KLE Society's

KLE Technological University



**Open Ended Activity Report**

**On**

**Cricket Tournament**

**Object Oriented Programming with C++ (20ECSC204) Object Oriented Programming with C++ Lab (20ECSP203)**

Submitted by

**Name Roll no SRN**

Abbasali Jamadar 466 01FE20BCS414

Neha Kardant 469 01FE20BCS422

Nihal Fernandis 437 01FE19BCS237

**Team Number: 4D18**

Faculty In-charge: Somashekhar Patil

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBLI – 580 031 (India).

Academic year 2020-21

**1. Introduction**

1.1 Overview of the Problem Statement

1.2 Features of Application

1.2.1 Auction of players

1.2.2 Matches between teams

1.2.3 Winner of the match

**2. Design**

2.1 Class Diagram

2.2 Description of Each Class

2.3 Main Function

2.4 Use of Standard Design Patterns

**3. Unit Test Plan**

3.1 Auction: void startauction(string players[],int mb[],string teams[],string role[])

3.2 Match: match(int choice)

3.3 Winner: winner( )

**4 Implementation**

4.1 Results

**1. Introduction**

**1.1 Overview of Problem Statement**

Cricket is immensely popular team game and played on a field that has 20-meters pitch and its center with one wicket of three stumps on each end. It is played between two teams of eleven players each. The objective is to score more run(points) than the opposing team. A match is divided into innings during which one team bats, two batsmen at time, and other team bowls. Here in this project we are simulating the cricket tournament, which involves auction of players(which will have players details), setting team players, organizing the matches(normal and final matches) between teams and at the end gives winner list and also declares final winner in the tournament.

**1.2 Features of Application**

**1.2.1 Auction of players**

A player can be selected for particular team by providing team no and bid amount. Then the provided team no and bid amount is checked and if eligible then the player is fixed to selected team.

**1.2.2 Matches between teams**

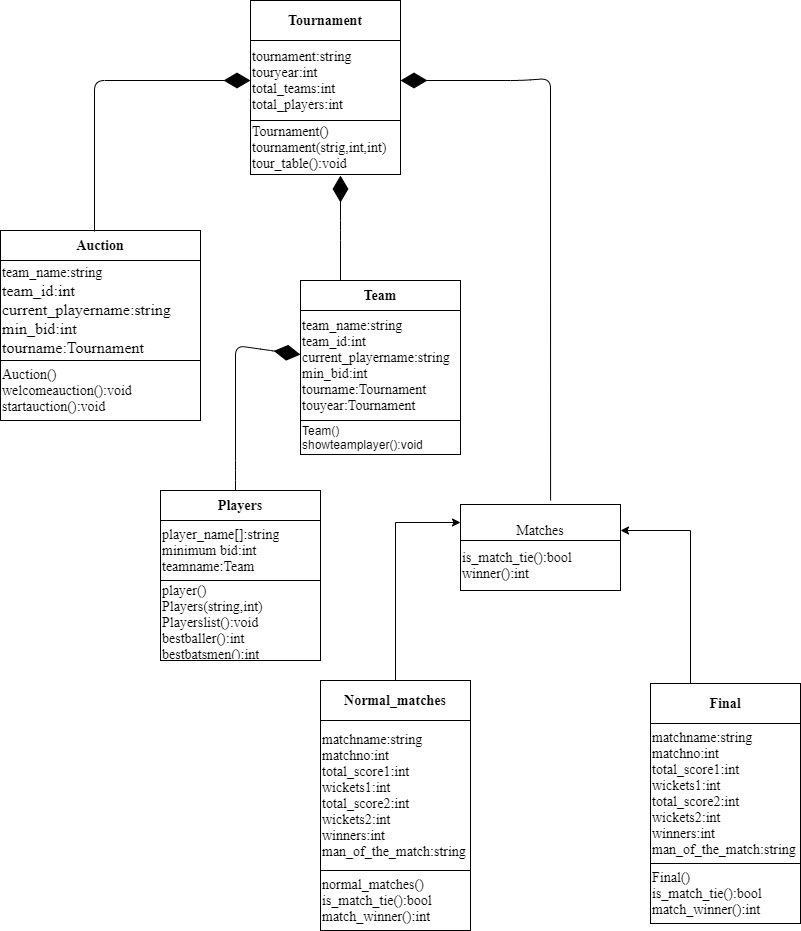
This feature the user can perform after players get selected for particular team then it comes turn of match where we should select between normal and final match with teams to play then the match starts.

**1.2.3 Winner of the match**

This feature gives us the final result that is it displays the winner teams name with match number.

**2. Design**

**2.1 Class Diagram**



**2.2 Description of Each Class**

**1. Tournament**

**Tournament**

+tournament:string

+touryear:int

+total\_teams:int

+total\_players:int

+Tournament( )

+tournament(string,int,int)

+tour\_table( ):void

* Tournament class : this class includes the information about tournament has name,year,total teams ,and total players
* The classes like auction and team from it.

