Training Project (Virtual Cricket Game - “CRIC-IN”)

1. virtual-cricket-game.cpp

```
#include "game.h"
using namespace std;

int main() {

        Game game;
        game.welcome();

        cout << "\nPress Enter to continue";
        cin.get();

        game.showAllPlayers();

        cout << "\nPress Enter to continue";
        cin.get();

        game.selectPlayers();
        game.showTeamPlayers();

        cin.ignore(numeric_limits<streamsize>::max(),'\n');

        cout<<"\nPress Enter to Toss:";
        cin.get();

        game.toss();
```

```
    game.startFirstInnings();
    game.playInnings();

    cin.ignore(numeric_limits<streamsize>::max(),'\n');

    game.startSecondInnings();
    game.playInnings();

    cin.ignore(numeric_limits<streamsize>::max(),'\n');

    game.showMatchSummary();

    return 0;
    }
```

---

2. **player.h**

```
#include<string>

class Player {

    public:
        Player();
        std::string name;
        int id;
        int runScored;
        int ballsPlayed;
        int ballsBowled;
        int runsGiven;
        int wicketsTaken;
```

```
};
```

3. **player.cpp**

```
#include "player.h"        // < string >

Player::Player() {

        runsScored = 0;
        ballsPlayed = 0;
        ballsBowled = 0;
        runsGiven = 0;
        wicketsTaken = 0;
}
```

4. **team.h**

```
#include<vector>
#include "player.h"          // <string>


class Team {

    public:
         Team();
        std::string name;
        int totalRunsScored;
        int wicketsLost;
```

```
        int totalBallsBowled;
        std::vector<Player> players;

};
```

5. **team.cpp**

```
#include "team.h"          // "player.h" , <string> , <vector>


Team::Team() {
        totalRunsScored = 0;
        wicketsLost = 0;
        totalBallsBowled = 0;
}
```

6. **game.h**

```
#include<iostream>
#include<cstdlib>
#include<ctime>
#include<limits>
#include "team.h"       // "player.h" , <string> , <vector>

class Game {

   public:
        Game();
        int playersPerTeam;
        int maxBalls;
```

```
    int totalPlayers;
    std::string Players[11];

bool isFirstInnings;
Team teamA, teamB;
Team *battingTeam, *bowlingTeam;
Player *batsman, *bowler;

void welcome();
void showAllPlayers();
int takeIntegerInput();
void selectPlayers();
bool validateSelectedPlayer(int);
void showTeamPlayers();
void toss();
void tossChoice(Team);
void startFirstInnings();
void initializePlayers();
void playInnings();
void bat();
bool validateInningsScore();
void showGameScorecard();
void startSecondInnings();
void showMatchSummary();
};
```

