

Lectures/Week_15/classwork0415.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  // prototypes
6  double calculateWeightedAverage();
7  void analyzeScores(int scores[][5], int threshold, char letter_grades[][5]);
8
9  int main()
10 {
11     //-----
12     //----- TODO 1 -----
13     //-----
14     // Declare the grades matrix
15     int grades[3][4] = {{85, 92, 78, 88}, {91, 73, 65, 94}, {76, 88, 59, 69}};
16     char option;
17     char letter[3][4];
18     FILE *output_file_ptr;
19
20     // Prompt the user for input
21     printf("Enter A for Average or L for Letter Grade: ");
22     scanf(" %c", &option);
23
24     if (option == 'A')
25     {
26         // Open the file for writing
27         output_file_ptr = fopen("grades_report.txt", "w");
28
29         // Write the grades matrix and calculate averages
30         for (int i = 0; i < 3; i++)
31         {
32             int sum = 0;
33             for (int j = 0; j < 4; j++)
34             {
35                 fprintf(output_file_ptr, "%d\t", grades[i][j]); // Write grades one by
one, separated by tabs
36                 sum = sum + grades[i][j];                        // Calculate sum of
grades of each row
37             }
38             fprintf(output_file_ptr, "%.2f\n", sum / 4.0); // Write average
39         }
40
41         fclose(output_file_ptr);
42     }
43     else if (option == 'L')
44     {
```

```

45 // Convert grades to letter grades
46 for (int i = 0; i < 3; i++)
47 {
48     for (int j = 0; j < 4; j++)
49     {
50         if (grades[i][j] >= 90)
51         {
52             letter[i][j] = 'A';
53         }
54         else if (grades[i][j] >= 80)
55         {
56             letter[i][j] = 'B';
57         }
58         else
59         {
60             letter[i][j] = 'C';
61         }
62     }
63 }
64
65 // Print the letter grade matrix
66 printf("Letter Grade Matrix:\n");
67 for (int i = 0; i < 3; i++)
68 {
69     for (int j = 0; j < 4; j++)
70     {
71         printf("%c\t", letter[i][j]);
72     }
73     printf("\n");
74 }
75 }
76 else
77 {
78     printf("Invalid option! Please enter A or L.\n");
79 } // end if-elseif-else
80
81 //-----
82 //----- TODO 2 -----
83 //-----
84
85 FILE *another_file;
86 char filename[100]; // filename
87
88 // Filename error check
89 do
90 {
91     printf("Please enter file name: \n");
92     scanf("%s", &filename); //filename =

```

```

93     another_file = fopen(filename, "r"); //read mode
94 } while (another_file == NULL); //undesired condition,
95
96 // Student ID input error check
97 int studentID;
98 do
99 {
100     printf("Please choose a student ID (from 1001, 1002, 1003 and 1004):\n");
101     scanf("%d", &studentID);
102 } while (studentID < 1001 || studentID > 1004); //undesired condition
103
104 int success = 1, idx = 0, sum_hr = 0, weighted_sum = 0;
105 int ID[100], hours[100], gds[100]; // assume you have no more than 100 rows
106
107 // while-loop to find all entries related to this student ID
108 while (success != EOF)
109 {
110     // scan from file every three numbers
111     success = fscanf(another_file, "%d %d %d", &ID[idx], &hours[idx],
&gds[idx]);
112     if (success != EOF) //if here to avoid the last item to be read
113     {
114         if (ID[idx] == studentID)
115         {
116             sum_hr = sum_hr + hours[idx];
117             weighted_sum = weighted_sum + hours[idx] * gds[idx];
118         }
119         idx = idx + 1;
120     }
121 }
122 fclose(another_file); // Close the file after reading
123
124 float weighted_avg = (weighted_sum + 0.0) / sum_hr;
125 printf("The weighted average grade is %06.2f\n", weighted_avg);
126 // if a function is built, call via weighted_avg = calculateWeightedAverage();
127
128 //-----
129 //----- TODO 3 -----
130 //-----
131
132 int scores[2][5] = {{88, 94, 76, 67, 90}, {85, 73, 82, 79, 95}};
133 char letter_grades[2][5];
134 int number = 85;
135
136 analyzeScores(scores, number, letter_grades);
137 printf("Letter Grades Matrix:\n");
138 for (int i = 0; i < 2; i++)
139 {

```

```

140     for (int j = 0; j < 5; j++)
141     {
142         printf("%c\t", letter_grades[i][j]);
143     }
144     printf("\n");
145 }
146 return 0;
147 } // end main
148
149 double calculateWeightedAverage()
150 {
151     FILE *another_file;
152     char filename[100];
153
154     // Filename error check
155     do
156     {
157         printf("Please enter file name: \n");
158         scanf("%s", filename);
159         another_file = fopen(filename, "r");
160     } while (another_file == NULL);
161
162     // Student ID input error check
163     int studentID;
164     do
165     {
166         printf("Please choose a student ID (from 1001, 1002, 1003 and 1004):\n");
167         scanf("%d", &studentID);
168     } while (studentID < 1001 || studentID > 1004);
169
170     int success = 1, idx = 0, sum_hr = 0, weighted_sum = 0;
171     int ID[100], hours[100], gds[100]; // assume you have no more than 100 rows
172
173     // for-loop to find all entries related to this student ID
174     while (success != EOF)
175     {
176         // scan from file every three numbers
177         success = fscanf(another_file, "%d %d %d", &ID[idx], &hours[idx],
178         &gds[idx]);
179         if (success != EOF)
180         {
181             if (ID[idx] == studentID)
182             {
183                 sum_hr = sum_hr + hours[idx];
184                 weighted_sum = weighted_sum + hours[idx] * gds[idx];
185             }
186             idx = idx + 1;
187         }
188     }

```

```
187     }
188     fclose(another_file); // Close the file after reading
189
190     float weighted_avg = (weighted_sum + 0.0) / sum_hr;
191     printf("The weighted average grade is %06.2f\n", weighted_avg);
192     return weighted_avg;
193 } // end calculateWeightedAverage
194
195 void analyzeScores(int scores[][5], int threshold, char letter_grades[][5])
196 {
197     for (int i = 0; i < 2; i++)
198     {
199         for (int j = 0; j < 5; j++)
200         {
201             if (scores[i][j] >= threshold)
202             {
203                 letter_grades[i][j] = 'H';
204             }
205             else
206             {
207                 letter_grades[i][j] = 'L';
208             }
209         }
210     }
211 } // end analyzeScores
```