**2. Auction**

**Auction**

+team\_name:string

+team\_id:int

+current\_playername:string

+min\_bid:int

+tournament:Tournament

+Auction( )

+welcomeauction( ):void

+startauction( ):void

* Auction class is to give the information about the bid and the team who bid for which player.
* It includes team name,team id,current player,and minimum bid

**3. Team**

**Team**

+team\_name:string

+team\_id:int

+current\_playername:string

+min\_bid:int

+tournament:Tournament

+touryear:Tournament

+Team( )

+showteamplayer( ):void

* Team class : this class has player and give information about the player present in each team and team name , team id, currentplayer
* This class also includes minimum bid

**4. Matches**

**Matches**

+is\_match\_tie( ):bool

+winner( ):int

* This is an Abstract class where we have a is\_match\_tie( ) function and winner( ) function.
* It creates the type of match as per demands of the user.
* It has 2 sub-classes: normal match and final match.

**5. Normal matches and final**

**Normal\_matches**

+matchname:string

+matchno:int

+total\_score1:int

+wickets1:int

+total\_score2:int

+wickets2:int

+winners:int1

+man\_of\_the\_match:string

+normal\_matches( )

+is\_match\_tie( ):bool

+match\_winner( ):int

**Final**

+matchname:string

+matchno:int

+total\_score1:int

+wickets1:int

+total\_score2:int

+wickets2:int

+winners:int1

+man\_of\_the\_match:string

+Final( )

+is\_match\_tie( ):bool

+match\_winner( ):int

* The above classes are used to create matches using factorydesign pattern.
* Matches is a interface between normal match ,final match and match class

**6. Players**

**Players**

+player\_name( ):string

+minimum bid:int

+teamname:Team

+player( )

+Players(string,int)

+Playerlist( ):void

+bestballer( ):int

+bestbatsmen( ):int

* Player class : player class gives information about the players with their name and bid amount

**2.3 Main Function**

In main function,

There is a switch menu for different options like displaying players list, auction process, displaying teams, match fixing, matches, displaying each match winners and finally displaying final match winner.

According to the user input we will be doing auction of players then the team will be formed. Once team is formed we can start the match where again there we will be selecting matches (normal match and final match). If matches get tie then there will be super overs and again if there is tie then each team will be give 1 point. Winner of the match displays winners of each match played and finally, at last final winner of the tournament will also be displayed.

**2.4 Use of Standard Design Patterns**

Factory design pattern is used for the above application.

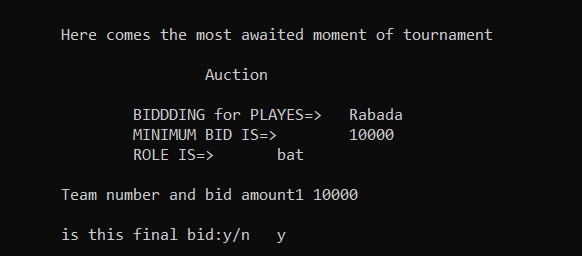
* **Factory method pattern(creational pattern):**
* **Definition:- The Factory Method pattern is a design pattern used to define a runtime interface for creating an object. It’s called a factory because it creates various types of objects without necessarily knowing what kind of object it creates or how to create it.**
* **Usage:- Factory method is suitable for this scenario because matches are to be created as per user demand, so to create the objects of the required matches type during the run time interface it becomes easier.**

**3. Unit Test plan**

**3.1 Auction: void startauction(string players[],int mb[],string teams[],string role[])**

The auction process involves giving team players details like name of the player, minimum bid amount and role of the player and then user should enter team no and bid amount then it checks if the team no entered and bid amount entered is eligible then finally if both the cases are passed then the player will be fixed to selected team.

Given below is how auction of players takes place: First team no and bid amount is stored and checked. Then if both case passed then ask if it is final bid.



**Test case 1: Passing teamno and bid amount**

**Input:**

Here comes the most awaited moment of tournament

Auction

BIDDDING for PLAYES=> Rabada

MINIMUM BID IS=> 10000

ROLE IS=> bat

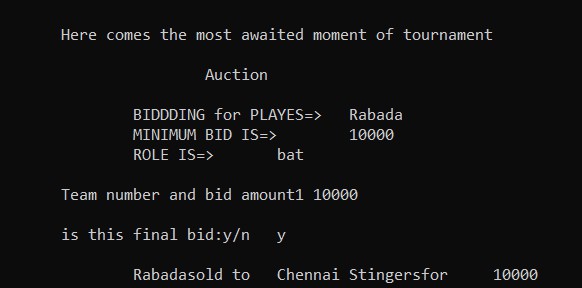
Team number and bid amount1 10000 is this final bid:y/n y

