An example to illustrate the processing of objects in the Driver class.

A nutritionist takes a survey on the composition of breakfast cereals consumed by South Africans on a daily basis.
The following information based on four types of cereals was recorded. The information is stored in a text file called **nutrition.txt**.

Eat&Go:800:550:100
RocketMind:600:1000:120
Wanna Win:600:900:200
Chocky Bits:900:650:500

Field1: Name of cereal,
Field2: Amount of Vitamins,
Field3: Amount of Iron,
Field4: Amount of Fat

5.1 Create an object class called Nutrition.
5.2 Define the following private fields using appropriate data types.
cerealName, amtVitamins, amtIron, amtFat

5.3 Create a parameterized constructor that will pass values to all fields.
5.4 Write a toString method that will return all details of each cereal.
5.5 Write a method that will return the name of the cereal.
5.6 Write a method that will receive three new values and change the composition of the cereals accordingly.
5.7 Write a method to return the nutrient with the highest value. (For each cereal)
5.8 Write a method to return a boolean value True if the "Fat" component of the cereal is less than 200. If greater than or Equal to 200, return false.
5.9 **Write a TestNutrition Class to:**
5.9.1 read information from the text file nutrition.txt.

Use a menu with the following options:

"A - Display Cereals" "B - Display Dietary Statistics"
"C - Search and Update" "Q - QUIT"

.

5.9.2 **Option A:** display all details of cereals under suitable headings.

Cereal Name	Amount of Vit.	Amount of Iron	Amount of Fat
Eat&Go	800	550	100
RocketMind	600	1000	120
Wanna Win	600	900	200
Chocky Bits	900	650	500

5.9.3 **Option B:** display the name of the cereal, the highest nutrient, a message : "Low" if the "fat" nutrient is less than 200, otherwise display "High".

Sample

Cereal	Highest Nutrient	Level of Fat
Eat&Go	Vitamins	Low
RocketMind	Iron	Low
Wanna Win	Iron	High
Chocky Bits	Vitamins	High

5.9.4 **Option C:** Allow the user to input the name of a cereal.
Search for the cereal, terminate the search process as soon as he cereal is found. If the cereal is found, prompt the user to input new values for the amount of vitamins, iron and fat, update the file accordingly. Display a suitable message if the cereal being searched for is not found.

Answer:

```
class Nutrition {
    private String cerealName = "";
    private int amtVitamins   = 0;
    private int amtIron       = 0;
    private int amtFat        = 0;
   public Nutrition(String name, int aV, int aI, int aF)
    {
      cerealName=name;
      amtVitamins = aV;
       amtIron = aI;
      amtFat=aF;
    }
     public String toString( )
    {
     String each_Object="";
     each_Object = cerealName + "\t\t\t"+ amtVitamins+ "\t\t\t" +
                amtIron+ "\t\t\t"+amtFat;
     return each_Object;
    }
     public String getCerealName( )
         {
      return cerealName;
    }
     public void setComponents(int aV, int aI, int aF)
    {
      amtVitamins = aV;
      amtIron     = aI;
      amtFat      = aF;
        }
   public String getHighestComponent( )
  {
      int  max = amtVitamins;
              String maxComp = "Vitamins";
      if (amt Iron > max)
              {
             max = amtIron;
        maxComp = "Iron";
      }
      if (amtFat > max)
      {
        max = amtFat;
        maxComp = "Fat";
      }
      return maxComp;
    }
     public boolean getLowFat ( )
     {
      boolean checkFat = false;
      if (amtFat < 200)
      {
        checkFat = true;
      }
      return checkFat; } }
```

Driver class

```
 import java.io.*;
 import java.util.*;
  public class TestNutrition
 {
   Nutrition [] arr = new Nutrition [50];
   int size = 0;
    public TestNutrition( )
    {
      try
       {
 Scanner scFile = new Scanner (new FileReader("nutrition.txt"));   while (scFile.hasNext( ))
       {
            String line = scFile.nextLine( );
            Scanner sc = new Scanner(line).useDelimiter(":");
            String cereal = sc.next( );
            int amtV = sc.nextInt( );
            int amtI = sc.nextInt( );
                        int amtF = sc.nextInt();
            sc.close();
            arr [size] = new Nutrition(cereal,amtV,amtI,amtF);
            size++;
        }
      }
         catch (Exception e)
         {
 System.out.println(e.getMessage( ));
         }
     }
       public void displayCereals()
     {
      System.out.println("Cereal Name\t\t"+ "Amount Vitamins\t\t"+   "Amount of Iron\t\t"+"Amount of Fat");
       for (int i = 0; i < size; i++)
       {
         System.out.println(arr[i].toString( ));
       }
     }
          public void displayDietStats( )
     {
       String message ="";
       System.out.printf("%-20s%-20s%-20s","Cereal Name","Highest
                    Component","Level of Fat");
       System.out.println();
        for (int i = 0; i < size; i++)
       {
         boolean t = arr[i].getLowFat();
         if (t)
         {
            message = "Low";
         }
         else
         {
            message = "High";
         }
System.out.printf("%-20s%-20s%20s",arr[i].getCerealName(),
        arr[i].getHighestComponent(),message);
        System.out.println();
       }
     }
            public void updateComponents()
           {
Scanner kb = new Scanner(System.in);
System.out.println("Please enter the name of cereal to update");
String name = kb.nextLine();
boolean found = false; int i=0;
                while (!found && i < size)
       {
         if (arr[i].getCerealName().equalsIgnoreCase(name))
         {
           found = true;
           System.out.println("Enter the amount of Vitamins");
           int vit = kb.nextInt( );
           System.out.println("Enter the amount of Iron");
           int iron = kb.nextInt( );
           System.out.println("Enter the amount of fat");
           int fat = kb.nextInt( );
          arr[i].setComponents(vit,iron,fat);
         }
         i++;
       }
       if (!found)
       {
        System.out.println(name+" is not in file");
            }
   }
    public static void main(String args [ ])
  {
     Scanner Kb = new Scanner (System.in);   char ch = ' ';
     TestNutrition diet = new TestNutrition( );   while (ch != 'Q')
     {
System.out.println("    Menu\n")
       System.out.println("A - Display Cereals");
       System.out.println("B - Display Dietary Statistics");
       System.out.println("C - Search and Update");
       System.out.println("Q - QUIT");
       System.out.print("\nEnter Your Choice:");
       ch = Kb.next().toUpperCase().charAt(0);    switch (ch)
       {
         case 'A':
           {
             diet.displayCereals( );
             break;
           }
          case 'B':
           {
             diet.displayDietStats( );
             break;
           }
         case 'C':
           {
             diet.updateComponents( );
              diet.displayCereals();
             break;
           }
         case 'Q':
           {
             System.exit(0);
           }
       }
     }
   }
}
```