COMPLEX ENGINEERING ACTIVITY

PROJECT REPORT



OBJECT ORIENTED PROGRAMMING

**Name:** Ammar Nadeem

**Roll no:** 220600

5/28/2023

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**Objective:**

The objective of the given code is to implement a cricket scorekeeping program. The code allows users to enter the details of a cricket match, such as the number of overs, player names, and their performance in terms of runs, fours, sixes, and balls faced. The code keeps track of the current score, wickets, overs, and other statistics by allowing the user to input the number of runs scored on each ball and then update the score accordingly. The code also includes functions to output the match summary to a file. The program uses classes to organize the code and stores player data in form of arrays.

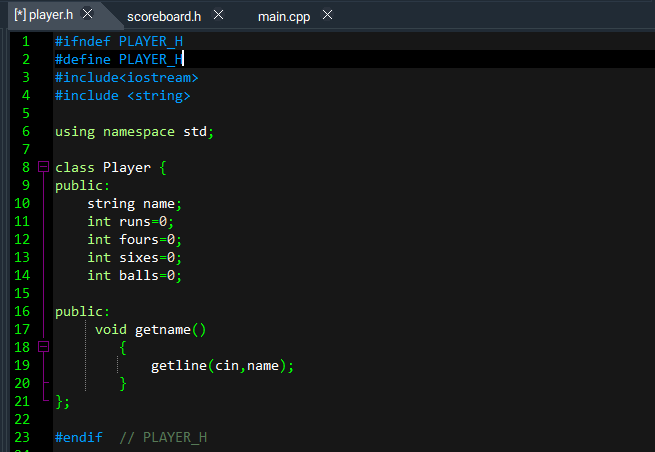
**Implementation:**

Here is an overview of the implementation:

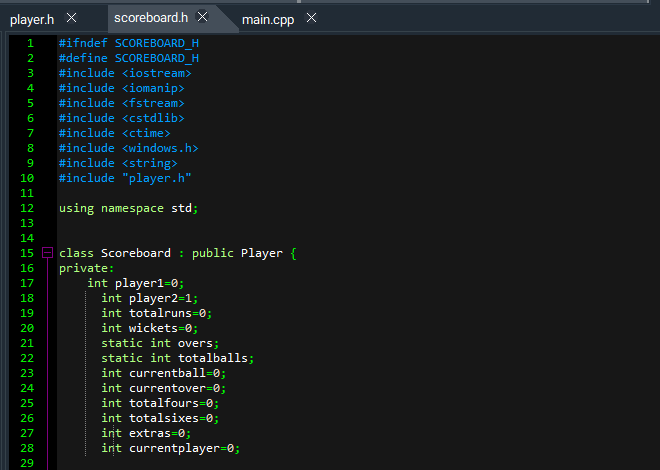
1. The code starts by including necessary header files for input/output, string manipulation, file handling, and Windows-specific functions.
2. The code defines a class called "Player," which represents a cricket player. It has member variables to store the player's name, runs, fours, sixes, and balls faced. It also includes a member function to get the player's name.
3. The code defines another class called "Scoreboard," which inherits from the "Player" class. It represents the cricket scoreboard and contains additional variables and functions related to the game.
4. Within the "Scoreboard" class, there are several private member variables, such as player indices which is an array of objects of the player class (namely, composition is used), total runs, wickets, overs, balls, etc. These variables are used to keep track of the game's progress and statistics.
5. The "Scoreboard" class includes member functions to get the number of overs, get player names, display the current score, calculate runs based on user input, display player statistics, calculate the target score, write the game summary to a file, and more.
6. The implementation utilizes a combination of loops, conditionals, and input/output statements to interact with the user, update the game state, and display information on the console.
7. The code also uses Windows-specific functions to change the text color in the console window for better visual representation.
8. The program also have two options, one to enter teams manually and second to read the already existing teams in txt files. The program also gives the user, the option to perform a toss which uses srand function.
9. There are two variations of the "currentscore" function in the "Scoreboard" class. One is non-parameterized and displays the score for the 1st innings, while the other takes an instance of the "Scoreboard" class and an additional parameter to display the score in the context of a 2nd innings (with a target score).
10. The implementation includes error handling for invalid user inputs, such as incorrect runs or out-of-range values.
11. The program uses functions to and conditions to check if the team has achieved the target, upon which the program stops and displays the message that the corresponding team has won.
12. At the end of the game, the program writes the game summary to a file named "Score.txt" using file handling operations which contains of the statistics of different players and the number of runs they made, balls faced and their number of sixes as well as fours.

