SCHOOL OF COMPUTING
UNIVERSITI UTARA MALAYSIA

STIA1123 PROGRAMMING 2

ASSIGNMENT 2 (8%)

FIRST SEMESTER SESSION 2022/2023 (A 221)

NAME	MATRIC NUMBER
VINCENT BEH HUA EIK	279018

Date of Submission : 28th December 2022

Pet Interface Code

```
1 package my.animalApp;
2
3 interface Pet {
4     public String play();
5 }
```

Herbivore Interface Code

```
1 package my.animalApp;
2
3  interface Herbivore {
4     public String typeH = "";
5     public String foodH = "";
6     public String eat();
7 }
```

Carnivore Interface Code

```
1 package my.animalApp;
2
3 interface Carnivore {
4     public String typeC = "";
5     public String foodC = "";
6     public String eat();
7 }
```

Animal Class Code

```
1  package my.animalApp;
 2
 3 public abstract class Animal {
 4     protected String name;
 5     protected int numOfLegs;
 6
 7         public Animal(String name, int numOfLegs) {
 8         this.name = name;
 9         this.numOfLegs = numOfLegs;
10     }
11
12     public abstract String display();
13
14        // getter methods
15     public String getName() {
16         return name;
17     }
18
19     public int getNumOfLegs() {
20         return numOfLegs;
21     }
22 }
23
24
```

Cat Class Code

```
  package my.animalApp;
1
 2 public class Cat extends Animal implements Pet,Carnivore {
 3     private String sound;
 4     private String colour;
 5
 6     public Cat(String name, int numOfLegs, String sound, String colour)
 7 {
 8         super(name, numOfLegs);
 9         this.sound = sound;
10         this.colour = colour;
11     }
12
13         // implement abstract methods from Animal
14     public String display() {
15         return "Name: " + getName() + "\nNumber of legs: " +
16 getNumOfLegs() + "\nSound: " + sound + "\nColour: " + colour;
17     }
18
19         // implement methods from Carnivore interface
20     public String typeC() {
21         return "Carnivore";
22     }
23
24     public String foodC() {
25         return "Meat/Animal Source Eater";
26     }
27
28     @Override
29     public String play(){
30     return getName() + " likes to play with strings and cat toys.";
31     }
32
33     @Override
34     public String eat(){
35     return  "Cat likes to eat birds, mice, fish etc. ";
36     }
37
38
39
  }
```

Bear Class Code

```
package my.animalApp;

public class Bear extends Animal implements Carnivore, Herbivore {
    private String sound;
    private String colour;


     public Bear(String name, int numOfLegs, String sound, String
colour) {
        super(name, numOfLegs);
        this.sound = sound;
        this.colour = colour;
    }

         public String display() {
        return "Name: " + getName() + "\nNumber of legs: " +
getNumOfLegs() + "\nSound: " + sound + "\nColour: " + colour;
    }

    public String typeC() {
        return "Carnivore";
    }

    public String foodC() {
        return "Meat/Animal Sources Eater";
    }
    // implement methods from Herbivore interface
    public String typeH() {
         return "Herbivore";
    }


    public String foodH() {
        return "Plants";
    }

       public String typeO() {
         return "Omnivore";
    }


    public String foodO() {
        return "Plants and are also Meat/Animal Source Eater";
    }

    public String descO() {
        return "Omnivore is similar to both Herbivore and Carnivore";
    }

    @Override
    public String eat() {
        return "Bear like to eat berries, roots, fungi, grasses, fish,
carrion, small mammals, and insects. ";
    }
}
```

Elephant Class Code

```
package my.animalApp;

public class Elephant extends Animal implements Herbivore {
     private String sound;
    private String colour;

    public Elephant(String name, int numOfLegs, String sound, String colour) {
        super(name, numOfLegs);
        this.sound = sound;
        this.colour = colour;
    }

    // implement abstract methods from Animal
    public String display() {
        return "Name: " + getName() + "\nNumber of legs: " + getNumOfLegs() + "\nSound: " + sound + "\nColour: " + colour;
    }

    // implement methods from Herbivore interface
    public String typeH() {
         return "Herbivore";
    }


    public String foodH() {
        return "eats Plants";
    }

    @Override
    public String eat() {
        return "Elephants like to eat grasses, small plants, bushes, fruit, twigs, tree bark, and roots.";
    }
}
```