Rabada sold to Chennai Stingers for 10000

**Expected output:**

Rabada sold to Chennai Stingers for 10000

**Actual output:**



**Test case 2: Failing teamno and bid amount**

**Input:**

Here comes the most awaited moment of tournament

Auction

BIDDDING for PLAYES=> Rabada

MINIMUM BID IS=> 10000

ROLE IS=> bat

Team number and bid amount1 10000 is this final bid:y/n y

Rabada sold to Chennai Stingers for 10000

**Expected output:**

!!!!!!!!!! EXCEPTION: WRONG TEAM CALL

0->DELHI

1->CHENNAI

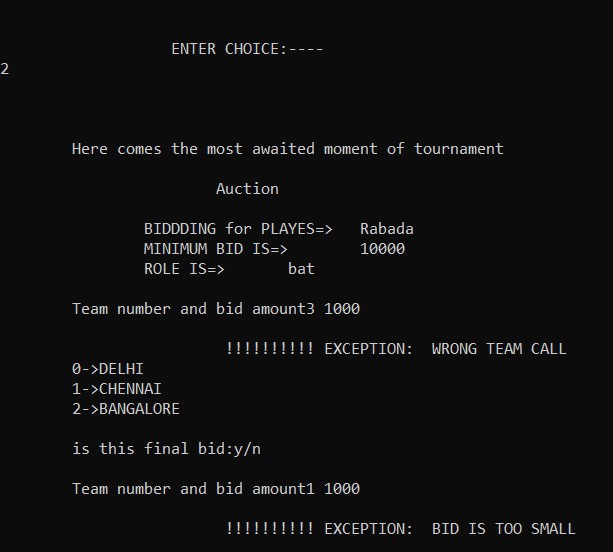
2->BANGALORE

is this final bid:y/n

Team number and bid amount1 1000

!!!!!!!!!! EXCEPTION: BID IS TOO SMAL

**Actual Output:**



**3.2 Match: match(int choice)**

After team is formed then comes matches which is of two types normal match and final match. So here user should select normal or final match.by entering team no Then the match starts which firstly ask for match no then it displays among which teams the match will be conducted then user will be entering score and wickets of both team and the innings are displayed and then finally it announces winner of the match and if match tie then super over will be conducted and if again there is tie then

1 point will be given to each team and final winner will be declared.

Given below is how match will take place:

First match no and match details(score, wicket)stored.

Then it gives the match winners.



**Test case 3: Normal matches:**

**Input:**

5

Normal Match(1) Final Match(2)

