```java
/*
   Gross Pay, Savings, and Investment Calculator data from txt file
   and output to a txt file.
   This program purpose is to calculate gross pay, savings, and investment from
   data stored in a txt file and then write the calculated data to a new file.
   Zachary Stall
   Program #4, CS 1050, Section 2
   jGRASP, Custom PC, Windows 10
   Assiduoud - Constant in application or attention; dilligen.
   "Great things are done by a series of small things brought together."
   -Vincent Van Gogh (1853 - 1890)
*/


import java.util.Scanner;           // Access the Scanner class
import java.io.*;                   // Access PrintWriter and related classes


public class ZacharyStall_2_04 {

   static Toolkit tools = new Toolkit();  // Toolkit allow for formatting

   public static void main(String[] args) throws IOException {


      // Files to get inputs and write outputs
      // All variable declared and described below

      final String INPUT_FILE  = "ZacharyStall_2_04_Input.txt";
      final String OUTPUT_FILE = "ZacharyStall_2_04_Output.txt";

      int numInputLines = 0;        // Number of lines in the input file
      int numValidLines = 0;        // Number of valid lines in the input file
      double grossPay = 0.0;        // Input file's gross pay
      double savingsRate = 0.0;     // Input file's savings rate
      double iraRate = 0.0;         // Input file's IRA investment rate
      double savingsAmount = 0.0;   // Calculated percentage saved
      double iraAmount = 0.0;       // Calculated percentage in ira
      double sumGrossPay = 0.0;     // Sum of all valid gross pay amounts
      double sumSavings = 0.0;      // Sum of all valid savings amounts
      double sumIra = 0.0;          // Sum of all valid IRA investment amounts
      double grossAverage = 0.0;    // Average of all valid gross salary
      double savingsAverage = 0.0;  // Average of all valid savings
      double iraAverage = 0.0;      // Average of all valid ira amounts

      // String vars for fromating and output

      String sumGrossPayStr;        // Sum of all valid gross pay
      String sumSavingStr;          // Sum of all valid savings
      String sumIraStr;             // Sum of all valid IRA
      String grossAverageStr;       // Average of all valid gross pay
      String savingsAverageStr;     // Average of all valid savings
      String iraAverageStr;         // Average of all valid IRA's

      // Access the input/output files

      File inputDataFile = new File(INPUT_FILE);
      Scanner inputFile  = new Scanner(inputDataFile);

      FileWriter outputDataFile = new FileWriter(OUTPUT_FILE);
      PrintWriter outputFile = new PrintWriter(outputDataFile);

      // Begin program execution

      System.out.println("Reading  file " + INPUT_FILE  + "\r\n" +
                         "Creating file " + OUTPUT_FILE + "\r\n");

      // Headers for the table of values for console and output file
```

```
69
70        outputFile.print(tools.padString("Grosspay", 12, " ", ""));
71        outputFile.print(tools.padString("Savings Rate", 15, " ", ""));
72        outputFile.print(tools.padString("Savings", 15, " ", ""));
73        outputFile.print(tools.padString("IRA Rate", 12, " ", ""));
74        outputFile.print(tools.padString("IRA", 15, " ", ""));
75        outputFile.println();
76
77        System.out.print(tools.padString("Grosspay", 12, " ", ""));
78        System.out.print(tools.padString("Savings Rate", 15, " ", ""));
79        System.out.print(tools.padString("Savings", 15, " ", ""));
80        System.out.print(tools.padString("IRA Rate", 12, " ", ""));
81        System.out.print(tools.padString("IRA", 15, " ", ""));
82        System.out.println();
83
84
85        // Read the input file and sum the numbers while there is data
86
87        while (inputFile.hasNext()) {
88           numInputLines++;                       // Adds the number of lines in input file
89           grossPay = inputFile.nextDouble();     // Gets grossPay from input file
90           savingsRate = inputFile.nextDouble();  // Gets savingRate from input file
91           iraRate = inputFile.nextDouble();      // Gets iraRate from input file
92
93           // Calculates the amount of money in savings and IRA
94
95           savingsAmount = (savingsRate / 100.0) * grossPay;
96           iraAmount = (iraRate / 100.0) * grossPay;
97
98           /*
99              If statement checks that the input is a valid line,
100             meaning that all values in the line are positive.If the
101             line is valid, it adds it to valid lines, each
102             variable respectively, and prints it in the outputfile
103             and console.
