|  |  |
| --- | --- |
| **AP Computer Science** | **TextLab10 Java Assignment** |
| **The Student Records File Program** | **80, 90, 100 & 110 Point Versions** |
| **Assignment Purpose:**  This program requires knowledge and manipulation of Java text file creation and retrieval. | |

Write a program that reads in a text file called **"students.dat"**. The file is organized with a separate character string on each line. This is a sequential file of characters, but the file is organized to store student records with three fields in each record. There is a **name** field, **age** field and a **gpa** field. Keep in mind that Java does not realize these are records. You, as programmer, need to convert the **age** string to an integer and the **gpa** string to a double. The first five records of the **"students.dat"** text file are shown below.

|  |
| --- |
| Bart Reagor  27  2.075  Kristyn Reckner  19  3.225  Paul Reiman  41  4.000  Andy Reitinger  20  3.525  William Reynolds  50  3.375 |

Using the **File** class, **FileReader** class and the **BufferedReader** class of the **java.io.\*** package, you need to read in each line and display the results. You have no knowledge of the actual size of the file. You will need to use a conditional loop to keep entering file data until the end or **null** is reached.

The provided student file (shown on the next page) has a few things done for you. Three **File** objects are created (*f1*, *f2*, & *f3*). The first is for the **students.dat** file. The other 2 are for files you create in the 100 and 110 point versions of this assignment – which are explained later. The program checks to see if the **students.dat** file exists and then sets up the **BufferedReader** and **InputStreamReader** objects for you.

NOTE: Unlike most programs, this program will be done entirely in the **main** method. While this might not be considered the best program design, the focus of the assignment is *File Handling*. Putting everything in the **main** method just made everything simpler.

|  |  |
| --- | --- |
| **TextLab10 Student Version** | **Do not copy this file, which is provided.** |
| // TextLab10st.java  // The Student Records File Program  // This is the student, starting version of the Lab17TEXT04 assignment.  import java.io.\*;  import java.text.DecimalFormat;  public class TextLab10st  {  public static void main (String args[]) throws IOException  {  System.out.println("\nTextLab10\n");  DecimalFormat df = new DecimalFormat("00.000");  File f1 = new File("students.dat");  File f2 = new File("passing.dat");  File f3 = new File("honors.dat");  f2.delete();  f3.delete();    if (f1.exists())  {  FileReader inFile = new FileReader(f1);  BufferedReader inStream = new BufferedReader(inFile);            }  else  {  System.out.println(f1.getName() + " does not exist");  }  System.out.println();  }  } | |

NOTE: While placing the entire program in the **main** method is not normally considered *proper programming design*, for this program we are simply focusing on handling and manipulating text files. Breaking this up into different methods would add extra complexity which we do not need at this time. In chapter 18, there will be another program that uses text files with methods.

**80-Point Version Specifics**

The 80-point version only requires that the file is retrieved and displayed. It is not necessary to process any record information for this version.

**80-Point Version Output**

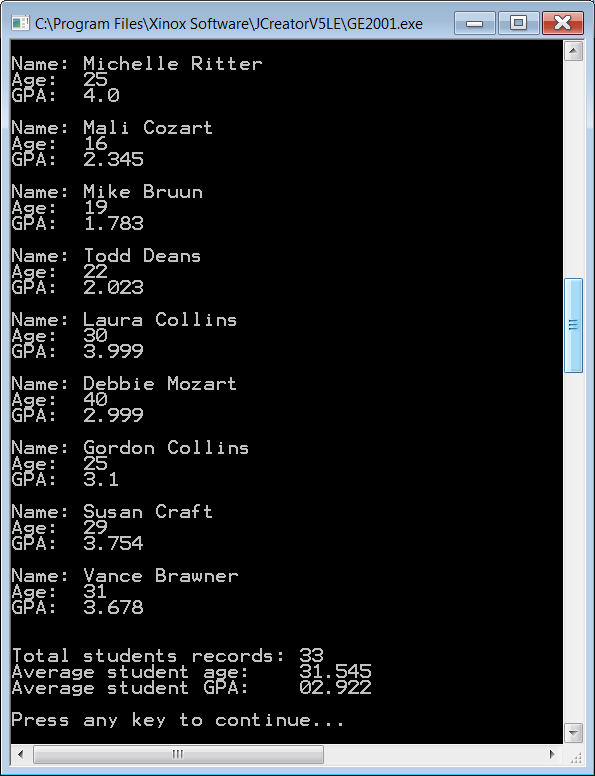
On the left is the beginning of the output. On the right is the end of the output.

