Comp 7013 –OOP Project

## Completion Date: 1st May 2020

## On completion please zip up your files including any documents used for drawing the class diagram etc. Upload to Canvas. Include screen shots, source code and executable jar file.

## You are to create a Soccer League application as outlined below.

A soccer league requires a (player, team, manager) management system. The league has several teams. Teams have several players. Teams are managed by Managers.

Write an application to manage the Player, Team and Managers in such a League.

We are not recording the score in matches played.

You are to write the following java classes:

1.

A Name class. This stores details of a name. Both Players and Managers have names.

 firstName;

        middleName;

          lastName;

2.

A Person class. This is a super class for all people in the League system. Its attributes are:

 Name name;

String phone;

String emai l;

3.

A Player class. This class is a subclass of person and contains Player details. It also holds the number of goals scored in the current year. Players also have a Boolean attribute called Goalie, to indicate whether or not the player is a goalie. If the player is a goalie, then the goals attribute indicates the number of goals scored while this goalie was defending the goal.

Operations: Allow get and set goals.

A Player may become a goalie

4.

A Manager class. Manager: This is a subclass of Person.  Managers have in addition to name date of birth and a star rating (1 to 10). The manager knows which Team he/she manages.

Operations: Allow get and set rating.

5.

A Team class.  This has a Manager and a collection of Players objects and the jersey colour (unique to team).

Operations: Add, Remove Players and Manager.

Only allow one Manager.

6.

A League class. This is your collection of Teams.

Operations: Add remove, Teams

7.

Write a test class which creates a League, some Team objects, some Player/Manager objects and some Manager objects. Then adds some of the Player objects to the Teams, then give the Players some goals. Add Managers to Teams. Then display all Teams, Players and Managers. Display the Team managed by a particular Manager.

**PART 2**

Design and implement a GUI to allow a user Add or remove Teams , players and Managers.

The data is to be stored in a database.

Use JPA, you can use plain old jdbc but you will loose marks if you take this approach.

At a minimum get the GUI to provide the following functionality:

A players attributes can be edited. Goals can be updated. List all players (sorted on name) in a particular team along with their goal attributes. List all managers along with their star rating ordered on star rating or alphabetically.

Provide a uml diagram of the classes in this system. (2 Marks)

Gui. (10 Marks)

Classes. (10 Marks)

Functionality . (10 Marks)

Persist all information to a database. (40 Marks JPA) (20 for jdbc)

Document (Javadoc) and test (junit) some code. (5 Marks)

Use an appropriate package structure. (3 Marks)

Use MVC. (10 Marks)

Show memory usage. (5 Marks)

Test your classes (just show some testing) (5 marks)

Your screen shots should show the following options:

1. Add a new Team, Player or Manager and add Player/Manager to a Team.
2. Remove a Player.
3. Search for a Player by supplying the Player name. Display goals and Manager details.

4. Display all players in a particular Team.

5. Display all Teams in the League.

6. Display Managers with both ordering.