

การเขียนโปรแกรม ด้วยภาษา C#

Class Inheritance – Part 3

Constructors in Inheritance

Constructor execution

- ในการสร้างวัตถุจากคลาส จะมีการเรียกใช้งาน constructor เป็น method แรกเสมอ
 - ใน class ที่มีการสืบทอดเป็นลำดับชั้น จะมีการเรียก constructor ของ base class ตามลำดับการสืบทอด
 - ในกรณีที่เราไม่ได้เขียน code ของ constructor ไว้ ก็จะมีการเรียก implicit constructor โดยอัตโนมัติ

ตัวอย่าง

```
class BaseClass  
{  
    public BaseClass()  
    {  
        Console.WriteLine("Hello From BaseClass");  
    }  
}  
class DerivedClass : BaseClass  
{  
    public DerivedClass()  
    {  
        Console.WriteLine("Hello From DerivedClass");  
    }  
}
```

ลำดับการเรียก

ลำดับการค้นหา

```
using System;  
class Program  
{  
    static void Main()  
    {  
        DerivedClass derived = new DerivedClass();  
    }  
}
```

```
C:\Windows\system32\cmd.exe  
Hello From BaseClass  
Hello From DerivedClass  
Press any key to continue . . .
```

ตัวอย่าง

```
using System;
class Program
{
    static void Main()
    {
        SecondDerivedClass sd = new SecondDerivedClass();
    }
}
```

```
class BaseClass
{
    public BaseClass()
    {
        Console.WriteLine("Hello From BaseClass");
    }
}

class DerivedClass : BaseClass
{
    public DerivedClass()
    {
        Console.WriteLine("Hello From DerivedClass");
    }
}

class SecondDerivedClass : DerivedClass
{
    public SecondDerivedClass()
    {
        Console.WriteLine("Hello From SecondDerivedClass");
    }
}
```

ลำดับการเรียก

ลำดับการค้นหา

```
C:\Windows\system32\cmd.exe
Hello From BaseClass
Hello From DerivedClass
Hello From SecondDerivedClass
Press any key to continue . . .
```

Virtual and override Constructor

```
class BaseClass
{
    virtual public BaseClass()
    {
        Console.WriteLine("Hello From BaseClass");
    }
}

class DerivedClass : BaseClass
{
    override public DerivedClass()
    {
        Console.WriteLine("Hello From DerivedClass");
    }
}

class SecondDerivedClass : DerivedClass
{
    override public SecondDerivedClass()
    {
        Console.WriteLine("Hello From SecondDerivedClass");
    }
}
```

error CS0106: The modifier 'virtual' is not valid for this item

error CS0106: The modifier 'override' is not valid for this item

error CS0106: The modifier 'override' is not valid for this item

จงเรียงลำดับการทำคำสั่ง a, b , c, d

```
using System;
class Program
{
    static void Main()
    {
        DerivedClass sd = new DerivedClass(); (a)
    }
}

class BaseClass
{
    public BaseClass() (b)
    {
        Console.WriteLine("Hello From BaseClass");
    }
}

class DerivedClass : BaseClass
{
    int Field1 = 10; (c)

    public DerivedClass() (d)
    {
        Console.WriteLine("Hello From DerivedClass");
    }
}
```


Constructor Initializers

- โดยทั่วไป constructor ที่ไม่มี parameters (เช่น default constructor) จะถูกเรียกโดยอัตโนมัติตอนสร้าง object
 - แต่ constructor สามารถทำ overloaded ได้
 - constructor สามารถมี signature ที่แตกต่างกันได้ (แต่ไม่มี return type)
- ถ้าต้องการเรียกใช้ constructor ที่มี parameters สามารถทำได้ 2 กรณี
 - ใช้ keyword 'base' เพื่อเรียกใช้ constructor ของ base class
 - ใช้ keyword 'this' เพื่อเรียกใช้ constructor ของ class นี้ (derived class)

Constructor implicitly & explicitly using base

```
class BaseClass
{
    public BaseClass()
    {
        ...
    }
}

class DerivedClass : BaseClass
{
    public DerivedClass()
    {
        ...
    }
}
```

implicitly

```
class BaseClass
{
    public BaseClass()
    {
        ...
    }
}

class DerivedClass : BaseClass
{
    public DerivedClass(): base()
    {
        ...
    }
}
```

explicitly

ตัวอย่าง

```
class Vehicle
{
    public string Brand { get; set; }
    public string Plate { get; set; }
    public Vehicle(string brand, string plate)
    {
        this.Brand = brand;
        this.Plate = plate;
    }
}
```

```
class Bike : Vehicle
{
    public Bike(string brand, string plate) : base(brand, plate)
    {
    }
    override public string ToString()
    {
        return $"This bike is {Brand}, plate = {Plate}";
    }
}
```

```
class Car : Vehicle
{
    public Car(string brand, string plate) : base(brand, plate)
    {
    }
    override public string ToString()
    {
        return $"This car is {Brand}, plate = {Plate}";
    }
}
```

ตัวอย่าง

```
using System;
class Program
{
    static void Main()
    {
        Bike harley = new Bike("Harley", "AB-123");
        Car toyota = new Car("Toyota", "EF-456");
        Console.WriteLine($"{harley}");
        Console.WriteLine($"{toyota}");
    }
}
```

C:\Windows\system32\cmd.exe

This bike is Harley, plate = AB-123
This car is Toyota, plate = EF-456
Press any key to continue . . .

ตัวอย่าง

```
using System;
class Program
{
    static void Main()
    {
        Myclass mc = new Myclass("abc");
        Console.WriteLine(mc);
        Myclass mc1 = new Myclass(100);
        Console.WriteLine(mc1);
    }
}
```

```
class Myclass
{
    public string Name;
    public int Id;
    private Myclass() // private constructor
    {
    }

    public Myclass(string name) : this()
    {
        Name = name;
        Id = -1; // default value
    }

    public Myclass(int id) : this()
    {
        Name = "Anonymous"; // default value
        Id = id;
    }

    public override string ToString()
    {
        return $"Name = {Name}, id = {Id}";
    }
}
```

