LAB 8 REQUIREMENTS

Problem:

Collecting data from past experiences can help us learn about how to make better decisions in the future. The longer we collect data, however, the harder it is to see a pattern developing because of the amount of information presented to us. There are many techniques to analyze data and obtain useful information from it. In this lab, we will use Visual C# to read in blood test results from patients who have a form of the Hepatitis virus and, in addition, we will link to the hospital's database to check whether each patient lived or died. We will perform some simple statistical processing on some of the data and hopefully will be able to draw some conclusions about interpreting blood tests from a Hepatitis patient and whether that patient will most likely live or die.

Hepatitis is a virus that attacks the liver. There are five main types of Hepatitis. Below is a brief description of the five most common types (there is a sixth very rare type).

Hepatitis A: is a liver disease caused by the hepatitis A virus (HAV). Hepatitis A can affect anyone. In the United States, hepatitis A can occur in situations ranging from isolated cases of disease to widespread epidemics.

Hepatitis B: is a serious disease caused by a virus that attacks the liver. The virus, which is called hepatitis B virus (HBV), can cause lifelong infection, cirrhosis (scarring) of the liver, liver cancer, liver failure, and death.

Hepatitis C: is a liver disease caused by the Hepatitis C virus (HCV), which is found in the blood of persons who have the disease. HCV is spread by contact with the blood of an infected person.

Hepatitis D: is a defective virus that needs the hepatitis B virus to exist. Hepatitis D virus (HDV) is found in the blood of persons infected with the virus.

Hepatitis E: is a virus (HEV) transmitted in much the same way as hepatitis A virus. Hepatitis E, however, does not often occur in the United States.

The most common form of spreading the Hepatitis virus person to person is by putting something in the mouth that has been contaminated with the stool of a person with hepatitis A. For this reason, the virus is more easily spread in areas where there are poor sanitary conditions or where good personal hygiene is not observed. More information is available from many different resources; one helpful resource is the CDC (Center for Disease Control).

PART A

Download the text and XML files (BloodTests.txt and BloodTests.xml) containing patient blood test data obtained from two different Blood Testing Centers (Qwest and Oracle) used by the Beaumont hospital's physicians. The two centers collect the same information but they store the data in two different formats: plain text and XML. Open the lab results from these two centers in notepad, WordPad or Internet Explorer and take a look at them. Notice that there are several fields. These fields are listed and briefly described below.

In the Blood Test File from the Blood Testing Center:

Patient Information

0. PatientID ***(Primary Key)***

Blood Tests for Certain Blood Characteristics

1. BILIRUBIN
2. ALK PHOSPHATE
3. SGOT
4. ALBUMIN

Now download the hospital medical records database (MedicalRecords1400.mdb) containing information about whether each patient lived or died from the class web site. Open the database in Microsoft Access. You will notice that there is a table in the database called ‘MedicalRecords’. This is the only table. Open it by double clicking on it, you will see several fields described below.

In the Hospital Medical Records database table ‘MedicalRecords’

Patient Information

0. PatientID ***(Primary Key)***
1. ALIVE: 1 = DIE, 2 = LIVE
2. AGE: Number of years
3. SEX: 1 = male, 2 = female

Symptom Information

4. STEROID: 1 = no, 2 = yes
5. ANTIVIRALS: 1 = no, 2 = yes
6. FATIGUE: 1 = no, 2 = yes
7. MALAISE: 1 = no, 2 = yes
8. ANOREXIA: 1 = no, 2 = yes
9. LIVER BIG: 1 = no, 2 = yes
10. LIVER FIRM: 1 = no, 2 = yes
11. SPLEEN PALPABLE: 1 = no, 2 = yes
12. SPIDERS: 1 = no, 2 = yes
13. ASCITES: 1 = no, 2 = yes
14. VARICES: 1 = no, 2 = yes

We won't be looking at every attribute. We are interested mainly in the data relating to the blood tests from the blood testing centers and whether the patient lived or not from the medical record database.

To start, create a new project and use the steps demonstrated in class and detailed step-by-step in your lecture notes and book in chapter 11 to connect to the database. Next, place a list box on the form and an 'Analyze' button. When the user presses the button, we will make some calculations and display the results in the list box. The calculations are explained in part B.

PART B:

Now we are ready to write the code for the 'Analyze' button on the main form. This button will perform some basic statistical processing using the blood test results in the text and XML files and cross reference that with whether that patient lived or died from data in the database. We will use the Average and Standard Deviation for the four blood test characteristics. These are Bilrubin, Alk Phosphate, SGot, and Albumin for the patients who lived and the patients who died, separately. We are looking for a pattern.

