Object-Oriented Programming
Assignment 3 (200 Marks)

Cairo University, Faculty of Computers and Artificial Intelligence

Instructions

1- Students will form teams of **3** students **from the same lab group.**
2- Deadline of submission is **Monday Apr.29th at 11:55 pm.**
3- Submission will be on google classroom.
4- No late submission is allowed.
5- No submission through e-mails.
6- Submit a single cpp file named as LabGroup_ID1_ID2_ID3.cpp. For example, S3_20202020_22112211_22101010.cpp.
   The character separating the group name and the ids is an Underscore character. Do not use any other characters like a space, a dash, or a bracket,…etc.
   Do not include your name in the file name.
   Do not submit more than 1 file.
   Do not submit multiple files.
   Do not submit project files (like cbp and sln)
   Do not submit exe files (your file will be considered a virus and won't be downloaded).
   Do not submit zip or rar files.
7- Failing to follow submission instructions will cost you the grade of the assignment.
8- **In case of Cheating, you will get a negative grade whether you give the code to someone, take the code from someone/internet, or even send it to someone for any reason.**
9- You have to write clean code and follow a good coding style including choosing meaningful variable names.

Task

A class **Player** is a class that holds the info of a football player including the following private member variables:
1- Name (string)
2- Year of Birth (int)
3- Height (int)
4- International Goals (int)
5- Nationality
Add appropriate setters and getters for the class.

Object-Oriented Programming

Assignment 3 (200 Marks)

Cairo University, Faculty of Computers and Artificial Intelligence

The class **NationalTeam** is a class holding the players in a national football team of a certain country. The class has an array of Player objects [**players**] of size 18, holding all the players including substitutes (fixed size).
The class has another array of Booleans [**active**] of size 18, where an element at index i in the **active** array is true if there's a player in the players array at index i, and false otherwise. Both arrays are private members of the class.
Other members of the class include country.
The array of players is initially empty, and the active array's elements are set to false.
We can add to the array using the **addPlayer(const Player&)** member function, which will add the parameter player object in the first empty slot in the **players** array. The function will also update the **active** array to indicate that a player is added at that index. The function returns true if the player was added, false otherwise.
We can also remove players using **removePlayer(string)** from the players array by giving the player's name and the player will be removed (the player object will not actually be removed but, the active array will be updated to indicate this). Make sure that the name entered is not case sensitive. The function will return true if the player was found and removed, false otherwise.
There other public member functions in the class including:

- getNumOfPlayers // to return total number of players in the team
- AvgHeight() // to return the average height of the team players
- OldestPlayer() // to return the oldest player object
- YoungestPlayer() // to return the youngest player object
- AvgIntGoals() // to return the average number of international goals per player in the team
- highestIntGoals() // to return the player with the highest number of international goals
- operator<< // to output a team's country and its players info.
- writeStatistics(oftream&)
  // writes team statistics to a file including:
  - Team Country
  - Number of players
  - Average height of players
  - Name and age of oldest player
  - Name and age of youngest player
  - Average number of international goals of players
  - Name of the player with the highest international goals and the number of goals scored by this player.

Object-Oriented Programming

Assignment 3 (200 Marks)

Cairo University, Faculty of Computers and Artificial Intelligence

Ex.

Country:		
# of Players:		
Avg. Height:		
Oldest Player:	Age:	
Youngest Player:	Age:	
Avg. Int. Goals:		
Player with highest goals:		Goals:

Create a global function **readPlayersFromFile(ifstream&)** which reads all players info from a file named "players.txt" into an array of player objects. Note that this file will have all of the players of all the teams that will be added later. The file will have the number of players in the file at the first line of the file.

An example file is shown:

```
100
Mohamed Salah
1992
173
48
Egyptian
//////same for the rest of players until player 100//////
```

Finally, create a main that will:

1- read players from players.txt into an array of player objects.

2- Ask the user to enter a number of national teams where the user will just enter their country names.

3- Next Display the players read from the file and the national teams entered by the user.

4- Now in a loop:

Ask the user to add/remove players to teams as many times as they wish

You can display players and teams numbered in step 3 such that the user can choose to add a player to a team by their displayed numbers.

To remove a player the user has to enter their name to be removed.

5- After finishing step 4, show all of the national teams added by the user including the country and the players in this team using the operator<<.

6- Finally write the statistics of each of the teams to the file statistics.txt. All teams statistics should be in this single file.

Object-Oriented Programming
Assignment 3 (200 Marks)

Cairo University, Faculty of Computers and Artificial Intelligence

Writing Good Quality Code

No program stays the same. It will need to change to fix bugs, add new features, etc. So, it is very important to write high quality readable code, so that you or other developers can be able to review and modify this code in the future. In this task, you will:

1. Add a comment/header at the beginning of your program saying who the author is, the purpose of the program, etc.
2. Add a comment/header for every function explaining what it does, what parameters it takes and what value it returns.
3. Add comments to any part that is difficult to understand.

Grading criteria (Total 200 marks)

Good Quality Code (10 marks)
Player Class (30 marks)
NationalTeam Class (100 marks)
readPlayersFromFile(20 marks)
main (40 marks)