines total memory spaces consume by a object(variable,method) and which type of value it can store.

1.Value Type

------------------

Simple type

Signed Integral Value: short ,byte ,int ,long

Unsigned Integral Value: ushort , ubyte, uint ,ulong

Unicode Character: char

Floating Type Value: float , double , decimal

Boolean Value: bool

Structure :struct {---------}

Enum:- enum e {------------}

Nullable Type:- null

2.Reference Type

---------------------

Class type

Interface

Array

task1-

----------

string name;

name = Console.ReadLine();

string[] breakstring = name.Split('a', 2);

System.Console.WriteLine(breakstring.Length);

Program:-

using System;

namespace App2

{

class Program

{

static void Main(string[] args)

{

string password="suraj";

string passuser = Console.ReadLine();

int result = password.CompareTo(passuser);

Console.WriteLine(result);

Console.WriteLine(password == passuser);

}

}

}

---------------------------------------------------------------------------- 08/12/2021 ------------------------------------------------------------------------------------

Conditional Statement:-conditional statement are used to apply condition before execution of block

missing

..............

Conditional statement are used to

1.by using if keyword

2. switch keyword.

1. By using if keyword : By using if keyword conditions may be applied before the statement.Based on the structer in many way you

can apply condition:

a. Only/simple if

b.else-if /else after if

c. ladder else-if / multiple option

a. Only if: If there is only on condition is given with few statement . if given condition is true then statement will execute, if condition is false then if block will skip and further lines will execute.

Syntax:

if(condition)

{

//statements

}

b. if-else statement: there is two block of statement is given with one condition,one block of if statement and second block for else statement.if given condition is true then if will execute otherwise else will execute without checking another condition.

Syntax:-

if()

{

//code block

}

else

{

// code block

}

else -if ladder: if many condition are given as option and each condition has its own statement if condition will true then statement will execute .

within given all condition with else-if ladder at one time only one block will execute.

Syntax:-

if(condition)

{

// code block

}

else -if

{

//code block

}

3.switch statement:-switch statement is used as optional of if-else ladder statement.if multiple conditions have to applied based on the equality operator then instead of else-if ladder,you can use switch statement.that is easy to use and manage.

switch(value/variable/expression)

{

case value:

...................... code statement

break;

-

-

-

}

Example:-

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

while (true)

{

string batch;

Console.WriteLine("Enter Your Batch");

batch = Console.ReadLine();

switch (batch)

{

case "b1":

case "B1":

{

Console.WriteLine("This is b1 batch for .NET");

Console.WriteLine("This is b1 batch not for all student");

break;

}

case "b2":

case "B2":

Console.WriteLine("This is b2 batch php");

break;

case "b3":

case "B3":

Console.WriteLine("This is b3 batch python");

break;

default:

Console.WriteLine("Wrong Input");

break;

}

}

}

}

}

Array: Array is the collection of multiple values of similar data type.it can store multiple value in a single variable on different index.

index of array always starts from 0 and max index of array size -1

Syntax to declare Array:

Data\_type[] var\_name=new data\_type[size];

Ex: Let's create a int type array with size 10:

int [] students=new int[10]

Initialization

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

int[] arr = new int[10];

int sum = 0;

for (int i = 0; i <= (arr.Length) - 1; i++)

{

arr[i] = int.Parse(Console.ReadLine());

};

for (int i = 0; i <= (arr.Length) - 1; i++)

{

sum = sum + arr[i];

};

Console.WriteLine($"Sum of All Numbers is:{sum}");

}

}

}

//task:-

//wap to input 10 element of array and seprate all positive and negative value in different array

//wap to input 10 students name and seprate name and seprate all student who has no title in different array

----------------------------------------------------------------- 10/12/2021--------------------------------

One dimensional array:-One dimensional array also called 1D array is used to store multiple values in a series on different index of array.

int[] arr\_name=new int[size];

Length: length is property of array which is used to get total number of element in array.

Many function are used on the Array type can be called by using Array class.

for the example:

Array.Sort(array);

Array.IndexOf();

Array.Reverse();

two dimensional Array: Also Called 2D Array. it is used to store values in form of rows and columns ,like a table, a matri. In 2D array number of columns in each rows are same .It can not be different.

