Development of C++ Application Program for League Equation using Object-Oriented Programming Technique (CAP-LEQ)

Gogo Tamuno-omie, Onokpegu Thompson

ABSTRACT---A research was carried out to determine the number of games to be played with a number teams in any sporting event and using a particular point system or value the maximum point to be earned by a team is also determined. The function was actualized using excel's spreadsheet application. This paper is purely an academic paper which focuses on the development of programs and applications using purely object-oriented approach. C++ is used as the programming language as this helps in the study of object-oriented programming with C++.

Key words: League, Points, Matches, Object-Oriented Programming, games

1 INTRODUCTION

This application is a tool to further enhance and add to the examples so far developed in the study of Object-Oriented Programming Techniques in the world of programming and most importantly in al citadels learning. Technology has driven into all sectors of the world's economy and so will not be out of place to improve and develop better skills in programming since programs are used in all spheres of life and in all disciplines and that is why it is also expedient and important to also consider the sporting world and develop a working application to manage the activities there in, for example, the football leagues across the world. A proper organization of sports could lead to a better society as sports contributes to the unity of people from different background. The learning of programming is also interesting when practical examples and problems are solved since some approaches seem to be abstract in the ears and eyes of students. This is a practical example which people see every day. This paper has bi-focal points of achievement; it explains briefly how OOP is studied and applied in the class and also coming up with a handy solution to manage a fresh league or updating an existing league when necessary.

2 ADVANTAGES OF THE APPLICATION (CAP-LEQ)

Immediate feedback, the ability to run on all windows platforms and it is also capable of reducing paper work in management

3 MOTIVATION

This paper can be a source of learning whenever the study of OOP is enacted in any citadel of learning

4 DEVELOPMENT OF THE APPLICATION (CAP-LEQ)

The CAP-LEQ package was written using C++ Object-Oriented Programming Technique. Other computer tools adopted were; Dev-C++ IDE and Windows 7 Operating System as the platform. The program was broken down into smaller aspect considered to be; the first module takes care of the input methods, the computation methods and the output methods by using the function as given as $Tn = 2[(n^2) - n]$. The second module is the control structures that enable the program run until a quit value is entered.

5 ANALYSIS

The equation is stated as $Tn = 2[(n^2) - n]$. Where;

Tn is the total number of matches to be played

n is the total number of teams that will participate in the league for a season

points is the adopted point to award a team that wins a match(draws not considered)

loose point to drop when a team loses a match or draws a match.

Matchplay a match played and a question is asked if a team wins, draws or loses.

Maxpoints is the total point earned by a team at the end of the season.

TeamRecord this displays the current status of a team (game played, wins, draws, loses and current points)

5.1 CLASS DIAGRAM

LeagueEquation
-Tn:int=0 -n:int=0 -points:int=0 -wins:int=0 -draws:int=0 -loses:int=0
+LeagueEquation(Tn:int,n:int,points:int) +setNumberOfMatches(Tn:int,n:int) +getNumberOfMatches():int +setdrawPoint(dp:int) +setMatchResults():int +setResults():int +setPoints(points:int, n:int) +getPoints():int +getIdividualMatches():int +getPoints():int +getNumberOfMatches():int +toString():string

