

```

using System;

namespace Task6_1_p
{
    /*
     * We can see that the method is hidden, and a new version of the method in a new subclass is implemented with the "new" keyword*.
     * This is a version of static polymorphism.
     * Beyond that, the code has the following problems (PDF) */
    class Class1
    {
        abstract public class Animal
        {
            abstract public void Greeting();
        }
        public class Cat : Animal
        {
            override public void Greeting()
            {
                Console.WriteLine("Cat: Meow!");
            }
        }
        public class Dog : Animal
        {
            override public void Greeting()
            {
                Console.WriteLine("Dog: Woof!");
            }
            public void Greeting(Dog another)
            {
                Console.WriteLine("Dog: Wooooooooooof!");
            }
        }
        public class BigDog : Dog
        {
            override public void Greeting()
            {
                Console.WriteLine("BigDog: Woow!");
            }
        }
    }
}

```

```

        {
            Console.WriteLine("BigDog: Woow!");
        }
        new public void Greeting(Dog another)
        {
            Console.WriteLine("Woowooooooooow!");
        }
    }

    public class TestAnimal
    {
        public static void Main(String[] args)
        {
            // Using the subclasses
            Cat cat1 = new Cat();
            cat1.greeting();
            Dog dog1 = new Dog();
            dog1.greeting();
            BigDog bigDog1 = new BigDog();
            bigDog1.greeting();
            // Using Polymorphism
            Animal animal1 = new Cat();
            animal1.greeting();
            Animal animal2 = new Dog();
            animal2.greeting();
            Animal animal3 = new BigDog();
            animal3.greeting();
            Animal animal4 = new Animal();
            // Downcast
            Dog dog2 = (Dog)animal2;
            BigDog bigDog2 = (BigDog)animal3;
            Dog dog3 = (Dog)animal3;
            Cat cat2 = (Cat)animal2;
            dog2.greeting(dog3);
            dog3.greeting(dog2);
            dog2.greeting(bigDog2);
            bigDog2.greeting(dog2);
            bigDog2.greeting(bigDog1);
        }
    }
}

```

```

        BigDog bigDog2 = (BigDog)animal3;
        Dog dog3 = (Dog)animal3;
        Cat cat2 = (Cat)animal2;
        dog2.greeting(dog3);
        dog3.greeting(dog2);
        dog2.greeting(bigDog2);
        bigDog2.greeting(dog2);
        bigDog2.greeting(bigDog1);
    }

}

/*
 * 1) There is a scrolling problem with some 41 code: Greeting() implisented uses upper case 'G' in the class, * however, all catis in TestAnimalClass use lower case 'g '
 * * This does not change because of case sensitivity in C#
 * 2) Attempts to create objects on abstract classes are failing.
 * 3) Animal animal4 = new Animal(); because abstract classes cannot be instantiated
 * 3) The casting of animal2 to a cat object fails because it has been cast in our sprooras to a dog object, which exists under a dirterent - a branch under the inheritance tree
 * 4) Hide the overloadedas greeting method in the BigDog class using the "new" keyword in its parent class.
 */

public class TestAnimal
{
    public static void Main(String[] args)
    {
        // Using the subclasses
        Cat cat1 = new Cat();
        cat1.Greeting();
        Dog dog1 = new Dog();
        dog1.Greeting();
        BigDog bigDog1 = new BigDog();
        bigDog1.Greeting();
        // Using Polymorphism
        Animal animal1 = new Cat();
        animal1.Greeting();
        Animal animal2 = new Dog();
        animal2.Greeting();
        Animal animal3 = new BigDog();
        animal3.Greeting();
        // (2) Animal animal4 = new Animal();
    }
}

```

```

public class TestAnimal
{
    public static void Main(String[] args)
    {
        // Using the subclasses
        Cat cat1 = new Cat();
        cat1.Greeting();
        Dog dog1 = new Dog();
        dog1.Greeting();
        BigDog bigDog1 = new BigDog();
        bigDog1.Greeting();
        // Using Polymorphism
        Animal animal1 = new Cat();
        animal1.Greeting();
        Animal animal2 = new Dog();
        animal2.Greeting();
        Animal animal3 = new BigDog();
        animal3.Greeting();
        // (2) Animal animal4 = new Animal();
        // Downcast
        Dog dog2 = (Dog)animal2;
        BigDog bigDog2 = (BigDog)animal3;
        Dog dog3 = (Dog)animal3;
        // (3) Cat cat2 = (Cat)animal2;
        dog2.Greeting(dog3);
        dog3.Greeting(dog2);
        dog2.Greeting(bigDog2);
        bigDog2.Greeting(dog2);
        bigDog2.Greeting(bigDog1);
    }
}

```