7. **game.cpp**

```
#include "game.h"
using namespace std;
```

```
Game::Game() {

        playersPerTeam = 4;
        maxBalls = 6;
        totalPlayers = 11;

        players[0] = "Virat";
        players[1] = "Rohit";
        players[2] = "Dhawan";
        players[3] = "Pant";
        players[4] = "Karthik";
        players[5] = "KLRahul";
        players[6] = "Jadeja";
        players[7] = "Hardik";
        players[8] = "Bumrah";
        players[9] = "BKumar";
        players[10] = "Ishant";

        isFirstInnings = false;
        teamA.name = "Team-A";
        teamB.name = "Team-B";
}

void Game::welcome() {

        cout << "----------------------------------------" << endl;
        cout << "|================= CRIC-IN
================|" << endl;
        cout << "|                                      |" << endl;
        cout << "|    Welcome to Virtual Cricket Game     |" << endl;
        cout << "----------------------------------------" << endl;
```

```
        cout << endl << endl;
        cout << "-------------------------------------------------" <<
endl;
        cout << "|===================== Instructions
=====================|" << endl;
        cout << "-------------------------------------------------" <<
endl;
        cout << "|                                                 |" << endl;
        cout << "|  1. Create 2 teams ( Team-A and Team-B with 4
|" << endl;
        cout << "|     players each) from a given pool of 11 players.   |"
<< endl;
        cout << "|  2. Lead the toss and decide the choice of play.     |"
<< endl;
        cout << "|  3. Each innings will be of 6 balls.                 |" <<
endl;
        cout << "-------------------------------------------------" <<
endl;

}

void Game::showAllPlayers() {

        cout << endl;
        cout << "-----------------------------------------" << endl;
        cout << "|================= Pool of Players
===============|" << endl;
        cout << "-----------------------------------------" << endl;
        cout << endl;

        for (int i = 0; i < totalPlayers; i++) {
```

```
            cout << "\t\t[" << i << "] " << players[i] << endl;
        }
}

int  Game::takenIntegerInput() {

        int n;

        while(!(cin >> n)) {

                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                cout << "Invalid input! Please try again with valid
input: ";
        }

        return n;
}

bool Game::validateSelectedPlayer(int index) {

        int n;
        vector<Player> players;

        players = teamA.players;
        n = players.size();
        for(int i = 0; i < n; i++) {
                if (players[i].id == index) {
                        return false ;
                }
        }
```

```
        players = teamB.players;
        n = players.size();
        for (int i = 0; i < n; i++){
                if (players[i].id == index) {
                        return false ;
        }

}
        return true;
}

void Game::selectPlayers() {

        cout << endl;
        cout << "-----------------------------------------" << endl;
        cout << "|========== Create Team-A and
Team-B==========" << endl;
        cout << "-----------------------------------------" << endl;

        for (int i = 0; i < playersPerTeam; i++) {

                // Add player to team A
                teamASelection:
                    cout << endl << "Select Player" << i + 1 << " of
Team A -" ;
           int playerIndexTeamA = takeIntegerInput();

                if(playerIndexTeamA < 0 || playerIndexTeamA > 10) {
                        cout << endl << "Please select from the given
players" << endl;
                        goto teamASelection;
                }else if ( !validateSelectedPlayer(playerIndexTeamA)) {
```

```
                    cout << endl << "Player has been already selected.
Please select other player" << endl;
                    goto teamASelection;
                }else {
            Player teamAPlayer ;
                    teamAPlayer.id = playerIndexTeamA;
                    teamAPlayer.name = players[playerIndexTeamA];
                    teamA.players.push_back(teamAPlayer);
                }

                // Add player to team B
                teamBSelection:
                    cout << endl << "Select Player" << i + 1 << " of
Team B -";
            int playerIndexTeamB = takeIntegerInput();