5.2 CODING

```
/*
An Application written in C++ to solve the league
management problem
By: GOGO, Tamunoomie/Thompson Onokpegu
Date: 20/01/2016
School: RIVERS STATE POLYTECHNIC, BORI
*/
#include<iostream>
#include<string>
#include<iomanip>
using namespace std;
bool quit=false;
class LeagueEquation {
        private:
                /*keeps count of loses, draws and wins
                Their values can be set with either
setMatchResults()
                or setResults() functions*/
                int numberOfTeams;
                //Takes the number of teams that will
participate in the league
                int points;
                //Explains or takes the points as decided
by the management team
                int drawPoint;//represents the number of
points awarded in a draw
        public:
                int wins,draws,loses;
        //START
                LeagueEquation(){//default constructor
                        numberOfTeams = 0;
                        points = 0; //points is initialize to
the contemporary point system
                        drawPoint = 0;//drawPoint is
initialize to the contemporary point system
                        loses = 0;
                        draws = 0;
                        wins = 0;
                }
                void setPoints(int point){
                        points=point;
                }
                void setdrawPoint(int dp){
                        drawPoint=dp;
                }

                void setMatchResults(){
                        cout<<"Enter Number of Wins: ";
                        cin>>wins;
                        cout<<"Enter Number of Draws: ";
                        cin>>draws;
                        cout<<"Enter Number of Loses: ";
                        cin>>loses;
                }

                void setResults(){
                        bool quit = false;
                        int counter = 1;
                        char result;
                        cout<<"ENTER MATCH
RESULTS:Win (W)|Draw (D)| Lose (L)";
                        do{
                                cout<<"\nMatch "<<
counter <<" Result (W,D,L): ";
                                cin>>result;
                                if (result =='W'|| result
== 'w'){
                                        wins = wins + 1;
                        }else if (result =='D'|| result ==
'd'){
                                        draws = draws +
1;
                        }else if (result =='L'|| result == 'l'){
                                        loses = loses + 1;
                        }else if (result == '0'|| result ==
'o'||result == 'O'){
                                        quit = true;
                                }
                                counter +=1;
                }while(counter <=
getIdividualMatches());

        }

        int teamPoint(){
                int wpoints,dpoints;
                wpoints = wins * points;
    //calculate points obtained from all wins
                dpoints = draws * drawPoint;
    //calculate points obtained from all draws
                return (wpoints + dpoints);
    //return the sum of all points obtained from wins
and draws
        }

        int gamesPlayed(){//Calculates number of
games played
                int numberOfGames;
                numberOfGames = wins + draws
+ loses;
                return numberOfGames;
        }

        //END

        void setNumberOfMatches(int n){
                numberOfTeams=n;
        }
        int getNumberOfMatches(){
        double
x=2*((numberOfTeams*numberOfTeams)-
numberOfTeams);
                x = x/2;
                return x;
        }
        void setPoints(int p, int t){
                numberOfTeams=t;
                points=p;
        }
        double getPoints(){
                double
x=2*((numberOfTeams*numberOfTeams)-
numberOfTeams);
                double v=x/numberOfTeams;
                return v*points;
        }
        double getIdividualMatches(){
                return 2*((numberOfTeams-1));
        }
        string toString(){
                cout<<this-
>numberOfTeams<<this->points<<endl;
```

```
                }
};
//custom design data type: team
struct teamRecord{
        string name;
        int matches;
        int wins;
        int draws;
        int loses;
        int points;
};

void manageLeague(){

        char response;
        int points,drawPoint,teams;

        cout<<"\n================================"<<e
ndl;
                cout<<"TO QUIT, ENTER 0
PLEASE"<<endl;

        cout<<"================================"<<endl
;
                cout<<"\n";
                cout<<"ENTER THE NUMBER OF
TEAMS: ";
                cin>>teams;
                if(teams==0 || teams==1){
                        cout <<"TEAMS CANNOT BE
ONE(1) OR DO YOU WANT TO QUIT(Y/N)";
                cin>>response;
        if(response=='Y'||response=='y'){
                        quit=true;
                        cout<<"THANK YOU FOR
USING THE SYSTEM.BYE!";
                        exit(0);
                         }else{
                                        quit=false;
                                        cout<<"ENTER
THE NUMBER OF TEAMS: ";
                                        cin>>teams;
                         }

                }
                        cout<<"ENTER THE POINT
SYSTEM: ";
                  cin>>points;
                  cout<<"ENTER THE DRAW POINT: ";
                  cin>>drawPoint;

                        //START NEW
```