For example, suppose that we find for Bilrubin that the average Bilrubin level in patients who lived was 0.6 and the std. dev. was 0.2. Additionally, suppose that we find that the average Bilrubin level in patients who died was 0.65 and the std. dev was 0.18. This means that there really is no information that we can infer from this. For example, what would you tell a patient who had a Bilrubin level of 0.5? Hopefully, you would not tell them anything ☺

On the other hand, suppose that we find that the average SGot level in patients who lived was 27 with a std. dev. of 8. Additionally, suppose that we find that the average SGot level in patients who died was 44 with a std. dev. of 6, we could make some good projections. For example, what would you tell a Hepatitis patient who had an SGot level of 42? Outlook not good... ☹

So, in this lab, we are going to calculate the average and std. dev. of each of the 4 blood levels for the patients who lived and the patients who died, separately and write the results to the list box.

This should help us make some observations about these blood characteristics for patients with Hepatitis and whether they lived or died. From this, we may be able to generalize something about Hepatitis patients and whether they will most likely live or die. The results displayed in the list box should look something like this:

```
Average Bilrubin (Lived):  ###.##

Std. Dev Bilrubin (Lived):  ###.##

Average Bilrubin (Died):  ###.##

Std. Dev Bilrubin (Died):  ###.##

Average SGot (Lived):  ###.##

Std. Dev SGot (Lived):  ###.##

Average SGot (Died):  ###.##

Std. Dev SGot (Died):  ###.##

…
```

Display 2 decimal places for each value. For each record, the 'Alive' field will tell you if the data in that record is for someone who lived or died. For Alive = 1, the patient died and for Alive = 2, the patient lived.

As you go through the records and add the values up to calculate the mean, you will notice that you may run across a record with a –99 for a value. This is not a valid piece of data. A –99 has been put in places where the information was not available. You must skip these records. This means that if you have, for example, 50 records, and 2 with –99 as the value for Bilrubin, we will skip 2 records, thus the average will be calculated using the sum of the values in the 48 good records divided by 48 (the number of good records). Notice that records with –99 values for one attribute may have good data for other attributes. That is why it is best to use a different loop for each attribute. The pseudo-code for such a loop is written below:

```
Use LINQ to create a structured array called BloodTests from the text and XML files with the
PatientID, and Bilirubin.
For Each testresult in BloodTests
    If the value for Bilirubin is not –99
        Use LINQ to create an array that contains whether or not the patient lived or died
        from the record in the database with the same PatientID as this testresult
        For Each patient in Patients              (there should be exactly one)
            If Alive = 1 (Died) then
                Add the value to Bilirubin_died_sum
                Add the value2 to Bilirubin_died_square-sum (for the std. dev.)
                Add one to the 'good' Bilirubin_died_record_count
            Otherwise (must have survived)
                Add the value to Bilirubin_survive_sum
                Add the value2 to Bilirubin_survive_square-sum (for the std. dev.)
                Add one to the 'good' Bilirubin_survive_record_count
        Next patient
    End If
Next testresult
Compute Bilirubin_died_Average = the Bilirubin_died_sum divided by the 'good'
Bilirubin_died_record_count
Compute Bilirubin_died_Std. Dev. = Bilirubin_died_squared-sum divided by the 'good'
Bilirubin_died_record_count minus the Bilirubin_died_average squared all raised to the (1/2)
power.
Compute Bilirubin_survive_Average = the Bilirubin_survive_sum divided by the 'good'
Bilirubin_survive_record_count
Compute Bilirubin_survive_Std. Dev. = Bilirubin_survive_squared-sum divided by the 'good'
Bilirubin_survive_record_count minus Bilirubin_survive_average squared all raised to (1/2) power.
Add Info to List box
```

Do this for each of the 3 remaining blood characteristics.

There are two main ways to compute the standard deviation. The typical method shown below would require you to loop through all of the values in the database and keeping a running sum to compute the mean and then go back through them all again to subtract each value from the mean you calculated to compute the standard deviation. Using the alternative method shown below you can keep a running sum of squared values *(x^2)* in the same loop in which you are keeping a running sum of values *(x)* and therefore you can compute the mean and standard deviation after looping through the values only once.

