**Task no 1:**

Construct a class Player which contains the player name, player number, number of matches, highest scores of all matches, no of 100s, no of 50s. Implement the following functionalities:

* A parametrized constructor that initialize all the data members
* Stream Insertion operator for player class

Construct another class Team that contains 11 players and team’s total scores and wickets. Implement the following functionalities in it:

* **readPlayers** function to read the data of all the players and assign them accordingly
* Highest Scorer will return the Player having height scores
* Most 100s returns the player with most 100s if two players have same highest 100s than compare 50s

Construct a class MATCH having any no of teams greater than five

* Compare among the teams having highest scores and least wicket and call it winner

Instructions about file handling(Choose any Name of the file):

* You will store updated array into file in binary form at the time of program’s termination and at the start of the program you need to read data from file in binary form and updates array.

**Perform This task with both weak/strong aggregations.**

**Task no 2:**

Consider an office system, where employees are assigned some cabins according to their designations.

Create two classes Cabin (id, ...) and Employee (id, name, ...)

Create other two classes named Cabin List (holding dynamic array of available cabins) and Employee List (holding dynamic array of registered Employees)

Identify weak/strong aggregation relation and perform selected operations

Your Program should be able to:

* Add a new employee, to employee list
* Remove an employee from employee list
* Add new cabin to cabin list
* Remove a cabin from cabin list
* Assign a cabin to an employee
* Cancel assignment /take away the cabin from employee
* id of cabins/employees must be unique

One cabin can be assigned to one employee only, and one employee can have one cabin only!

Your program should be able to show appropriate error message if we try to assign already assigned cabin to an employee

Instructions about file handling(Choose any Name of the file):

* You will store updated array into file in binary form at the time of program’s termination and at the start of the program you need to read data from file in binary form and updates array.

Your program should be able to show appropriate error message if we try to assign a cabin to an employee who already own a cabin