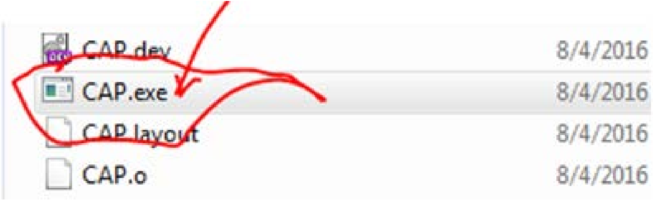
```
                    /*Creates an Array of
LeagueEquation Object (leq[teams])-
                    the number of teams represents
the size of the array*/
                    LeagueEquation leq[teams];
                    /*Creates an Array of the
structure teamRecord -
                    number of teams represents the
size of the array*/
                    teamRecord newTeam[teams];

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

          leq[i].setPoints(points,teams);
                              cout<<"ENTER NAME
OF TEAM: ";
                              cin>>newTeam[i].name;
                               //if(newTeam[i].name)
                              leq[i].setResults();
                              newTeam[i].matches =
leq[i].gamesPlayed();
                              newTeam[i].wins =
leq[i].wins;

                              newTeam[i].draws =
leq[i].draws;

                              newTeam[i].loses =
leq[i].loses;

                              newTeam[i].points =
leq[i].teamPoint();

                    }


          cout<<"===============================\nLEA
GUE TABLE\n==============================="<<endl;

          cout<<"Team"<<setw(8)<<"M"<<setw(5)<<"W"<<set
w(5)<<"D"<<setw(5)<<"L"<<setw(5)<<"P"<<endl;
                    cout<<"--------------------------------
\n";
                    for(int j=0;j<teams;j++){

          cout<<newTeam[j].name<<setw(8)<<newTeam[j].m
atches <<setw(5)<<newTeam[j].wins;

          cout<<setw(5)<<newTeam[j].draws<<setw(5)<<new
Team[j].loses<<setw(5)<<newTeam[j].points<<endl;
                    }
}
void analyseLeague(){//to decide and understand the
implication of adding or removing a team to or from
existing league.
int points, teams;
```

```
          char response;
          cout<<"ENTER THE NUMBER OF TEAMS
"<<endl;
          cin>>teams;
          if(teams==0 || teams==1){
   cout<<"TEAMS CANNOT BE 0 OR 1. PLEASE ENTER A
VALID NUMBER OF     TEAMS FROM 2"<<endl;

                                        }else{
                                        quit=false;
                                        cout<<"ENTER THE
POINT SYSTEM "<<endl;
          cin>>points;

          LeagueEquation leq;

          leq.setNumberOfMatches(teams);
          int numberofmat=leq.getNumberOfMatches();
          leq.setPoints(points, teams);
          double ind=leq.getIdividualMatches();
          double numberofpoints=leq.getPoints();


          cout<<"================================="<<endl
;
                         cout<<"MATCHES PER SEASON
/-"<<numberofmat<<"-/"<<endl;

          cout<<"================================="<<endl
;
                         cout<<""<<endl;

          cout<<"================================="<<endl
;
                         cout<<"MATCHES BY
INDIVIDUAL TEAM /-"<<ind<<"-/"<<endl;

          cout<<"================================="<<endl
;
                         cout<<""<<endl;

          cout<<"================================="<<endl
;
                         cout<<"MAXIMUM POINTS
OBTAINABLE/-"<<numberofpoints<<"-/"<<endl;

          cout<<"================================="<<endl
;
                    cout<<"\n";
                              }
```

```
        }
//menu function which is a void function
void menu(void){
                cout<<"|-----------------*MENU*---------------
|\n";
                cout<<"
=====================================\n";
                cout<<"|ENTER(1) TO DECIDE
MATCHES AND POINTS.|\n";
                cout<<"|ENTER(2) TO RECORD MATCH
OUTCOMES    |\n";
                cout<<"|ENTER(0) TO EXIT
|\n";
                cout<<"
=====================================\n";
        }
int main(){
        do{
           menu();
                char choice;
                char response;
                cout<<"ENTER ANY OF THE NUMBERS
TO START"<<endl;
                cin>>choice;

           switch(choice){
              case '1' :
                          analyseLeague();
                          cout << "DO YOU WANT TO
QUIT(Y/N)" << endl;
                   cin>>response;

                        if(response=='Y'||response=='y'){

                                quit=true;
                                cout<<"THANK YOU
FOR USING THE SYSTEM.BYE!";
                                exit(0);
                        }else{
                                        quit=false;
                                        //DoAnother();
                                 }
                          break;
                                case '2' :
                         manageLeague();
                         cout << "DO YOU WANT TO
QUIT(Y/N)" << endl;
                         cin>>response;
                  if(response=='Y'||response=='y')
                          {
                        quit=true;
                        cout<<"THANK YOU FOR
USING THE SYSTEM.BYE!";
                        exit(0);
                         }
                         else
                                {
                                quit=false;
                                //doSomething();
                                }
                   break;

         case '3':
                   cout << "DO YOU WANT TO
QUIT(Y/N)" << endl;
          cin>>response;

          if(response=='Y'||response=='y')
                          {
              quit=true;
                         cout<<"THANK YOU FOR
USING THE SYSTEM.BYE!";
                         exit(0);
                }

                else{
                     quit=false;
                }
                   break;
                   case '0' :
                          cout << "DO YOU WANT TO
QUIT(Y/N)" << endl;
                   cin>>response;

                  if(response=='Y'||response=='y')
                  quit=true;
                          else
                          quit=false;

          break;

                    default :
             cout << "ENTER A VALID CHOICE" << endl;

          break;
     }

    }while(!quit);

                        //END NEW
        return 0;
}
```