AnimalGUI.java Initialization

```
1  package my.animalApp;
 2 import javax.swing.JOptionPane;
 3 public class AnimalGUI extends javax.swing.JFrame {
 4     // create animal array
 5             private static Animal[] animals = new Animal[3];
 6             private int count = 0;
 7
 8     /**
 9      * Creates new form AnimalGUI
10      */
11     public AnimalGUI() {
12         initComponents();
13     }
```

AnimalGUI.java main method

```
    public static void main(String args[]) {

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new AnimalGUI().setVisible(true);
            }


        });
    }
```

AnimalGUI.java Clear All Button

```
private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this, "Do you want to
clear all the fields?", "Confirm", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);

        if(response == JOptionPane.YES_OPTION){
        animalButtonGroup1.clearSelection();
        nameTF.setText("");
        colourTF.setText("");
        legsTF.setText("");
        soundTF.setText("");
        display1.setText("");
        }
        else if (response == JOptionPane.NO_OPTION){
        JOptionPane.getRootFrame().dispose();
        }
    }
```

AnimalGUI.java Save Button Part 1

```
private void saveBtnActionPerformed(java.awt.event.ActionEvent evt) {
        display1.setText("");

        if(count == animals.length){
        display1.setText("Error. Sorry, list is Full!");
        animalButtonGroup1.clearSelection();
        nameTF.setText("");
        colourTF.setText("");
        legsTF.setText("");
        soundTF.setText("");
        return;
        }

        if (catButton1.isSelected()){
        //reading input
        String nameText = nameTF.getText();
        String colourText = colourTF.getText();
        String legsText = legsTF.getText();
        String soundText = soundTF.getText();
        int legs =0;
        try {
             legs = Integer.parseInt(legsText);

            }
        catch ( NumberFormatException e ) {
            display1.setText("Error. Invalid input for legs.");
            return;
            }

        animals [count] = new Cat (nameText, legs, soundText, colourText);
        count++;
        animalButtonGroup1.clearSelection();
        nameTF.setText("");
        colourTF.setText("");
        legsTF.setText("");
        soundTF.setText("");
        }
```

AnimalGUI.java Save Button Part 2

```
else if (bearButton2.isSelected()){
        //reading input
        String nameText = nameTF.getText();
        String colourText = colourTF.getText();
        String legsText = legsTF.getText();
        String soundText = soundTF.getText();
        int legs =0;
        try {
             legs = Integer.parseInt(legsText);

            }
        catch ( NumberFormatException e ) {
            display1.setText("Error. Invalid input for legs.");
            return;
            }

        animals [count] = new Bear (nameText, legs, soundText, colourText);
        count++;
        animalButtonGroup1.clearSelection();
        nameTF.setText("");
        colourTF.setText("");
        legsTF.setText("");
        soundTF.setText("");
        }
        else if (elephantButton3.isSelected()){
        //reading input
        String nameText = nameTF.getText();
        String colourText = colourTF.getText();
        String legsText = legsTF.getText();
        String soundText = soundTF.getText();
        int legs =0;
        try {
             legs = Integer.parseInt(legsText);

            }
        catch ( NumberFormatException e ) {
            display1.setText("Error. Invalid input for legs.");
            return;
            }

        animals [count] = new Elephant (nameText, legs, soundText,
colourText);
        count++;
        animalButtonGroup1.clearSelection();
        nameTF.setText("");
        colourTF.setText("");
        legsTF.setText("");
        soundTF.setText("");
        }
    }
```

