```
/* Percentage Grade Processor

   The purpose of this program is to read student names and
   percentage scores. Based on the percentage score the student
   earns, the program will classify it, and print the
   student name, score, and a message. The scores are recorded
   from a .txt file, and written to a .txt file (and echode to
   console).

   Zachary Stall
   Program #7, CS 1050, Section 2
   jGRASP, Custom PC, Windows 10

   Egalitarian - Asserting, resulting from, or characterized
   by belief in the equality of all people, especially in
   political, economic, or social life.

   "Freedom is never give; it is won"
   -A. Philio Randolph (1889 - 1979)
*/

import java.util.Scanner;  // For console input
import java.io.*;          // Access PrintWriter and related classes


public class ZacharyStall_2_07 {

   static Toolkit tools = new Toolkit();
   static Scanner console = new Scanner(System.in);

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

      // Access the input/output method
      final String INPUT_FILE  = "ZacharyStall_2_07_Input.txt";
      final String OUTPUT_FILE = "ZacharyStall_2_07_Output.txt";

      // Declare all var
      int grade      = 0;    // grade for each student
      int dataValues = 0;    // Total number of input values processed
      int totalCsBs  = 0;    // Total number of grades between 70 - 89

      double sumCsBs          = 0.0;   // Sum of CsBs to calc avg
      double sumAllScore      = 0.0;   // Sum of all grade to calc avg
      double avgCsBs          = 0.0;   // Average of grades between 70 - 89
      double avgAllScores     = 0.0;   // Average of all grades

      String name;      // Var to store names of students
      String message;   // Var to store the message for grade
      String table;     // Data output str

      // 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");

      // Prints the headers for the table
      displayHeader(outputFile);

      // Process the data from the input file
      while(inputFile.hasNext()) {

         grade = inputFile.nextInt();  // First # always a grade
```

```
69          name  = inputFile.nextLine();  // Next is student name
70          name  = name.trim();          // Trims leading and trailing spaces
71          dataValues ++;                // counts number of grades processed
72          sumAllScore += grade;         // sums all grades
73
74          if(grade >= 90) {
75             message = "OUTSTANDING";
76          }
77          else if(grade >= 70) {
78             message = "Satisfactory";
79             totalCsBs ++;
80             sumCsBs += grade;
81          }
82          else {
83             message = "Failing";
84          }
85
86          table = tools.padString(name, 20, "", " ") +
87                  tools.leftPad(grade, 5, "##0") +
88                  tools.padString(message, 18, " ", " ");
89
90          outputFile.println(table);
91          System.out.println(table);
92
93       }
94
95       // Get the average of C's & B's and Total
96       avgCsBs = getAverage(totalCsBs, sumCsBs);
97       avgAllScores = getAverage(dataValues, sumAllScore);
98
99       // Output information to console and file
100      outputMessage(outputFile, dataValues, totalCsBs, avgCsBs, avgAllScores);
101
102
103      inputFile.close();
104      outputFile.close();
105
106      System.exit(0);
107
108   } // End Main
109
110   // ********************************************************************
111   // Method for the header of the program
112   public static void displayHeader(PrintWriter output) {
113      String str;
114      str = tools.padString("Name", 22, "", " ") +
115            tools.padString("Grade", 7, "", " ") +
116            tools.padString("Message", 11, " ", "") +
117            "\r\n" +
118            "---------------------- " +
119            "------- " +
120            "---------------";
121
122     System.out.println(str);
123     output.println(str);
124
125   } // End headers
126
127   // ********************************************************************
128   public static double getAverage(int numberItems, double sum) {
129      return (double) sum / numberItems;
130   } // End getAverage
131
132   // ********************************************************************
133   public static void outputMessage(PrintWriter output,
134                                    int dataLine,
135                                    int numSat,
136                                    double avgSat,
```

```
137                                     double avgAllGrade)
138    {
139       String str;
140       str = "\r\n" +
141             "The number of grades processed is: " +
142             dataLine + "\r\n" +
143             "The number of scores that are \"Satisfactory\": " +
144             numSat + "\r\n" +
145             "The average of the \"Satisfacotry\" Scores: " +
146             tools.leftPad(avgSat, 0, "##0.0") + "%" + "\r\n" +
147             "The average of all the scores: " +
148             tools.leftPad(avgAllGrade, 0, "##0.0") + "%" + "\r\n";
149
150       System.out.print(str);
151       output.print(str);
152     } // End outputMessage
153
154    // ************************************************************************
155
156
157 } // End class
```