**Snippets of Code:**

1. **“player.h” header file:**

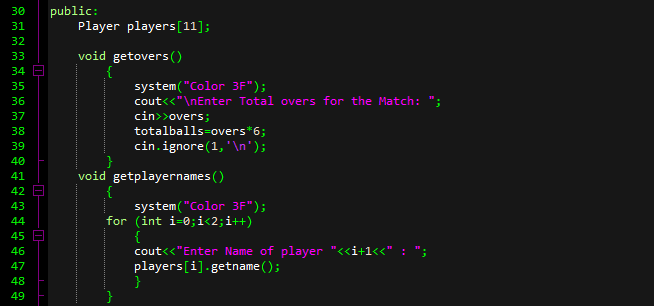


1. **“Scoreboard.h” header file:**

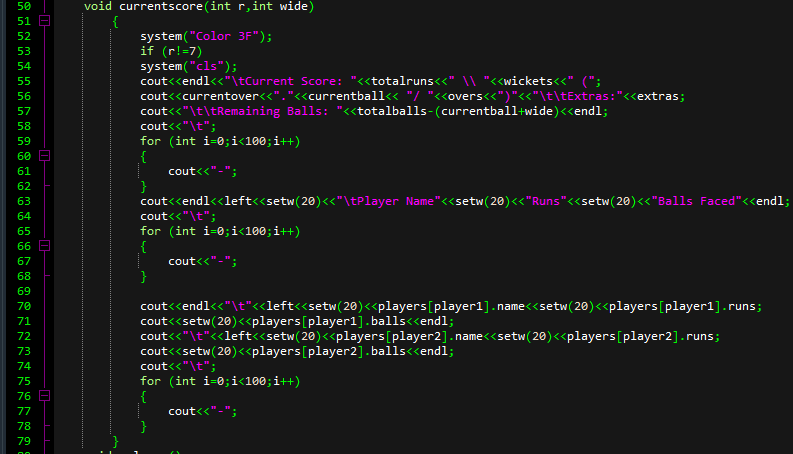


1. **Public functions to get names, overs and other details for the match:**

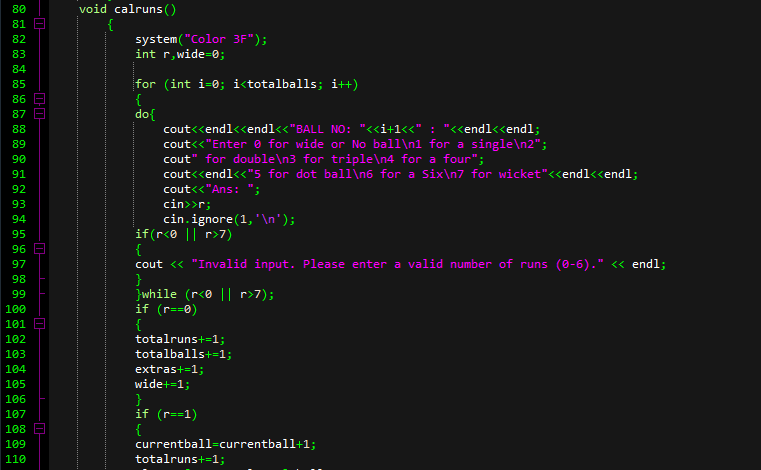
It also contains composition in form of an array of objects from player class.