Typical Method

$$\sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}}$$

First, you would need to loop through each value to compute the mean

Then, you would need to loop through a second time to subtract each value from the mean

Alternative Method

$$\sqrt{\frac{\sum_{i=1}^{n}(x_i)^2}{n}-\mu^2}$$

You would only need to loop through each value once where you can keep a running sum of ***x*** and a running sum of ***x^2*** and compute the mean and standard deviation after that single loop.

You should use the alternative method to make it easier for you. Add the information computed to the list box.

IMPORTANT: You will find that once you have written the small program described above for computing the Average and Std. Dev. for one attribute that with a slight modification, the entire thing can be copied and pasted and used for another attribute. If you want (it is not required) you may even want to make a method out of this code.

PART C:

Now that you've built the program, run the program and analyze the data. Using a piece of graph paper, neatly graph the Normal Distributions for an attribute for both patients who lived and who died on the same graph. This means that you will have 4 graphs. Each graph will have two 'bell-shaped' curves on them. Don't worry about labeling the y-axis, but make sure that your bell-shapes are scaled with respect to each other, that is, the areas under the bell shapes should be equal. Discuss which attributes have more useful information and which graphs have information that is not as useful and why.

What would you tell a patient with the following blood test results? Why?

Patient Name: William I. Survive

Bilrubin	1.3	Alk Phosphate	90
SGot	99	Albumin	3.7

Be sure that your code is carefully commented and that you understand what is going on in the program.

STEPS FOR SUBMITTING YOUR LAB:

For each lab and following comments must be added at the beginning of your Visual C# code.

/* ‘LAB #

‘SEMESTER NAME

‘STUDENT’S FIRST NAME, LAST NAME

‘I fully understand the following statement.

‘OU PLAGIARISM POLICY

‘All members of the academic community at Oakland are expected to practice and uphold ‘standards of academic integrity and honesty. An instructor is expected to inform and instruct ‘students about the procedures and standards of research and documentation required of students ‘in fulfilling course work. A student is expected to follow such instructions and be sure the rules ‘and procedures are understood in order to avoid inadvertent misrepresentation of her/his work. ‘Students must assume that individual (unaided) work on exams and lab reports and documentation ‘of sources is expected unless the instructor specifically says that is not necessary.

‘The following definitions are some examples of academic dishonesty:

- **‘Plagiarizing from work of others. Plagiarism is using someone else's work or ideas without ‘giving the other person credit; by doing this, a student is, in effect, claiming credit for ‘someone else's thinking. Whether the student has read or heard the information he/she uses, ‘the student must document the source of information. When dealing with written sources, ‘a clear distinction would be made between quotations (which reproduce information from ‘the source word-for-word within quotation marks) and paraphrases (which digest the ‘source information and produce it in the student's own words). Both direct quotations and ‘paraphrases must be documented. Just because a student rephrases, condenses or selects ‘from another person's work, the ideas are still the other person's, and failure to give ‘credit constitutes misrepresentation of the student's actual work and plagiarism of ‘another's ideas. Naturally, buying a paper and handing it in as one's own work is ‘plagiarism.**
- **‘Cheating on lab reports falsifying data or submitting data not based on student's own work.**

***/**

All labs will be submitted electronically, no paper copies will be given to Lab mentors.

Before submission:

- Please create a folder named as Lab8_FName_LName:
- **Place all your files and subfolders under this folder.**
- **Zip the folder** then upload through Moodle. You will not be able to upload unless you zip, 7zip or rar the folder.

GETTING READY FOR AN INTERVIEW with your Lab Mentor:

The interview is 40% of your lab grade. Make sure to be prepared for your mentor's questions about your program.

When it is your turn to explain your lab to your Lab mentor follow these steps **while your lab mentor is present**:

1. Log on to Moodle.
2. Find your submission link for this lab.
3. Download your Lab on your computer
4. Find your lab wherever you downloaded it to.
5. Make sure to Unzip, (or extract) your folder
6. Open the solution file to demo your lab.

You must follow these steps each time you are being graded for your lab. Your lab mentor must confirm that you downloaded what was submitted on Moodle. You should be graded on what was uploaded on Moodle, not on a local copy obtained from your C drive or external drives (i.e. memory sticks).

HOW WILL YOU BE GRADED BY YOUR LAB MENTOR AND WHAT IS THE GRADING CRITERIA?

1. The application works and was fully tested from what was downloaded and demonstrated from the copy uploaded to Moodle and not from a local copy or any external drive. (50 points)

2. Proper naming conventions were followed as explained in class (10 points)

3. Grade assigned based on oral examination of the students understanding of their solution and the overall quality of the solution (40 points)

GRADE: ________ out of 100