var names=new[]{"kis","Amrit","kum","kumar"};

ReadOnlyCollection<string> ro=Array.AsReadOnly(names);

ro.Add  //cant add in readonlycollection

//names is an array of names.

int index=Array.BinarySearch (names,""); //get the index of the entered value

int index=Array.FindIndex (names, x=> x.startsWith ("Amrita");

bool value=Array.Exists (names, x=>x=="Amrita");

string lastAmrit=Array.FindLast (name, x=>x.startsWith ("Amrita"));

//If we want to itterate an Array we can use ForEach Method

Array.Foreach(names, x=>

               {

                  Console.WriteLine(x);

   Console.WrityeLine(x.Length);

               });

//if we want to know if all elements satisfy a particular condition

bool value=Array.TrueForAll(names,x=>x.StartsWith("ki"));

//Cast

object name="";

Person p=(Person) name;  //it gives cast exception

Person p=name as Person; //it won’t give cast exception but we have to handle null check here

Lazy execution or deferred execution:

when we have lots of data to process and we need only few value with certain condition

then should use deferred execution so that till the time we get the required

number of execution, the processing should stop.

http://bit.ly/pscsyield  -->for detail yield and IEnumerable

using NestedClass1=Demo.demoSuportCode.Class1;

//Constructor

Here by giving just name as input we call the constructor having three

argument (through this keyword) and set the value in person class

public Person (string name)

   : this (name, int.MinValue, DefaultGender)

{

}

public Person (string name, int age, string gender)

{

   \_name=name;

   \_age=age;

   \_gender=gender;

}

//overload of constructor... giving two inputs value and setting up

 the class value

public Person (string name, string age)

   : this (name, age, DefaultGender)

{

}

https://www.dotnetcurry.com/microsoft-azure/1459/serverless-azure-functions