MS DOS > operating system , character based interface

Copy file1 file2

Copy con

Ren oldname newname

Windows > Graphical Interface

Languages > C , C++ , Pascal , Basic , COBOL , Foxpro

character based interface

Visual C++ , Visual Foxpro, Visual Basic

Combined multiple languages > DotNet Framework

Framework > It’s a collection of classes , libraries , etc.

What are the benefits of DotNet Framework

1. Supports multiple languages
2. It gives you lots of inbuilt classes , libraries

Components of DotNetFramework

CLR > Common Language Runtime > Runtime environment for DotNet Pragrams

CTS > Common Type System

CLS > Common Language Specification

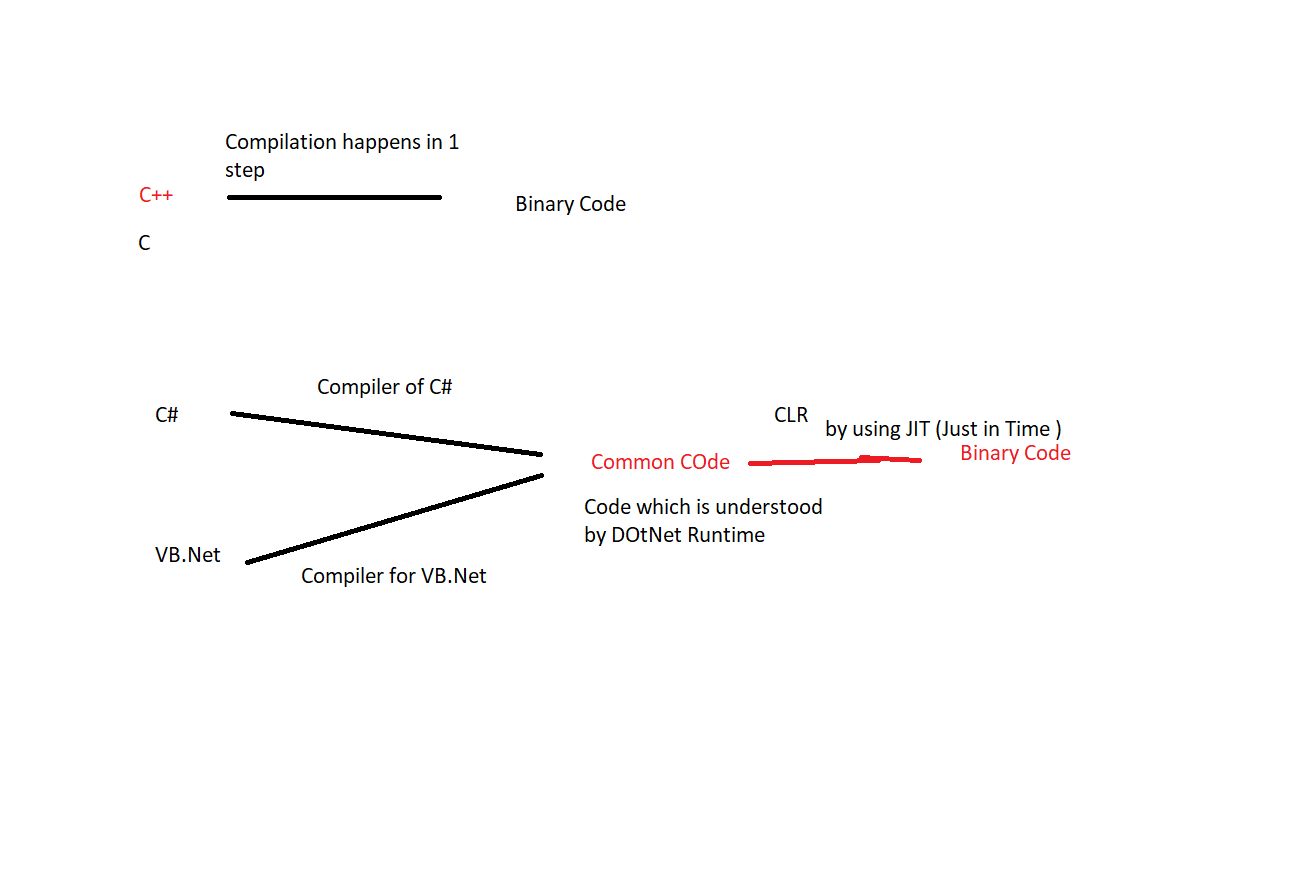
CLR performs lots of functions

1. Memory allocation
2. Garbage Collection
3. Type Safety
4. Code Compilation

Compiler > Translation of code from one language to other language

Compilation in other languages happens in 1 step

Compilation in other languages happens in 2 steps



Arrays : Structure which contains elements of same type. All the elements have same name but we can distinguish them by their position within the array. Index or position starts from 0

using System;

class Program

{

static void Main(string[] args)

{

int[] num = new int[10];

Console.WriteLine("Enter Elements");

for(int i=0;i<10;i++)