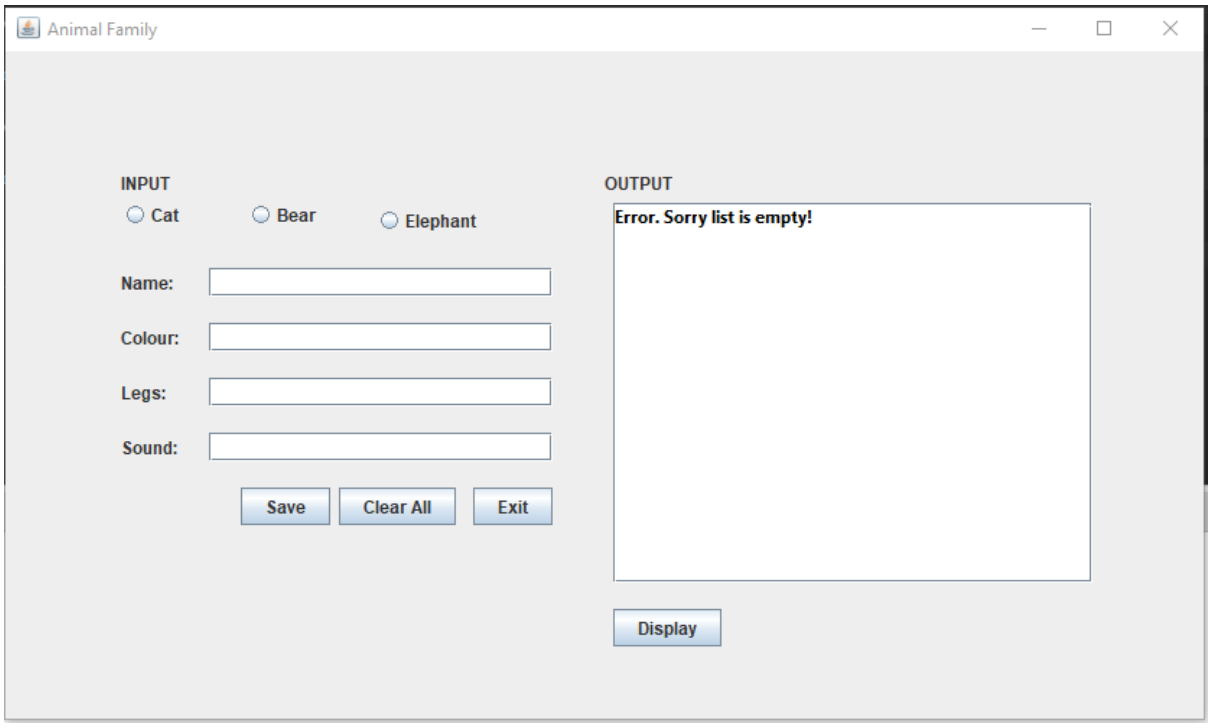
AnimalGUI.java Display Button

```
private void displayBtnActionPerformed(java.awt.event.ActionEvent evt)
{
            display1.setText("");
            display1.setLineWrap(true);
            display1.setWrapStyleWord(true);

            if(count == 0){
                display1.setText("Error. Sorry list is empty!");
                animalButtonGroup1.clearSelection();
                nameTF.setText("");
                colourTF.setText("");
                legsTF.setText("");
                soundTF.setText("");
                }

            else
            for (int i = 0;i<count; i++){


                if(animals[i] instanceof Cat){
                display1.append("---------------CAT---------------\n");
                display1.append(animals[i].display()+ "\n");
                display1.append(animals[i].getName()+ " is a " +
((Cat)animals[i]).typeC() + " and " + ((Cat)animals[i]).foodC() + "\n");
                display1.append(((Cat)animals[i]).eat() + "\n");
                display1.append(((Cat)animals[i]).play() + "\n");

                }
                else if (animals[i] instanceof Bear){
                display1.append("---------------BEAR---------------\n");
                display1.append(animals[i].display()+ "\n");
                display1.append("Bear is an " + ((Bear)animals[i]).typeO()+
"\n");
                display1.append(((Bear)animals[i]).descO() + "\n");
                display1.append(((Bear)animals[i]).eat() + "\n");
                }

                else if (animals[i] instanceof Elephant){
                display1.append("---------------ELEPHANT---------------
\n");
                display1.append(animals[i].display()+ "\n");
                display1.append(animals[i].getName()+ " is a " +
((Elephant)animals[i]).typeH() + " and " + ((Elephant)animals[i]).foodH() +
"\n");
                display1.append(((Elephant)animals[i]).eat() + "\n");
                }
    }
    }
```