|  |  |
| --- | --- |
|  |  |

**90-Point Version Specifics**

The 90-point version requires everything from the 80-point version but adds processing. Your program needs to **count** the number of records, compute the **meanAge** and the **meanGPA** for students. Keep in mind that both **age** and **gpa** are retrieved as strings from the text file, and require conversions. Both **meanAge** and **meanGPA** need to be displayed rounded off to three digits beyond the decimal point. This can be achieved with the **DecimalFormat** class and the **format** method.

**90-Point Version Output**

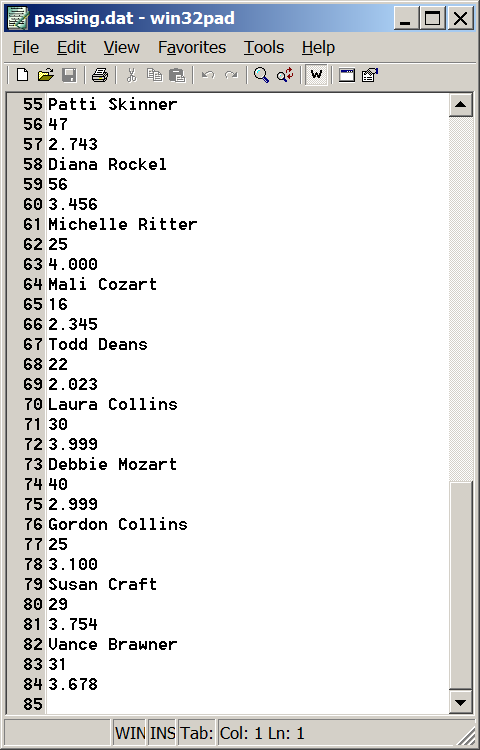


**100-Point Version Specifics**

The 100-point version requires everything from the 90-point version but adds file writing. Your program needs to create a file called **passing.dat** and as it reads in each student record, it needs to check if the student is passing (**gpa >= 2.0**). The **name**, **age**, and **gpa** of each passing student needs to be written to this file on a separate line.

**100-Point Version Output**

The monitor output of this file is the same as the 90-point version as will not be shown here; however, this program also has file output. When having this version of the program graded, you will need to show the monitor output first, and then show the file output in **Notepad**.



**110-Point Version Specifics**

The 110-point version requires everything from the 90-point version but also writes to an additional file. In addition to creating a file called **passing.dat**, you also need to create a file called **honors.dat**.

Just like **passing.dat**, as your program reads in each student record, and checks to see if the student is passing (**gpa >= 2.0**), it also needs to check to see if the student is on the honor roll

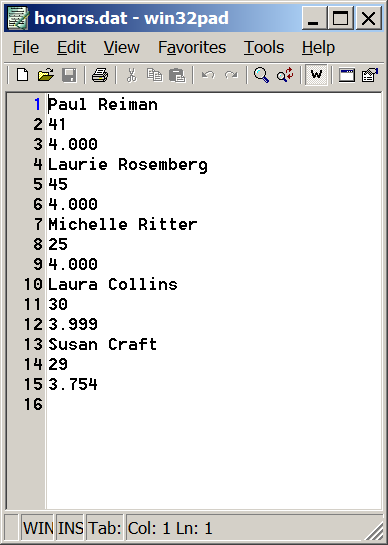
(**gpa >= 3.75**). The **name**, **age**, and **gpa** of each passing student needs to be written to this file on a separate line.

NOTE: All of the students who are written to the **honors.dat** file will also be written to the **passing.dat** file.

**110-Point Version Output**

The monitor output of this file is the same as the 90 and 100-point version as will not be shown here; however, this program also has file output. The file output for **passing.dat** is the same as the 100-point version and will also not be shown here.

What will be shown is the file output for **honors.dat**, which is considerable smaller than **passing.dat**.



When having this version of the program graded, you will need to show the monitor output first, and then show the file output for both **passing.dat** and **honors.dat** in **Notepad**.