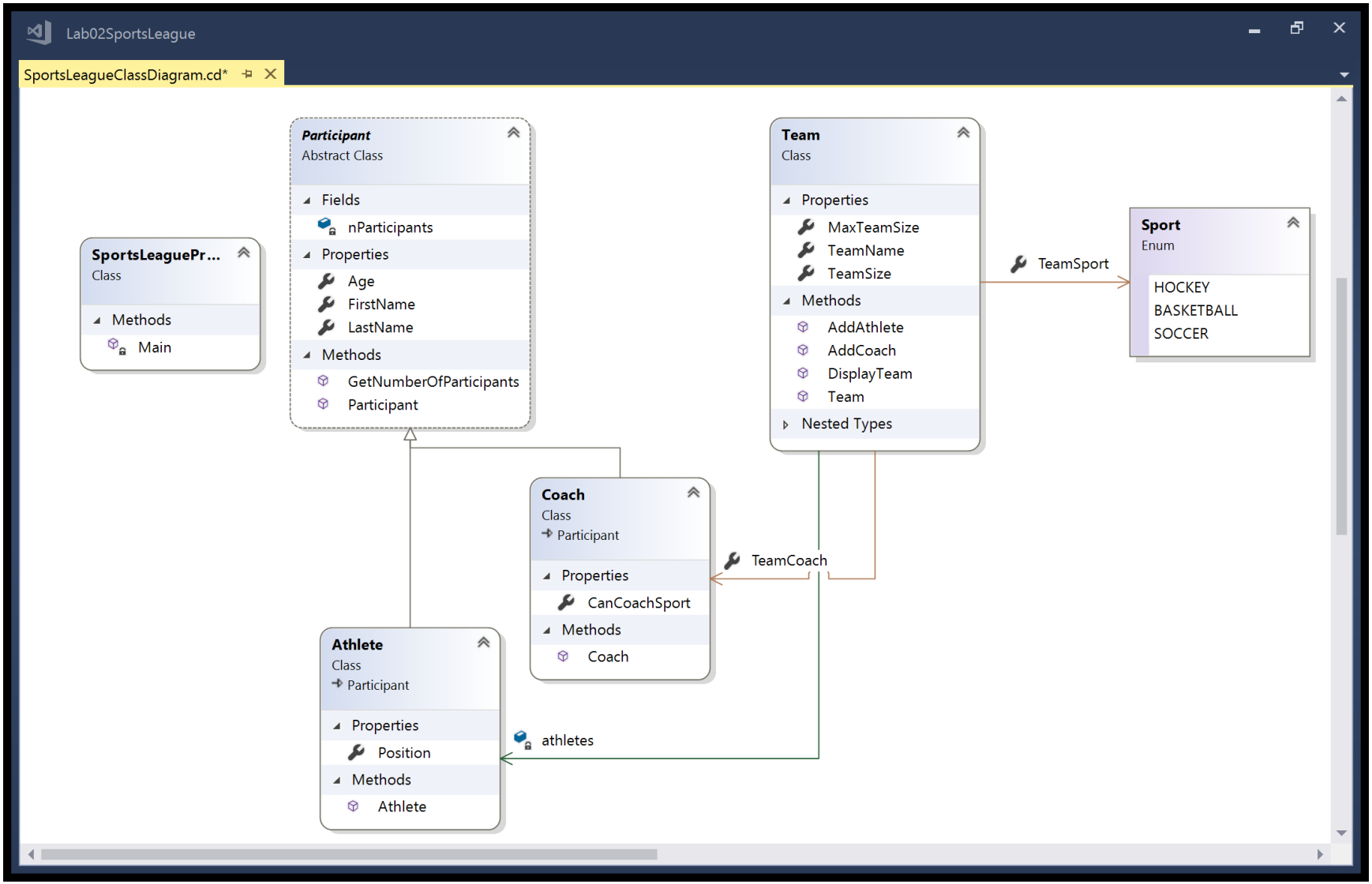
Lab 02
Sports League

Use of Arrays and Classes

Sports League Application

- Create Visual Studio Project
  - Solution and Project Name: **Lab02SportsLeague**
  - Console Application: **Lab02SportsLeague**
  - Download Lab02SportsLeague.zip from Blackboard, and unpack it to your USB drive or OneDrive in your CSIS3540 projects folder
- Description
  - Program should display the total number of participants at the beginning.
  - Program creates different sports teams with their coach and athletes, then displays each team's information and roster.
  - Implements classes for sports teams, containing an array of athletes and a coach per team.
- Setup Notes
  - Code for Main() is given and contains basic "driver" code with some example Athlete and Coach objects.
  - The code is incomplete, so there are errors.
  - Put all required classes in the SportsLeagueClasses.cs file in the project.
  - Class Diagram is shown in the following slide. You can also create classes using the Class Diagram in VS. Your choice, but it will likely be easier to do by hand.
- Project requirements
  - Implement classes given in class diagram
  - Output must display information for each team and its roster. Roster does not need to be sorted. See sample output in later slides.
- Show instructor completed lab for credit. Submit the VS solution (zipped) to Blackboard by due date for full credit.

Class Diagram



Participant, Athlete, Coach classes

- Participant is an abstract base class with the following members
  - static int nParticipants
    - The constructor should increment this, and it will let us know how many participants are in the sports league
  - Properties: Age, FirstName, LastName
  - Method: GetNumberOfParticipants() returns nParticipants
  - This is an abstract class, which means it can not be created on its own – used only by derived (child) classes.
