**Class Performance 3:**

**Question 1 (Structs):**

Suppose you want to determine the lowest goal scorer among three football teams. Each team will have 3 players. Consider each team as a structure and each team will have three properties: goal scored by each member of the team, name of the team, ID of the team. These properties will be taken as input from the user. Your task is to print the name of the team that scored the lowest goal. **[Hint: You need to take the goal of each member of the team in an array] [5 marks]**

***Example:***

**Score of Team1:**

Player 1-3

Player 2- 4

Player 3- 1

Total score= 8

**Score of Team2:**

Player 1-2

Player 2- 5

Player 3- 0

Total score= 7

**Score of Team3:**

Player 1-3

Player 2- 1

Player 3- 1

Total score= 5

Output:

Team 3 scored lowest goals.

**Class Performance 4:**

**Question 2 (Syscalls):**

Create an empty text file called **a.txt** using the touch command.

Now create a c program called **another.c** which will write the string "Hello from {PID}" to the file called a.txt, then sleep for 1 second, and then exit. Here, in place of {PID}, there should be the process ID of the currently executing program. To format the string to insert the PID, you can use the **sprintf()** function, more on that below.

After that, create another c program called **main.c**. This program will load & run the previously written program as a **child process**. After running it, it (from the **parent process**) should read the **a.txt** file and print its content into the console.

All file related operations should be done using system calls. You’re **not allowed** to use fopen(), fprintf(), fscanf(), fclose() functions.

**Example prompt and output:**

$ touch a.txt

$ gcc -o another another.c

$ gcc -o main main.c

$ ./main

Hello from 1872

*Remember, the PID is assigned by the OS, so don't panic if your PID doesn't match with the example output.*

**sprintf function usage:**

#include <string.h> // don't forget to include this for sprintf

char buffer[100];

int num = 42;

sprintf(buffer, "Hi this is %d", num);

// Now the string "Hi this is 42" will be stored in buffer