



...Default Constructor...

“ Default Constructor ทำหน้าที่เดียวกับ constructor กล่าวคือ ใช้สำหรับกำหนดค่าใน fields ให้กับวัตถุ แต่ Defalut Constructor แตกต่างออกไป โดยที่จะไม่รับ parameters ใดๆ ซึ่งจะกำหนดค่าเริ่มต้นให้กับฟิลด์ของวัตถุ ”

เห็นได้ว่าคลาส Person ไม่มี
constructor ใดๆ เมื่อมีการ
สร้างวัตถุของ Person ตัว
compiler จะกำหนดค่าเริ่มต้น
ให้ ผลลัพธ์เมื่อสร้างวัตถุ

----- ตัวอย่าง -----

```
class Person
{
    public string name;
    public int age;
    // Implicit default constructor
}
```

```
Person p = new Person();
Console.WriteLine($"Name: {p.name}");
Console.WriteLine($"Age: {p.age}");
```

----- ผลลัพธ์ -----

```
Name:
Age: 0
```

Implicit default constructor

เมื่อไม่มีการประกาศ constructor ใดๆ ในคลาส compiler
จะสร้าง default constructor ให้โดยอัตโนมัติ

----- ตัวอย่าง -----

```
class Person
{
    public string name;
    public int age;
    // Implicit default constructor
}
```

```
Person p = new Person();
Console.WriteLine($"Name: {p.name}");
Console.WriteLine($"Age: {p.age}");
```

----- ผลลัพธ์ -----

```
Name:
Age: 0
```

Explicit default constructor

เป็นการสร้าง default constructor โดยที่ผู้เขียนกำหนดข้อมูลเริ่มต้นด้วยตัวเอง

----- ตัวอย่าง -----

```
class Person
{
    public string name;
    public int age;

    // Explicit default constructor
    public Person()
    {
        name = "_";
        age = 20;
    }

    public static void Main(string[] args)
    {
        Person p = new Person();
        Console.WriteLine($"Name: {p.name}");
        Console.WriteLine($"Age: {p.age}");
    }
}
```

----- ผลลัพธ์ -----

```
Name: _
Age: 20
```

ค่าเริ่มต้นที่กำหนด

ค่าเริ่มต้นที่กำหนดให้ขึ้นอยู่กับชนิดข้อมูลของฟิลด์แต่ละตัว

Numeric เช่น `int`, `float`, `double`, `decimal` จะถูกกำหนดค่าเริ่มต้นเป็น `0`

Boolean จะถูกกำหนดเป็น `false`

Character จะถูกกำหนดเป็นค่า `null`

Reference เช่น `string`, คลาสต่าง ๆ, อาร์เรย์ จะถูกกำหนดเป็น `null`

Enum ค่าเริ่มต้นจะเป็น สมาชิกตัวแรก

Type	Default value
reference type	null
numeric type	0 (zero)
floating-point numeric type	0 (zero)
bool	false
char	'\0' (U+0000) null character
enum	สมาชิกตัวแรก
struct	สมาชิกแต่ละตัว จะมีค่า default values



คณะวิทยาศาสตร์
มหาวิทยาลัยสกลนคร

----- ตัวอย่าง -----

```
class Person
{
    public string name;
    public int age;
    public bool isAlive;
    public char code;

    public enum Gender
    {
        Male,
        Female
    }

    public static void Main(string[] args)
    {
        Person p = new Person();
        Console.WriteLine($"Name: {p.name}");
        Console.WriteLine($"Age: {p.age}");
        Console.WriteLine($"IsAlive: {p.isAlive}");
        Console.WriteLine($"Code: {p.code}");
        Console.WriteLine($"Gender: {p.gender}");
    }
}
```

----- ผลลัพธ์ -----

```
Name: //null
Age: 0
IsAlive: False
Code: // null character
Gender: Male
```


ข้อสำคัญ

ถ้าผู้เขียนใส่ constructor อย่างน้อยหนึ่งอัน compiler
จะไม่สร้าง default constructor ให้เสมอไป



คณะวิทยาศาสตร์
มหาวิทยาลัยศิลปากร


```
class Person
{
    public string name;
    public int age;

    public Person(string name, int age)
    {
        this.name = name;
        this.age = age;
    }

    public static void Main(string[] args)
    {
        Person p = new Person(); // compile-time error
    }
}
```

ดังนั้นเมื่อมีการประกาศ constructor ใดๆแล้ว ถ้าต้องการประกาศ default constructor จะต้องประกาศด้วยตัวเองเพิ่มลงไป

เปรียบเทียบกับภาษาอื่น

 เนื่องจากภาษา C เป็น Procedural programming language จึงไม่มี class หรือ object อย่าง OOP (object-oriented programming) และทำให้มีการทั้ง constructor, default constructor ได้ ซึ่งสามารถทำได้ในภาษาอื่นๆ ของ C-based นั่นคือ C++, C#

เปรียบเทียบกับภาษาอื่น

Java

ภาษา Java เป็นทั้งภาษาที่สนับสนุน OOP

จึงมี default constructor เช่นเดียวกับ C#

แต่มีข้อแตกต่างในส่วนค่าเริ่มต้นของ enum ซึ่งใน Java ค่าเริ่มต้น

คือ null ส่วนอื่นๆทำงานเช่นเดียวกับ C#

----- ตัวอย่าง -----

```
class Person{
    public String name;
    public int age;
    public boolean isAlive;
    public char code;
    public Gender gender;

    public enum Gender
    {
        Male,
        Female
    }

    public static void main(String[] args)
    {
        Person p = new Person();
        System.out.println("Name : " + p.name);
        System.out.println("Age : " + p.age);
        System.out.println("Alive : " + p.isAlive);
        System.out.println("Code : " + p.code);
        System.out.println("Gender : " + p.gender);
    }
}
```

----- ผลลัพธ์ -----

```
Name : null
Age : 0
Alive : false
Code :
Gender : null
```



คณะวิทยาศาสตร์
มหาวิทยาลัยสกลนคร

เปรียบเทียบกับภาษาอื่น

Python

เป็นภาษาสนับสนุน OOP และ

การประกาศตัวแปรไม่จำเป็นต้องระบุชนิดข้อมูล(Dynamic Typing)
ซึ่งถ้าไม่ได้มีการกำหนดชนิดข้อมูล ก็จะไม่สามารถกำหนดค่าเริ่มต้น
ให้ตัวแปรได้ ส่วนใหญ่จะใช้วิธี Explicit default constructor แทน

Explicit default constructor (Python)

----- ตัวอย่าง -----

```
class Person:
    def __init__(self): # explicit default constructor
        self.name = "Unknown"
        self.age = 20

p = Person()
print(p.name)
print(p.age)
```

----- ผลลัพธ์ -----

```
Unknown
20
```

Implicit default constructor (Python)

----- ตัวอย่าง -----

```
class Person:
    name = str()
    age = int()

p = Person()
print(p.name)
print(p.age)
```

----- ผลลัพธ์ -----

```
#null
0
```

Python default values

Numeric Types

จำนวนเต็ม `int` → 0

จำนวนทศนิยม `float` → 0.0

จำนวนเชิงซ้อน `complex` → 0j

Set Types

เซต `set` → {}

เซตที่ไม่สามารถเปลี่ยนแปลงค่าได้

`frozenset` → `frozenset()`

Sequence Types

สตริง `str` → null

ลิสต์ `list` → ()

`tuple` → ()

Boolean Type

ค่าความจริง `bool` → false

None Type

ไม่ระบุชนิดข้อมูล `anyVar` → None

Mapping Type

คู่อันดับ `dict` → {}