Go(0): 1

0

NORMAL MATCH STARTED Enter Match No: 1

Delhi Warriors VS Chennai Stingers

Enter score of : Delhi Warriors230

Enter wickets of : Delhi Warriors3

First Inning Results :-> 230-3

Enter score of : Chennai Stingers240

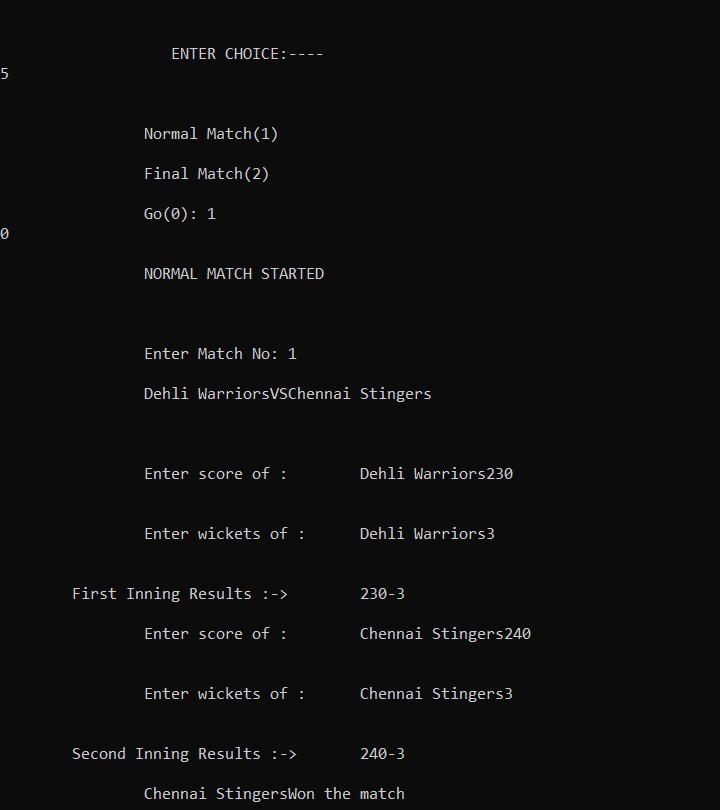
Enter wickets of : Chennai Stingers3

Second Inning Results :-> 240-3

**Expected Output:**

Chennai Stingers Won the match

**Actual Output:**



**Test case 4: During super overs:**

Super Over

Enter score of : Delhi Warriors12

Enter wickets of : Delhi Warriors1

First Inning Results :-> 12-1

Enter score of : Chennai Stingers12

Enter wickets of : Chennai Stingers0

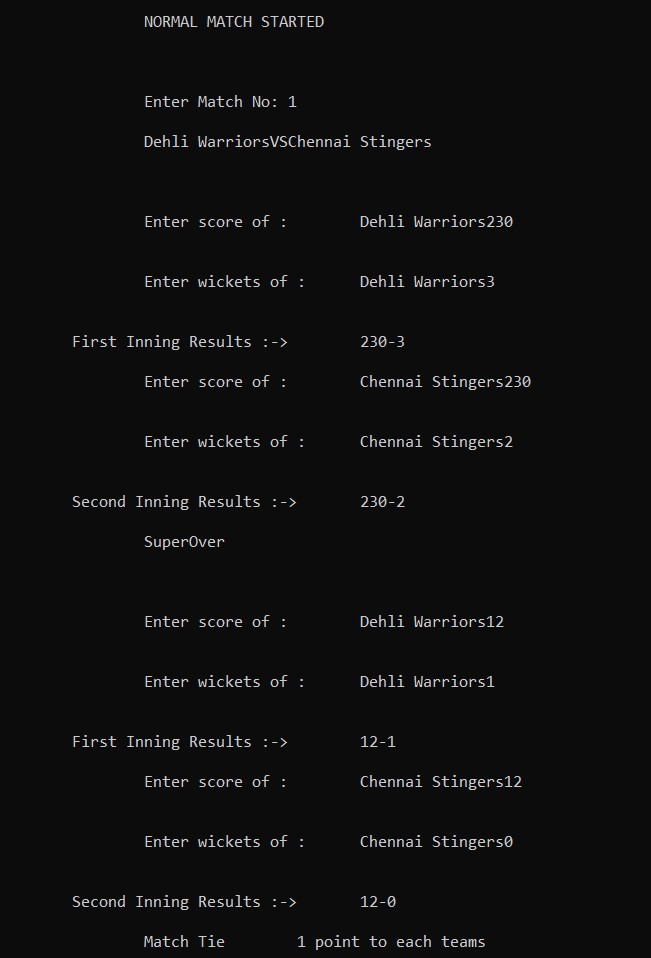
Second Inning Results :-> 12-0



**Expected Output:**

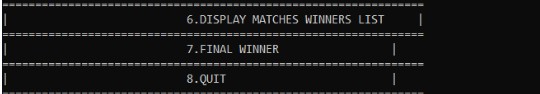
If super over also gets tie then output will be: Match Tie 1 point to each teams

**Actual Output:**



**3.3 Winner: winner( )**

Displays match winner and final winner according to choice selected



**Test case 5: Match Winner**

**Input:**

6

7

WINNERS OF TOURNAMENT MATCHES

**Expected Output:**

Match 1 > Chennai Stingers Match 1 > Chennai Stingers Match 2 > Bangalore knights Match 2 > Bangalore knights Match 3 > Delhi Warriors Match 3 > Delhi Warriors Match 3 > Delhi Warriors