            if(playerIndexTeamB < 0 || playerIndexTeamB > 10) {
            cout << endl << "Please select from the given players" <<
endl;
            goto teamBSelection;
        }else if(!validateSelectedPlayer(playerIndexTeamB)){
          cout<<endl<<"Player has been already selected.Please select
other player"<<endl;
          goto teamBSelection;
        }else{
             Player teamBPlayer ;
                    teamBPlayer.id = playerIndexTeamB;
                    teamBPlayer.name = players[playerIndexTeamB];
                    teamB.players.push_back(teamBPlayer);
        }
        }
}
```

```
void Game::showTeamPlayers() {

    vector<Player> teamAPlayers=teamA.players;
    vector<Player> teamBPlayers=teamB.players;

    cout<<endl<<endl;

cout<<"------------------------------\t\t--------------------------------"
<<endl;
    cout<<"|----------------TEAM A-------------\t\t-----------TEAM
B-------------|"<<endl;

cout<<"-----------------------------------\t\t-------------------------------"
<<endl;

    for(int i=0;i<playersPerTeam;i++){
    cout<<"|\t\t"<<"["<<i<<"] "<<teamAPlayers[i].name<<"\t\t
|"<<"\t\t"<<"|\t\t"<<"["<<i<<"] "<<teamBPlayers[i].name<<"\t\t
|"<<endl;
}
cout<<"-------------------------------\t\t---------------------------------
----"<<endl<<endl;
}

void Game::toss() {
    cout<<endl;
    cout<<"-----------------------------------------------"<<endl;
    cout<<"|-------------------------LET'S
TOSS----------------------|"<<endl;
    cout<<"-----------------------------------------------"<<endl<<endl;
    cout<<"TOSSING THE COIN....."<<endl<<endl;
```

```
    srand(time(NULL));
    int randomValue=rand()%2;
    switch(randomValue){
       case 0:
          cout<<"TEAM A WON THE TOSS!!" <<endl<<endl;
         tossChoice(teamA);
         break;
       case 1:
        cout<<"TEAM B WON THE TOSS!!"<<endl<<endl;
        tossChoice(teamB);
        break;
       }
}

void Game::tossChoice(Team tossWinnerTeam) {
        cout<<"ENTER 1 TO BAT OR 2 TO BOWL
FIRST:"<<endl<<"1. BAT "<<endl<<"2. BOWL"<<endl;

        int tossInput=takeIntegerInput();

        cin.ignore(numeric_limits<streamsize>::max(),'\n');

        switch(tossInput){
       case 1:
          cout<<endl<<tossWinnerTeam.name<<" won the toss and
chose to bat first."<<endl<<endl;
        if(tossWinnerTeam.name.compare("TeamA")==0){
                battingTeam=&teamA;
                bowlingTeam=&teamB;
        }else{
                battingTeam=&teamB;
```

```
                bowlingTeam=&teamA;
        }
          break;
        case 2:
         cout<<endl<<tossWinnerTeam.name<<"won the toss and
chose to bowl first."<<endl<<endl;
            if(tossWinnerTeam.name.compare("TeamA")==0){
              bowlingTeam=&teamA;
              battingTeam=&teamB;
            }else{
              bowlingTeam=&teamB;
              battingTeam=&teamA;
            }
            break;
            default:
           cout<<endl<<"Invalid input.Please try
again:"<<endl<<endl;
            tossChoice(tossWinnerTeam);
            break;
        }
    }

 void Game::startFirstInnings() {
         cout<<"\t\t|||FIRST INNINGS STARTED|||"<<endl<<endl;
                    isFirstInnings=true;
                    initializePlayers();
 }

void Game::initializePlayers(){
      //initializing *batsman and *bowler
         batsman=&battingTeam->players[0];
         bowler=&bowlingTeam->players[0];
```

```
            cout<<battingTeam->name<<" - "<<batsman->name<<" is
                                batting. "<<endl;
             cout<<bowlingTeam->name<<" - "<<bowler->name<<" is
                                bowling. "<<endl<<endl;
 }

void Game::playInnings(){

          for(int i=0; i<maxBalls; i++){
            cout<<"Press Enter to BOWL:";
            cin.get();
            cout<<"Bowling........"<<endl;
             bat();

             if(!validateInningsScore()){
                break;
       }
    }
}

void Game::bat(){
                srand(time(NULL));
                int runsScored=rand()%7;

 //Update batting team and batsman score
   batsman->runsScored=batsman->runsScored+runsScored;

battingTeam->totalRunsScored=battingTeam->totalRunsScored+runsSc
                                        ored;
                batsman->ballsPlayed=batsman->ballsPlayed+1;
```

```
            //Update bowling team and bowler score
          bowler->ballsBowled=bowler->ballsBowled+1;