{

num[i] = Convert.ToByte(Console.ReadLine());

}

Console.WriteLine("Elements are ");

for(int i=0;i<10;i++)

{

Console.WriteLine(num[i]);

}

}

}

Array Declaration and Initialization

using System;

class Program

{

static void Main(string[] args)

{

int[] num = new int[] { 1, 2, 3, 4, 5, 6, 7, 8 };

//Console.WriteLine("Enter Elements");

//for(int i=0;i<10;i++)

//{

// num[i] = Convert.ToByte(Console.ReadLine());

//}

Console.WriteLine("Elements are ");

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

{

Console.WriteLine(num[i]);

}

}

}

Sum and Average of elements of an Array

using System;

class Program

{

static void Main(string[] args)

{

int[] num = new int[] { 1, 2, 3, 4, 5, 6, 7, 8 };

int sum = 0;

int avg = 0;

Console.WriteLine("Elements are ");

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

{

sum += num[i];

Console.WriteLine(num[i]);

}

avg = sum / num.Length;

Console.WriteLine("Sum of all the elements of Array is " + sum);

Console.WriteLine("Avaerge is " + avg);

}

}

2-D Array

using System;

class Program

{

static void Main(string[] args)

{

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

int i, j;

Console.WriteLine("Enter the elements");

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

{

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

{

num[i, j] = Convert.ToByte(Console.ReadLine());

}

}

Console.WriteLine("Elements are");

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

{

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

{

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

}

Console.WriteLine();

}

}

}

Jagged Array

using System;

class Program

{

static void Main(string[] args)

{

// Declare the Jagged Array of four elements:

int[][] jagged\_arr = new int[4][];

// Initialize the elements

jagged\_arr[0] = new int[] { 1, 2, 3, 4 };

jagged\_arr[1] = new int[] { 11, 34, 67 };

jagged\_arr[2] = new int[] { 89, 23 };

jagged\_arr[3] = new int[] { 0, 45, 78, 53, 99 };

// Display the array elements:

for (int n = 0; n < jagged\_arr.Length; n++)

{

// Print the row number

System.Console.Write("Row({0}): ", n);

for (int k = 0; k < jagged\_arr[n].Length; k++)

{

// Print the elements in the row

System.Console.Write("{0} ", jagged\_arr[n][k]);

}

System.Console.WriteLine();

}

}

}

using System;

class Program

{

public static String GetTotalScore(int[][] array)

{

string res = "";

int score = 0;

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

{

score = 0;

for (int j = 0; j < array[i].Length; j++)

{

Console.WriteLine("Enter the score for team {0}", i+1);

array[i][j] = Convert.ToByte(Console.ReadLine());

score += array[i][j];

}

res += " Team " + (i + 1) + " Total Score is " + score +".";

}

return res;

}

static void Main(string[] args)

{

Console.WriteLine("Enter No of Teams");

int team = Convert.ToByte(Console.ReadLine());

int[][] array = new int[team][];

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

{

Console.WriteLine("No.of attempts for team {0}", i+1);

int attemp = Convert.ToByte(Console.ReadLine());

array[i] = new int[attemp];

}

string res = GetTotalScore(array);

Console.WriteLine(res);

}

}

// params Array , when no. of inputs is not fixed.

using System;

class Program

{

//static int add(int x, int y)

//{

// return x + y;

//}

//static int add(int x, int y , int z)

//{

// return x + y +z;

//}

static int add(params int[] array)

{

int sum = 0;

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

{

sum += array[i];

}

return sum;

}

static void Main()

{

Console.WriteLine(add(1,3));

Console.WriteLine(add(1,2,4));

Console.WriteLine(add(1,2,3,4,5,6,6,7,7,7,3));

}

}

Break continue statement

Break > is used to exit out of loop

Continue > is used to skip current pass of the loop and it takes you to the next iteration

// params Array

using System;

class Program

{

static void Main()

{

int sum = 0;

int num;

int i = 1;

do

{

Console.WriteLine("Enter No");

num = Convert.ToSByte(Console.ReadLine());

sum += num;

i++;

} while (i <= 10);

Console.WriteLine("Sum is " + sum);

}

}

Sum of only +ve numbers

// params Array

using System;

class Program

{

static void Main()

{

int sum = 0;

int num;

int i = 0;

do

{

i++;

Console.WriteLine("Enter No");

num = Convert.ToSByte(Console.ReadLine());

if (num < 0) continue;

sum += num;

} while (i <= 10);

Console.WriteLine("Sum is " + sum);

}

}

Sum of only +ve numbers, also exit when no. is 0

// params Array

using System;

class Program

{

static void Main()

{

int sum = 0;

int num;

int i = 0;

do

{

i++;

Console.WriteLine("Enter No");

num = Convert.ToSByte(Console.ReadLine());

if (num < 0) continue;

if (num == 0) break;

sum += num;

} while (i <= 10);

Console.WriteLine("Sum is " + sum);

}

}