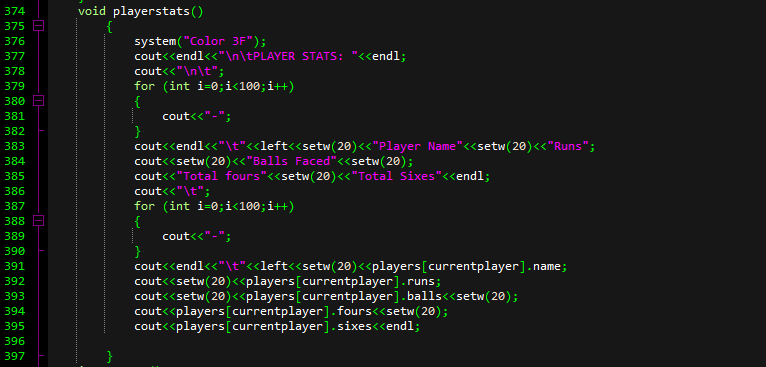
1. **Function to Display the Current Score:**



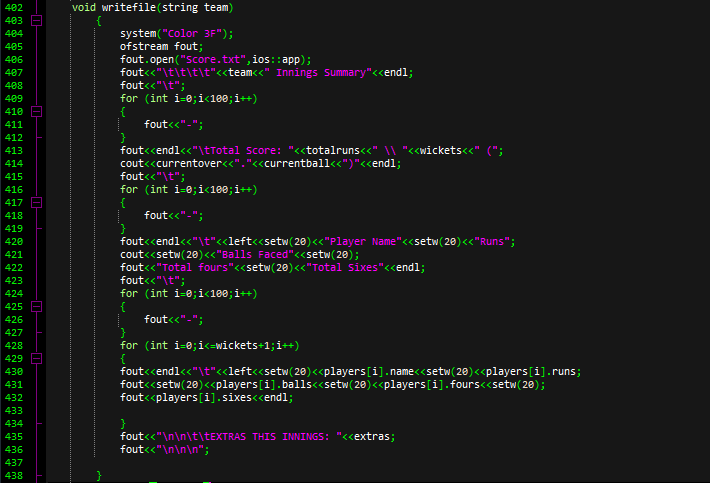
1. **Function that displays a menu to user and then checks particular conditions:**



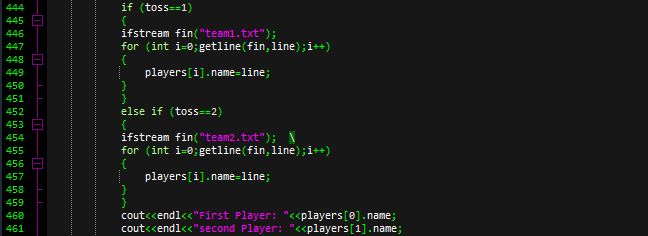
1. **Function to Display the Stats of a Players when they get out:**



