Module 3 Assignment
Fundamentals of Object Oriented Programming Using CPP

Hi, Welcome Back! Hope you had fun while coding your first assignment. Remember those school days when we used books to play cricket to pass time during boring lectures. Well, this assignment revolves around the same nostalgia where you would create a similar cricket game using the concepts of C++ which you have learned in this module. This assignment is a brief version of the final project, so we strongly recommend you to complete this and don't worry, we have added all the necessary details that will give you a headstart to code this assignment. **Happy Coding :)**

Problem Statement

Write a C++ program to build a mini cricket application named **Gully Cricket App**.

Project Requirements

1. There should be two teams: TeamA and TeamB
    a. Each team will have 3 players
    b. The player names will not be entered by the user. You can assign names to the players yourself
2. There should be two innings
    a. Each inning will be of 6 balls (one over)
        i. In each inning, one batsman from the batting team will bat for 6 balls and one bowler from the bowling team will bowl 6 deliveries
        ii. One batsman from the batting team and one bowler from the bowling team will be selected randomly for each inning
    b. TeamA will always bat first which means TeamB will always bowl first
3. In each delivery, runs can be scored from 0 to 6
4. There will be no criteria to get wickets. In simple words, once a batsman starts his inning, he will bat for all the 6 balls without getting out/dismissed/retired hurt etc
5. After completion of two innings i.e. after each team has done batting, scored runs will be compared to decide the winner or to decide if there is a tie

Note: The match will conclude only after each batting team has faced 6 deliveries. In the second inning, at any point of time during the match, if the score of TeamB is greater than the runs scored by TeamA (in the first inning) then the match should not end. The match should continue until TeamB has faced all the 6 deliveries.

Rough Algorithm

1. Create a class Team that can store the following data
    a. Team name
    b. The three players
    c. Total runs scored
2. Greet user with a welcome message
3. Initialize two teams
4. Display team details
5. Select a batsman and a bowler for the first inning
6. Start playing the first inning
7. Record total runs scored by the batting team in the first inning
8. Select a batsman and a bowler for the second inning
9. Start playing the second inning
10. Record total runs scored by the batting team in the second inning
11. Decide the winner

Approach

1. Create a class Team that can store the following data:
    a. Team name
    b. The three players
    c. Total runs scored
2. Create two global variables to store the name of the current batsman and current bowler in the inning being played
3. Define functions to perform each task in the app. For example, define the following functions to perform a specific task
    a. A function to greet the user with a welcome message
    b. A function display players of each team
    c. A function to randomly select a batsman and a bowler from the respective teams

before the inning starts

   d. A function to start the inning with current batsman and current bowler details when the inning starts
   e. A function to play cricket in each inning
   f. A function to display runs scored by batting team at the end of each inning
   g. A function to decide the winner and print the final message

4. Use escape characters such as ‘**\t**’, ‘**\n**’ wherever needed to keep program output clean
5. Use **usleep()** function as applicable along with user-friendly messages

Note: In the above given ‘Approach’ section, each task may require more than one function. It is totally up to you how you write code to perform a task. You can take help from the given code template to build the Gully Cricket Application.

Code Template

The following code template is just an illustration of what could be the possible structure of your program in order to build the app. This template will guide you through how the entire program can be organized.

<CODE>

```
// Documentation

// Include required header files and define namespaces as required
// Define a class Team

// Function declarations (function prototypes) as used in the program

// Global declaration of variables

int main() {

        // Call functions (as required) defined in the subprogram section.
        // Greet the user with a welcome message

        // Initialize both teams with the required data
        // Display player names of both the teams

        // Play inning one
        // Display score

        // Play inning two
        // Display score
        // Decide winner

 return 0;
}


// Note: The following function definitions are just an illustration of what are the possible
functions that can be created to perform tasks. You may or may not use the following function
prototypes. You can have as many functions as you require.

/* Greet users with a welcome message */
void welcomeUsers() {
        // Code….
 }

/* Display all the players in both team A and team B */
void displayPlayers(string playersA[3], string playersB[3]) {
        // Code…
        // Loop can be used to print player names in an array
}

/* Select randomly a batsman and bowler from the batting team and bowler team respectively
for a inning */
void selectBatsmanAndBowler(Team batTeam, Team bowlTeam) {
        // Code...
}

/* Start inning by displaying initial inning details */
void startInning(int inning, string batTeam, string bowlTeam) {
        // Code…
        // Display details such as who is the batting team or who is the batsman/bowler etc.
}

/* Play one inning (6 balls) */
int playInning() {
        // Code…
        // Use random function to generate run in each ball
         return runs; // Return the final run scored by the batting team
}
```

```
/* Display runs scored by batting team in the inning */
void displayScore(Team team) {
        // Code...
}

        /* Decide winner by comparing final scores of both the teams */
void decideWinner(int teamAScore, int teamBScore) {
        // code…
}
```

</CODE>

Submission

After completing the assignment, upload the **zip file** containing the source code file.

You will be able to download the ideal solution from the next Module Test solution screen. Compare your program with the provided solution and explore how a complex program can be broken down into smaller components by defining functions for each task.