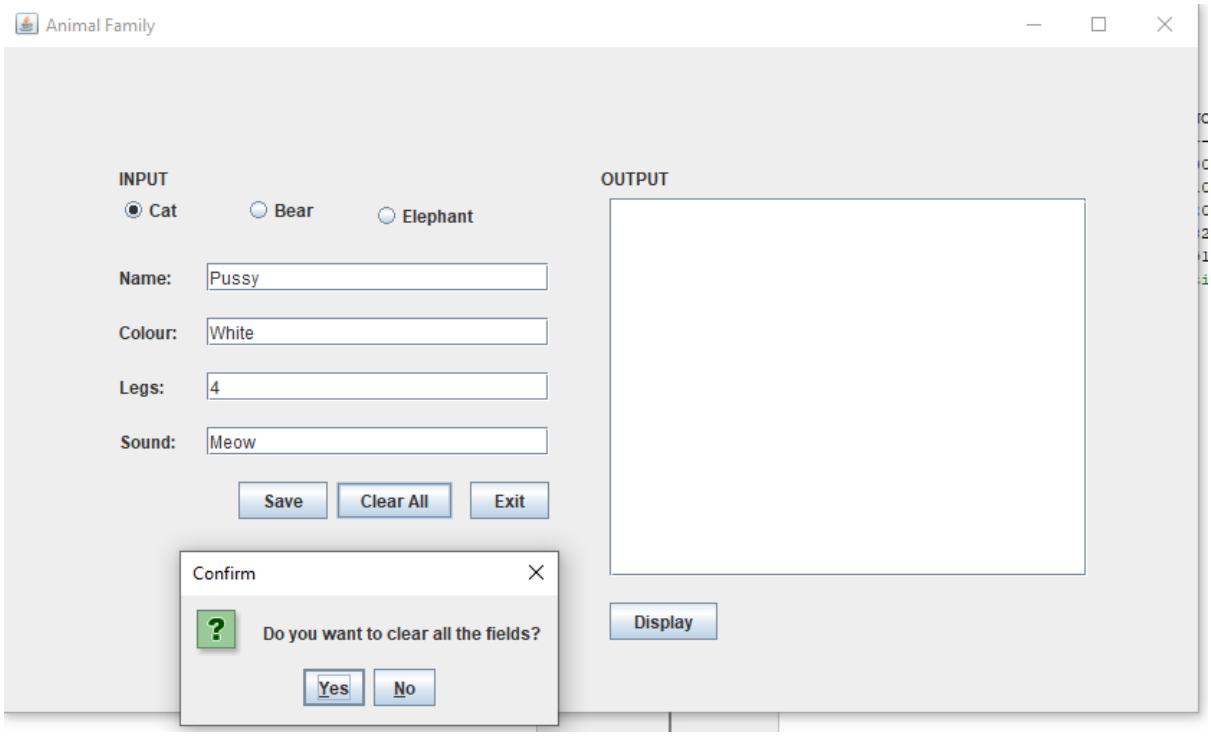
AnimalGUI.java Exit Button

```
private void exitBtnActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this, "Do you want to
quit?", "Confirm", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);

        if(response == JOptionPane.YES_OPTION){
        System.exit(0);
        }
        else if (response == JOptionPane.NO_OPTION){
        JOptionPane.getRootFrame().dispose();
        }

    }
```