1. **Function to write Summary of innings in txt file:**

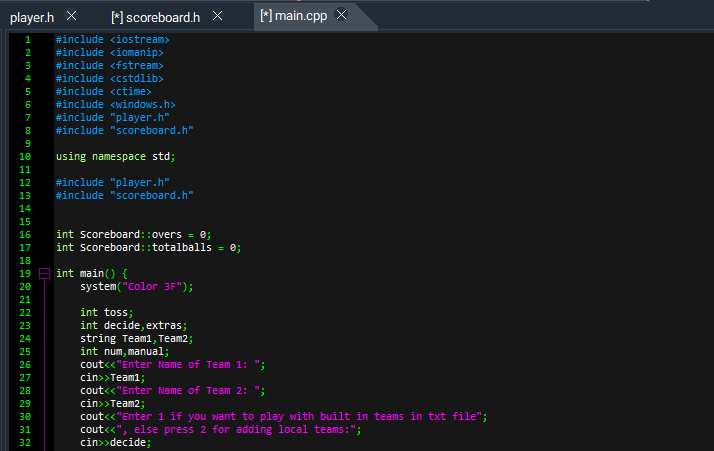


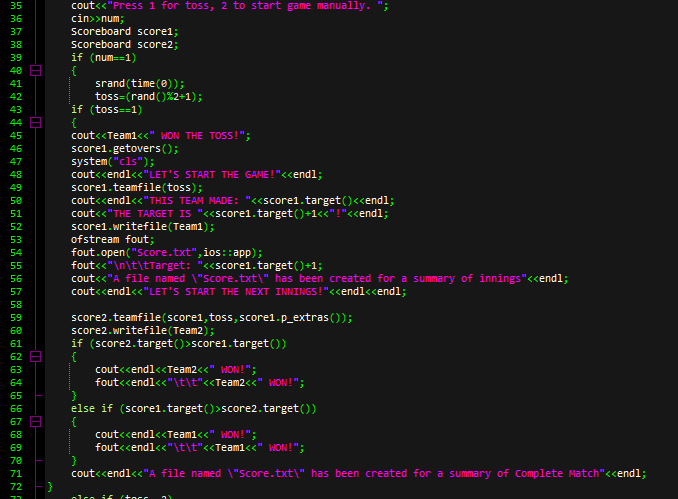
1. **Taking input of player names from txt file:**



1. **“Main.cpp” file:**

Main filecontains int main and all function calls along with object creations for scoreboard class. It also lets the user to perform toss and choose different options for the match like using txt files to create teams or create them manually.