bowlingTeam->totalBallsBowled=bowlingTeam->totalBallsBowled
+1;

bowler->runsGiven=bowler->runsGiven+runsScored;

          if(runsScored!=0){
            cout<<endl<<bowler->name<<" to
"<<batsman->name<<" "<<runsScored<<" runs!"<<endl<<endl;
               showGameScorecard();
           }else{
            cout<<endl<<bowler->name<<" to
"<<batsman->name<<"OUT!"<<endl<<endl;

            battingTeam->wicketsLost=battingTeam->wicketsLost+1;
            bowler->wicketsTaken=bowler->wicketsTaken+1;
            showGameScorecard();

            int nextPlayerIndex=battingTeam->wicketsLost;
            batsman=&battingTeam->players[nextPlayerIndex];
          }
    }

bool Game::validateInningsScore(){

     if(isFirstInnings){
       if(battingTeam->wicketsLost==playersPerTeam||
bowlingTeam->totalBallsBowled==maxBalls){
          cout<<"\t\t ||| FIRST INNINGS ENDS ||| "<<endl<<endl;
```

```
        cout<<battingTeam->name<<"
"<<battingTeam->totalRunsScored<<" - "<<
                battingTeam->wicketsLost<<"
("<<bowlingTeam->totalBallsBowled<<")"<<endl;
        cout<<bowlingTeam->name<<" needs
"<<battingTeam->totalRunsScored+1<<
                " runs to win the match."<<endl<<endl;

              return false;
            }
    }else{

if(battingTeam->totalRunsScored>bowlingTeam->totalRunsScored
){
         cout<<battingTeam->name<<" WON THE
MATCH"<<"\n\n";

            return false;
        }else
if(battingTeam->wicketsLost==playersPerTeam||bowlingTeam->tot
alBallsBowled==maxBalls){

if(battingTeam->totalRunsScored<bowlingTeam->totalRunsScored
){
     cout<<bowlingTeam->name<<" WON THE
MATCH"<<"\n\n";
         }else{
          cout<<"MATCH DRAW"<<"\n\n";
          }
             return false;
    }
 }
```

```
            return true;
    }

void Game::showGameScorecard(){

cout<<"-----------------------------------------------------------------------"
<<endl;
 cout<<"\t"<<battingTeam->name<<"
"<<battingTeam->totalRunsScored<<" - "<<
                        battingTeam->wicketsLost<<"
("<<bowlingTeam->totalBallsBowled<<")
""|"<<batsman->name<<" "<<batsman->runsScored<<"
("<<batsman->ballsPlayed<<") \t"<<bowler->name<<"
"<<bowler->ballsBowled<<"-"<<bowler->runsGiven<<"-"<<bowl
er->wicketsTaken<<"\t"<<endl;

cout<<"-----------------------------------------------------------------------"
<<endl<<endl;
}

 void Game::startSecondInnings(){
        cout<<"\t\t|||SECOND INNINGS STARTED|||"<<endl<<endl;

         isFirstInnings=false;

         Team  tempTeam=*battingTeam;
        *battingTeam=*bowlingTeam;
        *bowlingTeam=tempTeam;

        initializePlayers();
}
```

```
void Game :: showMatchSummary() {
    cout << "\t\t\t\t ||| MATCH ENDS ||| " << "\n\n";
    cout << battingTeam->name << " " <<
battingTeam->totalRunsScored << "-" <<
battingTeam->wicketsLost << " (" <<
bowlingTeam->totalBallsBowled << ")" << "\n";
cout << "\t\t\t\t==========================================="
<< "\n";
cout << "\t\t\t\t| PLAYER \t BATTING \t BOWLING |" << "\n";

    for (int j = 0; j < playersPerTeam; j++) {
        Player player = battingTeam->players[j];
    cout << "\t\t\t\t|-----------------------------------|" << "\n";
    cout << "\t\t\t\t| " << "[" << j << "] " << player.name << " \t ";
    cout << player.runsScored << "(" << player.ballsPlayed << ")
\t\t ";
    cout << player.ballsBowled << "-" << player.runsGiven <<
"-";
    cout << player.wicketsTaken << "\t |" << "\n";
}
    cout <<
"\t\t\t\t==========================================="
<< "\n\n";
    cout << bowlingTeam->name << " " <<
bowlingTeam->totalRunsScored << "-" <<
bowlingTeam->wicketsLost << " (" <<
battingTeam->totalBallsBowled << ")" < "\n";
    cout <<
"\t\t\t\t==========================================="
<< "\n";
    cout << "\t\t\t\t| PLAYER \t BATTING \t BOWLING |" <<
"\n";
```

```
        for (int i = 0; i < playersPerTeam; i++) {
              Player player = bowlingTeam->players[i];
             cout << "\t\t\t\t|-----------------------------------|" << "\n";
             cout << "\t\t\t\t| " << "[" << i << "] " << player.name << "
\t ";
             cout << player.runsScored << "(" << player.ballsPlayed
<< ") \t\t ";
             cout << player.ballsBowled << "-" << player.runsGiven <<
"-";
             cout << player.wicketsTaken << "\t |" << "\n";
 }
           cout <<
"\t\t\t\t=============================================="
<< "\n\n";

}
```