# Directions

1. Write the following code.
2. You may refer to lecture notes, book, and your old code if needed.
3. Do not search the Internet for guidance.
4. Submit your code attempts and screenshots showing the output
5. Submit to Blackboard.

## Class

public class Food implements Comparable<Food> {  
 protected String name;  
 protected int calories;  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getCalories() {  
 return calories;  
 }  
  
 public void setCalories(int calories) {  
 if(calories<0)  
 this.calories = 0;  
 else  
 this.calories = calories;  
 }  
  
 public Food(String name, int calories) {  
 this.name = name;  
  
 if(calories<0)  
 this.calories = 0;  
 else  
 this.calories = calories;  
 }  
  
 public Food() {  
 this.name = "";  
 this.calories=0;  
 }  
  
 @Override  
 public String toString() {  
 return this.name + " : " + this.calories + " calories";  
 }  
  
  
 @Override  
 public int compareTo(Food f) {  
 return this.name.compareTo(f.name);  
 }  
  
  
}

### Inherit

public class JunkFood extends Food{  
 protected double fat;  
 protected double sugar;  
  
 public double getFat() {  
 return fat;  
 }  
  
 public void setFat(double fat) {  
 this.fat = fat;  
 }  
  
 public double getSugar() {  
 return sugar;  
 }  
  
 public void setSugar(double sugar) {  
 this.sugar = sugar;  
 }  
  
 public JunkFood(String name, int calories, double fat, double sugar) {  
 super(name, calories);  
 this.fat = fat;  
 this.sugar = sugar;  
 }  
  
 public JunkFood() {  
 super();  
 this.fat=0.0;  
 this.sugar=0.0;  
 }  
  
 @Override  
 public String toString() {  
 return super.toString()+ " : "+ this.fat + " fat " + " : " + this.sugar + " sugar";  
 }  
}

### Comparators

public class NameComp extends Food implements Comparable<Food>{  
  
 public int compareNames(Food f1,Food f2)  
 {  
 return f1.getName().compareTo(f2.getName());  
 }  
}

public class CalorieComp extends Food implements Comparable<Food>{  
  
 public int compareCal(Food f1,Food f2)  
 {  
 return (int) (f1.getCalories()- f2.getCalories());  
 }  
}

### Array/ArrayList 1

import java.util.Arrays;  
  
public class List {  
 public static void main(String[] args) {  
 int[] nums = new int[20];  
 int count=0;  
  
 for(int x=0;x<nums.length;x++)  
 {  
 nums[x]=(int) (Math.*random*()\*10)+1;  
 }  
  
 for(int x=0;x<nums.length;x++) {  
 if (nums[x] == 5)  
 count++;  
 }  
  
 System.*out*.println("There are "+ count + " 5's in the array");  
 System.*out*.println(Arrays.*toString*(nums));  
  
 for (int i = 0; i < nums.length - 1; i++) {  
 for (int j = 0; j < nums.length - i - 1; j++) {  
 if (nums[j] > nums[j + 1]) {  
 // swap arr[j+1] and arr[j]  
 int temp = nums[j];  
 nums[j] = nums[j + 1];  
 nums[j + 1] = temp;  
 }  
 }  
 }  
  
 System.*out*.println(Arrays.*toString*(nums));  
  
 }  
}

Application

Description automatically generated with low confidence

### Array/ArrayList 2

import java.util.Arrays;  
  
public class List2 {  
 public static void main(String[] args) {  
 Food[] foods = new Food[5];  
  
 foods[0]= new Food("Hotdog",200);  
 foods[1]= new JunkFood("Chocolate",250,200,100);  
 foods[2]= new Food("Wrap",100);  
 foods[3]= new JunkFood("Pizza",300,200,100);  
 foods[4]= new Food("Salad",50);  
  
 for (int i = 0; i < foods.length - 1; i++) {  
 for (int j = 0; j < foods.length - i - 1; j++) {  
 if (foods[j].getCalories() > foods[j + 1].getCalories()) {  
 // swap arr[j+1] and arr[j]  
 Food temp = foods[j];  
 foods[j]=foods[j+1];  
 foods[j + 1]=temp;  
 }  
 }  
 }  
  
 System.*out*.println(Arrays.*toString*(foods));  
 }  
}

Text

Description automatically generated with medium confidence

### String

public class Sentence {  
 public static void main(String[] args) {  
 String s = "A dog has a tail";  
 s=s.toLowerCase();  
  
 System.*out*.println(s.indexOf("a"));  
  
 s=s.replaceAll("a","4");  
  
 System.*out*.println(s);  
  
 String s2= s.substring(0,s.length()/2);  
 System.*out*.println(s2);  
  
 }  
}

Text

Description automatically generated