Syntax to declare a 2D Array:

Data\_type[,] arry\_name=new data\_type[row\_size,column\_name]

Example:-Let's create a 2D Array with 2 rows and 3 columns to store int type value:

int[,] arr=new int[2,3];

Direct initialization in 2D Array:-

-------------------------------------------

int[,] arr=new int[,]{{4,6,7},{7,8,9}};

console.WriteLine(arr.Length);

int[,] arr=new int[,]{{4,6,7},{7,8,9}};

console.WriteLine(arr.Length);

foreach loop:-

wap to input a matrix of 3\*3 and print only digonal element of array

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

//wap to input a matrix of 3\*3 and print only digonal element of array

int[,] arr = new int[3, 3];

int i, j;

for(i=0;i<=2;i++)

{

for(j=0;j<=2;j++)

{

arr[i, j] = int.Parse(Console.ReadLine());

}

}

for (i = 0; i <= 2; i++)

{

Console.WriteLine(arr[i, i]);

}

}

}

}

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

//wap to input a matrix of 3\*3 and print only digonal element of array

int[,] arr = new int[3, 3];

int i, j,k=2;

for(i=0;i<=2;i++)

{

for(j=0;j<=2;j++)

{

arr[i, j] = int.Parse(Console.ReadLine());

}

}

for (i = 0; i <= 2; i++)

{

//Console.WriteLine(arr[i, k]);

//k--;

//for (j = 0; j <= 2; j++)

//{

// if (i + j == 2)

// Console.WriteLine(arr[i, j]);

//}

Console.WriteLine(arr[i, 2 - i]);

}

}

}

}

// wap to input 3\*4 matrix and print last element of each row

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

//wap to input a matrix of 3\*3 and print only digonal element of array

int[,] arr = new int[3, 4];

int i, j,k=2;

for(i=0;i<=2;i++)

{

for(j=0;j<=3;j++)

{

arr[i, j] = int.Parse(Console.ReadLine());

}

}

for (i = 0; i <= 2; i++)

{

Console.WriteLine(arr[i, 3 ]);

}

}

}

}

Task1:-

---------

//wap to input 3\*3 matrix and count how many elementare greater then 10

//wap to input 3\*3 matrix and change all digonal elements by 1

//wap to print "This is a identity matrix" if all right diagonal and left diagonal elements are 1

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Jagged Array :Jagged Array is same as 2D array but the only difference is number of columns in each row in two dimensional array is same but in jagged arraymeach row can have different number of columns.

Normally it is good to say that jagged array is the collection of multiple one dimensional array.

So jagged array is the collection of multiple rows where each rows is the collection of a array.And each array can have different number

of elements.

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

//int[][] marks = new int[3][];

//marks[0] = new int[] { 3, 5, 6, 7, 8 };

//marks[1] = new int[] { 2, 3, 4 };

//marks[2] = new int[] { 5, 6, 7, 8 };

//Console.WriteLine("Matrix is: ");

////for(int i=0;i<=2;i++)

////{

//// for (int j = 0; j <= marks[i].Length - 1; j++)

//// Console.WriteLine(marks[i][j] + " ");

//// Console.WriteLine();

////}

//foreach (int[] arr in marks)

//{

// foreach (int val in arr)

// {

// Console.WriteLine(val + " ");

// Console.WriteLine();

// }

//}

int[][] marks = new int[3][];

marks[0] = new int[] { 3, 5, 6, 7, 8 };

marks[1] = new int[] { 2, 3, 4 };

marks[2] = new int[] { 5, 6, 7, 8 };

int a = 1;

foreach (int[] arr in marks)

{

int sum = 0;

foreach (int val in arr)

{

sum += val;

}

Console.WriteLine($"sum of {a} matrix is :{sum}");

a++;

}

}

}

}

Sum of Jagged Array:-

-------------------------------

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

//int[][] marks = new int[3][];

//marks[0] = new int[] { 3, 5, 6, 7, 8 };

//marks[1] = new int[] { 2, 3, 4 };

//marks[2] = new int[] { 5, 6, 7, 8 };