104          */
105
106          if(grossPay > 0 && savingsRate > 0 && iraRate > 0) {
107             numValidLines ++;
108             sumGrossPay += grossPay;
109             sumSavings += savingsAmount;
110             sumIra += iraAmount;
111
112
113             // Formats numbers and writes them in the output file
114
115             outputFile.print(tools.leftPad(grossPay, 12, "##,##0.00"));
116             outputFile.print(tools.leftPad(savingsRate, 15, "#0.0"));
117             outputFile.print(tools.leftPad(savingsAmount, 15, "#,##0.00"));
118             outputFile.print(tools.leftPad(iraRate, 12, "#0.0"));
119             outputFile.print(tools.leftPad(iraAmount, 15, "#,##0.00"));
120             outputFile.println();
121
122             // Echos the above to the console
123
124             System.out.print(tools.leftPad(grossPay, 12, "##,##0.00"));
125             System.out.print(tools.leftPad(savingsRate, 15, "#0.0"));
126             System.out.print(tools.leftPad(savingsAmount, 15, "#,##0.00"));
127             System.out.print(tools.leftPad(iraRate, 12, "#0.0"));
128             System.out.print(tools.leftPad(iraAmount, 15, "#,##0.00"));
129             System.out.println();
130
131          } // End if
132
```

```
133       } // End while
134
135       /*
136       If there is data then the averages will be calculated and stored
137       into their respective variables. If there is not data then to avoid
138       dividing by zero, the if/else statement will assign zero to the averages
139       and warn the user that the input file is empty.
140       */
141
142       if(numValidLines > 0) {
143       grossAverage = sumGrossPay / numValidLines;
144       savingsAverage = sumSavings / numValidLines;
145       iraAverage = sumIra / numValidLines;
146       }
147       else {
148       System.out.println("ERROR: FILE CONTAINS NO DATA!");
149       outputFile.println("ERROR: FILE CONTAINS NO DATA!");
150       grossAverage = 0.0;
151       savingsAverage = 0.0;
152       iraAverage = 0.0;
153       }
154
155       /*
156       formats all the data collected, and outputs all the data to the file
157       and the console using tools.leftPad.
158       */
159
160       sumGrossPayStr = tools.leftPad(sumGrossPay, 17, "$###,##0.00");
161       sumSavingStr = tools.leftPad(sumSavings, 19, "$##,##0.00");
162       sumIraStr = tools.leftPad(sumIra, 11, "$##,##0.00");
163       grossAverageStr = tools.leftPad(grossAverage, 16, "$##,##0.00");
164       savingsAverageStr = tools.leftPad(savingsAverage, 11, "$##,##0.00");
165       iraAverageStr = tools.leftPad(iraAverage, 15, "$##,##0.00");
166
167       outputFile.println("\r\n" + "The number of input lines read: " + numInputLines
168       + "\r\n" + "The number of valid input lines read: " + numValidLines + "\r\n" + "\r\n"
169       + "The sum of gross pay: " + sumGrossPayStr + "\r\n"
170       + "The sum of savings: " + sumSavingStr + "\r\n"
171       + "The sum of IRA investments: " + sumIraStr + "\r\n" + "\r\n"
172       + "The average gross pay: " + grossAverageStr + "\r\n"
173       + "The average savings amount: " + savingsAverageStr + "\r\n"
174       + "The average ira amount: " + iraAverageStr + "\r\n");
175
176
177       // Does the same as above and echos the information to the console
178
179       System.out.println("\n" + "The number of input lines read: " + numInputLines
180       + "\n" + "The number of valid input lines read: " + numValidLines + "\n" + "\n"
181       + "The sum of gross pay: " + sumGrossPayStr + "\n"
182       + "The sum of savings: " + sumSavingStr + "\n"
183       + "The sum of IRA investments: " + sumIraStr + "\n" + "\n"
184       + "The average gross pay: " + grossAverageStr + "\n"
185       + "The average savings amount: " + savingsAverageStr + "\n"
186       + "The average ira amount: " + iraAverageStr + "\n");
187
188
189
190       inputFile.close();
191       outputFile.close();
192
193       System.exit(0);
194
195    } // End main
196 } // End class
```