6 TESTING/RESULTS

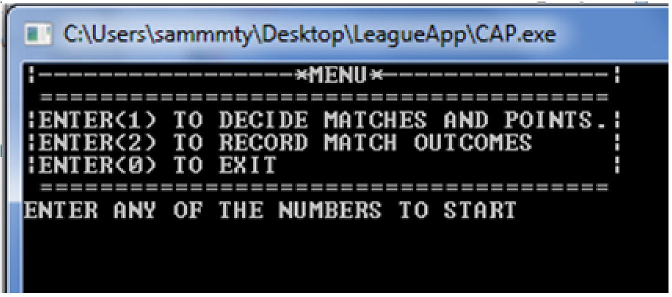
Running the program



```
CAP.dev      8/4/2016
CAP.exe      8/4/2016
CAP.layout   8/4/2016
CAP.o        8/4/2016
```

Figure: Executable file icon to run the program

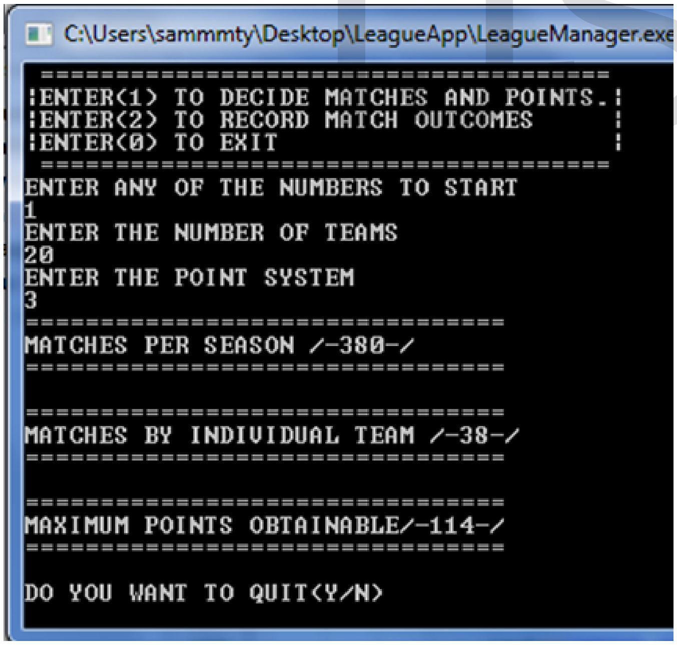
Below is the first menu screen that helps you to do what you want to



```
|-----------------*MENU*---------------|
 ======================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 ======================================
ENTER ANY OF THE NUMBERS TO START
```

Figure: Initial menu to select what to do

Initial output screen that prompts for user input to be able to know how many teams use and the points so as to plan and draw the time table for match days.

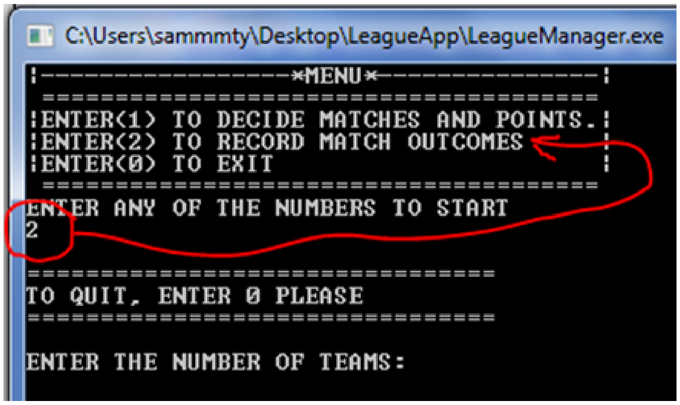


```
 ======================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 ======================================
ENTER ANY OF THE NUMBERS TO START
1
ENTER THE NUMBER OF TEAMS
20
ENTER THE POINT SYSTEM
3
================================
MATCHES PER SEASON /-380-/
================================

================================
MATCHES BY INDIVIDUAL TEAM /-38-/
================================

================================
MAXIMUM POINTS OBTAINABLE/-114-/
================================

DO YOU WANT TO QUIT(Y/N)
```