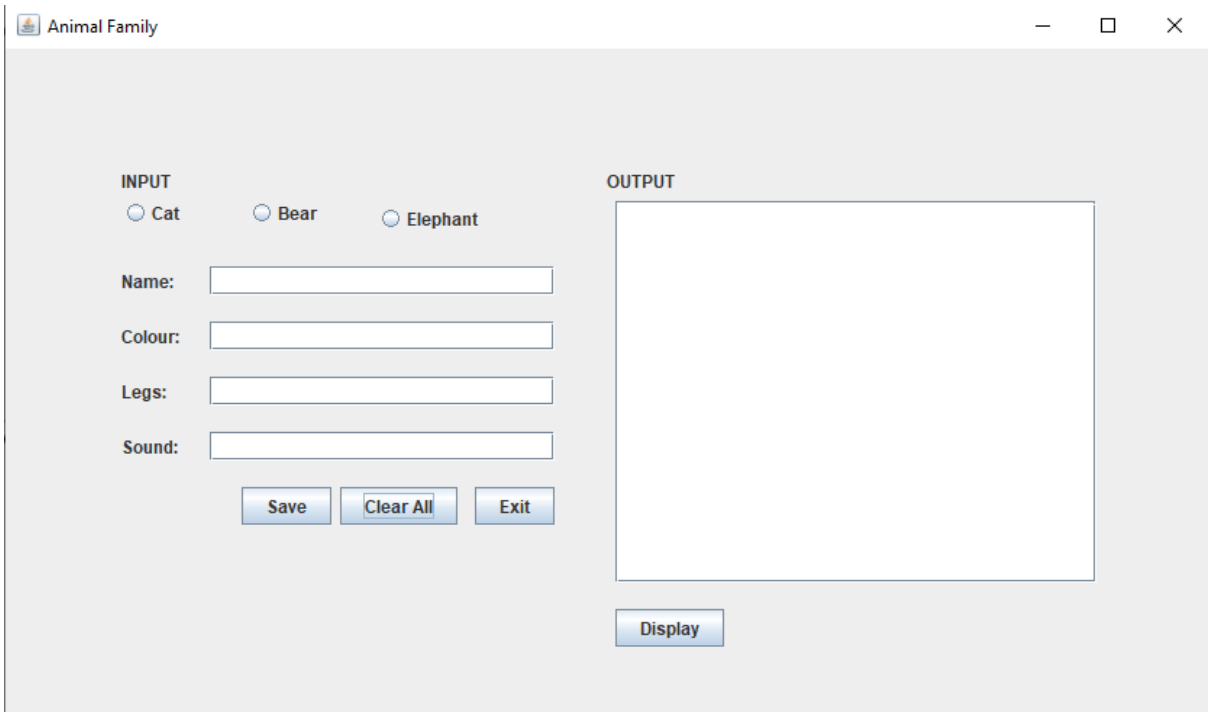
Sample Run



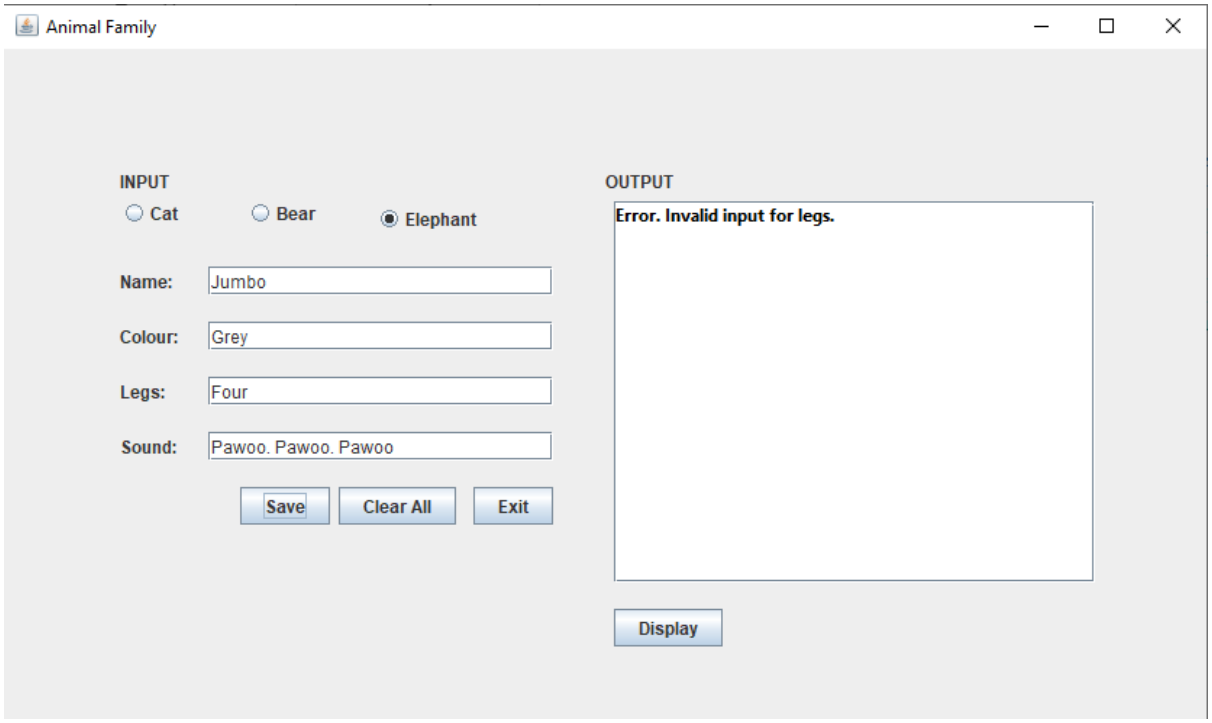
Action: Clicking Display without adding anything



