Program 1:

// program to demonstrate use of super keyword with constructors in multilevel inheritance.

package javaLab.javaPractical;

public class dog {

    public static dog speak;

    protected String name;

    protected int age;

    public dog(String name, int age){

        this.name = name;

        this.age = age;

    }

    public void speak(){

        System.out.println("I am " + this.name + " and my age is " + this.age);

    }

}

public class cat extends dog{

    private String food;

    public cat(String name, int age, String food){

        super(name, age);

        this.food = food;

    }

    public cat(String name, int age){

        super(name, age);

        this.food = "Food";

    }

    public void speak(){

        System.out.println("I am " + this.name + " and my age is " + this.age + " and i eat this " + this.food);

    }

}

public class dogCatMain {

    public static void main(String[] args) {

        dog dog01= new dog("Dog\_1", 3);

        dog01.speak();

        dog dog02= new dog("Dog\_2", 5);

        dog02.speak();

        dog dog03= new dog("Dog\_3", 1);

        dog03.speak();

        System.out.println("\nAfter Super Keyword this Cat class can also use the features of Dog Class >> \n");

        cat cat001 = new cat("Cat\_1", 2);

        cat001.speak();

        cat cat002 = new cat("Cat\_2", 1);

        cat002.speak();

        cat cat003 = new cat("Cat\_3", 2);

        cat003.speak();

        System.out.println("\nAfter Super Keyword this Cat class can also use the features of Dog Class and also modify it >> \n");

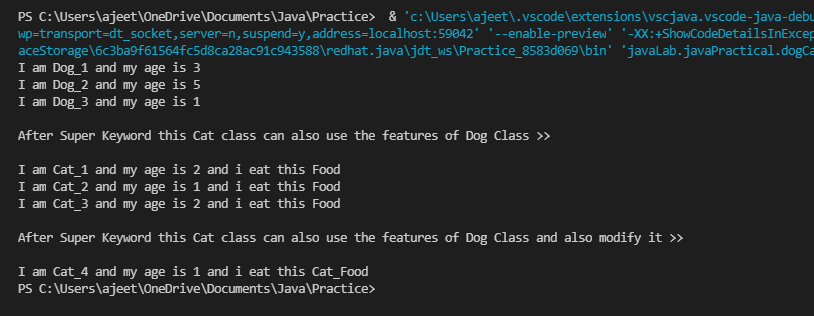
        cat cat004 = new cat("Cat\_4", 1, "Cat\_Food");

        cat004.speak();

    }

}

Output:



Program 2:

// Program to Handle Exception In a Super Class Methods

package javaLab.javaPractical;

public class superDogExH {

    void method() throws RuntimeException

    {

        System.out.println("\nSuperClass Dog\n");

    }

}

class subCatExH extends superDogExH {

    void method() throws ArithmeticException

    {

        System.out.println("\nSubClass Cat\n");

    }

    public static void main(String args[])

    {

        try{

            superDogExH s = new subCatExH();

            s.method();

        }

        catch(Exception e) {

            e.fillInStackTrace();

        }

    }

}

Output:

SubClass Cat