Figure: Analyzing the size of the league for decision making

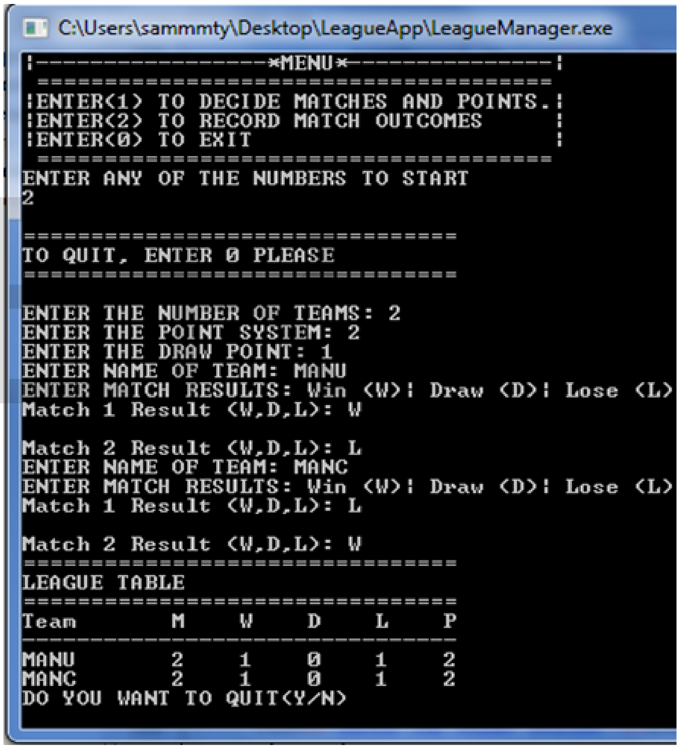
Final output screen displaying the two different inputs from the user and also the corresponding outputs after the operation is complete.

Output screen to enter match outcomes that shows the table of performance at the end



```
|-----------------*MENU*---------------|
 ======================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 ======================================
ENTER ANY OF THE NUMBERS TO START
2

================================
TO QUIT, ENTER 0 PLEASE
================================

ENTER THE NUMBER OF TEAMS:
```

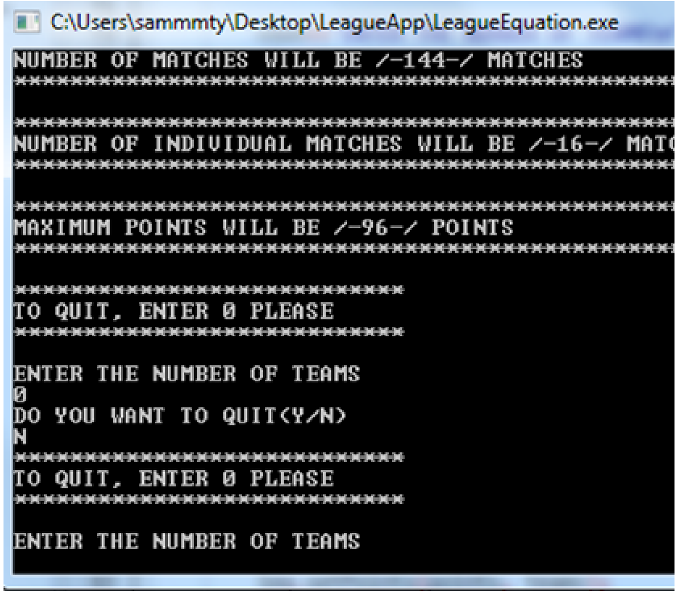
Figure: Capturing the different matches



```
|-----------------*MENU*---------------|
 ======================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 ======================================
ENTER ANY OF THE NUMBERS TO START
2

================================
TO QUIT, ENTER 0 PLEASE
================================

ENTER THE NUMBER OF TEAMS: 2
ENTER THE POINT SYSTEM: 2
ENTER THE DRAW POINT: 1
ENTER NAME OF TEAM: MANU
ENTER MATCH RESULTS: Win (W)| Draw (D)| Lose (L)
Match 1 Result (W,D,L): W

Match 2 Result (W,D,L): L
ENTER NAME OF TEAM: MANC
ENTER MATCH RESULTS: Win (W)| Draw (D)| Lose (L)
Match 1 Result (W,D,L): L

Match 2 Result (W,D,L): W
================================
LEAGUE TABLE
================================
Team         M     W     D     L     P
----------------------------------------
MANU         2     1     0     1     2
MANC         2     1     0     1     2
DO YOU WANT TO QUIT(Y/N)
```