**Actual Output:**



**Test case 6: Final Winner**

**Input:**

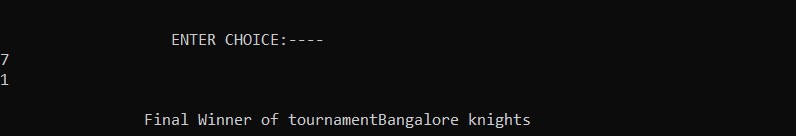
**7**

1

**Expected Output:**

Final winner is Bangalore knights

**Actual Output:**



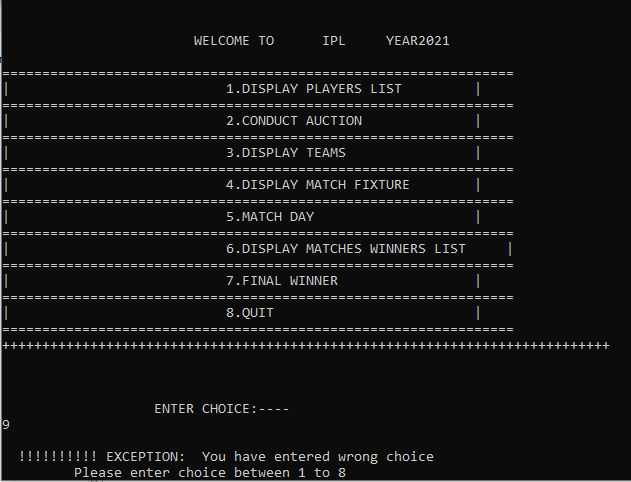
