**Question:** Create an abstract class named Animal which implements the interface Comparable. It will contain members such as size, number of feet, type of diet etc. It will have abstract methods such as breathe (), eat (), move () It will also override the compareTo () method of Comparable Now create these classes: Lion, Cow, Fish and Monkey Class and Implement the abstract methods in main method, create the above class objects and show some of their methods. Now add them to an array of Animals and sort the array according to size. Print the final array.

**Source Code:**

**Animal Class**

package Lab7Work;

import java.util.\*;

public abstract class Animal implements Comparable<Animal> {

int size; // in inch

int num\_of\_feet; // int var

String type\_of\_diet; // string var

abstract void breathe();

abstract void eat();

abstract void move();

public int compareTo(Animal Anml) {

if (size == Anml.size) {

return 0; }

else if (size > Anml.size) {

return 1; }

else {

return -1; } } }

**Lion Class**

package Lab7Work;

import java.util.\*;

public class Lion extends Animal {

Lion(int size, int num\_of\_feet, String diet){

this.size = size;

this.num\_of\_feet = num\_of\_feet;

this.type\_of\_diet = diet; }

void breathe(){

System.out.println("Lion is breathing..."); }

void eat(){

System.out.println("Lion is eating..."); }

void move(){

System.out.println("Lion is moving..."); } }

**Cow Class**

package Lab7Work;

public class Cow extends Animal{

Cow(int size, int num\_of\_feet, String diet){

this.size = size;

this.num\_of\_feet = num\_of\_feet;

this.type\_of\_diet = diet; }

void breathe(){

System.out.println("cow is breathing..."); }

void eat(){

System.out.println("cow is eating..."); }

void move(){

System.out.println("cow is moving..."); } }

**Fish Class**

package Lab7Work;

public class Fish extends Animal {

Fish(int size, int num\_of\_feet, String diet){

this.size = size;

this.num\_of\_feet = num\_of\_feet;

this.type\_of\_diet = diet; }

void breathe(){

System.out.println("fish is breathing..."); }

void eat(){

System.out.println("fish is eating..."); }

void move(){

System.out.println("fish is moving..."); } }

**Monkey Class**

package Lab7Work;

public class Monkey extends Animal{

Monkey(int size, int num\_of\_feet, String diet){

this.size = size;

this.num\_of\_feet = num\_of\_feet;

this.type\_of\_diet = diet; }

void breathe(){

System.out.println("monkey is breathing..."); }

void eat(){

System.out.println("monkey is eating..."); }

void move(){

System.out.println("monkey is moving..."); } }

**Main Class**

package Lab7Work;

import java.util.ArrayList;

import java.util.Collections;

public class MainMathod {

public static void main(String[] args) {

// Lion class object

ArrayList <Lion> Lion1 = new ArrayList<Lion>();

Lion1.add(new Lion(44, 3, "Meet"));

Lion1.add(new Lion(35, 3, "Meet"));

Lion1.add(new Lion(50, 3, "Meet"));

Lion1.add(new Lion(29, 3, "Meet"));

Lion1.add(new Lion(30, 3, "Meet"));

Collections.sort(Lion1);

for(Lion Ln: Lion1){

System.out.println(Ln.size+ " " + Ln.num\_of\_feet + " " + Ln.type\_of\_diet); }

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

// Cow class object

ArrayList <Cow> Cow1 = new ArrayList<Cow>();

Cow1.add(new Cow(77, 3, "Grass"));

Cow1.add(new Cow(70, 3, "Grass"));

Cow1.add(new Cow(43, 3, "Grass"));

Cow1.add(new Cow(59, 3, "Grass"));

Cow1.add(new Cow(54, 3, "Grass"));

Collections.sort(Cow1);

for(Cow Cw: Cow1){

System.out.println(Cw.size+ " " + Cw.num\_of\_feet + " " + Cw.type\_of\_diet);

}

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

// Fish class object

ArrayList <Fish> Fish1 = new ArrayList<Fish>();

Fish1.add(new Fish(23, 0, "Fish feed"));

Fish1.add(new Fish(22, 0, "Fish feed"));

Fish1.add(new Fish(5, 0, "Fish feed"));

Fish1.add(new Fish(23, 0, "Fish feed"));

Fish1.add(new Fish(6, 0, "Fish feed"));

Collections.sort(Fish1);

for(Fish Fs: Fish1){

System.out.println(Fs.size+ " " + Fs.num\_of\_feet + " " + Fs.type\_of\_diet);

}

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

// Monkey class object

ArrayList <Monkey> Monkey1 = new ArrayList<Monkey>();

Monkey1.add(new Monkey(33, 0, "Fruit"));

Monkey1.add(new Monkey(18, 0, "Fruit"));

Monkey1.add(new Monkey(22, 0, "Fruit"));

Monkey1.add(new Monkey(30, 0, "Fruit"));

Monkey1.add(new Monkey(9, 0, "Fruit"));

Collections.sort(Monkey1);

for(Monkey Mn: Monkey1){

System.out.println(Mn.size+ " " + Mn.num\_of\_feet + " " + Mn.type\_of\_diet);

}

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

}

}

**Output:**