- Athlete is derived from Participant
  - Has one additional Property – **Position**
    - String that contains the position on the team
- Coach is derived from Participant
  - Has one additional Property – **CanCoachSport**
    - Set to enum type Sport
    - It is the sport the coach is qualified to coach.

Team class members

- Constructor
  - public Team(string name, Sport sport, int maxTeamSize)
    - Set the name, the Sport (enum type) and create the internal array (athletes) of maxTeamSize
- Field: Athlete athletes[]
  - array of type Athlete
  - Should be created in the Team() constructor with maxTeamSize parameter
    - Hint: new Athlete[maxTeamSize]
- Properties
  - Coach TeamCoach
  - Sport TeamSport
    - Of type Sport (enum) that sets the type of sport (hockey, basketball, soccer)
    - Hint: sport = Sport.HOCKEY;
  - string TeamName
    - String containing the team name, as et
  - int TeamSize
    - Incremented every time an athlete is added to the team
  - Int MaxTeamSize
    - Set to size of the athletes array. Note that teamSize must be < maxTeamSize.
- Methods
  - AddAthlete(Athlete a)
    - Adds an Athlete object to the next free slot in the athletes array
    - Hint: athletes[teamSize] = a; teamSize++;
  - AddCoach(Coach c)
    - Adds a coach to the team, but the coach MUST be qualified to coach the sport.
  - DisplayTeam()
    - Output to the console listing team info (sport, number of players, coach) and the team roster (list of athletes)
    - Iterate over the athletes array to display each

Sport type (enum)

- Create an enum Sport with the following values
  - HOCKEY, SOCCER, BASKETBALL
- This is used to store the Sport in the Team class
  - Hint: TeamSport = Sport.HOCKEY;
- Also used to identify which sport a coach is qualified for.
  - Hint: CanCoachSport = Sport.SOCCER
    - Means the coach is qualified to coach soccer.