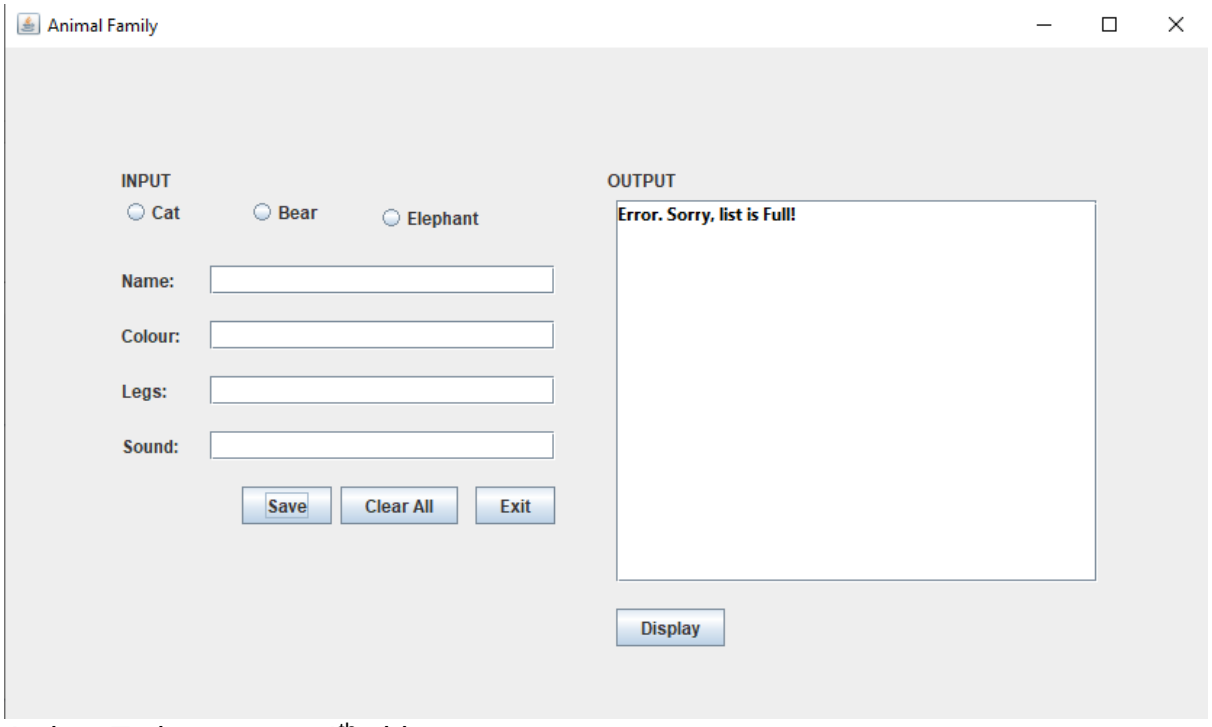
Action: Clicking Clear All button



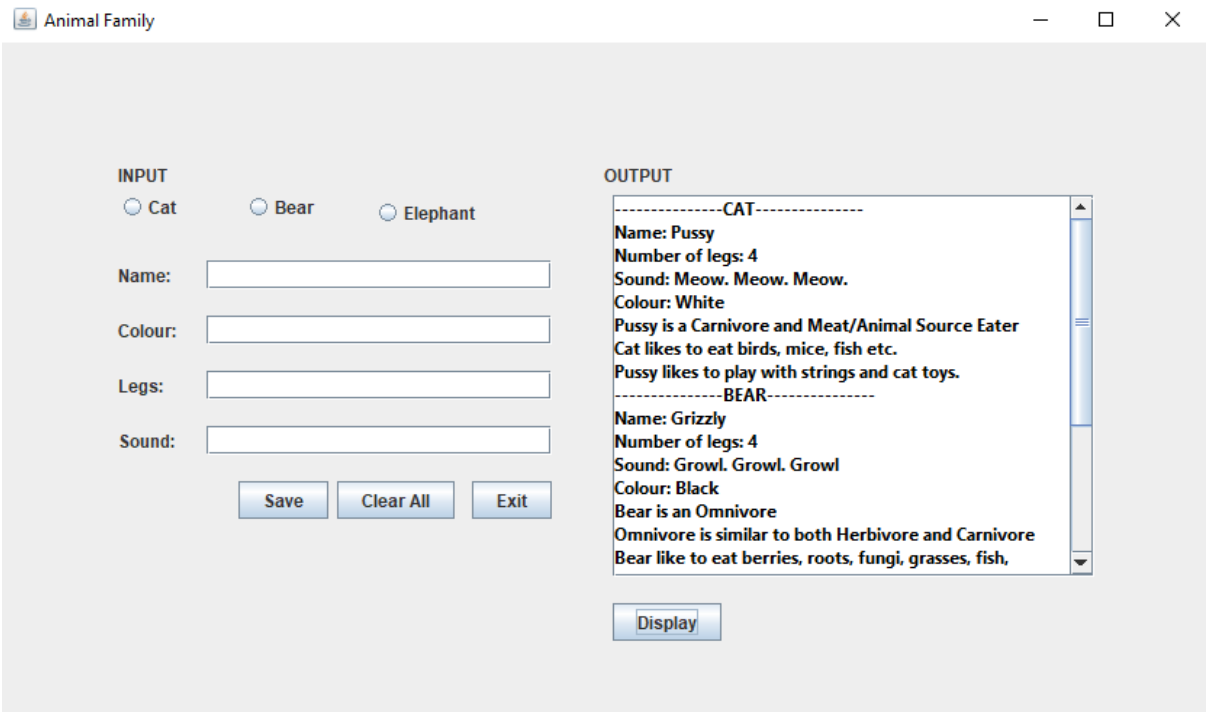
