

CODE SAMPLE: CONSTRUCTORS

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Inheritance
{
    class Inheritance
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Creating unnamed Pet");
            Pet unnamed = new Pet();
            Console.WriteLine("\r\nCreating Pet MyPet");
            Pet myPet = new Pet("MyPet");

            Console.WriteLine("\r\nCreating Cat Tiger, then asking for noise");
            Cat tiger = new Cat("Tiger");
            tiger.MakeNoise();
            Console.WriteLine("\r\nCreating Kitten Fluffy, then asking for
noise");
            Kitten fluffy = new Kitten("Fluffy");
            fluffy.MakeNoise();

            Pet[] arr = new Pet[4];
            arr[0] = unnamed;
            arr[1] = myPet;
            arr[2] = tiger;
            arr[3] = fluffy;
            Console.WriteLine("\r\nAsking everybody for noise");
            foreach (Pet p in arr)
            {
                p.MakeNoise();
            }
        }
    }

    class Pet
    {
        protected string _name;

        public Pet()
        {
            Console.WriteLine("Pet default constructor called");
        }

        public Pet(string name)
        {
            _name = name;
            Console.WriteLine("Pet string constructor called: " + _name);
        }

        virtual public void MakeNoise()
        {
            Console.WriteLine(_name + " :Noise from Pet class");
        }
    }
}
```

```

class Cat : Pet
{
    public Cat()
    {
        Console.WriteLine("Cat default constructor called");
    }

    public Cat(string name) : base(name)
    {
        Console.WriteLine("Cat string constructor called: " + name);
    }

    public override void MakeNoise()
    {
        Console.WriteLine(_name + " :Meow");
    }
}

class Kitten : Cat
{
    public Kitten(string name) : base(name)
    {
        Console.WriteLine("Kitten string constructor called: " + name);
        _name = name + "!";
    }
}

//Output with BOTH base(name) calls
//-----
//Creating unnamed Pet
//Pet default constructor called

//Creating Pet MyPet
//Pet string constructor called: MyPet

//Creating Cat Tiger, then asking for noise
//Pet string constructor called: Tiger
//Cat string constructor called: Tiger
//Tiger :Meow

//Creating Kitten Fluffy, then asking for noise
//Pet string constructor called: Fluffy
//Cat string constructor called: Fluffy
//Kitten string constructor called: Fluffy
//Fluffy! :Meow

//Asking everybody for noise
// :Noise from Pet class
//MyPet :Noise from Pet class
//Tiger :Meow
//Fluffy! :Meow

```

```
//Output with NEITHER base(name) call
//-----
//Creating unnamed Pet
//Pet default constructor called

//Creating Pet MyPet
//Pet string constructor called: MyPet

//Creating Cat Tiger, then asking for noise
//Pet default constructor called
//Cat string constructor called: Tiger
// :Meow

//Creating Kitten Fluffy, then asking for noise
//Pet default constructor called
//Cat default constructor called
//Kitten string constructor called: Fluffy
//Fluffy! :Meow

//Asking everybody for noise
// :Noise from Pet class
//MyPet :Noise from Pet class
// :Meow
//Fluffy! :Meow
}
```