Figure: Showing table standing

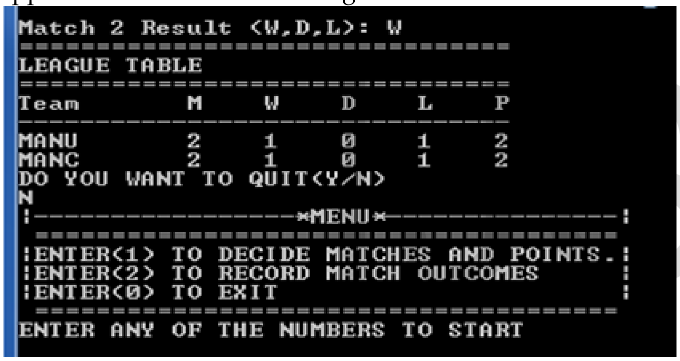
When user tries to enter the sentinel value 0 to quit



```
C:\Users\sammmty\Desktop\LeagueApp\LeagueEquation.exe
NUMBER OF MATCHES WILL BE /-144-/ MATCHES
*********************************************

*********************************************
NUMBER OF INDIVIDUAL MATCHES WILL BE /-16-/ MAT
*********************************************

*********************************************
MAXIMUM POINTS WILL BE /-96-/ POINTS
*********************************************

***************************
TO QUIT, ENTER 0 PLEASE
***************************

ENTER THE NUMBER OF TEAMS
0
DO YOU WANT TO QUIT(Y/N)
N
***************************
TO QUIT, ENTER 0 PLEASE
***************************

ENTER THE NUMBER OF TEAMS
```

Figure: System rejects and corrects wrong entry

When a decision has to be taken either to quit the application or to continue using it

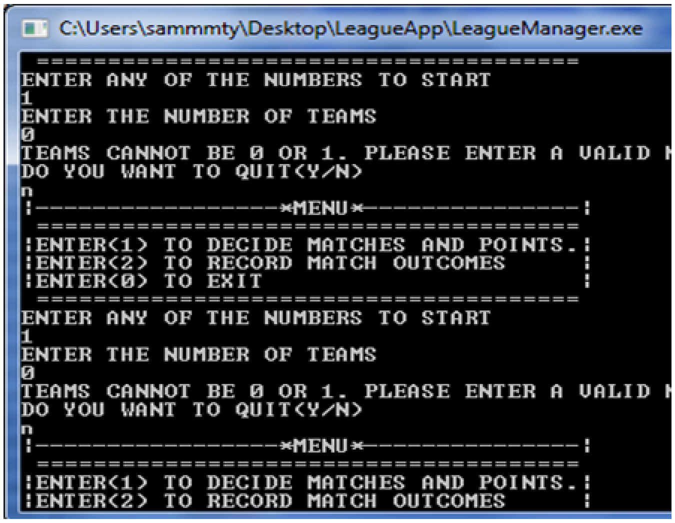


```
Match 2 Result (W,D,L): W
==================================
LEAGUE TABLE
==================================
Team         M     W     D     L     P
----------------------------------
MANU         2     1     0     1     2
MANC         2     1     0     1     2
DO YOU WANT TO QUIT(Y/N)
N
|-----------------*MENU*---------------|
 =====================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 =====================================
ENTER ANY OF THE NUMBERS TO START
```

To ensure that number of teams is not 0 or 1,

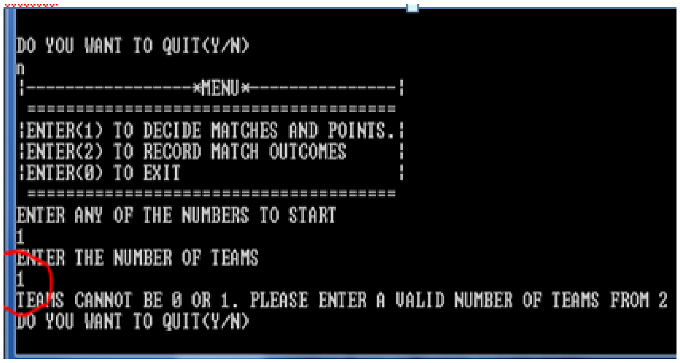
The output shows when a 0 is entered and the output becomes

