# CLASS DIAGRAM



**PlayerInitialization**

+ name[64]
+ batsman
+ bowler
+ wicketkeeper

+ GetData()
+ ShowData()

**Team**

+ TeamName[64]
+ Head* (Player)
+ pointerP* (Player)
+ TeamScore
+ AllWickets
+ Batted
+ CurrentOver
+ CurrentBall

+ InsertFromFile(fstream &file)

**Player**

+ PlayerInfo* (PlayerInitialization)
+ NoOverBalled
+ NoOfNoBalls
+ NoOfWideBalls
+ NoOfWickets
+ NoOfLBWs
+ NoOfCatches
+ BattedScore
+ out
+ onTheGround
+ NextPlayer* (Player)

**Cricket**

+ team1* (Team)
+ team2* (Team)
+ CurrentRunRate
+ RequiredRunRate

+ TeamNameInput(ifstream &file)
+ CheckingTheWinner(Team *teamBat, Team *TeamBall)
+ ScoreBoard(Team *TeamBat, Team *TeamBall)
+ CalCurrentRunRate(int i, int j, Team *TeamBat)
+ CalRequiredRunRate(int i, int j, Team *TeamBat, Team *TeamBall)
+ FirstSecondBatting(Team *teamBat, Team *TeamBall)
+ GemaFlow(Team *teamBat, Team *TeamBall)
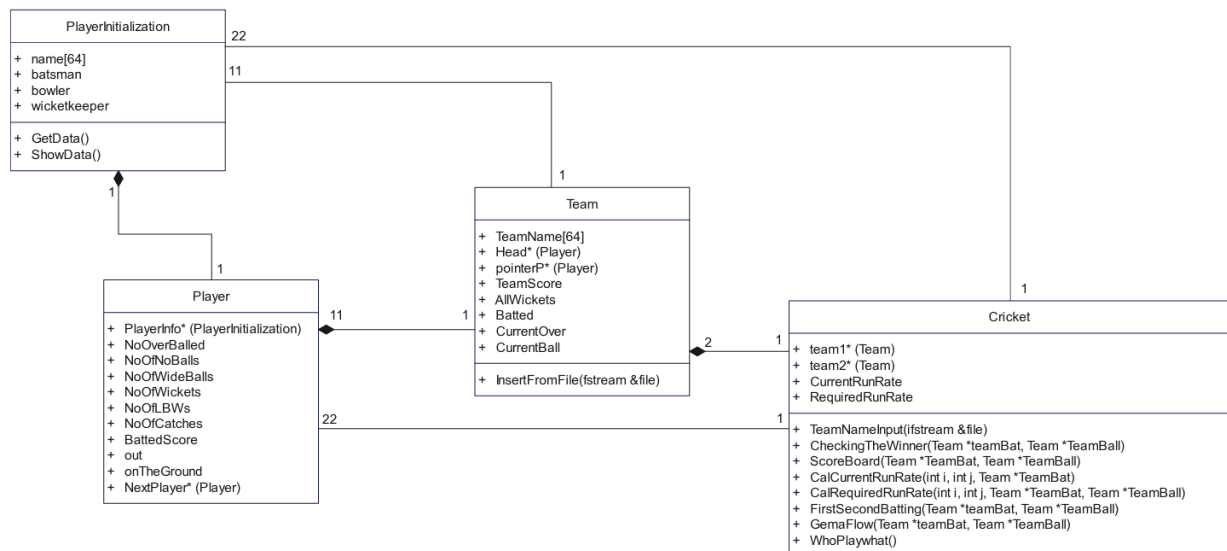+ WhoPlaywhat()