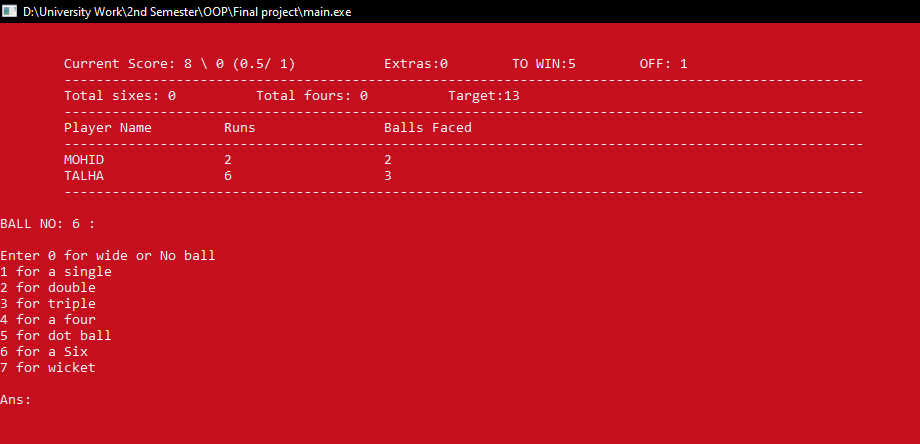
**Program outputs:**



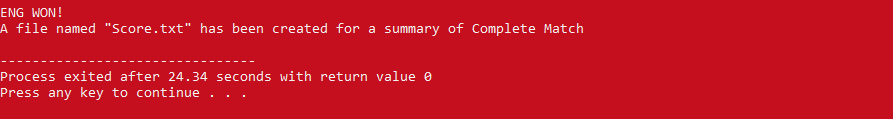
**Program design (1st innings):**



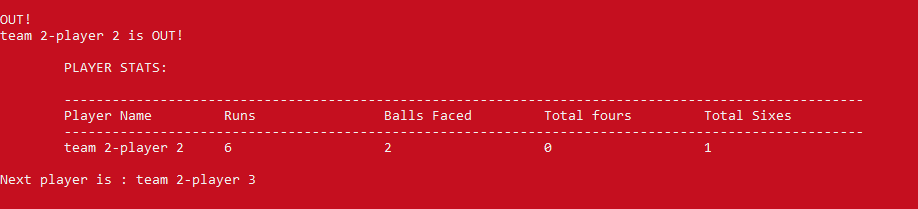
**2nd innings:**



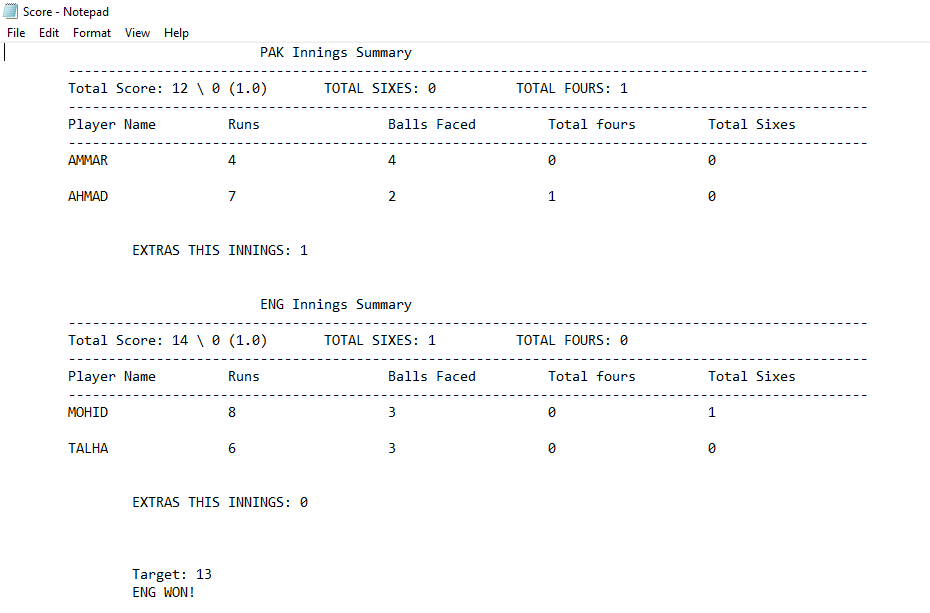
**If a team wins:**



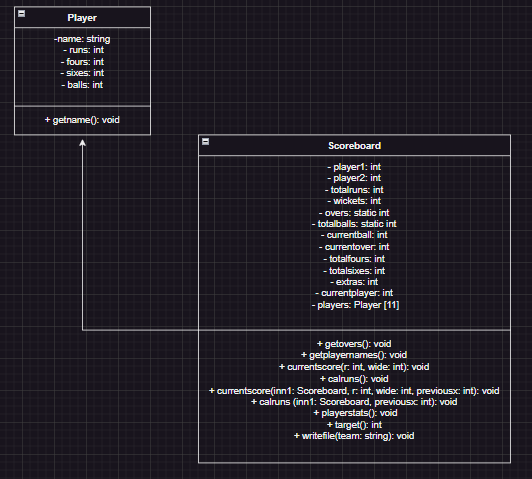
**If a player gets out:**



**Created Summary of Match in a txt file:**



**UML Diagram for code:**



**Application Software Used:**

Dev C++ is a user-friendly integrated development environment (IDE) for C and C++ programming. It offers a range of features such as code auto completion, debugging tools, and project management capabilities. With Dev C++, developers can efficiently write, compile, and debug their C and C++ applications.



**Conclusion:**

In conclusion, this project enabled us to implement all the concepts of object oriented programming into a single and complex program. Concepts such as encapsulation, inheritance, polymorphism and composition were used to create a clear and logical structure for the code. We also learned how different object oriented programming is useful for storing long information about different objects. Moreover, concepts such as file handling were also implemented in the program. We also learned to create and represent long codes with the help of UML diagrams.