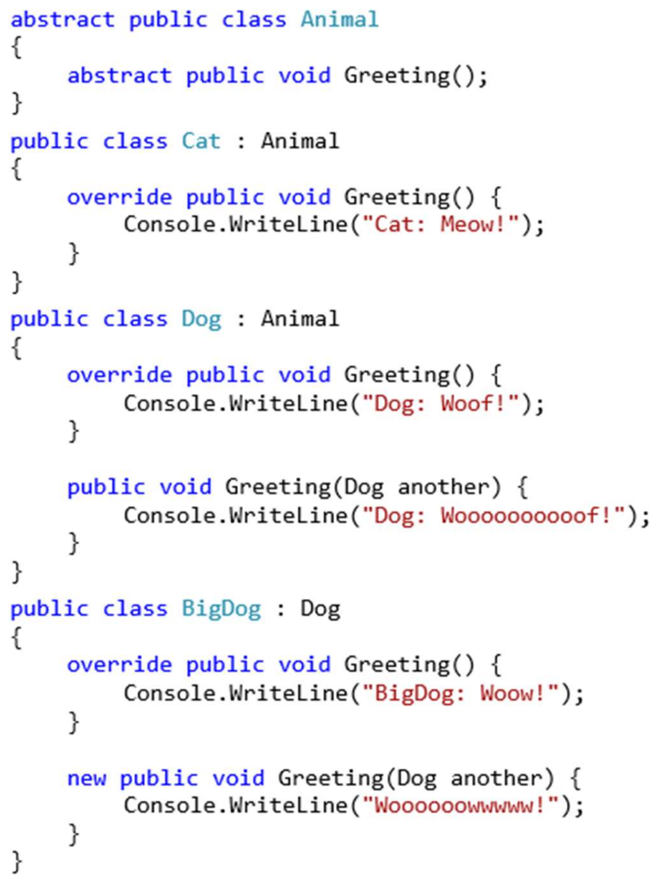
Student Name: Rakyan Satrya Adhikara

Student ID: 219548135

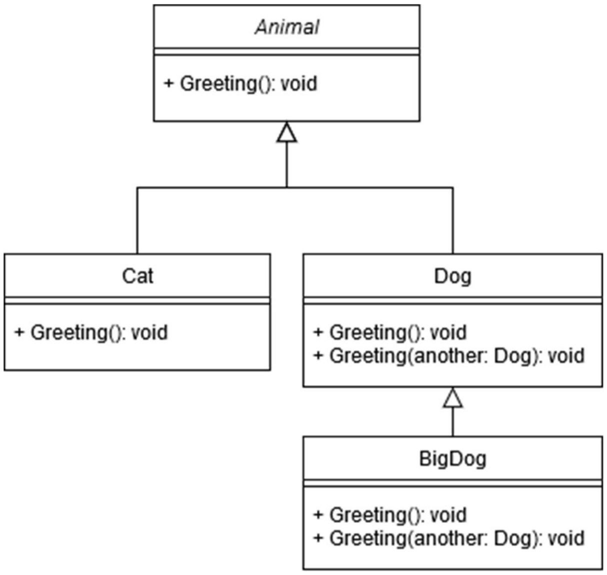
Practical Task 6.1 (Task 03 & 04)

- Task 03 (UML Diagram)
  - The code:

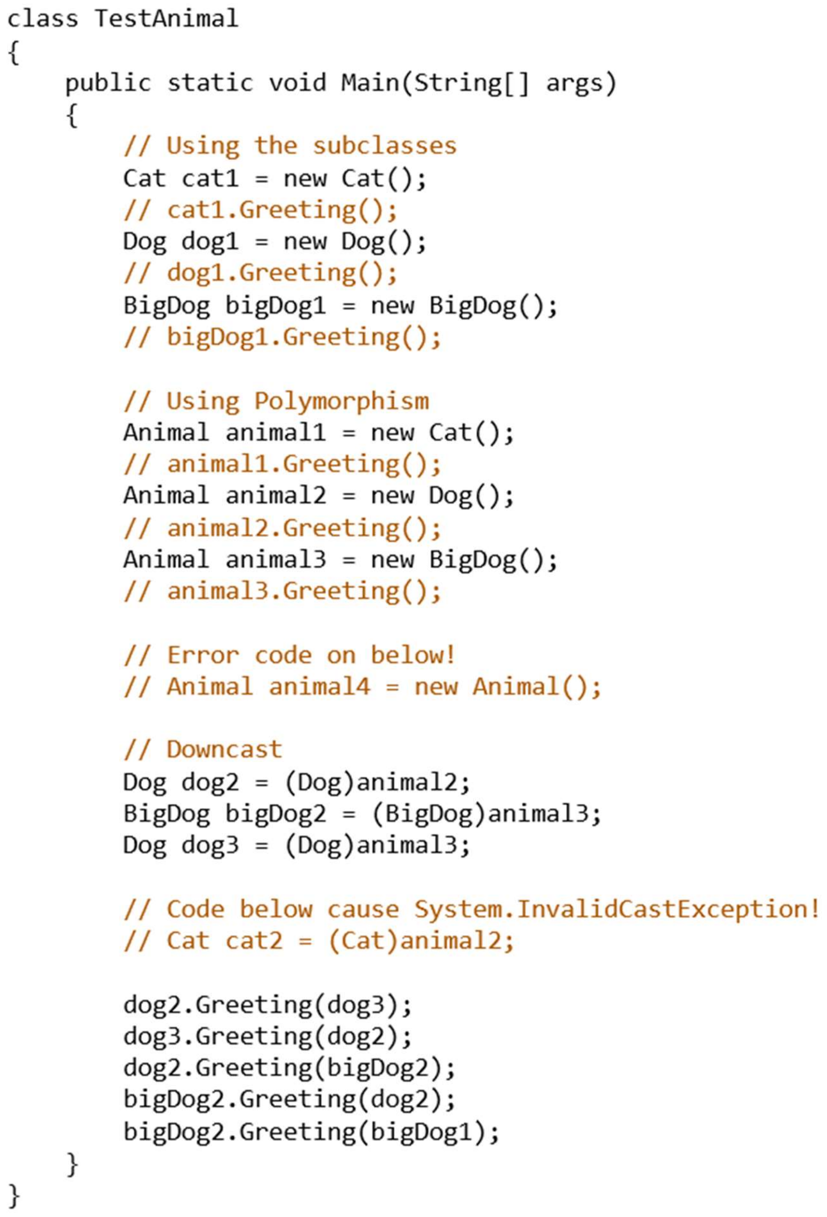


```
abstract public class Animal
{
    abstract public void Greeting();
}
public class Cat : Animal
{
    override public void Greeting() {
        Console.WriteLine("Cat: Meow!");
    }
}
public class Dog : Animal
{
    override public void Greeting() {
        Console.WriteLine("Dog: Woof!");
    }

    public void Greeting(Dog another) {
        Console.WriteLine("Dog: Woooooooooof!");
    }
}
public class BigDog : Dog
{
    override public void Greeting() {
        Console.WriteLine("BigDog: Woow!");
    }

    new public void Greeting(Dog another) {
        Console.WriteLine("Wooooowwwww!");
    }
}
```

- The UML diagram:

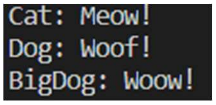


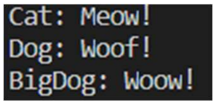
- Task 04
    - The code (with errors)

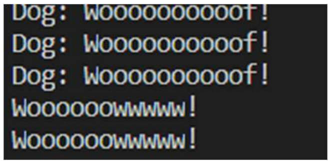


```
class TestAnimal
{
    public static void Main(String[] args)
    {
        // Using the subclasses
        Cat cat1 = new Cat();
        // cat1.Greeting();
        Dog dog1 = new Dog();
        // dog1.Greeting();
        BigDog bigDog1 = new BigDog();
        // bigDog1.Greeting();

        // Using Polymorphism
        Animal animal1 = new Cat();
        // animal1.Greeting();
        Animal animal2 = new Dog();
        // animal2.Greeting();
        Animal animal3 = new BigDog();
        // animal3.Greeting();

        // Error code on below!
        // Animal animal4 = new Animal();

        // Downcast
        Dog dog2 = (Dog)animal2;
        BigDog bigDog2 = (BigDog)animal3;
        Dog dog3 = (Dog)animal3;

        // Code below cause System.InvalidCastException!
        // Cat cat2 = (Cat)animal2;

        dog2.Greeting(dog3);
        dog3.Greeting(dog2);
        dog2.Greeting(bigDog2);
        bigDog2.Greeting(dog2);
        bigDog2.Greeting(bigDog1);
    }
}
```

        - Error #1:
            - Cause: program initiating a variable with an *abstract-type* of class
        - Error #2
            - Cause: System.InvalidCastException
            - Since Animal2 was originally a *dog*, the new Greeting function is not familiar with the *cat* type.

- The output
  - Subclass part:
    - 

      ```
      Cat: Meow!
      Dog: Woof!
      BigDog: Woow!
      ```
    - Meaning: it will print from the overridden functions
  - Polymorphism part:
    - 

      ```
      Cat: Meow!
      Dog: Woof!
      BigDog: Woow!
      ```
    - Meaning: like the subclass, it will print from the overridden function
  - Downcast part:
    - 

      ```
      Dog: Woooooooooof!
      Dog: Woooooooooof!
      Dog: Woooooooooof!
      Woooooowwwww!
      Woooooowwwww!
      ```
    - Meaning: Since the function was being called with a provided input (another *dog* variable), it will use the new *Greeting* function that have a *dog* parameter.