Action: Clicking yes



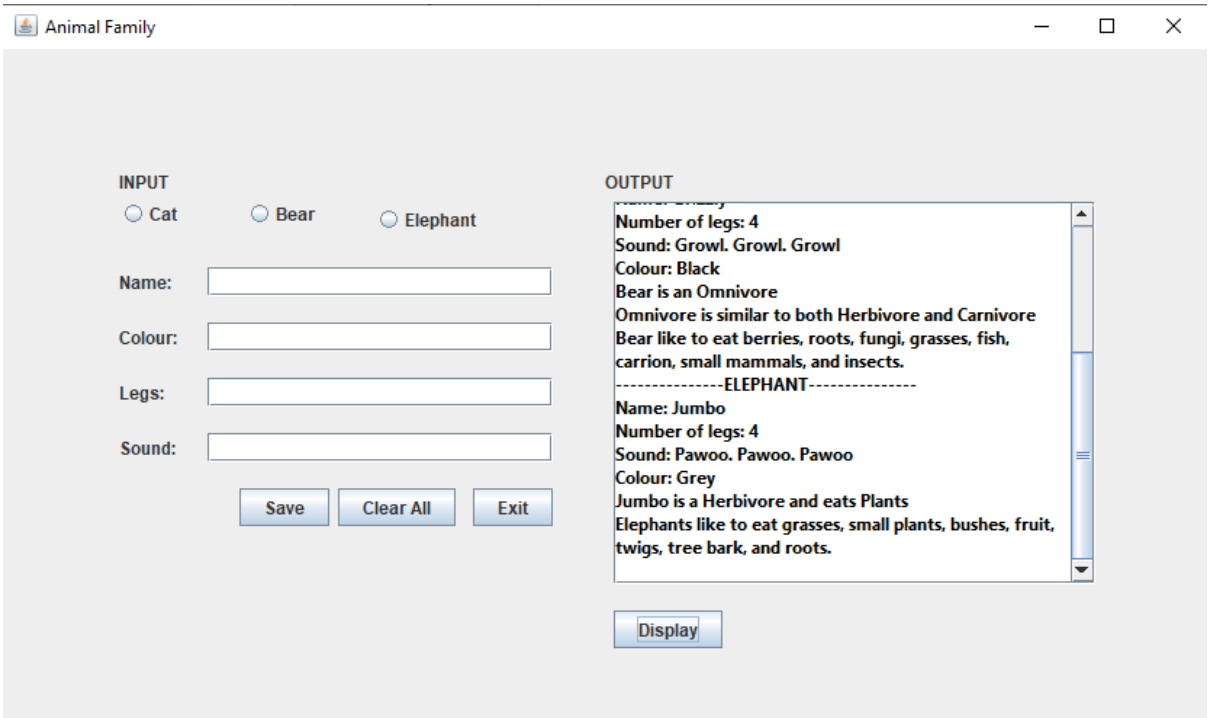
Action: Input legs is wrong