Initial Main() code

```
        static void Main(string[] args)
        {
            // create four teams

            Team mySoccerTeam = new Team("Kickers", Sport.SOCCER, 7);

            Team[] hockeyTeams = new Team[2];

            hockeyTeams[0] = new Team("Skaters", Sport.HOCKEY, 5);
            hockeyTeams[1] = new Team("Shooters", Sport.HOCKEY, 5);


            Team myBasketballTeam = new Team("Dribblers", Sport.BASKETBALL, 3);

            // add 11 athletes and one coach to the soccer team, note it only has room for 7

            mySoccerTeam.AddAthlete(new Athlete("Haspreet", "Kaur", "MidFielder", 19));
            mySoccerTeam.AddAthlete(new Athlete("Kamaljeet", "Singh", "Forward", 22));
            mySoccerTeam.AddAthlete(new Athlete("Mary", "Jones", "Goalkeeper", 23));
            mySoccerTeam.AddAthlete(new Athlete("Jen", "Loi", "MidFielder", 19));
            mySoccerTeam.AddAthlete(new Athlete("Dar", "Singh", "Forward", 22));
            mySoccerTeam.AddAthlete(new Athlete("Sim", "Baker", "Defense", 23));
            mySoccerTeam.AddAthlete(new Athlete("Tim", "Rauk", "MidFielder", 19));
            mySoccerTeam.AddAthlete(new Athlete("Red", "Nites", "Forward", 22));
            mySoccerTeam.AddAthlete(new Athlete("Mary", "Tines", "Goalkeeper", 23));
            mySoccerTeam.AddAthlete(new Athlete("Maury", "Siguras", "Defense", 30));
            mySoccerTeam.AddAthlete(new Athlete("Syd", "Calumphy", "Forward", 25));

            mySoccerTeam.AddCoach(new Coach("Sonny", "Rooster", 41, Sport.SOCCER));

            // add athletes and a coach to the first hockey team

            hockeyTeams[0].AddAthlete(new Athlete("Jack", "Gretzky", "Centre", 19));
            hockeyTeams[0].AddAthlete(new Athlete("John", "Crosby", "Forward", 22));
            hockeyTeams[0].AddAthlete(new Athlete("Jim", "Luongo", "Goalie", 23));

            hockeyTeams[0].AddCoach(new Coach("Alain", "Vineault", 50, Sport.HOCKEY));

            // add an unqualified coach and no players to the second hockey team

            hockeyTeams[1].AddCoach(new Coach("Mickey", "Mouse", 70, Sport.BASKETBALL));

            // add athletes and a coach to the basketball team

            myBasketballTeam.AddAthlete(new Athlete("Steph", "Curry", "Guard", 19));
            myBasketballTeam.AddAthlete(new Athlete("Koby", "Jones", "Centre", 23));
            myBasketballTeam.AddAthlete(new Athlete("Ken", "Bryant", "Forward", 22));

            myBasketballTeam.AddCoach(new Coach("Jerry", "West", 72, Sport.BASKETBALL));

            // show the number of participants in all sports

            Console.WriteLine("All Sports: {0} participants\n", Participant.GetNumberOfParticipants());

            // display each team

            mySoccerTeam.DisplayTeam();
            hockeyTeams[0].DisplayTeam();
            hockeyTeams[1].DisplayTeam();
            myBasketballTeam.DisplayTeam();

            Console.ReadLine();

        }
```

Console Output

- Notice that there are 21 participants, but only 13 players + 3 coaches on four teams
  - Why?
- Extra Credit
  - What can be done to improve the code in Main() to catch errors?
  - Sort rosters by last name (hint: use Compare)

```
All Sports: 21 participants

======= SOCCER Team =======
Team Name: Kickers - 7 players (7 max)
Coach: Sonny Rooster (age: 41) (CanCoach SOCCER)

Team Roster

Defense Sim Baker (age: 23)
Goalkeeper Mary Jones (age: 23)
MidFielder Haspreet Kaur (age: 19)
MidFielder Jen Loi (age: 19)
MidFielder Tim Rauk (age: 19)
Forward Kamaljeet Singh (age: 22)
Forward Dar Singh (age: 22)

======= HOCKEY Team =======
Team Name: Skaters - 3 players (5 max)
Coach: Alain Vineault (age: 50) (CanCoach HOCKEY)

Team Roster

Forward John Crosby (age: 22)
Centre Jack Gretzky (age: 19)
Goalie Jim Luongo (age: 23)

======= HOCKEY Team =======
Team Name: Shooters - 0 players (5 max)
Coach: None

======= BASKETBALL Team =======
Team Name: Dribblers - 3 players (3 max)
Coach: Jerry West (age: 72) (CanCoach BASKETBALL)

Team Roster

Forward Ken Bryant (age: 22)
Guard Steph Curry (age: 19)
Centre Koby Jones (age: 23)
```