**4. Implementation**

**4.1 Results**

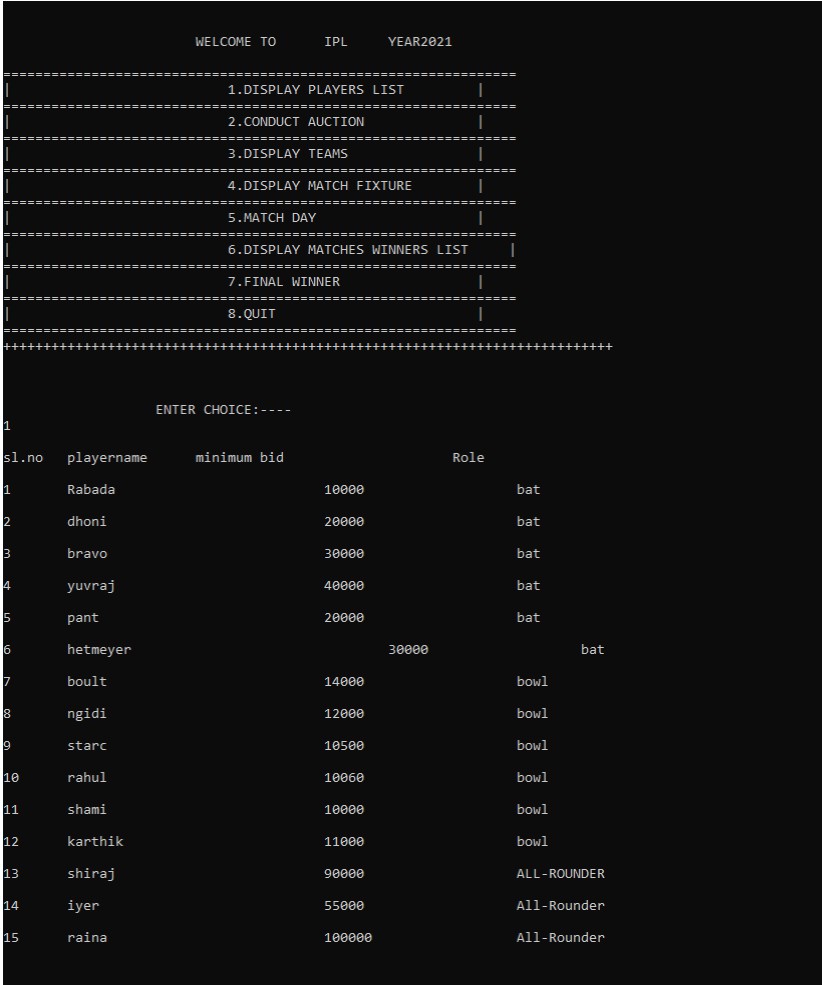
**1. Main Menu**



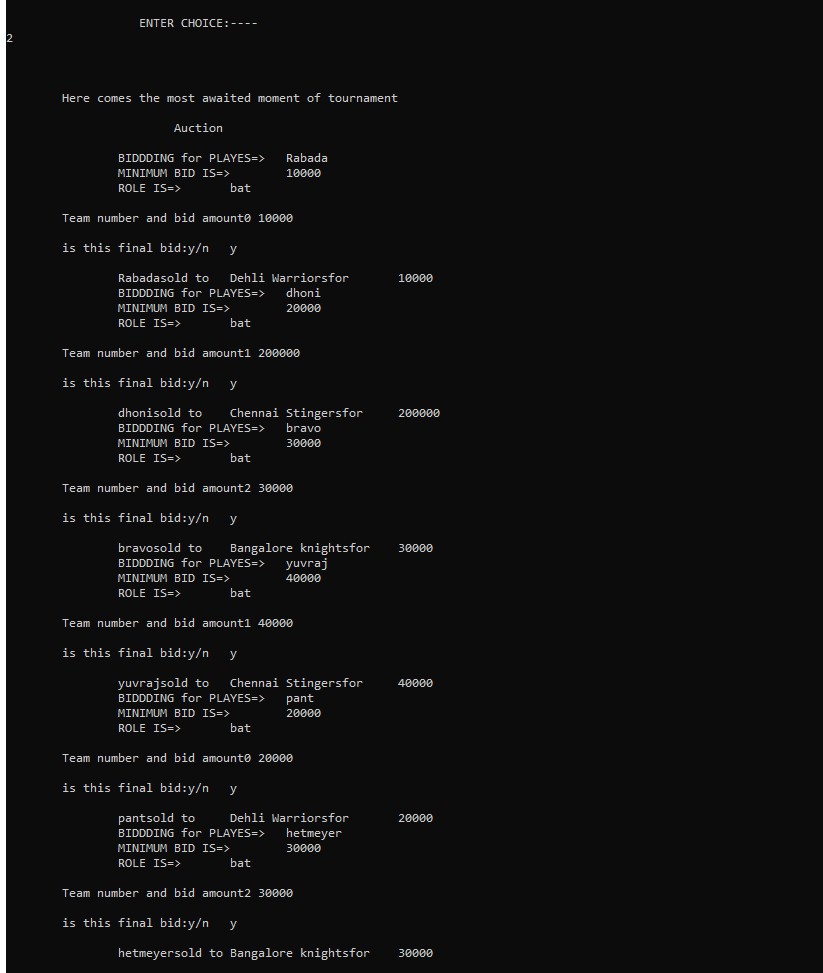
**1. Exception handling in main menu:**



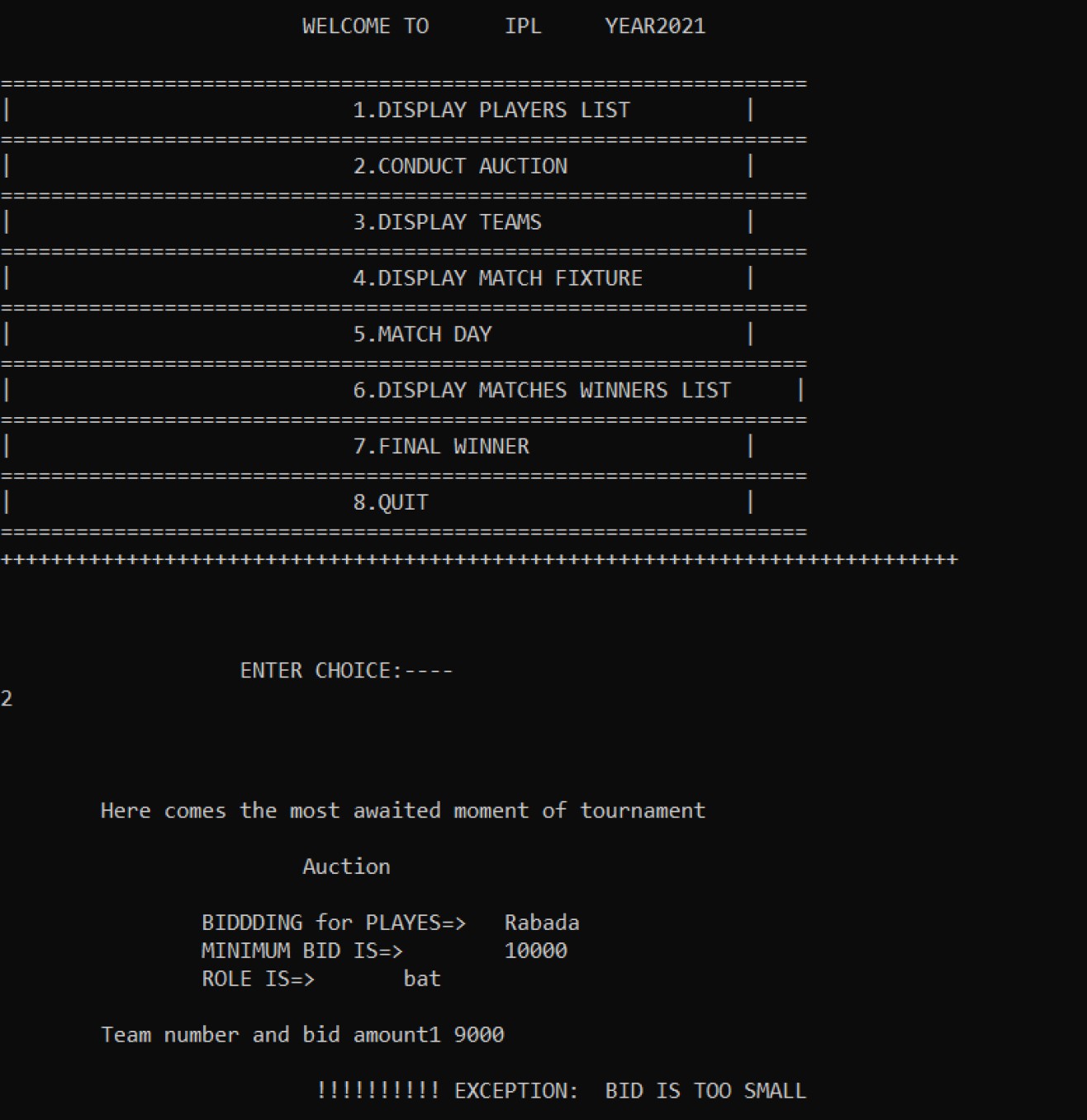