"Teams cannot be 0 or 1…"



```
C:\Users\sammmty\Desktop\LeagueApp\LeagueManager.exe
 =====================================
ENTER ANY OF THE NUMBERS TO START
1
ENTER THE NUMBER OF TEAMS
0
TEAMS CANNOT BE 0 OR 1. PLEASE ENTER A VALID N
DO YOU WANT TO QUIT(Y/N)
n
|-----------------*MENU*---------------|
 =====================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 =====================================
ENTER ANY OF THE NUMBERS TO START
1
ENTER THE NUMBER OF TEAMS
0
TEAMS CANNOT BE 0 OR 1. PLEASE ENTER A VALID N
DO YOU WANT TO QUIT(Y/N)
n
|-----------------*MENU*---------------|
 =====================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
```

When a 1 is entered as the number of teams, the output is "Teams cannot be 0 or 1…"



```
DO YOU WANT TO QUIT(Y/N)
n
|-----------------*MENU*---------------|
 =====================================
|ENTER(1) TO DECIDE MATCHES AND POINTS.|
|ENTER(2) TO RECORD MATCH OUTCOMES     |
|ENTER(0) TO EXIT                      |
 =====================================
ENTER ANY OF THE NUMBERS TO START
1
ENTER THE NUMBER OF TEAMS
1
TEAMS CANNOT BE 0 OR 1. PLEASE ENTER A VALID NUMBER OF TEAMS FROM 2
DO YOU WANT TO QUIT(Y/N)
```

Figure: System sends error message when 1 team is entered

……………………………………………………………………

Gogo Tamuno-omie is presently a lecturer in the Computer Science Department of Rivers State Polytechnic, Bori, Nigeria and heading towards a Doctorate Degree (sammmty@yahoo.com)

Thompson Onokpegu is currently pursuing his HND in computer Science in the same school

7 CONCLUSION

In order to implement the use of any type of instructional software, instructional technology and educational technology, teachers must feel confident in its operation and their own ability to integrate it into daily classroom practices (Bandele & Adekunle, 2015). The use of computer technology enhance teacher to student interaction, student to student interaction as well as it helps student to become more independent. Computer technology, when properly used, can enhance learning and has the potential to positively influence students' success rates (Carter, 2004). On the basis of the major findings of this study, it was conducted that the use of CAP-LEQ package is more positive and effective in improving the management leagues of any sporting event or games in any part of the world. Also, encourage or motivate students to practices more than they thought and improve the proficiency in computer use which will be valuable later in life.

8 REFERENCES

[1] Development of C++ Application Program for Solving Quadratic Equation in Elementary School in Nigeria Journal of Education and Practice ISSN 2222-1735 (Paper) ISSN 2222-288X (Online) Vol.6, No.28, 2015 accessed(June, 2015)

[2] Eunsuk K & JeanY () 6.088 Intro to C/C++ Day 4: Object-oriented programming in C++ http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-088-introduction-to-c-memory-management-and-c-object-oriented-programming-january-iap-2010/lecture-notes/MIT6_088IAP10_lec04.pdf accessed(June, 2015)

[3] http://www.cprogramming.com/tutorial/references.html

[4] Gogo, T(2015) Optimizing League Management Using Square Matrix Relation, International Journal of Current Research, vol7,Issue 12, pp23592-23595, December, 2015 http://www.journalcra.com/article/optimizing-league-management-using-square-matrix-relation accessed(June, 2015)

[5] Gillius's Programming, Learning Object Oriented Programming in C++ accessed(June, 2015)
[]Herbert .S (1998)C++ ***The complete reference*** Third Edition

http://www.uml.org.cn/c++/pdf/C%2B%2BComplete%20Reference%20(3rd%20Ed.).pdf
accessed(June, 2015)

[6] Object-Oriented Analysis and Design with Applications (G. Booch, et. al) 54

[7] C++ Programming Language Object-Oriented Programming (OOP) in C++ https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp3_OOP.html accessed(June, 2015)