Department of Information Systems and Technologies

CTIS 152 –Algorithms and Data Structure

FALL 2024 - 2025

Lab Guide #6 – Week 5 – 1

OBJECTIVE : Structures as Function Parameters, Nested Structures

Instructor : Burcu LIMAN
Assistant : Berk ÖNDER, Engin Zafer KIRAÇBEDEL

1. Consider a text file named “students.txt” that contains information about different students, including the student name, unit GPA, and class level.

Write a C program that reads the information from the file, stores it in an array of structures, calculates the total GPA of all students, and displays a report showing the student details and their total GPA.
Write the following functions;

- **`readFromFile:`** takes a file pointer and the student array, reads the information for all students from the file into the structure array, and returns the number of elements in the array.
- **`calculatewWeightedGpa`:** takes the student array and the number of students as parameters and calculates the weighted GPA (which is GPA * class level) for each student.
- **`calculateTotalWeightedGpa`:** takes the student array and the number of students as parameters and calculates the total weighted GPA of all students combined.

students.txt

```
Berk_Onder   3.5     12
Bengi_Onder  3.8     11
Dobi_Onder   2.9     10
Bora_Onder   4.0     12
```

Example Run:

```
 Student Name   GPA    Class Level
 ----------------------------------
 Berk_Onder      3.50    12
 Bengi_Onder     3.80    11
 Dobi_Onder      2.90    10
 Bora_Onder      4.00    12

 Student Name   GPA    Class Level   Weighted GPA
 -------------------------------------------------
 Berk_Onder      3.50    12              42.00
 Bengi_Onder     3.80    11              41.80
 Dobi_Onder      2.90    10              29.00
 Bora_Onder      4.00    12              48.00

The total weighted GPA of all students: 160.80
```

Project Name: LG6_Q1
File Name: Q1.cpp

2. **a)** Consider a text file named “smartphones.txt” that contains information about different smartphones, including their brand, model, battery score, camera score, and performance score.

Write a C program that reads the information from the file, stores it in an array of structures, calculates the total score for each smartphone based on the given weights, and displays a report showing the smartphone details and their total score. The program should use dynamic memory allocation for the array of structures.

Project Name: LG6_Q2a
File Name: Q2a.cpp

b) Add the following functions to your solution in part a
Write the following functions;

- **`readFromFile:`** takes a file pointer, an array of smartphones, and the number of smartphones as parameters. It reads the information for all smartphones from the file and stores it in the nested structure array.
- **`displayReport:`** takes the array of smartphones and the number of smartphones as parameters. It displays the detailed information about each smartphone, including their brand, model, battery score, camera score, performance score, and overall score.

Overall score = **battery (%25) + camera (%40) + performance (%35).**

Project Name: LG6_Q2b
File Name: Q2b.cpp

smartphones.txt

```
6
Apple iPhone14 90 85 95
Samsung GalaxyS21 80 90 80
Google Pixel5 75 95 90
OnePlus 9 70 85 88
Xiaomi Mi11 65 80 92
Sony Xperia 5 75 75 70
```

Example Run:

```
Brand              Model              Battery  Camera      Performance  Overall Score

------------------ ------------------ ------- ------------ ----------- --------------

Apple              iPhone14                90           85           95         89.75

Samsung            GalaxyS21               80           90           80         84.00

Google             Pixel5                  75           95           90         88.25

OnePlus            9                       70           85           88         82.30

Xiaomi             Mi11                    65           80           92         80.45

Sony               Xperia                   5           75           75         57.50
```

3. A college wants to determine the scholarship for the students depending on the following criteria. Registered students' information consists of **student name**, **surname**, **scholarship exam score** and **scholarship info.** Scholarship info has **a scholarship type**, **degree** and **percentage rate.**

Write a C program that will read all of the information from the text file named **"scholarship.txt"** into a **dynamically** created **nested structure array**. The percentage rate will be calculated while reading the data using the following criteria. The program displays the student information and calculates the school fee according to the student' scholarship. Finally, the program displays the total earnings of the school by getting the summation of the school fees like in the example run.

The college fee is 68.500 TL.

Scholarship Type	**Degree**		
	1	**2**	**3**
Martyr's Relative - R	100% Scholarship	20% Scholarship	---
Talented Person - I	50% Scholarship	30% Scholarship	15% Scholarship

Extra scholarship will be calculated. If the scholarship exam score is:

- greater than or equal to 90, extra scholarship is 15%
- less than 90 and greater than or equal to 70, extra scholarship is 10%
- less than 70 and greater than or equal to 50, extra scholarship is 5%

scholarship.txt

```
8
Zeynep Agca 87.5 R 2
Cagatay Revan 66.9 I 3
Nebi Sever 35.8 R 1
Basak Oduncu 98.0 I 2
Emre Sakaray 43.5 R 2
Vahide Poslu 100.0 R 1
Sennur Aldag 46.0 R 2
```

Write the following functions;

- **readFromFile** that reads the students' information from the file, calculates and stores the percentage rate into an array of structures.
- **displayReport** that displays the information of all students with the total earnings of the school as in the example run.

Example Run:

```
Student Name Surname  Scholarship Exam Score  Scholarship Type  Percentage  School Fee
--------------------  ----------------------  ----------------  ----------  ----------
Zeynep     Agca               87.5                   R             %30        47.950
Cagatay    Revan              66.9                   I             %20        54.800
Nebi       Sever              35.8                   R             %100        0.000
Basak      Oduncu             98.0                   I             %45        37.675
Emre       Sakaray            43.5                   R             %20        54.800
Vahide     Poslu              100.0                  R             %100        0.000
Sennur     Aldag              46.0                   R             %20        54.800
Elif       Okcu               91.5                   I             %65        23.975

Total earnings of the school -> 274.000 TL.
```

Project Name: LG6_Q3
File Name: Q3.cpp

Additional Question

Create a nested structure **applicantsOfII** and **grades** as follows:

```
typedef struct{
        int englishProficiency, jury, graduateExam;
 } grades_t;

 typedef struct{
             int id;
             grades_t gr;
             double overall;
}applicantsOfII_t;
```

Write the following functions:

- **readFile** that takes a set of application information from a text file named **"applicants.txt"** until the end of file is reached, also returns the size of the structure array (Do not forget to initialize the overall grade to 0 for each student).
- **calculate** that calculates the overall applicants' grades' average and the overall grade of each applicant with the loads of English proficiency being 30%, jury being 50%, and the graduate exam being 20%)
- **display** that displays the content of the structure array of **applicantsOfII_t** type.
- **findPassFail** that finds and displays the number of the applicants who fail and pass the elimination as well as displaying the average of all applicants' grades'. (An applicant passes if overall >= average, otherwise student fails).

Write a C program that reads the entirety of applicants' information from applicants.txt file into an array of structures, and displays all the information on the screen as necessary, as shown in the example run below.

applicants.txt

```
1111 49 65 94
2222 79 58 45
3333 76 88 95
```

Example Run :

```
Applicant ID: 1111

Scores:
Applicant English Proficiency: 49
Applicant Jury: 65
Applicant Graduate Examination: 94
Applicant Overall: 66.0

Applicant ID: 2222

Scores:
Applicant English Proficiency: 79
Applicant Jury: 58
Applicant Graduate Examination: 45
Applicant Overall: 61.7

Applicant ID: 3333

Scores:
Applicant English Proficiency: 76
Applicant Jury: 88
Applicant Graduate Examination: 95
Applicant Overall: 85.8

Average is 71.2
Number of the applicants who pass is 1
Number of the applicants who fail is 2
```