**2. Display players list**



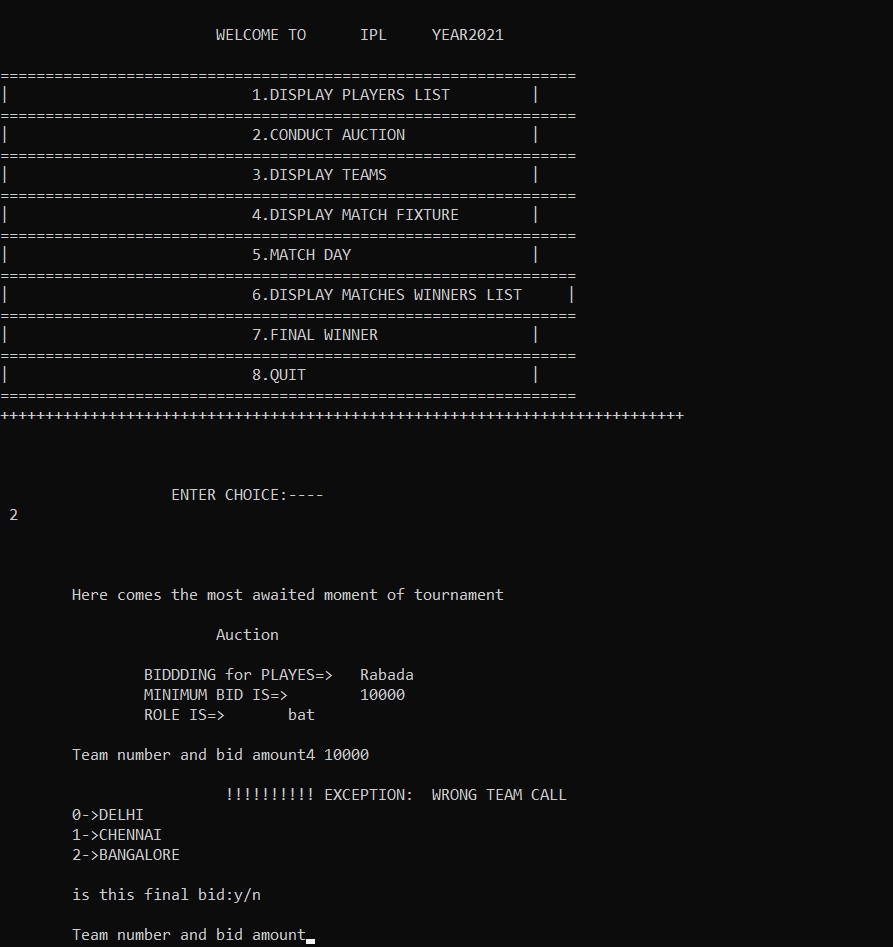
**3. Auction of players**



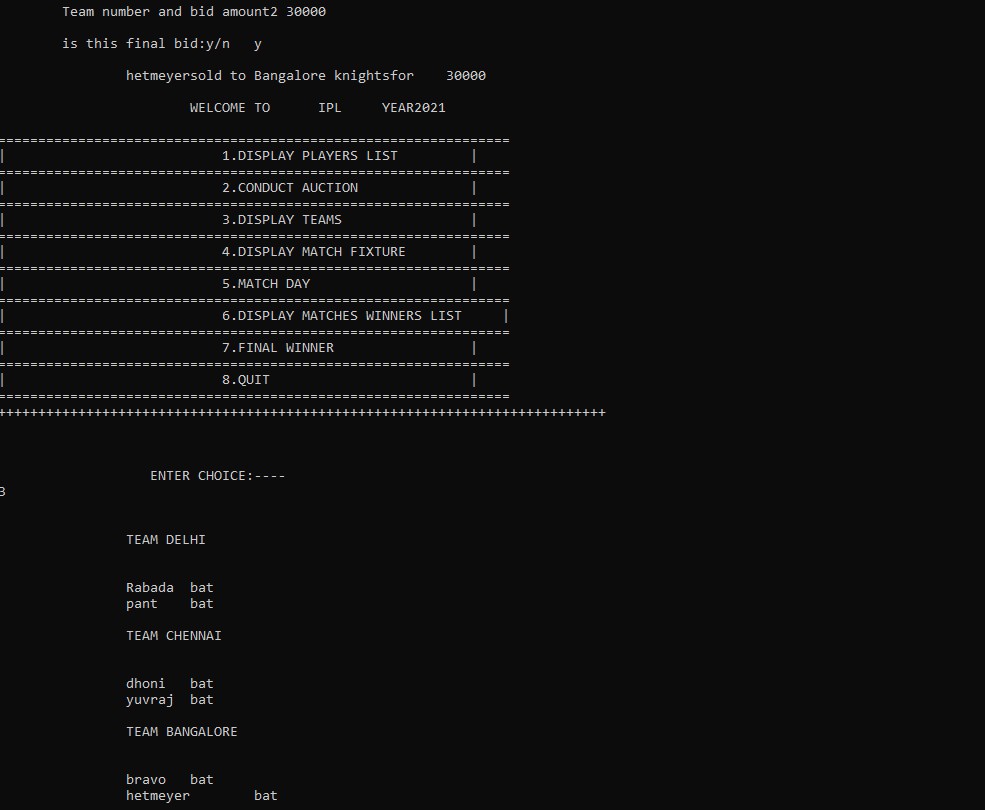
**2.Exception handling in auction:**



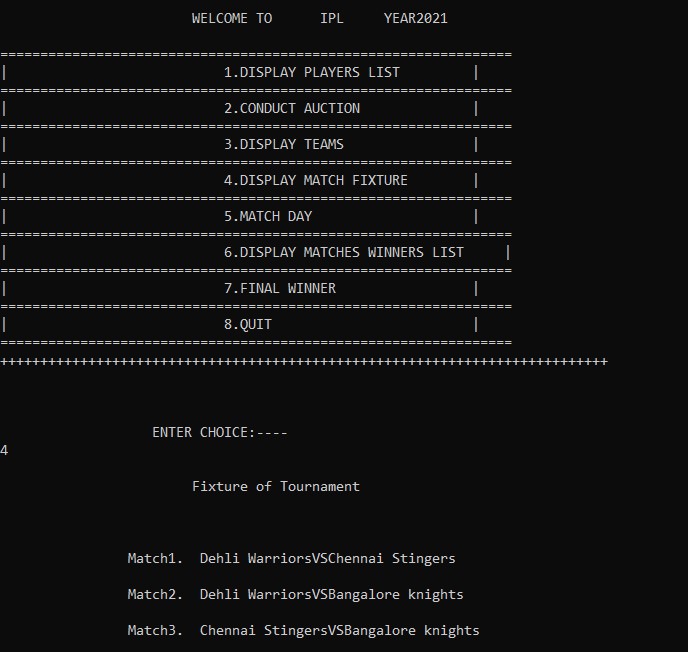
