/\*Create a C# application that demonstrates the concepts of method overloading, method overriding, delegates, and constructors. The application should include:

A base class Animal with a constructor that takes a name and a method Speak.

A derived class Dog that overrides the Speak method and adds an overloaded method Speak that takes a parameter to specify the number of times the dog barks.

A delegate AnimalSound that can reference the Speak method.

A class AnimalTrainer that uses the delegate to make animals speak.

A Main method to demonstrate these concepts by creating instances of Animal and Dog, using the delegate, and calling the overloaded method.\*/

using System;

namespace IceTask

{

//base class animal

public class Animal

{

public string Name { get; set; }

public Animal (string name)

{

Name = name;

}

public virtual void Speak()

{

Console.WriteLine($"{Name} makes a sound");

}

}

//deriveed class dog

public class Dog:Animal {

public Dog(string name):base(name) {

}

public override void Speak()

{

Console.WriteLine($"{Name} barks");

}

public void Speak(int times)

{

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

{

Console.WriteLine($"{Name} barks");

}

}

}

//delegate that references speak

public delegate void AnimalSounds();

//animal trainer class

public class AnimalTrainer

{

public void AnimalSpeak(AnimalSounds DelegateSound)

{

DelegateSound();

}

}

//main method

class Program

{

static void Main(string[] args) {

Animal animal = new Animal("The dog");

Dog dog = new Dog("Dog");

AnimalSounds animalSounds;

//demonstrate method overriding

animal.Speak();

dog.Speak();

//demonstrate method overloading

dog.Speak(6);

//demonstrate delegate

AnimalTrainer trainer = new AnimalTrainer();

animalSounds = new AnimalSounds(animal.Speak);

trainer.AnimalSpeak(animalSounds);

animalSounds = new AnimalSounds(dog.Speak);

trainer.AnimalSpeak(animalSounds);

}

}

}