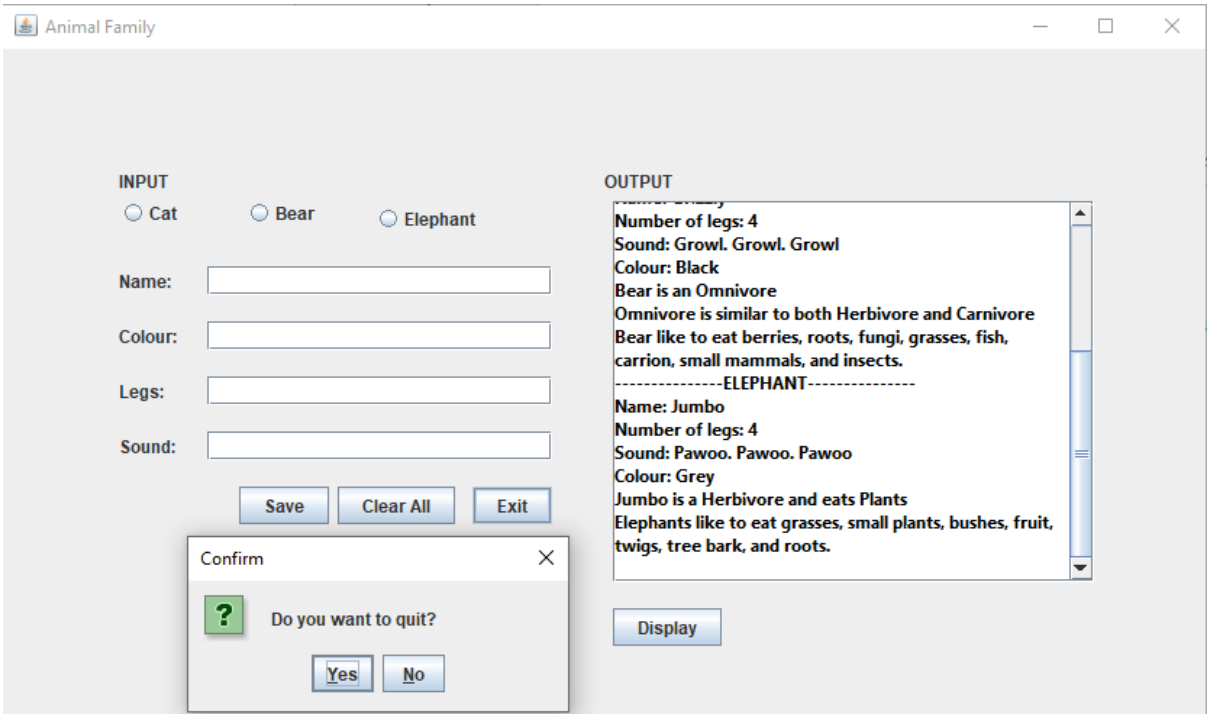
Action: Trying to save 4th object



Display Pt.1



Display Pt.2



Action: Clicking Exit button