# ***this* Keyword**

**this -** ключове слово використовується для посилання на поточний екземпляр класу. Воно використовується для доступу до членів з конструкторів, методів екземпляра та аксесорів екземпляра. Ключове слово ***this*** також використовується для відстеження екземпляра, який викликається для виконання деяких обчислень або подальшої обробки, пов'язаної з цим екземпляром.

Нижче наведено різні способи використання ключового слова ***this*** у мові C# :

Програма 1: Використання ключового слова ***this*** для посилання на поточні члени екземпляру класу

// C# program for using 'this' keyword to

// refer current class instance members

using System;

namespace geeksforgeeks {

class Geeks {

// instance member

public string Name;

public string GetName()

{

return Name;

}

public void SetName(string Name)

{

// referring to current instance

//"this.Name" refers to class member

this.Name = Name;

}

}

class program {

// Main Method

public static void Main()

{

Geeks obj = new Geeks();

obj.SetName("Geeksforgeeks");

Console.WriteLine(obj.GetName());

}

}

}

Програма 2: Використання ***this()*** для виклику конструктора в тому самому класі

// C# program for using 'this' keyword to

// invoke the constructor in same class

using System;

namespace geeksforgeeks {

class Geeks {

// calling another constructor

// calls Geeks(string Name)

// before Geeks()

public Geeks() : this("geeks")

{

Console.WriteLine("Non-Parameter Constructor Called");

}

// Declare Parameter Constructor

public Geeks(string Name)

{

Console.WriteLine("Parameter Constructor Called");

}

}

class program {

// Main Method

public static void Main()

{

Geeks obj = new Geeks();

// Warning: obj never used..

}

}

}

Програма 3: Використання ключового слова ***this*** для виклику поточного методу класу

// C# code for using this to invoke current

// class method

using System;

class Test {

void display()

{

// calling function show()

this.show();

Console.WriteLine("Inside display function");

}

void show()

{

Console.WriteLine("Inside show function");

}

// Main Method

public static void Main(String []args)

{

Test t1 = new Test();

t1.display();

}

}

Програма 4: Використання ключового слова ***this*** як параметра методу

// C# code for using 'this'

// keyword as method parameter

using System;

class Test

{

int a;

int b;

// Default constructor

Test()

{

a = 10;

b = 20;

}

// Method that receives 'this'

// keyword as parameter

void display(Test obj)

{

Console.WriteLine("a = " + a + " b = " + b);

}

// Method that returns current

// class instance

void get()

{

display(this);

}

// Main Method

public static void Main(String[] args)

{

Test obj = new Test();

obj.get();

}

}

Програма 5: Використання ключового слова ***this*** для оголошення індексатора

// C# code for using 'this'

// keyword to declare indexer

using System;

namespace geeksforgeeks {

class Geeks {

private string[] days = new string[7];

// declaring an indexer

public string this[int index]

{

get

{

return days[index];

}

set

{

days[index] = value;

}

}

}

class Program {

// Main Method

public static void Main()

{

Geeks g = new Geeks();

g[0] = "Sun";

g[1] = "Mon";

g[2] = "Tue";

g[3] = "Wed";

g[4] = "Thu";

g[5] = "Fri";

g[6] = "Sat";

for (int i = 0; i < 7; i++)

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

}

}

}

# **C# Namespace**

**Простір імен (namespace)** - це контейнер для класів і просторів імен. Простір імен також надає унікальні імена своїм класам, таким чином, ви можете мати одне і те ж ім'я класу в різних просторах імен.

**Простори імен (namespace)** відіграють важливу роль в управлінні спорідненими класами в C#. .NET Framework використовує простори імен для організації своїх вбудованих класів. Наприклад, у .NET є кілька вбудованих просторів імен, таких як System, System.Linq, System.Web тощо. Кожен простір імен містить пов'язані з ним класи.

У C# простір імен можна визначити за допомогою ключового слова **namespace**.

**Example: Namespace**

namespace School

{

// define classes here

}

Наступний простір імен містить класи **Student** і **Course**.

**Example: Namespace**

namespace School

{

class Student

{

}

class Course

{

}

}

Класи в одному просторі імен можна посилатися за допомогою синтаксису **namespace.classname**. Наприклад, до класу **Student** можна отримати доступ як **School.Student.**

**Example: Refer a Class with Namespace**

namespace CSharpTutorials

{

class Program

{

static void Main(string[] args)

{

School.Student std = new School.Student();

School.Course cs = new School.Course();

}

}

}

Щоб використовувати класи у просторі імен без повного імені, імпортуйте простір імен з ключовим словом **using** у верхній частині файлу класу C#.

using System; //built-in namespace

using School;

namespace CSharpTutorials

{

class Program

{

static void Main(string[] args)

{

Student std = new Student();

}

}

}

Простір назв може містити інші простори назв. Внутрішні простори назв можна відокремлювати за допомогою ***(.)***

namespace School.Education

{

class Student

{

}

}

У наведеному вище прикладі повне ім’я класу – **School.Education.Student.**

**File Scoped Namespace**

Ідея полягає у видаленні одного рівня відступу з вихідних файлів, якщо вони містять лише один простір імен. Мета - зменшити горизонтальну та вертикальну прокрутку і зробити код більш читабельним.

//Non-File Scoped Namespace

namespace MyProject

{

class Demo

{

}

}

// File Scoped Namespace

namespace MyProject;

class Demo

{

}

Більшість файлів мають лише один простір назв, тому до більшості з них слід застосовувати простір назв, обмежений файлом. Однак, ви не хочете оновлювати їх вручну один за одним.