**3.Exception handling in selecting team members:**



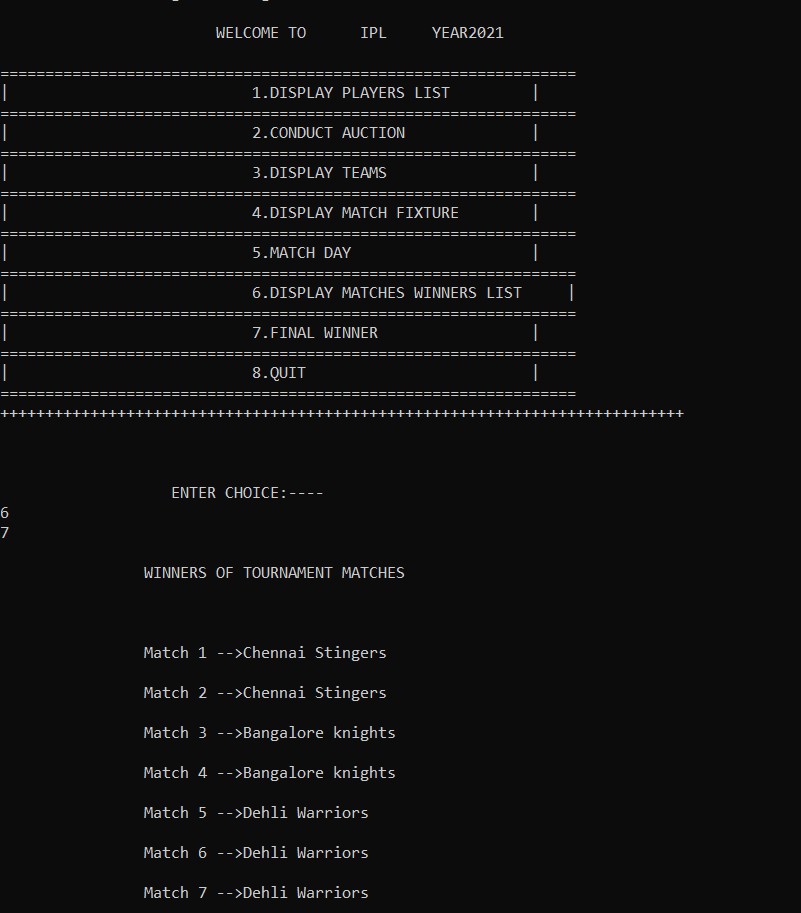
**4. Display teams**



**5. Match fixing**



**6. Display match winner**



**7. Display final winner**