//Console.WriteLine("Matrix is: ");

////for(int i=0;i<=2;i++)

////{

//// for (int j = 0; j <= marks[i].Length - 1; j++)

//// Console.WriteLine(marks[i][j] + " ");

//// Console.WriteLine();

////}

//foreach (int[] arr in marks)

//{

// foreach (int val in arr)

// {

// Console.WriteLine(val + " ");

// Console.WriteLine();

// }

//}

int[][] marks = new int[3][];

marks[0] = new int[] { 3, 5, 6, 7, 8 };

marks[1] = new int[] { 2, 3, 4 };

marks[2] = new int[] { 5, 6, 7, 8 };

int a = 1;

foreach (int[] arr in marks)

{

int sum = 0;

foreach (int val in arr)

{

sum += val;

}

Console.WriteLine($"sum of {a} matrix is :{sum}");

a++;

}

}

}

}

Task of jagged Array:-

---------------------------------------------

//wap to input 6,5,8 elements in a jagged array and print sum of all even element of each row.

//wap to input 5,4 elements in rows in jagged array and print all element of rowsin reverse order.

//wap to input 3,5,7,8 in jagged array and print each rows elements in assending order.

//wap to input 3,5,7,8 injagged array and print each row element in descending order.

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ARRAY:[ ], [ , ],[][] :-

Date:-14/12/2021

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Problem with array:-

1.fixed size

2.you can not insert any item in the middle of array

3.you can not delete any item from the middle of array

collection:-

collection is also the group of multiple elements which can hold variable size values and provide many function to insert,delete,add sort

and many more.

it was introduced in c# 1.0

many collection type are present like stack,queue ,ArrayList,LinkedList,HashTable etc.

The all type of collection is a class.

All classes of collection is present in a namespace:

System.Collections;

------------------------

using System;

using System.Collections;

namespace arraylistfunc

{

class Program

{

static void Main(string[] args)

{

ArrayList arraylist = new ArrayList();

arraylist.Add("Techpile");

arraylist.Add(10);

arraylist.Add(20);

arraylist.Add(30);

Console.WriteLine(arraylist.Count);

foreach (object val in arraylist)

{

Console.WriteLine(val);

}

}

void add(double a, object b)

{

Console.WriteLine(Convert.ToInt32(a) + Convert.ToInt32(b));

}

}

}

---------------

ArrayList: ArrayList is a class of collection (non-Generic Collection). it works same as Array that means it can store multiple values and each value

can be accessed by using index.

But the major difference between array and arraylist is array is fixed size collection but Arraylist is variable size collection.

It automatic sets the capacity of collection to the 4. And increase twice if capacity fails.

You can set capacity of ArrayList explicitly.

1st method: ArrayList=new ArrayList(30);

Now capacity of ArrayList is 30.

2nd Method:arraylist.capacity=40;

using System;

using System.Collections;

second difference between Array And ArrayList is,in ArrayList you can add,delete,values anywhere in the ArrayList

by using some pre-define functions Add(),Insert(),InsertAt(),Remove(),RemoveAt().

Count property is used to get number of total elements in ArrayList.

Console.WriteLine(arraylist.Cont);

Example:-

-----------------

namespace arraylistfunc

{

class Program

{

static void Main(string[] args)

{

ArrayList arraylist = new ArrayList(30);

arraylist.Add("Techpile");

arraylist.Add(10);

arraylist.Add(20);

arraylist.Add(30);

Console.WriteLine(arraylist.Capacity);

for (int i = 0; i <= arraylist.Count; i++)

{

Console.WriteLine(arraylist[i]);

}

}

}

}

HashTable:-

using System;

using System.Collections;

namespace arraylistfunc

{

class Program

{

static void Main(string[] args)

{

Hashtable hashtable = new Hashtable();

hashtable.Add("Name","Techpile Technology");

hashtable.Add("DOB","15 OCT");

hashtable.Add("Turn Over",1000000);

Console.WriteLine(hashtable["Name"].GetHashCode());

foreach (object k in hashtable.Keys)

{

Console.WriteLine(k + " " + hashtable[k]);

}

}

}

}