Figure 01: UML Diagram

1

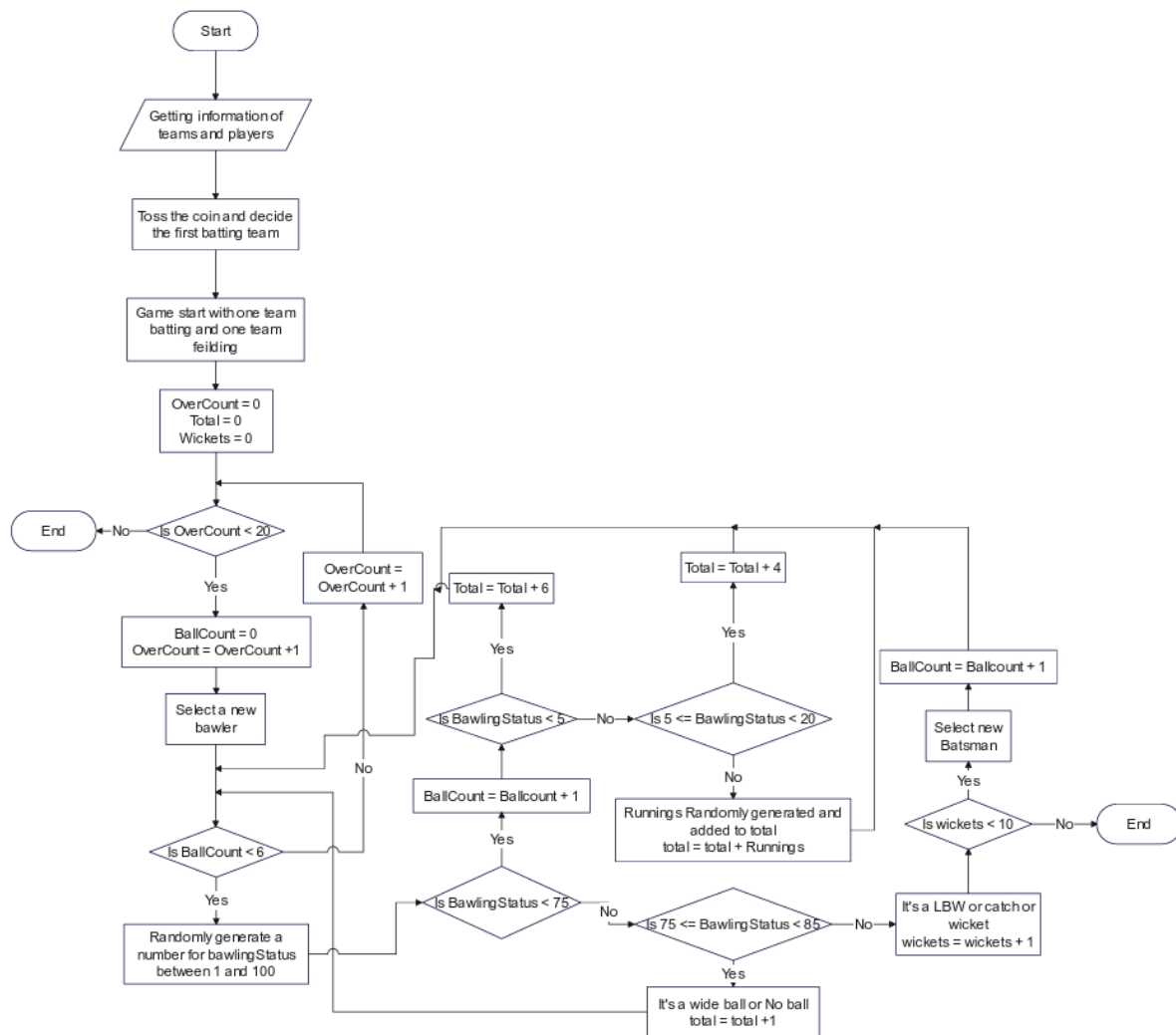# BASIC DESIGN OF THE PROJECT



Figure 02: Rough flow diagram of the first half of the game

In the second half of the game after every ball the scores of the batting team will be compared to the bowling team scores. If batting team get more than bawling team batting team wins.

If batting team's all players got out then before the game ended score comparison will be done. Then the winning team will be decided. Same goes when all the balls are over and game ended.

**Getting information of two teams and players**

Team names have been provided using a text file. That text file only contains the respective team names.

Player information has been provided using binary file streaming. Two separate binary files have been used to players of two teams.